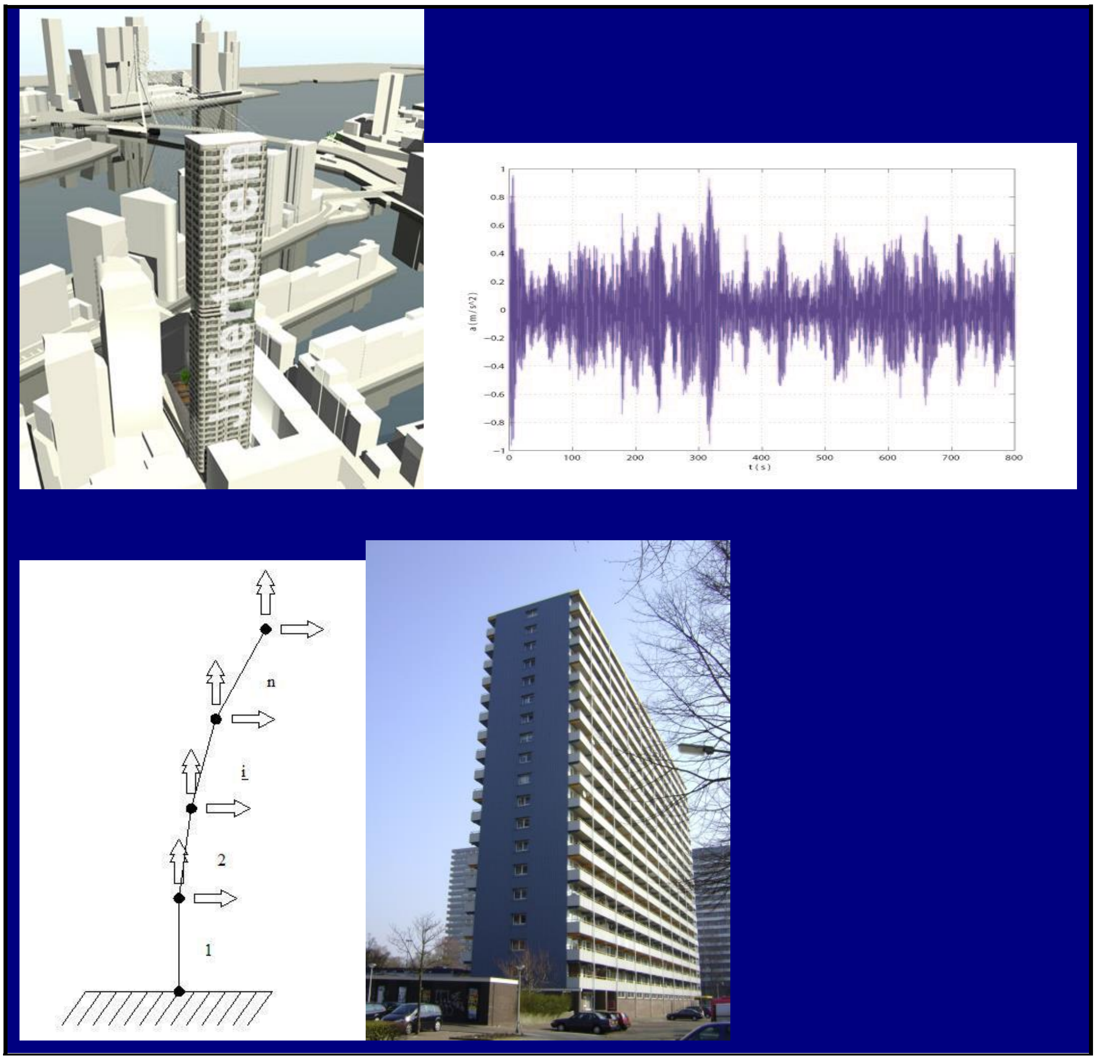
Master thesis



Torsion motions of high-rise buildings due to wind loading

16 May 2022

H.A.O.Richardson

TU Delft

Student: Hypolite Anthony Orlando Richardson 1113895

Student number:

Master thesis committee

Chairman dr.ir. P.C.J. Hoogenboom
Stevinweg 1
2628CN Delft

Committee member di.ir. W.A.A.M. Bierbooms
Klyverweg 1
2628HS Delft

Committee member em.prof.ir. A.C.W.M. Vrouwenvelder
TNO/ TUDelft
Stevinweg 1
2628CN Delft

Committee member ir. C. Kasbergen
Stevinweg 1
2628CN Delft

Preface

I have chosen this subject for my thesis, because I have always been fascinated with high-rise buildings as Taipei 101, World Trade centers, Burj Dubai, Millennium. For this same reason I chose to specialize in structural mechanics, even though the subjects are more challenging than those of other specializations. By working on this thesis, I have achieved my dream to learn and research how the loads on a building are transferred from the façade to the foundation. I was also able to research factors which influence the comfort of occupants of the building.

I must give thanks to the members of my thesis committee. I thank Professor Vrouwenvelder for his critical view on the thesis. Dr. Hoogenboom for his help with determining the structural characteristics of the buildings and proper textual presentation of this thesis. I also would like to thank Dr. Bierbooms for his help with questions that I posed pertaining to dynamics of wind, wind simulation and programming in Mathlab.

Finally, I would like to thank people along the way that helped me during this Master's thesis project and study at Delft University of Technology.

H.A.O. Richardson

Delft, 2022

Summary

This thesis is on wind induced motions of high-rise buildings. The research question was to make a modeling tool in Matlab to predict the bending and torsion oscillations of the Juffertoren and Voorhof student building in Delft and also see if these buildings satisfy the serviceability limit state. This research is a continuation of the thesis of Hans Breen [5].

The wind load causes translation in the wind direction and rotation around the vertical axis and also translation perpendicular to the wind direction. This leads to a moment at the base and a torsional moment around the vertical axis of the building. The created model does not take the translational motions perpendicular to the wind into account.

After in-depth reading of various articles on the subject of torsion and bending acceleration on buildings, no efficient tool could be found to determine the total acceleration. Presently, in the dimensioning stage of a building, static wind load values are assumed but this method has little accuracy. Therefore, a dynamic model has been made. This model has been applied to the Juffertoren in Rotterdam and the Voorhof student building in Delft. The most important output of this model are the acceleration and displacement of the building. The outcome shows that if the acceleration due to torsion is added to the acceleration due to bending, most buildings designed with the NEN 6702 (Dutch Norm) exceed the serviceability limit state for acceleration which was also the case by the Juffertoren and the Voorhof student building.

Most codes of practice neglect the torsional acceleration, which this thesis shows is not acceptable.

Index

List of symbols

A	area	$\left[m^2\right]$
A	the area loaded by wind	$\left[m^2\right]$
A	amplitude of the building movement	$\left[m\right]$
A_i	the area of the considerd wall or steel cross-section	$\left[m^2\right]$
A_i	amplitude of the homogenous solution	$\left[m\right]$
A_i	amplitude of the homogenous solution for torsion	$\left[m\right]$
A_{fl}	area of the floor	$\left[m^2\right]$
A_{ref}	reference area of the structure	$\left[m^2\right]$
a	limition demand of the acceleration	$\left[m/s^2\right]$
a	acceleration	$\left[m/s^2\right]$
a	maximum occurring acceleration due to bending and torsion motion	$\left[m/s^2\right]$
a_j	horizontal distance from node to the shear center	$\left[m\right]$
a_L	normalised limiting amplitude giving the deflection of structures with very low damping	$\left[m\right]$
a_p	horizontal distance from the node to the shear center of the building	$\left[m\right]$
a_r	calculation constant of the National building code of Canada	
$a_{\max}$	the characteristic peak acceleration	$\left[m/s^2\right]$
$a_{\max}$	the maximum acceleration	$\left[m/s^2\right]$
$a_{\max;NEN}$	limitation demand for the peak acceleration according to NEN 6702	$\left[m/s^2\right]$
a_y	the across wind acceleration of the building according to NBCC	$\left[m/s^2\right]$
a_0	constant determining the damping proportional to the mass	$\left[rad/s\right]$
a_1	constant determining the damping proportional to the stiffness	$\left[s/rad\right]$
a_{0ben}	constant determining the damping proportional to the mass	$\left[rad/s\right]$
a_{1ben}	constant determining the damping proportional to the bending stiffness	$\left[s/rad\right]$
a_{0tor}	constant determining the damping proportional to the polar moment of inertia	$\left[rad/s\right]$

a_{1tor}	constant determining the damping proportional to the torsion stiffness	$[s/rad]$
$a_{48,i}$	the acceleration of the top of the building at time point I (node 48)	$[m/s^2]$
$a_{48,\max}$	the maximum acceleration of the top of the building (node 48)	$[m/s^2]$
$a_{48;peak}$	peak acceleration of acceleration (node 48)	$[m/s^2]$
$a_{48;peak;600}$	peak acceleration of acceleration for 10 minute period (node 48)	$[m/s^2]$
$a_{48;peak;3600}$	peak acceleration of acceleration for hour long storm (node 48)	$[m/s^2]$
$a_{48;peak;21600}$	peak acceleration of acceleration for 6 hour long storm (node 48)	$[m/s^2]$
B	width of the building	$[m]$
B^2	background factor	$[m]$
b	width of the building	$[m]$
b	constant of coherence	$[m]$
b	width of the added concrete walls	$[m]$
b	the length of 3D model, distance between both applied forces	$[m]$
b	reference width of the cross-section at which resonant vortex shedding occurs	$[m]$
b_i	width of the considered wall segment	$[m]$
b_i	width of the considered steel section	$[m]$
b_m	width of the building perpendicular to the wind direction	$[m]$
b_m	width of the building	$[m]$
C	damping matrix	$[kg/s]$
C	Raleigh value	$[-]$
C	coherence decrement	$[m]$
$Coh_{jk}(f)$	coherence between the wind speed in points j and k	$[m]$
C_{ben}	the damping matrix for bending	$[kg/s]$
C_c	the aerodynamic constant dependat on cross-sectional shape	$[-]$
C_e	energy disipation factor	$[-]$
C_f	rotational stiffness of the foundation	$[Nm]$
C_h	summation of trust and suction factor	$[-]$
$C_{jk}(f)$	coherence decrement	$[-]$
C_r	rotational stiffness of the foundation	$[Nm]$

C_{tor}	the damping matrix for torsion	[kg/s]
C_t	summation of the shape factore	[−]
C_x	centre of gravity	[m]
C_y	coherence constant in longditudinal-direction	[−]
C_z	coherence constant in lateral-direction	[−]
C^*	modal damping matrix	[kg/s]
C^*_{tor}	modal torsional damping matrix	
C	the values of different types of fixations	[−]
$coh_{v_1v_2}(f)$	coherence between the wind speed in points 1 and 2 at frequency f	[−]
c_{alt}	height factor	[−]
c_{dir}	directional factor	[−]
c_f	force coefficent	[−]
c_{lat}	lateral force coefficient	[−]
$c_{lat,0}$	basic value of the lateral force coefficient	[−]
c_r	roughness factor	[−]
$c_r(z)$	roughness factor at height z	[−]
$c_r(z_e)$	roughness factor at height z_e	[−]
$c_o(z)$	orography factor at height z	[−]
$c_o(z_e)$	orography factor at height z_e	[−]
$c_s c_d$	structural factor	[−]
c_{season}	season factor	[−]
c_{temp}	season factor	[−]
c_y	decay constant	[−]
c_z	decay constant	[−]
d	depth of the building	[m]
d	displacement height	[m]
d_w	displacement height	[m]
E	Young's modulus	$[N/m^2]$
E_s	Young's modulus of steel	$[N/m^2]$
E_c	Young's modulus of concrete	$[N/m^2]$
E'_b	Young's modulus of concrete	$[N/m^2]$
EA	axial stiffness	[N]

EI	Bending stiffness	$[Nm^2]$
EI_x	Bending stiffness in x-direction	$[Nm^2]$
EI_y	Bending stiffness in y-direction	$[Nm^2]$
E	eigenmatrix containing the eigenvectors	$[-]$
E^T	transposed eigenmatrix	$[-]$
e	rotation in a node	$[rad]$
F	force vector	$[N]$
F	force matrix of mean and fluctuating wind speed for total realisation	$[N]$
F	force on structure in 3D model	$[N]$
F	force of one element on the structure	$[N]$
F	wind load on any arbitrary area	$[N]$
F	the nodal force matrix	$[N]$
F_D	reduced spectra	$[-]$
F_D	Davenport spectra	$[-]$
F_i	force in node i	$[N]$
$F_i(t)$	force of horizontal row of areas loaded by wind	$[N]$
$F_i(t)$	force summation on row of nodes per time instant	$[N]$
F_j	force in node j	$[N]$
F_k	force in node k	$[N]$
F_w	the wind force acting on a structure or structureal component	$[N]$
f	frequency	$[Hz]$
f	eigenfrequency	$[Hz]$
f	frequency of the building	$[Hz]$
f_c	coriolisparameter	$[s^{-1}]$
f_e	natural frequency of the building	$[Hz]$
f_e	eigenfrequency	$[Hz]$
f_e	frequency of the system	$[Hz]$
f_e	frequency of the building	$[Hz]$
f_{e_ben}	natural frequency of bending movement	$[Hz]$
f_{e_tor}	natural frequency of torsional movement	$[Hz]$
$f_{e\_Woudenberg}$	natural frequency of bending or torsional movement	$[Hz]$
f_L	constant of the wind power spectral density	

f_0	natural frequency	$[\text{Hz}]$
G	weight building	$[\text{N}]$
G	shear modulus	$[\text{N/m}^2]$
G_{ss}	shear modulus of structural steel	$[\text{N/m}^2]$
G_w	shear modulus of added concrete walls	$[\text{N/m}^2]$
G_y	constant of the size reduction fuction	
G_z	constant of the size reduction fuction	
GJ	torsional stiffness	$[\text{Nm}^2]$
g	gravitational acceleration	$[\text{m/s}^2]$
g	piekfactor	$[-]$
g_p	peak factor	$[-]$
H	height of the building	$[\text{m}]$
H	depth of the building	$[\text{m}]$
h	height of the building	$[\text{m}]$
h	height of structure	$[\text{m}]$
h	height of the concrete floor	$[\text{m}]$
h	length of the added concrete walls	$[\text{m}]$
h	story height	$[\text{m}]$
h_i	length of the considered wall segment	$[\text{m}]$
h_i	height of the considered steel section	$[\text{m}]$
h	height of the walls	$[\text{m}]$
h_i	height of the walls	$[\text{m}]$
h_{fl}	thickness of the floor	$[\text{m}]$
h_{story}	height of one story of the building	$[\text{m}]$
I	moment of inertia	$[\text{m}^4]$
I	second moment of Area	$[\text{m}^4]$
I	identitiy matrix	$[-]$
I	turbulence intensity	$[-]$
$I(h)$	turbulence intensity at height h	$[-]$
$I(z)$	turbulence intensity at height z	$[-]$
I_p	polar moment of inertia	$[\text{kgm}]$

I_p	polar moment of inertia	$[\mathrm{m}^4]$
I_{p_wall}	polar moment of inertia of a wall	$[\mathrm{m}^4]$
I_{p_floor}	polar moment of inertia of a floor	$[\mathrm{m}^4]$
I_v	turbulence intensity	$[-]$
$I_v(z)$	turbulence intensity at reference height z	$[-]$
$I_v(z_e)$	turbulence intensity at reference height $z = z_e$ above ground	$[-]$
I_x	second moment of inertia in x direction	$[\mathrm{m}^4]$
$I_{x;b}$	second moment of inertia in x direction before renovation	$[\mathrm{m}^4]$
$I_{x;ss}$	second moment of inertia in x direction of steel structure before renovation	$[\mathrm{m}^4]$
$I_{x;w}$	second moment of inertia in x direction of the added concrete walls	$[\mathrm{m}^4]$
I_y	second moment of inertia in y direction	$[\mathrm{m}^4]$
$I_{y;b}$	second moment of inertia in y direction before renovation	$[\mathrm{m}^4]$
$I_{y;w}$	second moment of inertia in y direction of the added concrete walls	$[\mathrm{m}^4]$
I	identity matrix	$[-]$
i	node number	$[-]$
J	torsional constant	$[\mathrm{m}^4]$
j	lateral node number	$[-]$
K	stiffness matrix	$[\mathrm{N/m}]$
K	mode shape factor	$[\mathrm{N/m}]$
K_a	aerodynamic damping parameter	$[\mathrm{N/m}]$
$K_{a,\max}$	maximum aerodynamic damping parameter	$[\mathrm{N/m}]$
K_b	spring stiffness	$[\mathrm{N/m}]$
K_{ben}	the stiffness matrix for bending	$[\mathrm{N/m}]$
K_e	effective stiffness of the system	$[\mathrm{N/m}]$
K_p	negative stiffness of the upside-down pendulum	$[\mathrm{N/m}]$
K_s	size reduction function	$[-]$
$K_s(n)$	size reduction function	$[-]$
K_{tor}	the stiffness matrix for torsion	$[\mathrm{N/m}]$
K^*	modal stiffness matrix	$[\mathrm{N/m}]$

K_{tor}^{*}	modal stiffness matrix	
K_w	effective correlation length factor	$[-]$
K_y	constant	$[-]$
K_z	constant	$[-]$
k	spring constant	$[N/m^2]$
k	factor	$[-]$
k	spring stiffness	$[N/m]$
k_c	factor of turbulence	$[-]$
k_I	turbulence factor	$[-]$
k_p	peak factor	$[-]$
$k_r(z)$	terrain factor depending on the roughness length z_0	$[-]$
k_t	terrain factor	$[-]$
L	height of the building	$[m]$
L	length of one element on the structure	$[m]$
L_j	correlation length	$[m]$
L_{gust}	characteristic length of a wind gust	$[m]$
L_t	reference length scale	$[m]$
$L(z_e)$	turbulence length scale	$[m]$
l	the building length	$[m]$
l	length of 1 element	$[m]$
l	length of the diagonal	$[m]$
l_j	the length of the structure between the nodes	$[m]$
$l_{cross\ sec}$	length from shear center to outer point of building	$[m]$
l_{floor}	length of 1 element floor height	$[m]$
l_p	pendulum length	$[m]$
M	mass matrix	$[kg]$
M	moment	$[Nm]$
M	moment vector	$[Nm]$
M	the nodal moment matrix	$[Nm]$
M	moment on structure in 3D model	$[Nm]$
M	moment matrix of mean and fluctuating wind speed for total realisation	$[N]$

M	mass of the floor and the walls of 1 storey	$[kg]$
M_{ben}	mass matrix for bending	$[kg]$
M_{tor}	polar moment of inertia matrix for torsion	$[kgm^2]$
M^*	modal mass matrix	$[kg]$
M^*_{tor}	modal torsional mass matrix	
M_i	moment of the node i	$[Nm]$
M_1	moment in node 1	$[Nm]$
M_2	moment in node 2	$[Nm]$
M_{2-1}	torsional moment in element number of local node number 1	$[Nm]$
$M_i(t)$	moment summation on row of nodes per time instant	$[N]$
M_j	moment matrix due to mean and fluctuationg wind speed	$[Nm]$
M_i	lumped mass in node i	$[kg]$
m	static moment on one node height	$[Nm]$
m	mass of the building	$[kg]$
m	the number of antinodes of the vibrating structure in the considered mode shape $\Phi_{i,y}$	$[-]$
$m_{building}$	mass of the building	$[kg]$
m_e	equivalent mass per unit length	$[kg/m]$
m_{eq}	equivalent mass of the building	$[kg]$
m_{eq_tor}	equivalent polar moment of inertia of the building	$[kg]$
m_i	moment applied on each node	$[Nm]$
$m_{i,e}$	the equivalent mass m_e per unit length for mode I as defined in F.4 (1)	$[Nm]$
m_o	static moment per meter height	$[Nm]$
m_{tot}	static moment on total building height	$[Nm]$
m_{vb}	the variable load on the floor	$[kg/m^2]$
N	the number of draws which follows from the number of local peaks in the total time range	$[-]$
n	natural frequency of the structure	$[Hz]$
n	mode number	$[-]$
n	the number of discrete time points for which a_{48} have been calculated	$[-]$
n	critical buckling factor	$[-]$
n	the number of steel cross-sections that are located in the building	$[-]$

n	the number of wall segments that the wall of the cross-section is divided into	$[-]$
n	the number of added concrete walls in the cross-section of the building	$[-]$
n_{floors}	the number of floors of the building	$[-]$
$n_{i,y}$	the natural frequency of the considerd flexural mode I of cross-wind vibration	$[-]$
n_{sec}	second order factor	$[-]$
$n.a.$	distance between the neutral axis and a parallel refernce line	$[m]$
$n_{1,x}$	natural frequency of the structure	$[Hz]$
p	load per area	$[N/m^2]$
p_{vb}	the variable load on the floor	$[N/m^2]$
$\tilde{p}_{w;1}$	value of the varying part of the wind pressure	$[N/m^2]$
Q	the building weight	$[N]$
Q_k	critical buckling force	$[N]$
Q_{optr}	actual buckling force	$[N]$
q	force per length of the structure	$[N/m]$
q	dead weight of the structure	$[N/m]$
$q_{g,rep}$	representive dead weight of the floor	$[N/m^2]$
$q_{g,d}$	dead weight of the structure	$[N/m]$
q_m	force per length of building weight	$[N/m]$
$q_p(z)$	peak velocity pressure at reference height z	$[N/m]$
$q_p(z_e)$	peak velocity pressure at reference height z_e	$[N/m]$
q_w	force per length	$[N/m]$
R	radius of the circle	$[m]$
R	square root of the resonant responce	
R^2	resonance response factor	
Sc	Scruton number	$[-]$
S_L	wind power spectral density fucnction	$[m^2/s^2]$
St	Strouhal number	$[-]$
S_v	velocity spectrum	$[m^2/s^2]$
S_{vv}	velocity spectrum	$[m^2/s^2]$

$S_w(f)$	velocity spectrum at frequency f	$[m^2/s^2]$
$S_w(\omega)$	velocity spectrum at frequency ω	
$S_{v_1v_2}(f)$	the cross-spectrum of the wind speeds in points 1 and points 2	$[m^2/s^2]$
$S_{v_iv_i}(f)$	auto-spectrum of the wind speeds in points i	$[m^2/s^2]$
s_i	the perpendicular distance between the reference line and the centre of gravity of the considerd wall or considered steel cross-section	$[m]$
T	vibration time	$[s]$
T	the averaging time for mean wind velocity	$[s]$
T	time range of signal	$[s]$
T_s	time range of the signal	$[s]$
T_1	period	$[s]$
t	time	$[s]$
t	thickness of the concrete floor	$[m]$
U	mean wind speed at reference height 10 meters	$[m/s]$
u	displacement	$[m]$
u	displacement vector	$[m]$
u	displacement calculated by the 3D program	$[m]$
$u(t)$	displacement of the forced damped equation	$[m]$
$u(t)$	natural vibration of bending or torsional movement at top building	$[m]$
$u_{b;top}$	displacement at the top of the structure due to bending	$[m]$
u_{floor}	maximum acepptable displacement due to interstory drift	$[m]$
u_i	displacement of node i	$[m]$
u_j	displacement of node j	$[m]$
u_k	displacement of node k	$[m]$
$u_m(z_i)$	mean wind speed of the wind speed at point z_i	$[m/s]$
$u_m(z_j)$	mean wind speed of the wind speed at point z_j	$[m/s]$
$u_{m,av}$	average of the mean wind speed of the wind speed at points z_i, z_j	$[m/s]$
u_{m_10}	mean wind speed at reference height 10 meters	$[m/s]$
u_p	mean wind speed at reference height 10 meters	$[m/s]$
u_{ref}	reference speed at height of 10 m	$[m/s]$
$u_{ref,0}$	onalterd reference speed at height of 10 m	$[m/s]$
$u_{static;top}$	displacement at the top of the structure for bending due to static load	$[m]$

u_{story}	displacement left for torsional motion	$[m]$
$u_{t;top}$	displacement at the top of the structure due to torsion	$[m]$
u_{top}	displacement at the top	$[m]$
$\hat{u}_h$	peak displacement at the top of the building	$[m]$
$\hat{u}_i$	eigenvector	$[-]$
u_*	friction velocity	$[m/s]$
$u_{*12,5}$	friction velocity for a return period of 50 years	$[m/s]$
$\dot{u}$	velocity	$[m/s]$
$\dot{u}$	velocity vector	$[m/s]$
$\ddot{u}$	acceleration	$[m/s^2]$
$\ddot{u}$	acceleration vector	$[m/s^2]$
v	the up-crossing frequency	$[Hz]$
$\bar{v}$	hourly-averaged part of wind speed	$[m/s]$
$\bar{v}(h)$	hourly-averaged part of wind speed at height h	$[m/s]$
$\bar{v}(z)$	hourly-averaged part of wind speed at height z	$[m/s]$
$\bar{v}(10)$	mean wind speed at reference height 10 meters	$[m/s]$
$\tilde{v}$	fluctuation part of wind speed	$[m/s]$
v_b	basic wind velocity	$[m/s]$
$v_b(z)$	basic wind velocity at height z	$[m/s]$
$v_{b,0}$	fundamental value of basic wind velocity	$[m/s]$
$v_{crit,i}$	critical wind velocity for mode i	$[m/s]$
v_h	mean wind speed at top of the building	$[m/s]$
v_m	the characteristic 10 minutes mean wind velocity specified in [10] 4.3.1(1) at the cross section where vortex shedding occurs	$[m/s]$
$v_m(z)$	mean wind velocity at height z	$[m/s]$
$v_m(z_e)$	mean wind velocity at height z_e	$[m/s]$
w	width of the concrete floor	$[m]$
x	height of the building at which the rotation angle is required	$[m]$
x	dimensionless frequency	$[-]$
x_1	the maximum displacement due to self weight applied in the vibration direction	$[-]$
y	coordinate in width direction of the building	$[m]$
y	lateral coordinate	$[m]$

$y_{F,\max}$	largest displacement calculated for cross-wind amplitude	$[\text{m}]$
y_r	node position relative to the left edge of the building	$[\text{m}]$
$y_{\max}$	the characteristic maximum displacement at the point with the largest movement	$[\text{m}]$
z	height above the surface of the earth	$[\text{m}]$
z	lateral coordinate	$[\text{m}]$
z_e	reference height or height of structure	$[\text{m}]$
z_j	lateral coordinate	$[\text{m}]$
z_k	vertical coordinate	$[\text{m}]$
z_m	value of the lateral an vertical coorinate divided by 2	$[\text{m}]$
$z_{\min}$	minimum height	$[\text{m}]$
z_0	roughness length	$[\text{m}]$
$z_{0,\text{II}}$	roughness length	$[\text{m}]$
z_r	height of chosen node point above the surface of the earth	$[\text{m}]$
z_t	reference height	$[m]$

Greek symbols

α	wind turbulence factor	
β_w	lift damping ratio	$[-]$
ϕ_2	dynamic amplification factor	$[-]$
ϕ_{ij}	nodal rotation between points i and j	$[-]$
ϕ_{jk}	nodal rotation between points j and k	$[-]$
ϕ_y	constant of the size reduction function	$[-]$
ϕ_z	constant of the size reduction function	$[-]$
$\Delta\omega$	equal distance in which the velocity spectrum is divided into	$[\text{rad/s}]$
$\Delta\omega_k$	equal distance k in which the velocity spectrum is divided into	$[\text{rad/s}]$
Δr_{jk}	the distance between points j and k	$[\text{rad/s}]$
δ	displacement	$[\text{m}]$
δ	the total logarithmic decrement of damping	$[-]$
δ_{C_f}	displacement at top of building due to foundation rotation	$[\text{m}]$
δ_{EI}	bending displacement at top of building due to wind	$[\text{m}]$
δ_{GA}	shear displacement at top of building due to wind	$[\text{m}]$

δ_{Top}	displacement at top of building	[m]
$\delta_{Top;sec}$	displacement at top of building including second order effect	[m]
δ_a	the total logarithmic decrement of aerodynamic damping	[−]
δ_d	the total logarithmic decrement of damping due to special devices	[−]
δ_s	the total logarithmic decrement of structural damping	[−]
ε_0	bandwidth factor	[−]
$\Phi(y,z)$	the mode shape	[−]
$\Phi_{i,y}(s)$	the cross-wind mode shape i	[−]
Φ_{max}	mode shape value at the point with maximum amplitude	[−]
ς	damping ratio	[−]
ς_i	damping ratio of the i -th mode	[−]
ς_i	damping ratio of the i -th mode for torsion	[−]
κ	curvature	$[1/m]$
κ	Von Karman constant	[−]
λ	length divided by width of structure	[−]
$\mu_{a;48}$	mean value of acceleration realization of node 48	$[m/s^2]$
μ_b	constant of coherence	
μ_{ref}	the reference mass per unit area	$[kg/m^2]$
ν	poisson ratio	[−]
ν_c	poisson ratio of the concrete for additional structural walls	[−]
ν_s	poisson ratio of the structural steel	[−]
ρ	mean mass density	$[kg/m^3]$
ρ	mean mass density of air	$[kg/m^3]$
ρ	average density of the building	$[kg/m^3]$
ρ	the air density under vortex shedding conditon	$[kg/m^3]$
ρ_c	mean mass density of reinforced concrete	$[kg/m^3]$
ρ_1	mass of the building per meter height	$[kg/m^3]$
φ	twist angle of element	[rad]
φ	twist angle of 3D model	[rad]
φ	random phase shift ; a number between 0 and 2π	[rad]

φ_k	random phase shift of the i-th mode	$[\text{rad}]$
φ_i	random phase shift ; a number between 0 and 2π	$[\text{rad}]$
λ	width degree of earth at location	$[°]$
Ω	diagonal matrix with the eigenfrequencies	$[\text{rad/s}]$
Ω	diagonal matrix with the eigenfrequencies for torsion	$[\text{rad/s}]$
Ω	rotation speed of the earth	$[\text{rad/s}]$
ω	cyclic frequency	$[\text{rad/s}]$
ω_b	frequency of spring stiffness	$[\text{rad/s}]$
ω_D	damped natural frequency	$[\text{rad/s}]$
ω_e	damped natural frequency	$[\text{rad/s}]$
ω_e	frequency of the effective frequency of the system	$[\text{rad/s}]$
$\omega_{e\_ben\_mathlab}$	damped natural frequency for bending motion out of mathlab	$[\text{rad/s}]$
$\omega_{e\_tor\_mathlab}$	damped natural frequency for torsional motion out of mathlab	$[\text{rad/s}]$
ω_{e_Breen}	damped natural frequency for bending for Breen's model	$[\text{rad/s}]$
$\omega_{e\_mathlab}$	corrected damped natural frequency for Breen's model	$[\text{rad/s}]$
$\omega_{e\_Woudenberg}$	natural frequency of bending or torsional movement	$[\text{rad/s}]$
ω_k	finite number of points of which the velocity spectrum is divided into	$[\text{rad/s}]$
ω_i	eigenfrequency of the i-th mode	$[\text{rad/s}]$
ω_i	eigenfrequency of the i-th mode for torsion	$[\text{rad/s}]$
ω_n	first natural frequency	$[\text{rad/s}]$
ω_n	first natural frequency for torsion	$[\text{rad/s}]$
ω_{n_tor}	first natural frequency of torsional movement	$[\text{rad/s}]$
ω_p	natural frequency of the upside-down pendulum	$[\text{rad/s}]$
ψ	angle of rotation of the nodes	$[\text{rad}]$
ψ	rotation of the cross section	$[\text{rad}]$
$\psi_{static;element}$	angle of rotation of static element	$[\text{rad}]$
ψ_{top}	angle of rotation at the top of the building	$[\text{rad}]$
ψ_i	angle of rotation of the node i	$[\text{rad}]$
ψ_1	angle of rotation of the node 1	$[\text{rad}]$
ψ_2	angle of rotation of the node 2	$[\text{rad}]$
$\dot{\psi}$	angular velocity of the nodes	$[\text{rad/s}]$
$\ddot{\psi}$	angular acceleration of the nodes	$[\text{rad/s}^2]$

$\sigma_{a;48}$	standard deviation of acceleration of node 48	$\left[m^2/s\right]$
σ_v	standard deviation of wind speed	$\left[m/s\right]$
σ_v^2	variance of velocity spectrum	$\left[m^2/s^2\right]$
σ_y	standard deviation of the displacement	$\left[m/s\right]$
ζ	damping ratio	$\left[-\right]$
ζ_i	damping ratio for torsion	$\left[-\right]$

1. Introduction

In this thesis, an investigation is presented of the bending and torsion oscillations of high-rise buildings due to wind loading. This research is a continuation of the thesis of Hans Breen [5]. The aim of this thesis is to make a modeling tool in Matlab to predict the bending and torsion oscillations of high-rise buildings due to wind load fluctuations in time and space. With the developed modeling tool more challenging designs are feasible, eliminating conservative approximations in wind loading in sevicibility limit state.

The European and Dutch standards are expected to be conservative when it comes to wind loading on tall buildings, which means that the buildings are over dimensioned. Many Dutch and European standards also specify that there must be enough sunlight in a building. This entails that a building's width is limited, which makes the building more susceptible to dynamic loading.

My research questions were:

> Does the Juffertoren comply in the serviceability limit state when looking at bending and torsional accelerations in the along wind direction?
>
> Does the structurally strengthened student building Voorhof in Delft comply in the serviceability limit state, when looking at bending and torsional accelerations in the along wind direction? After strengthening there were still complaints that occupants were experiencing motion sickness during storms.

This Master thesis researches whether the maximum acceleration of a building in the serviceability limit state is exceeded due to wind loading. Research was also carried out on how the natural frequencies vary due to different design rules and the effect this has on the predicted maximum acceleration in the serviceability limit state.

The used Juffertoren design was never built, due to the fact that design calculations of DHV showed that wind loading would cause accelerations that are too large by norm regulations. These accelerations would be detrimental to the comfort of the users of the building. A lower and less slender structure was built instead.

Due to the hign costs of a windtunnel aeroelastic model, companies are reluctant to build this model in the early design fase when the dimensions of the building have only been estimated. The outcomes of this model (the forces, moments and response) are only valid for the specific shape of the model. A structural firm will not invest a large sum of money which will be useless in a later stage.

This is why a computational design tool that can predict the occilations due to windload is important.

In this thesis, the new method is first explained. Then the time-domain analysis values (bending and torsional acceleration) for the Juffertoren are determined and are compared to the values out of design formulas (bending acceleration). The same is done for the student building Voorhof, before and after renovation. After which the frequency-domain analysis results for the Juffertoren and the Student building Voorhof after renovation are presented (bending acceleration). And finally the comparison between the bending acceleration results for the time-domain analysis and frequency-domain analysis for the Juffertoren and the student building Voorhof after renovation are presented.

2. Literature study

Literature was sought for determining the torsional accelerations in buildings. The outcome is that there have been many studies on wind on high rise buildings but very few guide lines or tools exist to determine the bending and torsional accelerations accurately in the early design stage.

2.1. Literature on dynamic modeling

Below is a list of publications that are relevant to the subject of dynamic modelling of high-rise buildings.

Bezeos N, & Beskos D.E. (1996). Torsional moments on buildings subjected to wind loads. Engineering Analysis with Boundary Elements. Elsevier Vol.18,No.4, pp. 305-310.

Geurts C.P.W. (1997). *Wind-Induced Pressure Fluctuations on Building Facades.* Dissertation: Eindhoven University of Technology, The Netherlands.

Holmes J, Rofial A, & Arelius L. (2003). High frequency base balance methodologies for tall buildings. Preceedings of the 11th international conference on Wind Engineering, Texas, USA (online 13 dec. 2012: http://windtech.com.au/wp-content/uploads/2012/08/09.pdf)

Hong S, Kang B, & Park J. (2003). *Dynamic Analysis of Bending-Torsional Coupled Beam structures Using Exact dynamic elements.* Kumi.

Kareem A, Gu M, & Zhou Y. (2000). *Equivilent static buffeting wind loads on structures.* Notre Dame, Shanghai: University of Notre Dame, Tongji University.

Kareem A., & Chen. X. (2005). *Validity of Wind Load Distribution based on High Frequency Force Balance Measurements.* Notre Dame, Indiana: University of Notre Dame.

Markarios T. (2008). Practical Calculations of the torsional stiffness radius of multistory tall buildings. *The Structural Design of Tall and Special Buildings*, 39-65.

Oosterhout G.P.C. van. (1996). *Wind-Induced Dynamic Behaviour of Tall Buildings.* dissertation Delft, University of Technology.

Steenbergen R.D.J.M. (2007). *Super elements in High-Rise Buildings under Stochastic Wind Load.* dissertation Delft, University of Technology.

Tamara Y. (n.d.). Wind Tunnel Tests and Full-scale Mesurements.

Tomasevic S., Grubisic R., & Senjanovic I. (2007). Coupled Horizontal and Torsional Vibrations of Container Ships.

Woudenberg I.A.R. (2001). *Wind en de hoogbouwdraagconstructie.* Thesis Delft, University of Technology.

2.2. Wind tunnel measurements

Wind induced torsion of a building is caused by a torsional moment due to fluctuating wind gusts on a building surface and vortex shedding on edges of the facade surface. Torsion in a building may also occur due to an irregular cross-section of a building.

There have been a lot of studies on motions of buildings in wind tunnels. The **force-balance approach** (section 2.2.1) is mostly used to determine the forces, moments and response on a structure. This model type is reasonably cheap and modifiable in comparison with the **pressure integration method** (section 2.2.2) and **aeroelastic model**, (section 2.2.3). For slender buildings an expensive aeroelastic physical model has to be used.

2.2.1. Force-balance approach [21]

A force-balance test measures overall shears and moments at the base of a stiff and lightweight model. The base moments are used to estimate the generalized forces for further dynamic analysis. The force-balance test has several advantages:

- The required model has to be stiff and lightweight but no other requirement is necessary except for the scaled geometry of the building.
- A force-balance model can be constructed quickly (typically within two to three weeks).
- The measurements of base moments inherently include the correlation of the wind forces on various parts of the building.
- Structurally-connected towers can be tested by collecting simultaneous measurements on each component using a multiple-balance setup.
- Another advantage is that relatively little (statistical) calculations are required to make the results usable for the desigh engineer.

On the other hand, the force-balance analysis relies on the assumption of nearly linear mode shapes. It is not suitable for flexible structures where the mode shapes may be significantly non-linear. It is also not suitable for flexible structures where the natural frequencies for higher modes lie in the range where significant wind energy is available.

2.2.2. Pressure integration method [22]

State-of-the-art pressure instrumentation allows simultaneous local pressure measurements at about 1,000 locations on a pressure model. These pressures can be converted to loads on the building and can be integrated at each sampled time interval to determine the overall structural load time histories. Generalized forces are calculated using the measured load distribution with the mode shape.

The information obtained from the pressure integration method is generally equivalent to that available from the force-balance tests. There are a number of differences between the force-balance and pressure tests:

- The integration of wind loads on all building surfaces includes the correlation and coherence of the force components in the x, y and torsional directions.
- The variation of wind loads with height is available in detail and can be used together with any variation of mode shapes.
- Estimates of the torsional contribution to the generalized forces are significantly improved over the force-balance analysis.
- A pressure test is inherently more robust.
- The construction of a pressure model is more time consuming; hence, the information on overall loads takes longer to develop than by using a force-balance model.

For very complex geometry or for multiple towers, a large number of pressure taps is required to cover all building surfaces. It requires a large interior space in the pressure model, and hence sometimes integration resolution can be compromised.

In general, the force-balance test and the pressure integration method can be treated as equivalent. It is generally recommended that a force-balance study be carried out if structural load information is required urgently or if there is good possibility of geometry changes. The pressure integration method is more cost effective since only one model and one set of tests are required to obtain both structural loads and local pressure information.

2.2.3. Aeroelastic model [23]

Aeroelastic model tests are suitable for buildings that are unusually slender, flexible or have significant structural coupling. Any of these characteristics may lead to significant wind-induced motion. With this motion the building interacts with the surrounding air creating additional forces not accounted for directly in the rigid model testing techniques. These forces, commonly referred to as aerodynamic damping can enhance or mitigate building motion.

An aeroelastic model is designed with correctly-scaled dynamic properties of the full scale structure. Making this scale model is a challenge in itself. For some effects (cables on a bridge) only a full size model is suitable.The wind induced responses (deflections, accelerations, bending moments) may then be measured directly in the wind tunnel. The responses inherently include the additional effect of the motion-induced forces.

If properly carried out, the aeroelastic model tests provide the most accurate information on structural responses due to wind. They do require extensive design and construction of the model and a more intensive test program.

3. Idealisation of the Juffertoren building, bending and torsional model

In this chapter, Juffertoren is idealized and the structural characteristics are determined.

3.1. Design and idealisation

This design model of DHV (structural firm) in the early design process is used, because there were doubs that it would exceed the servicibility limit state requirement. A thesis study with this design model was done by a student earlier [5] in which only the bending accelerations are taking into account. In this model bending and torsional accelerations occur simultaneously. The used Juffertoren design of DHV (structural firm) has a height of 144 m, a width of 26.34 m and a depth of 15.44 m. The design was planned to be built in the Wijnhaven district of Rotterdam in the Netherlands. (Fig. 3.1) The final design for the Juffertoren differs from this design.

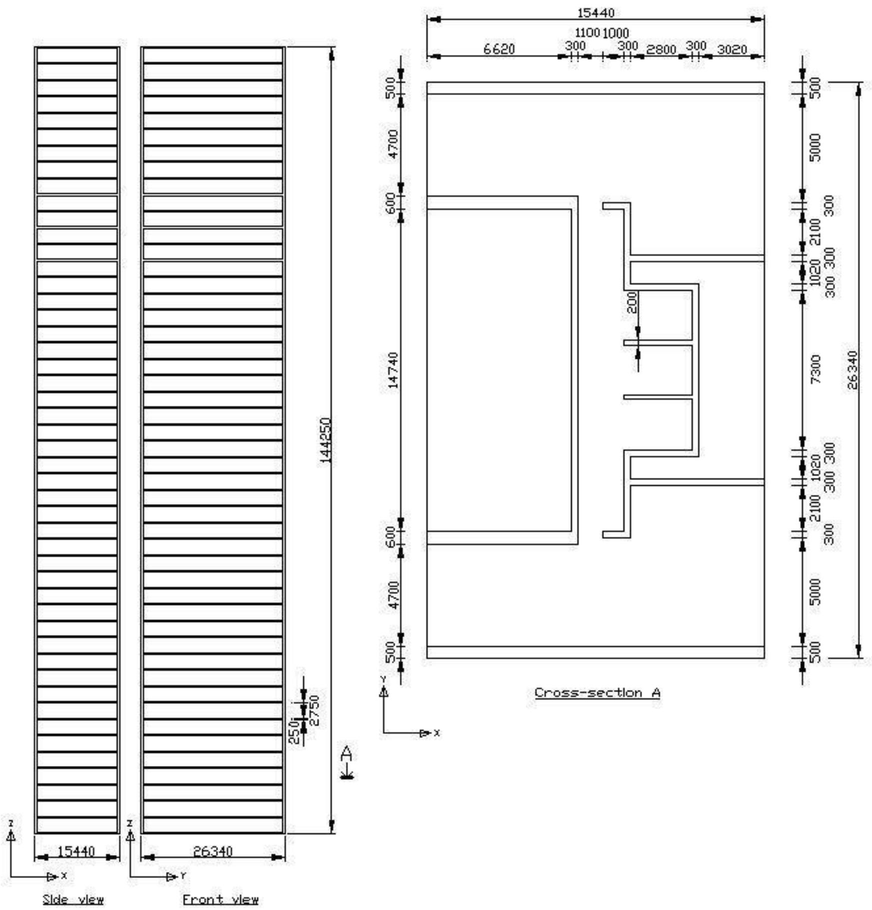


Figure 3.1: Drawing of the building

The Juffertoren building is modeled by 48 elements, 1 for each floor. The elements are connected at nodes. Each node has two degrees of freedom, which represent the displacement and the rotation of a floor. (Fig. 3.2)

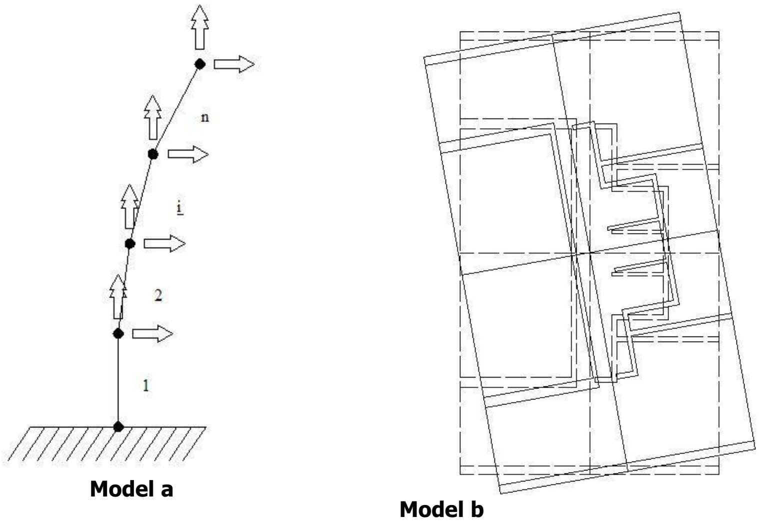
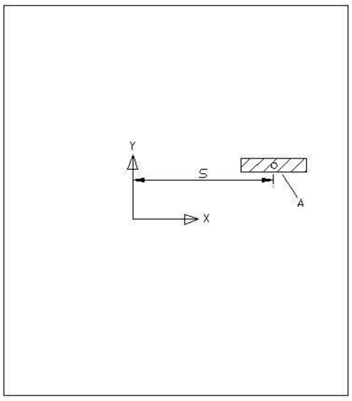


Figure 3.2: a) Displacements and rotations of the floors b) Rotation of a representative cross-section.

3.2. Cross-section properties

Centre of gravity

The location of the neutral axis of the cross-section in x and y direction of the building structure can be determined with the formula:



$$C_x = \frac{\sum s_i A_i}{\sum A_i}$$

Figure 3.3: Computation of the centre of gravity

Where:

s_i The perpendicular distance between the reference line and the centre of gravity of the considered wall, or column.

A_i The area of the considered wall, or column.

The calculation of the location of the neutral axis in x and y direction is performed in Matlab (Appendix 2). The neutral-axis of the Juffertoren design is located in the x direction at -2.571 m from the centerline and located in the y direction at 0.021 m from the centerline.

The second moments of the cross-section are

$$I_x = \sum_{i=1}^{n} \frac{b_i h_i^3}{12} + \sum_{i=1}^{n} s_{x,i}^2 A_i \quad I_y = \sum_{i=1}^{n} \frac{h_i b_i^3}{12} + \sum_{i=1}^{n} s_{y,i}^2 A_i$$

where:

n The number of wall segments that the wall of the cross-section is divided into.

b_i The width of the considered wall segments.

h_i The length of the considered wall segments.

$s_{x,i}$ The perpendicular distance between the reference line and the centre of gravity of the considered wall, or column for x direction.

$s_{y,i}$ The perpendicular distance between the reference line and the centre of gravity of the considered wall, or column for y direction.

A_i The area of the considered wall, or column.

The second moment of inertia in x and y direction were calculated in Matlab (Appendix 2). The outcome is $I_x = 3409.3$ m^4 and $I_y = 666.4$ m^4.

The shear modulus is defined as

$$G = \frac{E}{2(1+\nu)}$$

We assume $\nu = 0.15$ for uncracked concrete. We use a concrete strength for the Juffertoren of C35/B65. ([25]) We assume $E'_b = 30000$ N/mm^2 $= 3.00 * 10^{10}$ N/m^2 which leads to a shear modulus of $E/2.3 = 13043$ N/mm^2 $= 1.304 * 10^{10}$ N/m^2

Torsion Constant

The torsional constant is calculated as:

$$J = \frac{1}{3}\sum_{i=1}^{n} b_i^3 h_i = 2.7544 \text{ m}^4$$

The torsional stiffness of the cross section is:

$$GJ = 3.592 * 10^{10} \text{ Nm}^2$$

Polar Moment of Inertia

The polar moment of inertia of the wall cross section is

$$I_{p\_wall} = \int_A r^2 dA = I_x + I_y$$

The polar moment of inertia of the floor cross section is

$$I_{p\_floor} = \int_A r^2 dA = I_x + I_y$$

Storey Mass

The mass of the floor and the walls of one storey can be calculated with the formula:

$$M = \rho_c \sum_{i=1}^{n} h_i A_i + \rho_c h_{fl} A_{fl} + m_{vb} A_{fl} = 1.479*10^7 \text{ kg}$$

where:

ρ_c	specific gravity of reinforced concrete, $\rho_c = 2500$ kg/m^3;
h_i	height of the walls, $h = 2.75$ m
A_i	area of the walls
h_{fl}	thickness of the floor, $h = 0.25$ m
A_{fl}	area of the floor
m_{vb}	The variable load on the floor , $m_{vb} = 70$ kg/m^2 = 0.7 kN/m^2;

Mass of the furniture, decorations and inhabitants are estimated at $m_{vb} = 70$ kg/m^2 $= 0.7$ kN/m^2 [1]

Shear Center

The *shear center* is an imaginary point on a section, where a shear force can be applied without introducing any torsion. If the section is twisted, it also rotates around the shear center. It is not easy to compute the shear center. For now it is assumed that the shear center is equal to the center of gravity. This is a good approximation due to the symmetry of the floor plan.

[1] When the mass becomes less the smallest natural frequency becomes larger. This can be closer to a loading frequency. It is not possible to choose a safe value beforehand.

4. Structural Matrices

The essence of this chapter is to determine the different matrices and values for bending and torsion and to show these are assembled in the global stiffness matrices. For the Raleigh damping method [2] the matrices and values are also determined.

4.1. Field bending element

The model is built up of 48 elements. The flexibility of the elements is concentrated in springs at the nodes. (**Figure 4.1**) The relation between the moment *M* and the rotation *e* is:

$$M = Ke$$

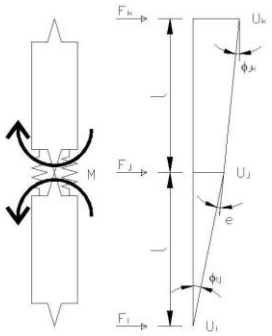


Figure 4.1: Bending element

The relation between the rotation and nodal displacements can be found with:

$$\left.\begin{array}{l} \phi_{ij} = \dfrac{1}{l}\left(u_j - u_i\right) \\ \phi_{jk} = \dfrac{1}{l}\left(u_k - u_j\right) \end{array}\right\} \rightarrow e = \phi_{ij} - \phi_{jk} = \frac{1}{l}\left(-u_i + 2u_j - u_k\right)$$

Hence:

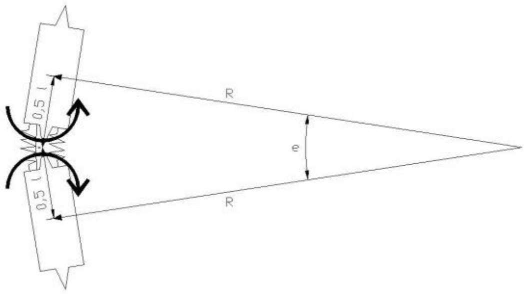


Figure 4.2: Bending element deformation

$$M = EI\kappa = EI\frac{1}{R}$$

$$e = \frac{l}{R} \rightarrow R = \frac{l}{e}$$

So,
$$M = EI\frac{e}{l}$$

Moment equilibrium gives:

$$F_i = -\frac{M}{l}$$

$$F_k = -\frac{M}{l}$$

Horizontal equilibrium gives:

$$F_i + F_j + F_k = 0 \rightarrow F_j = -F_i - F_k = 2\frac{M}{l}$$

This provides the stiffness matrix of a bending element:

$$\begin{bmatrix} F_i \\ F_j \\ F_k \end{bmatrix} = \frac{EI}{l^3}\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}\begin{bmatrix} u_i \\ u_j \\ u_k \end{bmatrix}$$

4.2. Field torsional element

The notation of the torsional moments will be written as the letter M followed by the element number and the local node number, separated by a dash. Example: $M_{3\text{-}1}$

$$\varphi = (\psi_1 - \psi_2)$$

$$M_{..-2} = -\frac{GJ}{l}\varphi$$

$$M_{..-1} = -M_{..-2}$$

Where ψ is the angle of rotation of the nodes and φ is the twist angle of the element.

$$\begin{bmatrix} M_1 \\ M_2 \end{bmatrix} = \frac{GJ}{l}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} \psi_1 \\ \psi_2 \end{bmatrix}$$

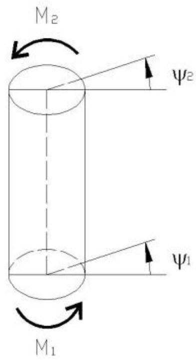


Figure 4.3: Field torsional element

4.3. Bending stiffness matrix

The derivation of the bending stiffness matrix for the last 3 nodes of the structure is shown below

$$\begin{bmatrix} . \\ . \\ . \\ . \\ . \\ . \\ . \\ F_{47} \\ F_{48} \\ F_{49} \end{bmatrix} = \frac{EI}{l^3}\begin{bmatrix} . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & 1+4+1 & -2-2 & 1 & 0 & 0 \\ . & . & . & . & . & -2-2 & 1+4+1 & -2-2 & 1 & 0 \\ . & . & . & . & . & 1 & -2-2 & 1+4+1 & -2-2 & 1 \\ . & . & . & . & . & 0 & 1 & -2-2 & 1+4 & -2 \\ . & . & . & . & . & 0 & 0 & 1 & -2 & 1 \end{bmatrix}\begin{bmatrix} . \\ . \\ . \\ . \\ . \\ . \\ . \\ u_{47} \\ u_{48} \\ u_{49} \end{bmatrix}$$

The derivation of the bending stiffness matrix for the first 3 nodes of the structure is shown below

$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} = \frac{EI}{l^3} \begin{bmatrix} +4+1 & -2-2 & 1 & 0 & \cdot\ \cdot\ \cdot \\ -2-2 & 1+4+1 & -2-2 & 1 & \cdot\ \cdot\ \cdot \\ 1 & -2-2 & 1+4+1 & -2-2 & \cdot\ \cdot\ \cdot \\ 0 & 1 & -2-2 & 1+4+1 & \cdot\ \cdot\ \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot\ \cdot\ \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot\ \cdot\ \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot\ \cdot\ \cdot \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}$$

K_{ben}

In the global stiffness matrix, the first row and column can be omitted from the matrix because the displacement of node 0 is set to zero.

$$\begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} = \frac{EI}{l^3} \begin{bmatrix} 1+4+1 & -2-2 & 1 & \cdot\ \cdot\ \cdot\ \cdot \\ -2-2 & 1+4+1 & -2-2 & \cdot\ \cdot\ \cdot\ \cdot \\ 1 & -2-2 & 1+4+1 & \cdot\ \cdot\ \cdot\ \cdot \\ \cdot & \cdot & \cdot & \cdot\ \cdot\ \cdot\ \cdot \\ \cdot & \cdot & \cdot & \cdot\ \cdot\ \cdot\ \cdot \\ \cdot & \cdot & \cdot & \cdot\ \cdot\ \cdot\ \cdot \\ \cdot & \cdot & \cdot & \cdot\ \cdot\ \cdot\ \cdot \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}$$

4.4. Torsional stiffness matrix

The derivation of the torsion stiffness matrix for the last 3 nodes of the structure is shown below

$$\begin{bmatrix} \cdot \\ \cdot \\ M_{47} \\ M_{48} \\ M_{49} \end{bmatrix} = \frac{GJ}{l} \begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1+1 & -1 & 0 \\ \cdot & \cdot & -1 & 1+1 & -1 \\ \cdot & \cdot & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} \cdot \\ \cdot \\ \psi_{47} \\ \psi_{48} \\ \psi_{49} \end{bmatrix}$$

The derivation of the bending stiffness matrix for the first 3 nodes of the structure is shown below

$$\begin{bmatrix} M_0 \\ M_1 \\ M_2 \\ \cdot \\ \cdot \end{bmatrix} = \frac{GJ}{l} \begin{bmatrix} 1+1 & -1 & 0 & \cdot & \cdot \\ -1 & 1+1 & -1 & \cdot & \cdot \\ 0 & -1 & 1+1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} \psi_0 \\ \psi_1 \\ \psi_2 \\ \cdot \\ \cdot \end{bmatrix}$$

K

In the global stiffness matrix, the first row and column can be omitted from the matrix because the rotation of node 0 is set to zero.

$$\begin{bmatrix} M_1 \\ M_2 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} = \frac{GJ}{l} \begin{bmatrix} 1+1 & -1 & \cdot & \cdot & \cdot \\ -1 & 1+1 & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} \psi_1 \\ \psi_2 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}$$

4.5. Mass matrix

The structure is modeled as discontinuous. This entails that the mass of the structure is lumped in the nodes. The mass of each element is put on the diagonal in the mass matrix.

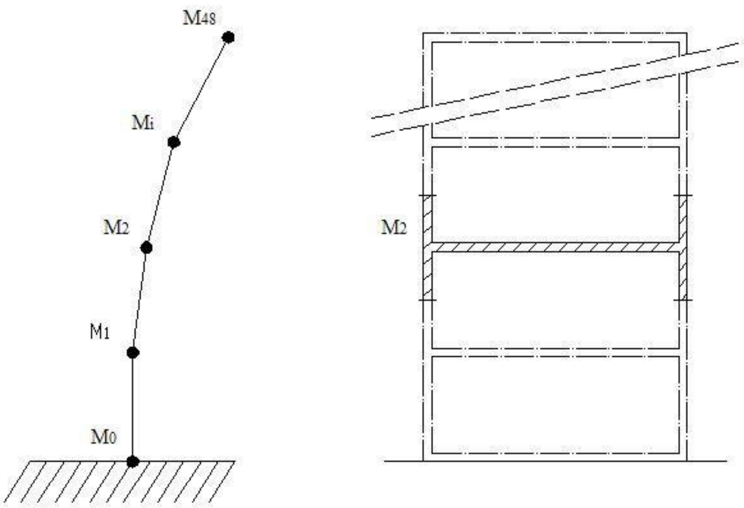


Figure 4.4: Lumping the mass in nodes

4.6. Polar Moment of Inertia

The structure is modeled as discontinuous. This entails that the polar moment of inertia of the structure is lumped in the nodes. The polar moment of inertia of each element is put on the diagonal in the polar moment of inertia matrix.

$$I_{p\_floor} = \left(\frac{1}{12}BH^3 + \frac{1}{12}HB^3\right) = 31593 \text{ m}^4$$

$$I_{p\_wall} = I_x + I_y = 4077 \text{ m}^4$$

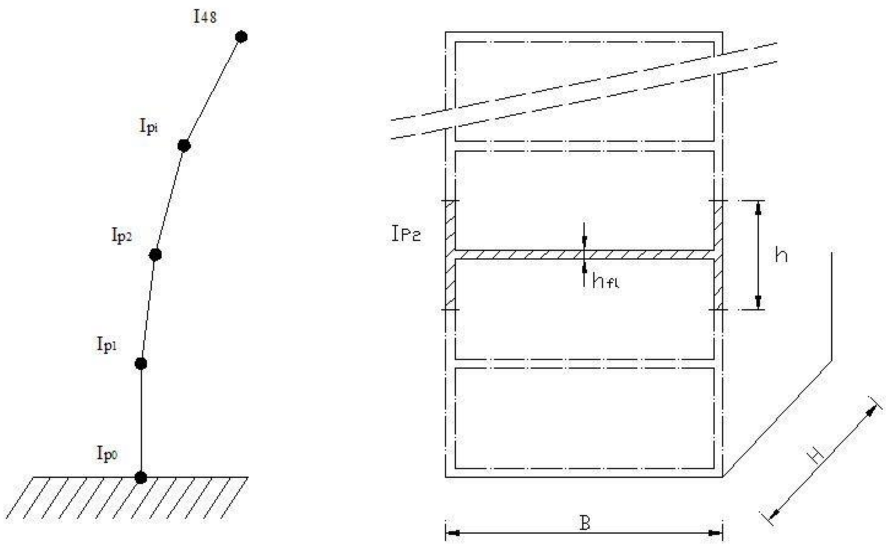


Figure 4.5: Lumping the polar moment of inertia in nodes

$$I_p = I_{p\_wall} h\rho + I_{p\_floor}\left(h_{fl}\rho + m_{vb}\right) = 50521317 \text{ kgm}^2$$

story height	$e = h = 3$ m	mean density ρ_c	$= 2400$ kg/m^3
floor height	$h_{fl} = 0.25$ m	width building B	$= 26.34$ m
depth building	$H = 15.44$ m	variable mass m_{vb}	$= 70$ kg/m^2

$$I_{p48} = I_{p\_wall}\frac{h}{2}\rho + I_{p\_floor}\left(h_{fl}\rho + m_{vb}\right) = 35844192 \text{ kgm}^2$$

4.7. Assembling the global stiffness matrix

The mass and polar moment of inertia matrixes for the entire building are assembled with the use of Mathlab. The mass matrix is 49 x 49 before reduction.

$$
\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ F_{47} \\ F_{48} \end{bmatrix}
=
\begin{bmatrix}
M & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & & & & & 0 & 0 \\
0 & M & 0 & 0 & 0 & 0 & 0 & \cdots & & & & & 0 & 0 \\
0 & 0 & M & 0 & 0 & 0 & 0 & \cdots & & & & & 0 & 0 \\
0 & 0 & 0 & M & 0 & 0 & 0 & \cdots & & & & & 0 & 0 \\
0 & 0 & 0 & 0 & M & 0 & 0 & \cdots & & & & & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & M & 0 & \cdots & & & & & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & M & \cdots & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & M & 0 & 0 & 0 & 0 & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & 0 & M & 0 & 0 & 0 & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & 0 & 0 & M & 0 & 0 & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & 0 & 0 & 0 & M & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \cdot & \cdots & 0 & 0 & 0 & 0 & M & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \cdot & \cdots & 0 & 0 & 0 & 0 & 0 & M_{48}
\end{bmatrix}
\begin{bmatrix} \ddot{u}_0 \\ \ddot{u}_1 \\ \ddot{u}_2 \\ \ddot{u}_3 \\ \ddot{u}_4 \\ \ddot{u}_5 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \ddot{u}_{47} \\ \ddot{u}_{48} \end{bmatrix}
$$

Since $\ddot{u}_0 = \ddot{u}_{-1} = 0$ the reduced mass matrix is

$$
\begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ F_{47} \\ F_{48} \end{bmatrix}
=
\begin{bmatrix}
M & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & & & & & 0 & 0 \\
0 & M & 0 & 0 & 0 & 0 & 0 & \cdots & & & & & 0 & 0 \\
0 & 0 & M & 0 & 0 & 0 & 0 & \cdots & & & & & 0 & 0 \\
0 & 0 & 0 & M & 0 & 0 & 0 & \cdots & & & & & 0 & 0 \\
0 & 0 & 0 & 0 & M & 0 & 0 & \cdots & & & & & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & M & 0 & \cdots & & & & & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & M & \cdots & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & M_{ben} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & M & 0 & 0 & 0 & 0 & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & 0 & M & 0 & 0 & 0 & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & 0 & 0 & M & 0 & 0 & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & 0 & 0 & 0 & M & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \cdot & \cdots & 0 & 0 & 0 & 0 & M & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \cdot & \cdots & 0 & 0 & 0 & 0 & 0 & M_{48}
\end{bmatrix}
\begin{bmatrix} \ddot{u}_1 \\ \ddot{u}_2 \\ \ddot{u}_3 \\ \ddot{u}_4 \\ \ddot{u}_5 \\ \ddot{u}_6 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \ddot{u}_{47} \\ \ddot{u}_{48} \end{bmatrix}
$$

The polar moment of inertia matrix for the entire building is assembled with the use of Mathlab. The matrix is 49 x 49 before reduction.

$$
\begin{bmatrix} M_0 \\ M_1 \\ M_2 \\ M_3 \\ M_4 \\ M_5 \\ \cdot \\ \vdots \\ \cdot \\ M_{47} \\ M_{48} \end{bmatrix}
=
\begin{bmatrix}
I_p & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & \cdot & \cdot & \cdot & \cdot & 0 & 0 \\
0 & I_p & 0 & 0 & 0 & 0 & 0 & \cdots & \cdot & \cdot & \cdot & \cdot & 0 & 0 \\
0 & 0 & I_p & 0 & 0 & 0 & 0 & \cdots & \cdot & \cdot & \cdot & \cdot & 0 & 0 \\
0 & 0 & 0 & I_p & 0 & 0 & 0 & \cdots & \cdot & \cdot & \cdot & \cdot & 0 & 0 \\
0 & 0 & 0 & 0 & I_p & 0 & 0 & \cdots & \cdot & \cdot & \cdot & \cdot & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & I_p & 0 & \cdots & \cdot & \cdot & \cdot & \cdot & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & I_p & \cdots & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & I_p & 0 & 0 & 0 & 0 & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & 0 & I_p & 0 & 0 & 0 & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & 0 & 0 & I_p & 0 & 0 & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & 0 & 0 & 0 & I_p & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \cdot & \cdots & 0 & 0 & 0 & 0 & I_p & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \cdot & \cdots & 0 & 0 & 0 & 0 & 0 & I_{p48}
\end{bmatrix}
\begin{bmatrix} \ddot{\psi}_0 \\ \ddot{\psi}_1 \\ \ddot{\psi}_2 \\ \ddot{\psi}_3 \\ \ddot{\psi}_4 \\ \ddot{\psi}_5 \\ \cdot \\ \vdots \\ \cdot \\ \ddot{\psi}_{47} \\ \ddot{\psi}_{48} \end{bmatrix}
$$

Since $\ddot{\psi}_0 = \ddot{\psi}_{-1} = 0$ the reduced matrix is

$$
\begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ M_4 \\ M_5 \\ M_6 \\ \cdot \\ \vdots \\ \cdot \\ M_{47} \\ M_{48} \end{bmatrix}
=
\begin{bmatrix}
I_p & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & \cdot & \cdot & \cdot & \cdot & 0 & 0 \\
0 & I_p & 0 & 0 & 0 & 0 & 0 & \cdots & \cdot & \cdot & \cdot & \cdot & 0 & 0 \\
0 & 0 & I_p & 0 & 0 & 0 & 0 & \cdots & \cdot & \cdot & \cdot & \cdot & 0 & 0 \\
0 & 0 & 0 & I_p & 0 & 0 & 0 & \cdots & \cdot & \cdot & \cdot & \cdot & 0 & 0 \\
0 & 0 & 0 & 0 & I_p & 0 & 0 & \cdots & \cdot & \cdot & \cdot & \cdot & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & I_p & 0 & \cdots & \cdot & \cdot & \cdot & \cdot & 0 & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & M_{tor} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & I_p & 0 & 0 & 0 & 0 & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & 0 & I_p & 0 & 0 & 0 & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & 0 & 0 & I_p & 0 & 0 & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & 0 & 0 & 0 & I_p & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \cdot & \cdots & 0 & 0 & 0 & 0 & I_p & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \cdot & \cdots & 0 & 0 & 0 & 0 & 0 & I_{p48}
\end{bmatrix}
\begin{bmatrix} \ddot{\psi}_1 \\ \ddot{\psi}_2 \\ \ddot{\psi}_3 \\ \ddot{\psi}_4 \\ \ddot{\psi}_5 \\ \ddot{\psi}_6 \\ \cdot \\ \vdots \\ \cdot \\ \ddot{\psi}_{47} \\ \ddot{\psi}_{48} \end{bmatrix}
$$

4.8. Damping matrix

The damping matrix is determined with the use of the Raleigh damping method [2]. The Raleigh damping method or proportional damping method implicates that the damping matrix is proportional to the mass and stiffness matrix. The Raleigh damping method is used because it is nearly impossible to determine the damping of each individual mode of a structure.

The mass matrix for bending is denoted as M_{ben} .

The polar moment of inertia matrix for torsion is denoted as M_{tor} .

The stiffness matrix for bending is denoted as K_{ben} matrix.

The stiffness matrix for torsion is denoted as K_{tor} matrix.

The general equations for proportional damping for bending and torsion are:

$$C_{ben} = a_{0ben} M_{ben} + a_{1ben} K_{ben}$$
$$C_{tor} = a_{0tor} M_{tor} + a_{1tor} K_{tor}$$

From here on the subscripts M_{ben} and M_{tor} will not be added to matrices or coefficients pertaining to bending or torsion.

The proportional damping constants are [2]:

$$a_0 = \frac{2\omega_1\omega_2\left(\varsigma_1\omega_2 - \varsigma_2\omega_1\right)}{\omega_2^{\ 2} - \omega_1^{\ 2}} \quad \left[\frac{rad}{s}\right]$$
$$a_1 = \frac{2\left(\varsigma_2\omega_2 - \varsigma_1\omega_1\right)}{\omega_2^{\ 2} - \omega_1^{\ 2}} \quad \left[\frac{s}{rad}\right]$$

The proof is given below:

By using modal analysis we can prove the derivation of the constants of proportional damping. The eigenmatrix will be noted as the matrix E and $\hat{u}_i$ as eigenvector. [2]

$$\mathrm{E} = \sum_{i=1}^{n} \hat{u}_i$$

The modal stiffness matrix and modal mass matrix are now introduced, both matrices are diagonal by definition because of the orthogonality conditions.

$$\mathrm{K}^* = \mathrm{E}^T \mathrm{KE}$$
$$\mathrm{M}^* = \mathrm{E}^T \mathrm{ME}$$

The term modal damping matrix will also be introduced now. The modal damping matrix will also be diagonal and gives the following result:

$$C^* = \mathrm{E}^T C\mathrm{E} = a_0\mathrm{M}^* + a_1\mathrm{K}^* = a_0\mathrm{E}^T\mathrm{ME} + a_1\mathrm{E}^T\mathrm{KE}$$

A modal damping ratio is assumed, which entails that every mode *i* has its own damping ratio ς_i . (This is a fully decoupled equation)

$$\left[2\varsigma_i\omega_i\right] = \mathrm{M}^{*-1}C^*$$

Making use of two equations above results in:

$$\left[2\varsigma_i\omega_i\right] = \mathrm{M}^{*-1}\left[a_0\mathrm{E}^T\mathrm{ME} + a_1\mathrm{K}^*\right] = a_0\mathrm{I} + a_1\mathrm{M}^{*-1}\mathrm{K}^* = a_0\mathrm{I} + a_1\Omega^2$$

In which Ω represents the eigenfrequencies of the undamped system on the diagonal, consequently the damping ratios can be written as:

$$\varsigma_i = \frac{a_0}{2\omega_i} + \frac{a_1}{2}\omega_i$$

This derivation above has proven this. The constants of proportional damping a_0 and a_1 can be determined by the rearranging of the formula:

$$\frac{1}{2}\begin{bmatrix} 1/\omega_i & \omega_i \\ 1/\omega_j & \omega_j \end{bmatrix}\begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \varsigma_i \\ \varsigma_j \end{bmatrix} \rightarrow \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = 2\begin{bmatrix} 1/\omega_i & \omega_i \\ 1/\omega_j & \omega_j \end{bmatrix}^{-1}\begin{bmatrix} \varsigma_i \\ \varsigma_j \end{bmatrix}$$

The damping ratios of mode **1** and mode **2** and also mode **1** and mode **10** will be determined later in this paragraph. The damping ratios of mode **1** and mode **2** are chosen to be used in this calculation and from hereon after this chapter. The chosen 2 modes are dominant for wind on the building.

The damping ratio for concrete we can calculate as: [17]

$$\varsigma = 0.003 + 0.0076 f_e + C_e \frac{\hat{u}_h}{h} \quad (f_e \text{ in Hz})$$

With

$$C_e = 10^{(0.3\sqrt{d})}$$

In which:

ς	damping ratio for concrete	[-]
f_e	natural frequency of the building structure	[Hz]
C_e	energy dissipation factor	[-]
$\hat{u}_h$	peak displacement at the top of the building	[m]
h	height of the building	[m]
d	depth of the building	[m]

For the values of b = 26.4 m; $f_e = \omega_e/2\pi = 1{,}409/2\pi = 0.224$ Hz ;

(see 5.4.1)

$\hat{u}_h$ = 0.1784 m ; h = 144 m

We attain the values for $C_e = 10^{(0.3\sqrt{15.4})} = 11.8 \approx 12$

$$\varsigma = 0.003{+}0.0076{*}0.224{+}12{*}\frac{0.1784}{144}{=}0.019 \ (1.9\%)$$

The damping ratio can be calculated from:[13]

$$\varsigma = 0.01 f_e + C_e \frac{\hat{u}_h}{h}$$

$$\varsigma = 0.01{*}0.224{+}12{*}\frac{0.1784}{144}{=}0.017 \ (1.7\%)$$

In [3] a damping ratio is proposed of $\varsigma = 0.02$ for a concrete structure but this value is only valid in the Ultimate limit state. In the serviceability limit state there will be less cracking of concrete. For this situation a conservative choice for the damping ratio is $\varsigma = 0.01$.

For the Juffertoren the damping ratio is selected for mode **1** $\varsigma_1 = 0.01$ and for mode **2** $\varsigma_{10} = 0.01$.

The Raleigh damping coefficients for bending are a_{0ben} = 0.0243 rad/s and a_{1ben} = 0.0002 s/rad for mode 1 and mode 2. The Raleigh damping coefficients for bending are a_{0ben} = 0.0281 rad/s and a_{1ben} = 0.0001 s/rad for mode 1 and mode 10. When interpreting the values of the Raleigh damping constants for bending, one can see that the damping acts totally on the Modal mass matrix (low frequencies). $\omega_{e\_ben\_matlab}$ = 1.409 rad/s

The Raleigh damping coefficients for torsion are a_{0tor} = 0.013 rad/s and a_{1tor} = 0.0058 s/rad for mode 1 and mode 2. The Raleigh damping coefficients for torsion are a_{0tor} = 0.0165 rad/s and $a_{1tor} = 0.0012$ s/rad for mode 1 and mode 10. When interpreting the values of the Raleigh damping constants for torsion, one can see that the damping acts totally on the Modal mass matrix (low frequencies). $\omega_{e\_tor\_matlab} = 0.502$ rad/s

[2]

---

[2] When reviewing the Mathlab code of Breen [5] the Raleigh damping coefficients for bending were a_0 = 0.0256 rad/s and a_1 = 0.0018 s/rad . In the newly written code the corrected Raleigh damping coefficients for bending were a_0 = 0.0243 rad/s and a_1 = 0.0020 s/rad . The reason for this is that Breen included the foundation stiffness in the modal. The K[2,2] in this study is $6\frac{EI}{l^3}$ and in Breen is $5\frac{EI}{l^3}$. ω_{e_Breen}= 1,48 rad/s ω_{e_matlab} = 1,41 rad/s My assumption is that the actual displacement, velocity and acceleration would have been smaller than what Breen displayed, when taking these aspects into consideration. Secondly, even though Breen's model also takes the foundation stiffness into account, this model does not, which is a conservative approach. It is rigidly connected at the foundation. Breen also includes the foundation stiffness.

5. Dynamic Simulations with Simulink

In this chapter the dynamic model for bending and torsion is determined and validated in Simulink.

5.1. Simulink and equations of motion

Simulink is a sub program of Matlab in which graphical programming can be done for modeling, simulating, and analyzing dynamic systems.

The equation of motion for bending and torsion can be written as: [12]

$$\mathrm{M}_{ben}\ddot{u} + C_{ben}\dot{u} + \mathrm{K}_{ben}u = F$$
$$\mathrm{M}_{tor}\ddot{\psi} + C_{tor}\dot{\psi} + \mathrm{K}_{tor}\psi = \mathrm{M}$$

The number of elements for this building model is 48, with at each node 2 degrees of freedom. The bending and torsion equations of motions are loaded in separate matrices in Simulink. The solution of the eigenmatrices required two separate matrices for bending and torsion. The M, C and K matrices for bending and torsion are (48 x 48) matrices and u, ψ and F, M are (48 x1) vectors. State space formulation is introduced for the bending and torsion equation of motion because Simulink cannot process a second order differential equation.

5.2. State space formulation for bending

As explained before the equation of motion for bending of the building can be written as:

$$\mathrm{M}_{ben}\ddot{u} + C_{ben}\dot{u} + \mathrm{K}_{ben}u = F$$

Further on in Chapter 6.6, F is dicretised as a matrix. The bending state space formulation is done by introducing a vector:

$$x = \begin{bmatrix} u \\ \dot{u} \end{bmatrix}$$

Substitution of the equation above into the bending equation of motion results in:

$$\begin{bmatrix} I & 0 \\ 0 & M_{ben} \end{bmatrix}\left[\dot{X}\right]+\begin{bmatrix} 0 & -I \\ K_{ben} & C_{ben} \end{bmatrix}\left[X\right]=\begin{bmatrix} 0 \\ F \end{bmatrix}$$

Multiplication and rearranging with $\begin{bmatrix} I & 0 \\ 0 & M_{ben}^{-1} \end{bmatrix}$ results in:

$$\left[\dot{X}\right]=\begin{bmatrix} 0 & I \\ -M_{ben}^{-1}K_{ben} & -M_{ben}^{-1}C_{ben} \end{bmatrix}\left[X\right]+\begin{bmatrix} I & 0 \\ 0 & M_{ben}^{-1} \end{bmatrix}\begin{bmatrix} 0 \\ F \end{bmatrix}$$

The state space equation in Simulink can be written as:

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

In which u is the input and y is the output. The output of interest is the displacement u and the velocity $\dot{u}$, because of this y will be defined as:

$$\left[y\right]=\left[X\right]$$

Rearranging of two equations above results in:

$$\left[\dot{X}\right]=\begin{bmatrix} 0 & I \\ -M_{ben}^{-1}K_{ben} & -M_{ben}^{-1}C_{ben} \end{bmatrix}\left[X\right]+\begin{bmatrix} 0 \\ M_{ben}^{-1} \end{bmatrix}\left[F\right]$$

$$\left[y\right]=\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}\left[X\right]+\begin{bmatrix} 0 \\ 0 \end{bmatrix}\left[F\right]$$

The derived matrices above can be placed into Simulink as written below:

$$A = \begin{bmatrix} 0 & \mathrm{I} \\ -\mathrm{M}_{ben}^{-1} K_{ben} & -\mathrm{M}_{ben}^{-1} C_{ben} \end{bmatrix} \quad (96x96)$$

$$B = \begin{bmatrix} 0 \\ \mathrm{M}_{ben}^{-1} \end{bmatrix} \quad (96x48)$$

$$C = \begin{bmatrix} \mathrm{I} & 0 \\ 0 & \mathrm{I} \end{bmatrix} \quad (96x96)$$

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (96x48)$$

$$u = \begin{bmatrix} F \end{bmatrix} \quad (48xti)\ (ti = \text{number of timesteps})$$

The displacement vector u and the damping matrix C are not identical to the matrix C and the vector u in Simulink.

The Simulink model can be viewed in **Figure 5.1**. The different contents of the model will be explained in the text below. While Simulink runs, the state is recalculated for every instant in time.

The contents of the blocks in the Simulink bending model will be given from left to right. The first block contains the nodal forces as defined by 3 equations on previous page. The output of this block is the nodal forces for every time step of the 48 nodes. In the second block a transformation occurs which transforms the elements of the first block into a vector. The third block contains the matrices defined in the equations on previous page. The inputs of this block are the nodal forces and the outputs of this block are the displacements and velocities of the 48 nodes. The next 2 blocks which are selecting only the displacement and velocity of the top of the building as output from the total vector with all degrees of freedom. The 3 output blocks at the right store the output in combination with the time of which Matlab can make plots. The block between output velocity top and acceleration top differentiates the velocity in time with outcome the acceleration of the top.

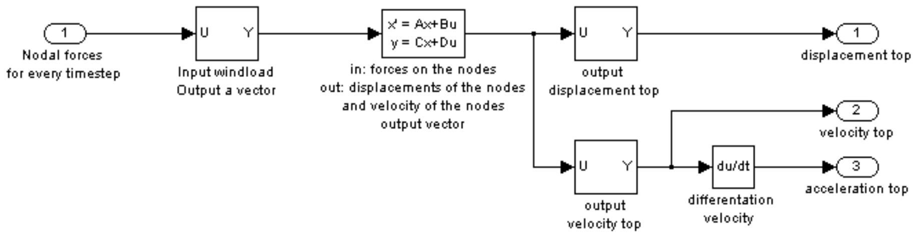


Figure 5.1: Model for bending of the MDF system in Simulink.

5.3. Verification of static bending behavior

The mass matrix has elements only on the diagonal, therfore verification is not necessary. The stiffness matrix can be verified by placing equal static loads at all the nodes of the structure. If a static load of $q = 62.1$ kN/m (arbitrarily chosen) is placed on the structure, the total static displacement of the top of the building will be:

$$u_{static;top} = \frac{qL^4}{8EI} = 0.1666 \text{ m}$$

The stiffness matrix can be validated in Matlab by using the formula:

$$u = K^{-1}F$$

with:

K The stiffness matrix is derived in Appendix E according to section 4.7

F The force vector is a (48x1) matrix. A force of 62.1 kN per meter height multiplied by the height of one story is applied on each node of the structure. The load applied on each node is F_i = 3*62.1= 186.3 kN. The load on the top node is half the value of the previous ones.

Evaluation of the equation above in Matlab results in a static deflection at the top of 0.1784 m, the deflection of the structure can be viewed in (**Figure 5.2**). The outcome verifies that the stiffness matrix is well defined. The absolute difference is 0.1784 – 0.1666 = 0.0118 m
The difference in percentage from the actual deflection is 0.0118 / 0.1666 *100 % = 7 %

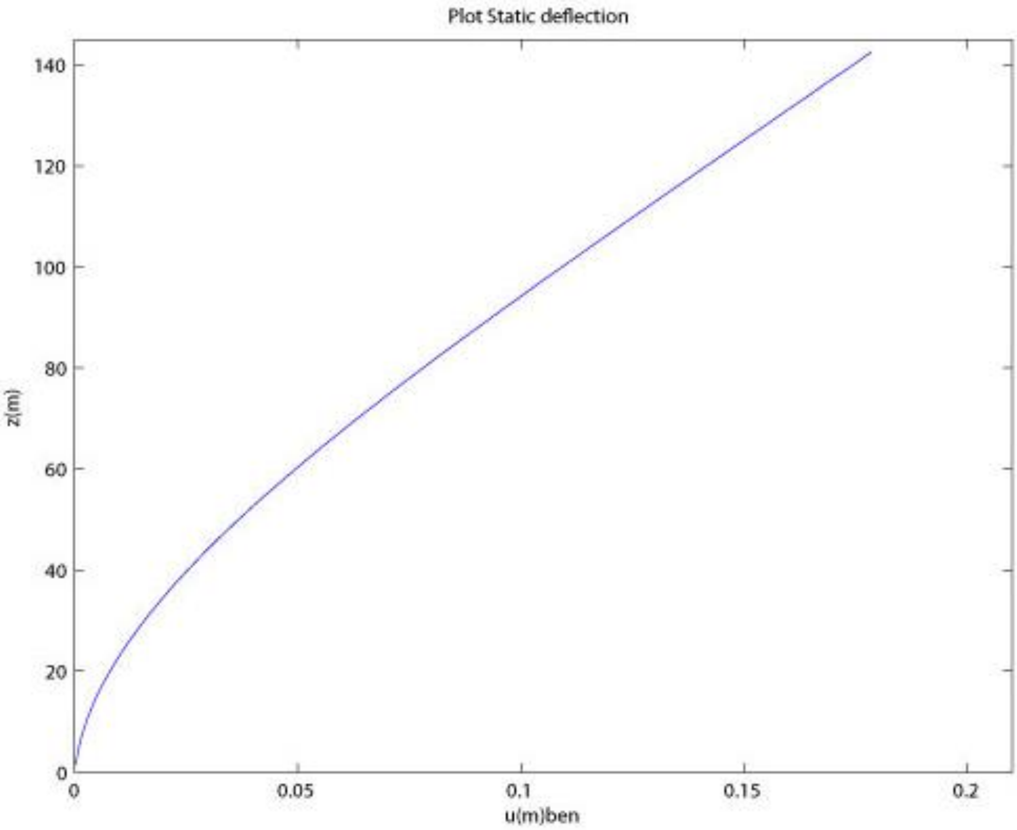


Figure 5.2: Static deflection of the building (bending)

5.4. Verification of natural frequency

5.4.1. Computation of Natural Frequencies

The damping matrix and dynamic behavior can be verified by applying step function to the system in the Simulink model. When the same force is applied as in the static displacement control, this will result in a vibration equal to the defection of the static deflection round the origin: (**Figure 5.2**)

The natural frequencies of the equation of motion can be determined by using the homogeneous part of the equation of motion [2] ;

$$M\ddot{u} + C\dot{u} + Ku = F$$

With the use of modal analysis this can be rewritten as:

$$E^T ME\ddot{u} + E^T CE\dot{u} + E^T KEu = M^*\ddot{u} + C^*\dot{u} + K^*u = 0$$

Modal analysis implies that this differential equation above is a totally decoupled system. Multiplying this equation by $\left(M^*\right)^{-1}$ and substituting the relation $\left(M^*\right)^{-1} K^* = \Omega^2$ we conclude that:

$$\ddot{u} + \left(M^*\right)^{-1} C^*\dot{u} + \Omega^2 u = 0$$

In which Ω is a diagonal matrix with $\omega_i \quad (i = 1,2,...,48)$. And substitution of $\left(M^*\right)^{-1} C^* = 2\zeta_i\omega_i$ in this decoupled system can be written as follows: [2]

$$\ddot{u}_i + 2\varsigma_i\omega_i\dot{u}_i + \omega_i^2 u_i = 0 \quad (i = 1,2,...,48)$$

The homogeneous solution of the equation of motion is [2] [11]

$$u_i\left(t\right) = A_i e^{(-\varsigma_i\omega_i t)} \sin\left(\omega_i t\sqrt{1-\varsigma_i^{\,2}} + \varphi_i\right) \quad (i = 1,2,...,48)$$

The solution for the first mode or natural frequency is:

$$u_1(t) = A_1 e^{(-\varsigma_1 \omega_1 t)} \sin\left(\omega_1 t \sqrt{1-\varsigma_1^{\,2}} + \varphi_1\right)$$

$$\cos(x) = \sin(-x + \frac{\pi}{2})$$

With $\left(M^*\right)^{-1} K^* = \Omega^2$, Matlab can determine ω_1 (the natural frequency). With $\varsigma_1 = 0.01$ and $\omega_1 = 1.41\ rad\ /\ s$ the eigenfrequency of the damped system can be determined:

$$\omega_e = \omega_1 \sqrt{1-\varsigma_1^{\,2}} = 1.41\sqrt{1-0.01^2} = 1.40 \text{ rad/s}$$

Natural frequencies Bending [rad/s]							
1.409	8.821	24.666	48.236	79.521	118.384	164.668	218.179
278.698	345.974	419.728	499.654	585.422	676.673	773.029	874.090
979.433	1088.619	1201.193	1316.684	1434.610	1554.476	1675.781	1798.016
1920.668	2043.222	2165.164	2285.980	2405.163	2522.210	2636.629	2747.938
2855.667	2959.362	3058.586	3152.919	3241.963	3325.342	3402.704	3473.722
3538.095	3595.552	3645.849	3688.773	3724.144	3751.811	3771.657	3783.598

Table 1: Natural frequencies bending

5.4.2. Manual calculation of the bending natural frequency

A hand calculation is performed to check that the eigenfrequency calculated by Simulink is correct. The first natural frequency can be calculated [2] p.63

$$\omega_n = C\sqrt{\left(EI\ /\ \rho A l^4\right)}$$

([2], p.80)

which can be derived from the Raleigh quotient.

$C = 3.52$ modulus of elasticity $E = 3.0E^{10}$ N/m²

Second moment of inertia $I = 667$ m^4 mean mass density $\rho = 675.54$ kg/m^2

Area building $A = 405.02$ m^2 height building $l = 144$ m

$$\omega_n = C\sqrt{\left(EI / \rho A l^4\right)} = 3.52\sqrt{\left(\left(3.0E^{10}*667\right)/\left(675.5366*405.02*144^4\right)\right)} = 1.451 \text{ rad/s}$$

This has a reasonable agreement with the first natural frequency from Mathlab. The difference is due to approximation when using the Raleigh quotient.

5.5. Response to a sinus load

5.5.1. Verification of the MDF system for bending

To further verify the Simulink model we have to prove that the displacement of the single degree of freedom (SDF) and multi degree of freedom (MDF) are the same. In Simulink we make a diagram to test the response of the multi degree of freedom system and see if the displacement corresponds to the displacement of a single degree of freedom system. On the MDF system for bending, a sinusoidal load will be placed on the top node. $\left(F = 1000000 * \sin\left(1 * t\right)\right)$

The responses of both systems are almost the same. (Compare **Figure 5.5** to **Figure 5.3**) The difference can be accounted for by the higher modes of the MDF system which are neglected in the SDF system.

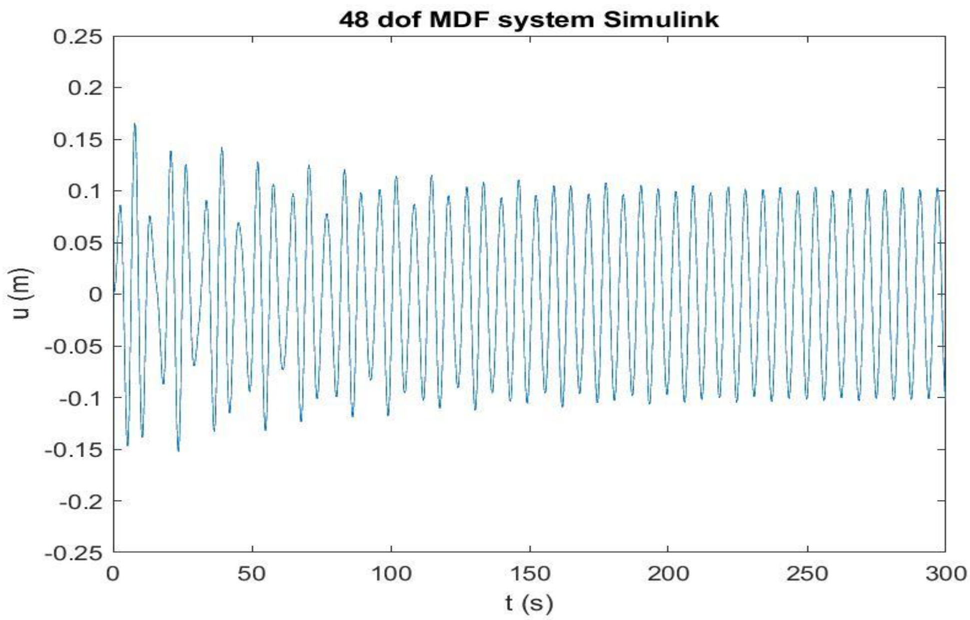


Figure 5.3: Response to sinusoidal load of MDF system (bending)

5.5.2. Manual calculation of the dynamic bending behaviour

The dynamic displacement of the structure will mosty be due to the first mode of vibration, if the forcing frequency is close to the natural frequency. The contribution of the higher modes for the displacements are relatively small and can be neglected in this analysis. By knowing the damping ratio of the first eigenmode (Section 4.8) and the natural frequency, the behaviour of the multi degree of freedom (MDF) system can be approached by the behaviour of a single degree of freedom (SDF) system. (**Figure 5.4**) On the SDF system for bending, a sinusoidal load will be placed on the equivalent system.

$$\left(F = 1000000 * \sin(1 * t)\right)$$

$$m_{eq} = 0.24 * rho * A * l = 0.24 * 675.54 * 405.02 * 144 = 9455865 \text{ kg}$$ [2] pp.79-80

$$k_{eq} = \frac{3 * EI}{L^3} = \frac{3 * 2.003 * E^{13}}{144^3} = 2.012 * E^7 \frac{N}{M}$$

$$c_{eq} = \zeta * c_{kr} = \zeta * 2 * \sqrt{\frac{k_{eq}}{m_{eq}}} = 0.01 * 2 * \sqrt{\frac{2.012 * E^7}{9455865}} = 2.759 * E^5 \left(\frac{N}{kg * M}\right)^{0.5}$$

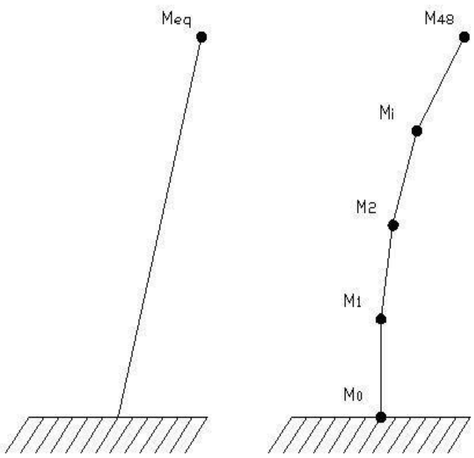


Figure 5.4: Static deflection of building.

Schematizing the structure as an SDF system with the displacement at the top as the only degree of freedom with $u(0) = 0$ m and $\dot{u}(0) = 0$ m/s gives [26] pp. 72-73.

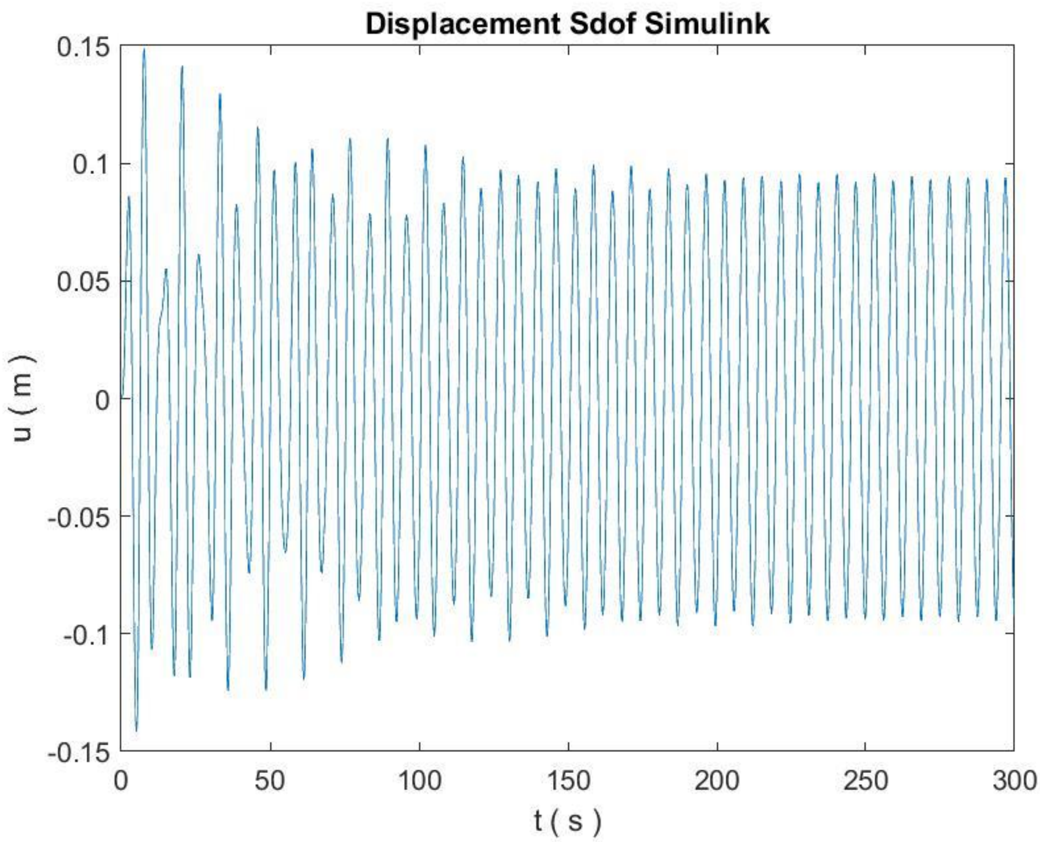


Figure 5.5: Response to sinusoidal load of analytical SDF system (bending)

5.6. State space formulation for torsion

The equation of motion for torsion of the building can be written as:

$$M_{tor}\ddot{\psi} + C_{tor}\dot{\psi} + K_{tor}\psi = M$$

Further on in Chapter 6.7 M is dicretised as a matrix. The torsion state space formulation is done by introducing a vector:

$$X = \begin{bmatrix} \psi \\ \dot{\psi} \end{bmatrix}$$

Substitution of the equation above into the torsion equation of motion results in:

$$\begin{bmatrix} I & 0 \\ 0 & M_{tor} \end{bmatrix}\left[\dot{X}\right]+\begin{bmatrix} 0 & -I \\ K_{tor} & C_{tor} \end{bmatrix}\left[X\right]=\begin{bmatrix} 0 \\ M \end{bmatrix}$$

Multiplication and rearranging with $\begin{bmatrix} I & 0 \\ 0 & M_{tor}^{-1} \end{bmatrix}$ results in:

$$\left[\dot{X}\right]=\begin{bmatrix} 0 & I \\ -M_{tor}^{-1}K_{tor} & -M_{tor}^{-1}C_{tor} \end{bmatrix}\left[X\right]+\begin{bmatrix} I & 0 \\ 0 & M_{tor}^{-1} \end{bmatrix}\begin{bmatrix} 0 \\ M \end{bmatrix}$$

The state space equation in Simulink can be written as:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

In which u is the input and y is the output. The output of interest is the displacement u and the velocity $\dot{u}$, because of this y will be defined as:

$$[y]=[X]$$

Rearranging of two equations above results in:

$$\begin{aligned} \left[\dot{X}\right] &= \begin{bmatrix} 0 & I \\ -M_{tor}^{-1}K_{tor} & -M_{tor}^{-1}C_{tor} \end{bmatrix}[X]+\begin{bmatrix} 0 \\ M_{tor}^{-1} \end{bmatrix}[M] \\ [y] &= \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}[X]+\begin{bmatrix} 0 \\ 0 \end{bmatrix}[M] \end{aligned}$$

The derived matrices above can be placed into Simulink as written below:

$$A=\begin{bmatrix} 0 & \mathrm{I} \\ -\mathrm{M}_{tor}^{-1}K_{tor} & -\mathrm{M}_{tor}^{-1}C_{tor} \end{bmatrix} \qquad (96\text{x}96)$$

$$B=\begin{bmatrix} 0 \\ \mathrm{M}_{tor}^{-1} \end{bmatrix} \qquad (96\text{x}48)$$

$$C=\begin{bmatrix} \mathrm{I} & 0 \\ 0 & \mathrm{I} \end{bmatrix} \qquad (96\text{x}96)$$

$$D=\begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad (96\text{x}48)$$

$$\mathrm{u}=[\mathrm{M}] \qquad (48\text{xti})\ (\text{ti = number of timesteps})$$

The displacement vector u and the damping matrix C are not identical to the matrix C and the vector u in Simulink. The Simulink model can be viewed in **Figure 5.6**.
The different contents of the model will be explained in the text below. While Simulink runs the state is recalculated for every time step.

The contents of the blocks in the Simulink torsional model will be given from left to right on the next page. The first block contains the nodal moments as defined in Section 6.7 The output of this block is the nodal moments for every time step of the 48 nodes. In the second block a transformation occurs, which transforms the elements of the first block into a vector. The third block contains the matrices, defined in the equations above and written on the previous page. The inputs of this block are the nodal moments and the outputs of this block are the angular rotation and angular velocity of the 48 nodes. The next 2 blocks only selects the angular displacement and angular velocity of the top of the building as output from the total vector with all degrees of freedom. The 3 output blocks at the right store the output in combination with the time of which Matlab can make plots. The block between output velocity top and acceleration top differentiates the velocity in time with outcome the acceleration of the top.

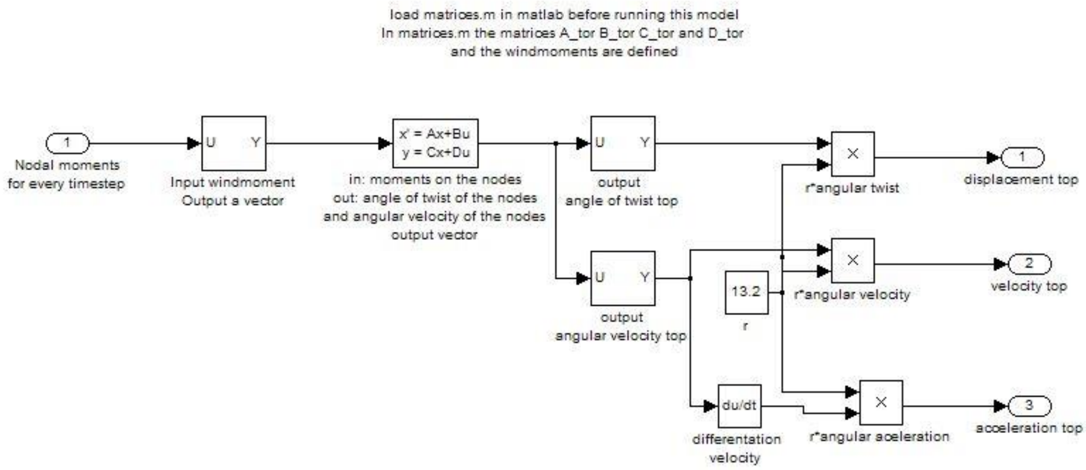


Figure 5.6: Model for torsion of the MDF system in Simulink

5.7. Verification of the static torsional behaviour

The mass matrix for torsion motion, as the bending matix, only has elements on the diagonal. Verification of this matrix is therefore not necessary.

The stiffness matrix can be verified by placing a moment on each node of the structure. The static angle of twist of the building has to be can be verified in Matlab by using the formula:

$$\psi = K_{tor}^{-1} M$$

with:

K_{tor} The stiffness matrix is derived in according to section 4.4. The moment vector M is a (48x1) matrix. Each element of this vector is 10 KNm. The torsion stiffness of the K_{tor} matrix is $\frac{GJ}{l} = \frac{3.592E^{10} Nm^2}{3m} = 1.197E^{10} Nm$

Executing the equation above in Matlab gives $\psi = 0.982E^{-3} rad$ at the top**.**

For the hand calculation, i the formula below will be used to determine the angle of twist at the top of the building. The angle of twist multiplied by the distance from the outer node to the shear center (23.4/2) results in an angular static deflection.

$$\psi(x) = \left(\frac{m_0 * x}{G * J}\right) * (l - 0.5 * x)$$

where:

$\psi(x)$	angle of twist at height x	$[rad]$
m_0	applied moment	$[Nm/m]$
x	distance from foundation to reference height	$[m]$
G	shear modulus	$[N/m^2]$
J	polar moment of inertia	$[m^4]$
l	height of the building	$[m]$

the formula with values:

$$\psi(144) = \left(\frac{10000/3 * 144}{3.592E^{10}}\right) * (144 - 0.5 * 144) = 0.962E^{-3} rad$$

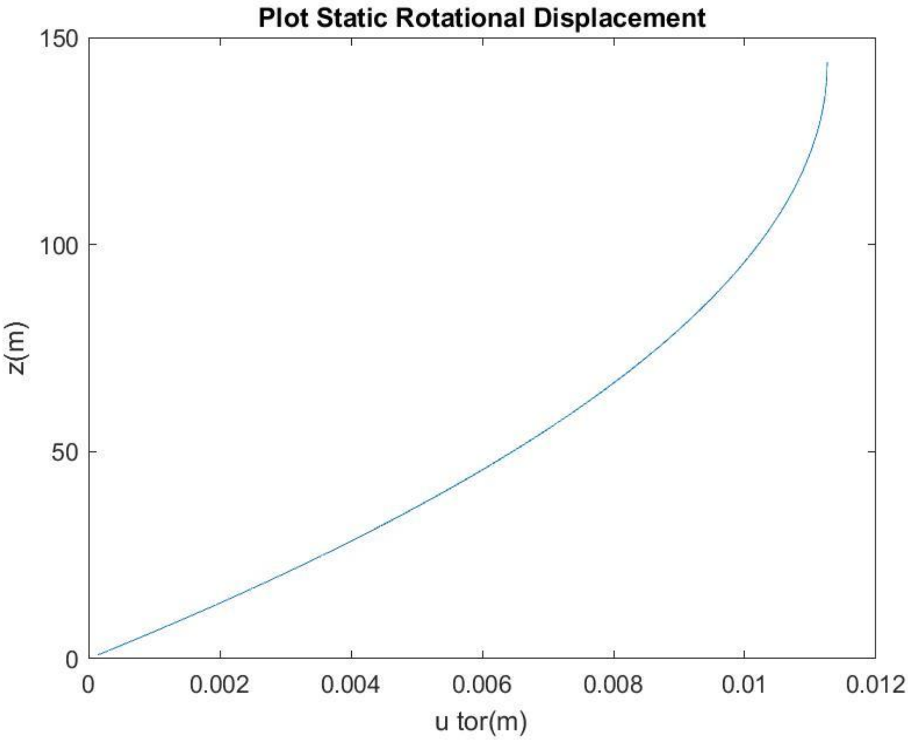


Figure 5.7: Static deflection of building. (torsion)

The outcome matlab and hand calculation above validates that the stiffness matrix is well defined.

5.8. Verification of the natural frequency

5.8.1. Verification of MDF system for torsion

The damping matrix and dynamic behavior can be verified by applying step function to the system in the Simulink model. When one uses the same moment as applied in the static angular control, this will result in a vibration equal to the defection of the static deflection around the origin (**Figure 5.7**)

The natural frequencies of the equation of motion can be determined by using the homogeneous part of the equation: [2]

$$M_{tor}\ddot{\psi} + C_{tor}\dot{\psi} + K_{tor}\psi = M$$

With the use of modal analysis this can be rewritten as:

$$E^T M_{tor} E\ddot{\psi} + E^T C_{tor} E\dot{\psi} + E^T K_{tor} E\psi = M^*_{tor}\ddot{\psi} + C^*_{tor}\dot{\psi} + K^*_{tor}\psi = 0$$

Modal analysis implies that this differential equation above is a totally decoupled system. Multiplying this equation by $\left(M^*_{tor}\right)^{-1}$ and substituting the relation $\left(M^*_{tor}\right)^{-1} K^*_{tor} = \Omega^2$ we conclude that:

$$\ddot{\psi} + \left(M^*_{tor}\right)^{-1} C^*_{tor}\dot{\psi} + \Omega^2\psi = 0$$

In which Ω is a diagonal matrix with ω_i $(i = 1, 2, ..., 48)$. And substitution of $\left(M^*_{tor}\right)^{-1} C^*_{tor} = 2\zeta_i\omega_i$ in this decoupled system can be written as follows: [2]

$$\ddot{\psi}_i + 2\varsigma_i\omega_i\dot{\psi}_i + \omega_i^2\psi_i = 0 \quad (i = 1, 2, ..., 48)$$

The homogeneous solution of the equation of motion is [2] [11]

$$\psi_i(t) = A_i e^{(-\varsigma_i\omega_i t)} \sin\left(\omega_i t\sqrt{1-\varsigma_i^2} + \varphi_i\right) \quad (i = 1, 2, ..., 48)$$

The solution for the first mode or natural frequency is:

$$u_1(t) = \left(A_1 e^{(-\varsigma_1\omega_1 t)} \sin\left(\omega_1 t\sqrt{1-\varsigma_1^2} + \varphi_1\right)\right) * l_{cross\,sec}$$

With $\left(M_{tor}^{*}\right)^{-1} K_{tor}^{*} = \Omega^2$, Matlab can determine ω_i (the natural frequency). With $\varsigma_1 = 0.01$ and $\omega_1 = 0.502$ rad/s the eigenfrequency of the damped system can be determined:

$$\omega_e = \omega_1 \sqrt{1-\varsigma_1{}^2} = 0.502\sqrt{1-0.01^2} = 0.502 \text{ rad/s}$$

Natural frequencies Torsion [rad/s]							
0.502	1.504	2.505	3.504	4.499	5.488	6.472	7.449
8.419	9.379	10.329	11.268	12.195	13.109	14.009	14.894
15.763	16.615	17.450	18.266	19.062	19.838	20.593	21.326
22.036	22.723	23.385	24.023	24.635	25.221	25.780	26.311
26.815	27.290	27.737	28.154	28.542	28.899	29.226	29.523
29.789	30.024	30.227	30.400	30.541	30.651	30.729	30.776

Table 2: Natural frequencies torsion

5.8.2. Manual calculation torsional natural frequency [2]

$$\omega_n = \left(\frac{(2*n-1)\pi}{2l}\right)\sqrt{\left(\frac{GJ}{I_p}\right)}$$

mode $n = 1$ shear modulus $G = 1.3043E^{10}$ N/m^2

polar moment of inertia $I_p = \dfrac{50521317}{3} = 16840439$ kgm

building height $l = 144$ m torsion constant $J = 2.7544$ m^4

$$\omega_n = \left(\frac{(2*n-1)\pi}{2l}\right)\sqrt{\left(\frac{GJ}{I_p}\right)} = \left(\frac{(2*1-1)\pi}{2*144}\right)\sqrt{\left(\frac{1.3043E^{10}*2.7544}{16840439}\right)} = 0.504 \text{ rad/s}$$

This has a reasonable agreement with the first natural frequency from Mathlab. The difference is due to approximation when using the Raleigh quotient. The Raleigh quotient is a upperbound calculation.

5.9. Response to a sinus load

5.9.1. Verification of MDF system for torsion

On the MDF system for torsion, a sinusoidal load will be placed on the top node. $\left(M = 1E7 * \sin\left(0.4 * t\right)\right)$

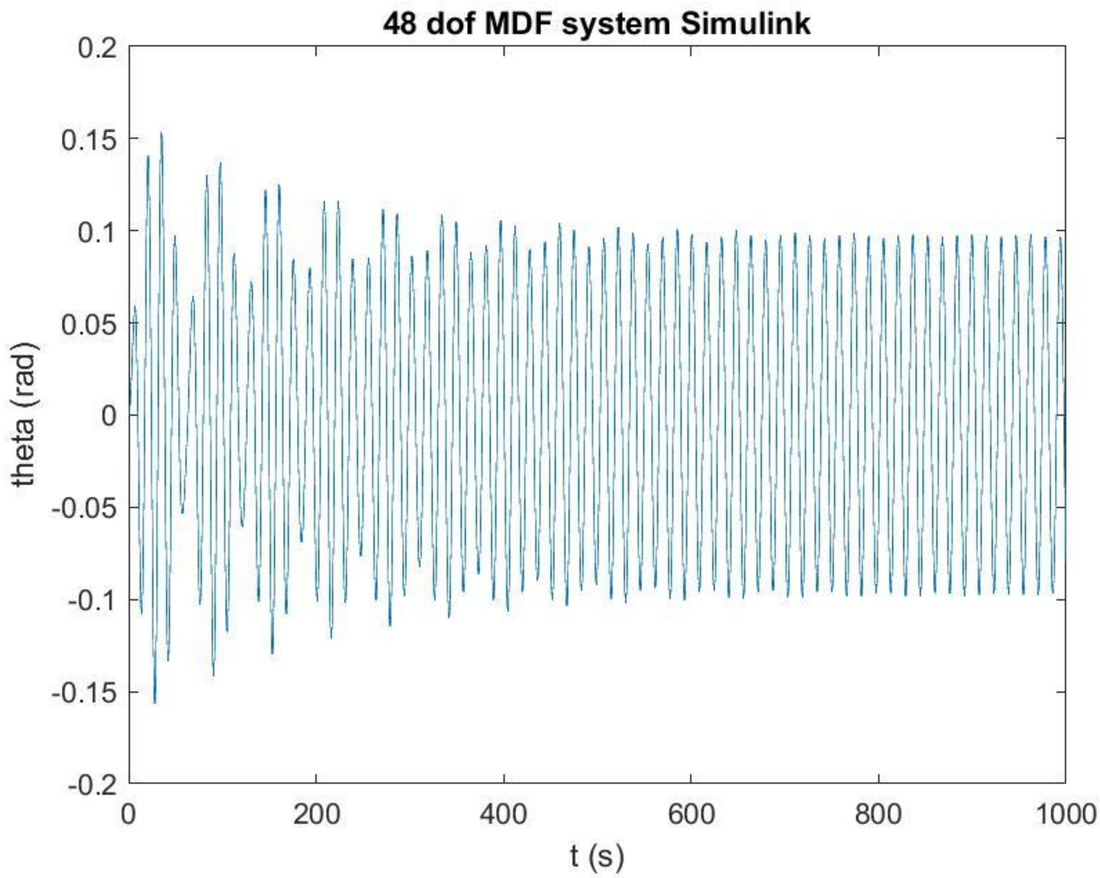


Figure 5.8: Angular Response to sinusoidal load of MDF system (torsion)

5.9.2. Manual calculation of the dynamic torsional behaviour

On the SDF system for torsion, a sinusoidal load will be placed on the equivalent system. $\left(M = 1E7 * \sin(0.4 * t)\right)$

$$\omega_n = \sqrt{\frac{k_{eq}}{m_{eq}}} = 0.502\,\frac{rad}{s}$$

$$k_{eq} = \frac{GJ}{L} = \frac{3.5926 * E^{10}}{144} = 2.4949 * E^{8}\,Nm$$

$$m_{eq} = \frac{k_{eq}}{\omega_n^2} = 9.951\text{E}^8\ \text{kgm}^2$$

$$c_{eq} = \zeta * c_{kr} = \zeta * 2 * \sqrt{\frac{k_{eq}}{m_{eq}}} = 0.01 * 2 * \sqrt{\frac{2.4949 * E^{8}}{9.951\text{E}^{8}}} = 9.9473 * E^{8}\left(\frac{N}{\text{kgm}}\right)^{0.5}$$

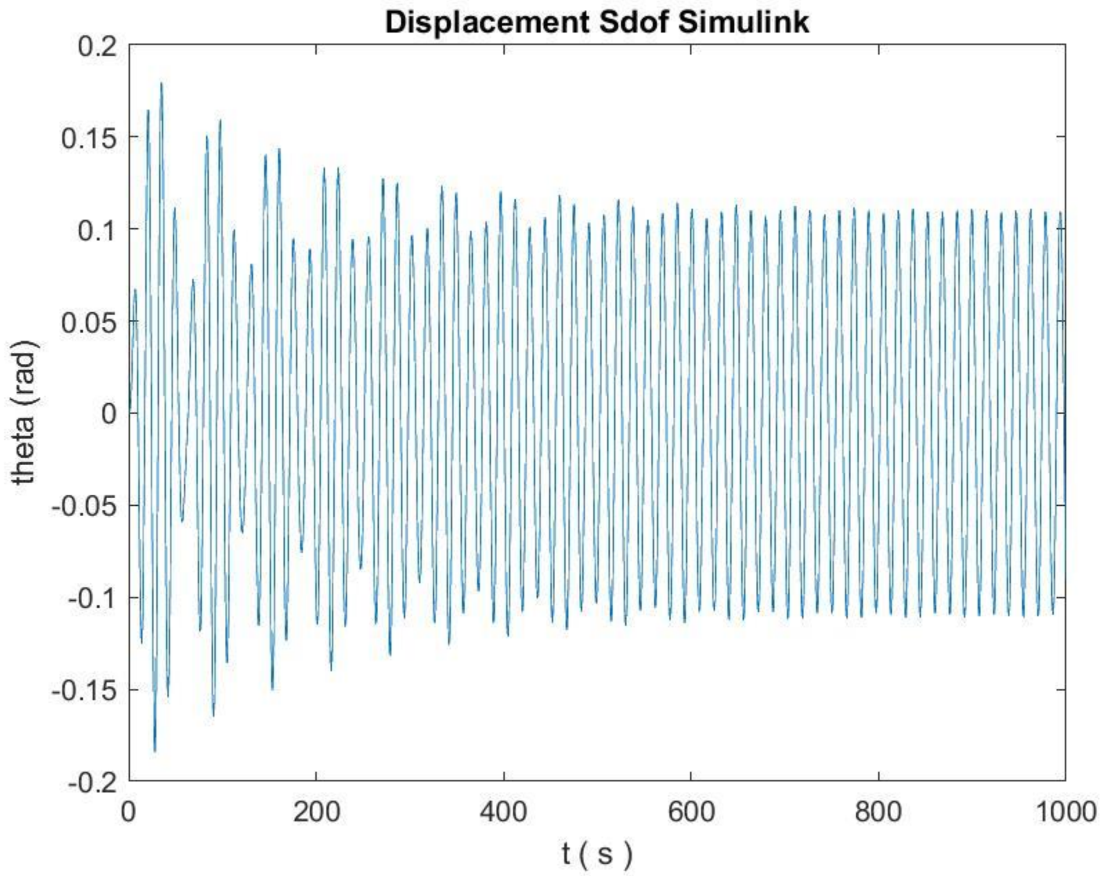


Figure 5.9: Angular response to sinusoidal load of analytical SDF system (torsion)

5.10. Simulink diagram for bending and torsion

The contents of the blocks in the Simulink bending and torsional model are the same as the bending model (section 5.2) and torsional model (section 5.6). The only difference is that the total acceleration due to bending and torsion is summated and stored in block 7.

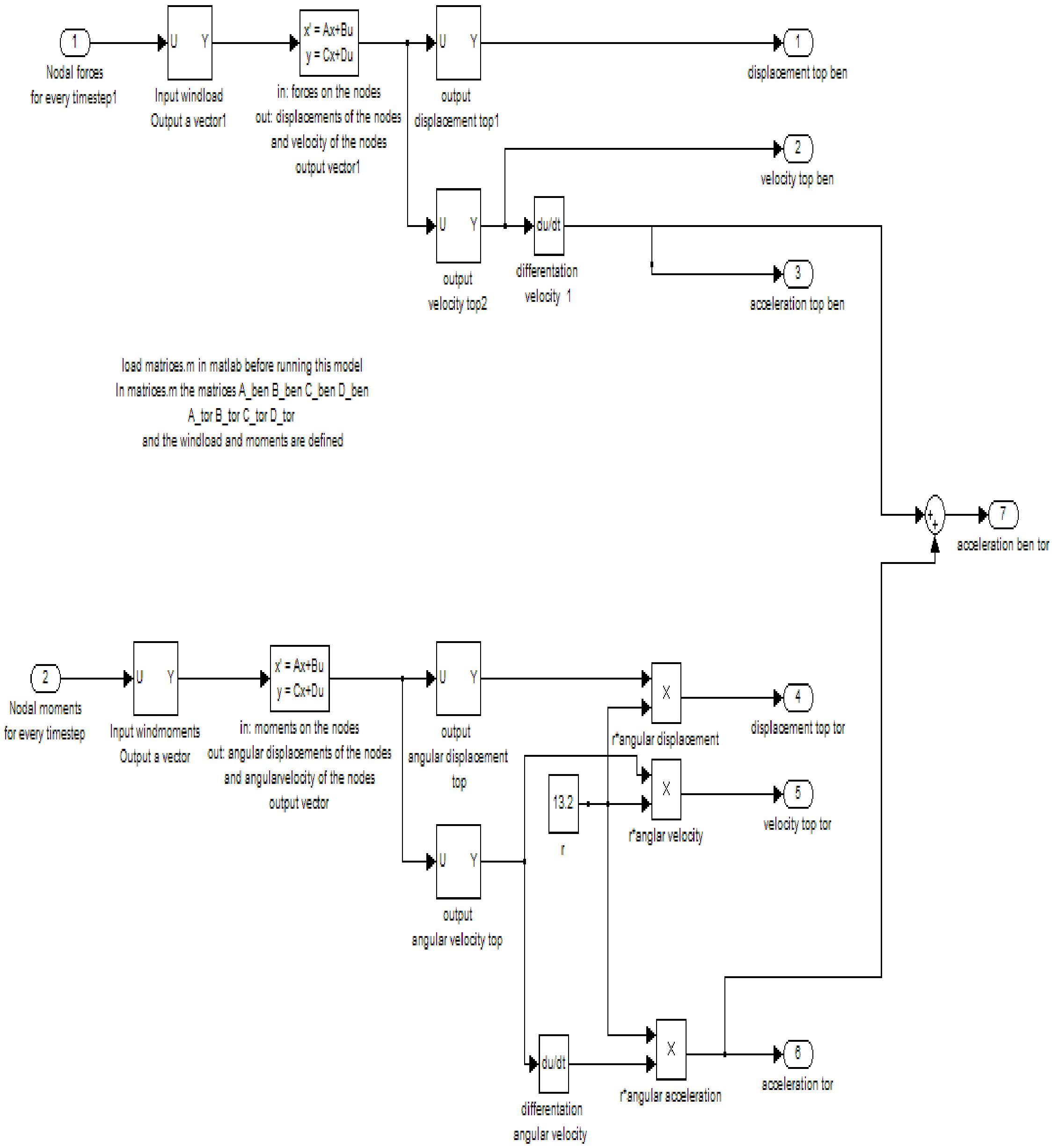


Figure 5.10: Model for bending and torsion of the MDF system in Simulink

6. Wind

In this chapter the mean and fluctuating wind velocity are determined in space and time, out of which the wind load and moments are determined on the structure.

6.1. Wind load and parameters for Juffertoren

Most building standards, NEN (Dutch), NBCC (Canadian), AS (Australian), AIJ (Japanese), EN (Euro code), BS (British), ASCE (American), CNS (Chinese) split the velocity of the wind into an hourly-averaged part $\bar{v}$ and a fluctuating part v [24].

The wind load on any arbitrary area can be written as

$$F = \frac{1}{2}\rho\left(\bar{v} + v\right)^2 AC_h$$

In which:

A	the area loaded by wind
ρ	density of air
C_h	summation of thrust and suction factor
$\bar{v}$	hourly-averaged part of the wind speed
v	fluctuating part of the wind speed

The Dutch norm NEN 6702 [3] indicated that the Juffertoren would have been located in Urban area II. This is true for one half of the building facing the city of Rotterdam. The other half of the building, which is facing the Meuse River, would have to use Vacant area II of the Dutch norm. For simplicity we use the values for **rural area II** of the NEN 6702 [3], these would lead to the highest forces on the structure [13]. The coefficients are shown in the table below.

	Rural	Urban
	II	II
$u_{*12,5}$	2.3 m/s	2.82 m/s
z_0	0.2 m	0.7 m
d	0 m	3.5 m
k	1 -	0.9 -

Table 3: Parameters for hourly averaged wind speed, NEN 6702 page 128 Table 6.1

[13] (p. 72)

The stability system of the Juffertoren has 2 main directions; because of this the use of the reference period of 12.5 years is valid.

6.2. Hourly-averaged part of the wind speed for return period of 12.5 years

The NEN 6702 describes the hourly-averaged wind speed which varies with the height as: ([3])

$$\bar{v}(z) = \frac{u_*}{\kappa} \ln\left(\frac{z - d_w}{z_0}\right)$$

In which:

u_*	Friction velocity [m/s]	= 2.30	[m/s]
κ	Von Karman constant [1];	= 0.4	[-]
z	height above the surface of the earth [m]	= 144	[m]
d_w	displacement height [m]	= 0	[m]
z_0	roughness length [m]	= 0.2	[m]

From the values in Table 6.1 we acquired the profile for the extreme hourly wind speed according to the NEN 6702 ([3])

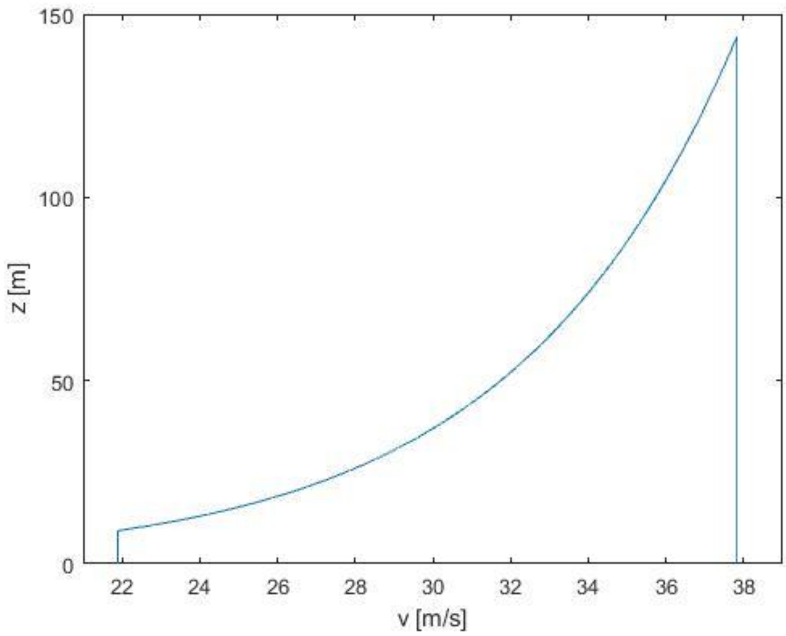


Figure 6.1: Extreme hourly-averaged wind speed profile for return period of 12.5 years or return period of 50 years under unfavorable direction in Rotterdam

Harris and Deaves describes the hourly-averaged wind speed which varies the height: [13]

$$\bar{v}(z) = \frac{u_*}{\kappa}\left(\ln\left(\frac{z-d}{z_0}\right) + 5.75a - 1.88a^2 - 1.33a^3 + 0.25a^4\right)$$

With: $a = \left(\frac{z-d}{z_g}\right)$, $z_g = \frac{u_*}{6f_c}$ and $f_c = 2\Omega \sin\lambda$

In which:

u_*	Friction velocity [m/s]	= 2.30	[m/s]
κ	Von Karman constant [1];	$= 0.4$	[-]
z	height above the surface of the earth [m]	$= 144$	[m]
d	displacement height [m]	$= 0$	[m]
z_0	roughness length [m]	$= 0.2$	[m]
f_c	Coriolisparamenter [s^{-1}]		
Ω	rotation speed of the earth [rad/s]	$(2\pi/24*60*60 = 7.2722E^{-5}\ rad/s)$	
λ	width degree [$^\circ$]	$\lambda = 51.75^\circ$	(Rotterdam)

From the values in Table 6.1 we acquired the wind profile for the extreem hourly wind speed of Harris and Deaves.

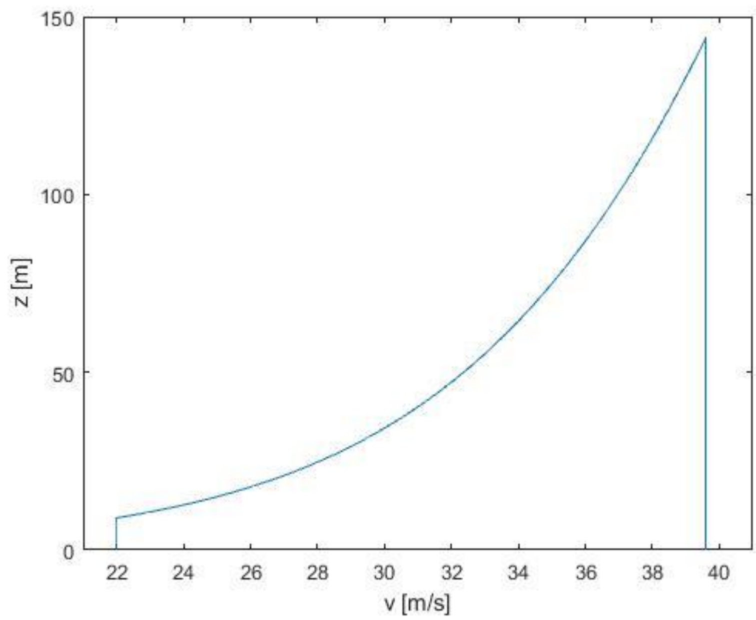


Figure 6.2: Extreme hourly-averaged wind speed profile for return period of 12.5 years or for return period of 50 years under unfavorable direction in Rotterdam

Eurocode describes the hourly-averaged wind speed which varies the height: [13]

$$\bar{v}(z) = u_{ref} k_t \ln\left(\frac{z}{z_0}\right)$$

With: $u_{ref} = c_{dir} * c_{temp} * c_{alt} * u_{ref,0}$, $k_t = 0.19 * \left(\frac{z_0}{z_{0,\mathrm{II}}}\right)^{0.0706}$

In which:

$\bar{v}(z)$	mean wind speed at height z [m/s]	=	[m/s]
u_{ref}	reference speed at height of 10 m [m/s]	$= 27.5$	[m/s]
k_t	terrain factor [-]	$= 0.21$	[-]
z	height above the face of the earth [m]	=	[m]
z_0	measure for the roughness of the terrain [m]	$= 0.2$	[m]
c_{dir}	direction factor [-]	$= 1$	[-]
c_{temp}	season factor [-]	$= 1$	[-]
c_{alt}	height factor [-]	$= 1$	[-]
$u_{ref,0}$	onalterd reference speed at height of 10 m [m/s]	$= 27.5$	[m/s]
$z_{0,\mathrm{II}}$	roughness length	$= 0.05$	[m]

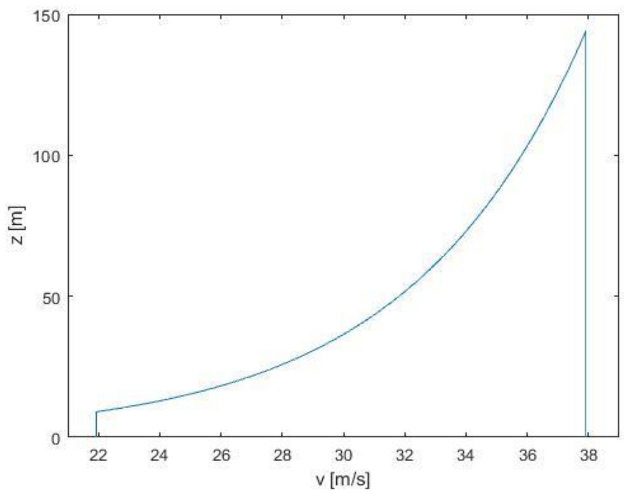


Figure 6.3: Extreme hourly-averaged wind speed for eurocode for return period of 12.5 years or return period of 50 years under unfavorable direction in Rotterdam

For this thesis the extreme hourly-averaged wind of Harris and Deaves will be used from now on in Matlab. This wind profile would lead to a larger overturning moment, stresses and strains in the structure. Harris and Deaves is more precise.

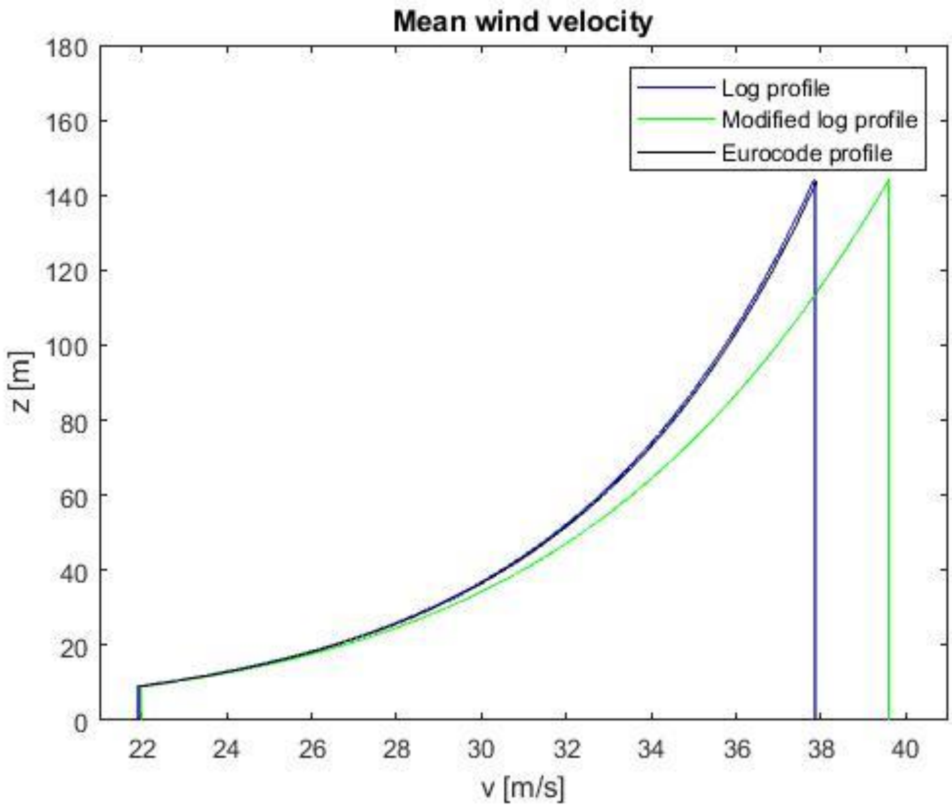
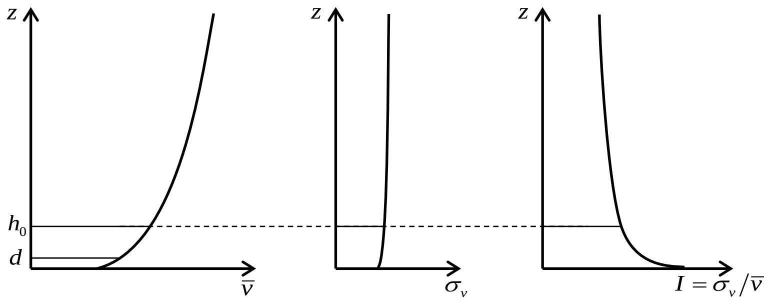


Figure 6.4: Extreme hourly-averaged wind speed for NEN, Harris and Deaves and Eurocode for return period of 12.5 years or return period of 50 years under unfavorable direction in Rotterdam

6.3. Fluctuating part of wind speed for return period of 12.5 years

The magnitude of the fluctuating part of the wind speed v can be described in terms of the standard deviation σ_v [1].

The literature says that the standard deviation of the wind speed varies per height z. An example where the standard deviation of the wind speed varies is shown below.



Variation of $\bar{v}$, σ_v and I as a function of the height z.

Figure 6.5: Standard deviation of the wind speed [1]

In the [3] the fluctuating part of the wind is defined as $\sigma = \bar{v} * I = \bar{v}(h) * I(h)$

In which:

I	Turbulence intensity	[-]
$\bar{v}$	Mean wind speed	[m/s]

The turbulence intensity is highest at the bottom of the structure and is lowest at the top of the structure this is because the wind gusts reduce with the height of the structure.
The standard deviation of the wind speed is lowest at the bottom of the structure and is highest at the top of the structure.

With:

$$I(z) = \frac{k_c}{\ln\left(\frac{z - d_w}{z_0}\right)}$$ Turbulence intensity factor;

For the Juffertoren the standard deviation of wind speed at reference height 10 m is determined below:

$$\sigma_v = \frac{ku_*}{\kappa} = \frac{1.0 * 2.30}{0.4} = 5.75 \text{ m/s}$$

In which:

u_*	Friction velocity	= 2.30	[m/s]
κ	Von Karman constant	$= 0.4$	[-]
k	Shape factor	$= 1.0$	[-]

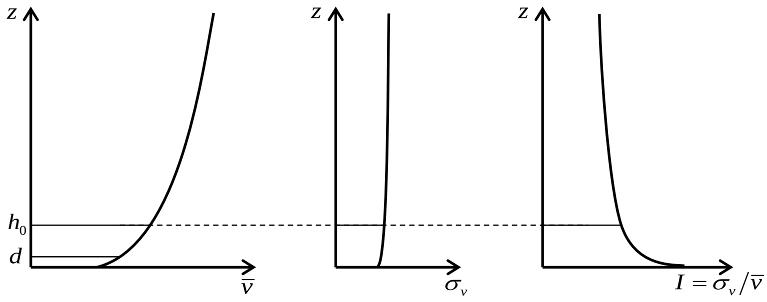
The standard deviation of the wind speed varies per height. The standard deviation of the wind speed at reference height has been determined above. The standard deviation of the wind speed on the top of the building is determined with formula:

$$\sigma_v(z) = \sigma_v(h_0)\left(\frac{z-d}{h_0}\right)^{\delta}$$ ([1] Ch.6 p.10).

In which:

$\sigma_v(h_0)$	Standard deviation of wind speed at reference height	= 5.75	[m/s]
h_0	reference height	= 10	[-]
z	height above the surface of the earth	= 144	[m]
d	displacement height	= 0	[m]
δ	power	= 0.03	[-]

For the Juffertoren the values for the wind speed, standard deviation of wind speed and turbulence intensity are shown for the top of the building and reference height of 10 meters. The wind speed at reference height and standard deviation of wind speed on the top of building are put into Matlab in the wind generator. (Appendix 2)



Variation of $\bar{v}$, σ_v and I as a function of the height z.

Figure 6.6: Standard deviation of the wind speed [1]

Height m	V Harris & Deaves m/s	σ_v m/s	I -
144	39.6	6.23	0.16
10	22.62	5.75	0.25

[13] (p. 84)

Table 4: Mean wind speed, standard deviation and turbulence intensity

The researchers, Davenport, Harris, and Simiu have determined spectra for fluctuations of the wind speed in 3 directions.

The reduced spectra's are

$$F_D = \frac{2}{3}\frac{x^2}{\left(1+x^2\right)^{4/3}} \qquad \text{(Davenport)}$$

$$F_D = \frac{3}{5}\frac{x}{\left(2+x^2\right)^{5/6}} \qquad \text{(Harris)}$$

$$F_D = \frac{2}{3}\frac{x}{\left(1+x\right)^{5/3}} \qquad \text{(Simiu)}$$

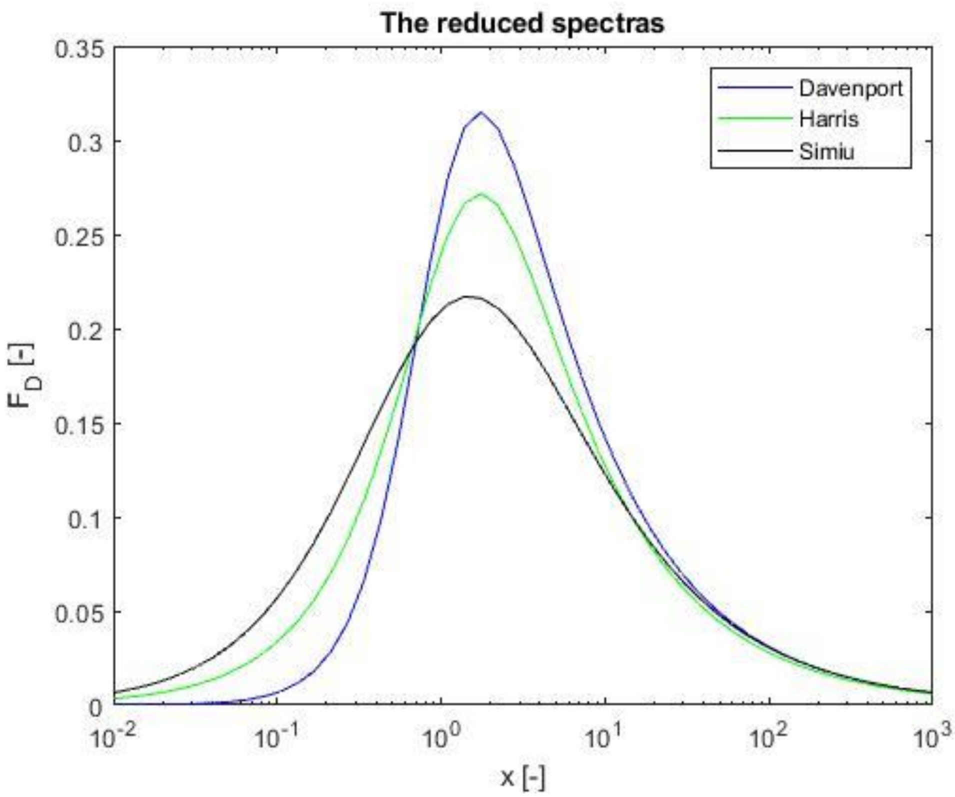


Figure 6.7: The reduced spectras

The NEN uses the Davenport spectrum [1] to determine the fluctuating part of the wind speed v. In [3], the fluctuating part of the wind speed v can be approximated by the formula $v \approx 2gI$ in which $g = \sqrt{2\ln(Tf_0)}$ the piekfactor and $I(z) = k_c \Big/ \ln\left(\frac{z-d_w}{z_0}\right) = \sigma_v / \bar{v}$ the turbulence intensity.

The fluctuating part of the wind speed v when defined as above is an overestimation of the actual fluctuation wind speed. v is conservative, because we use total correlation which entails that the fluctuating part of the wind speed is the same everywhere. In reality the fluctuating part of the wind speed varies in space and time.

In this thesis, a program in matlab will be used to approximate the fluctuating part of the wind speed more accurately in longitudinal and latitudinal direction. The façade loaded by wind in the along wind direction is split into a grid of areas. For each time instant the program will produce a randomly determined fluctuating wind speed on each area on the façade. Before we can execute this program, the coherence, which is the standard deviation of fluctuating wind speed and reduce spectra, have to be determined.

The velocity spectrum of the wind can be written as a function of the Davenport spectrum by:

$$S_w(f) = \frac{F_D {\sigma_v}^2}{f}$$

with:

$$F_D = \frac{2}{3}\frac{x^2}{\left(1+x^2\right)^{4/3}}$$ Davenport spectrum

σ_v standard deviation of the wind speed

$$x = \frac{fL_{gust}}{\bar{v}(10)}$$ dimensionless frequency

This gives the variance spectrum of the wind speed as a function of the frequency f :

$$S_{vv}(f)=\frac{2}{3}\frac{\left(\frac{fL_{gust}}{\bar{v}(10)}\right)^2}{\left(1+\left(\frac{fL_{gust}}{\bar{v}(10)}\right)^2\right)^{4/3}}\frac{\sigma_v{}^2}{f}$$

with:

f frequency;

$\bar{v}(10)$ mean wind speed at 10 m height $=22.62$ m/s (**Figure 6.2**)

L_{gust} characteristic length of a wind gust.

The velocity spectrum $S_{vv}(f)$ can be written as a function of the circle-frequency $S_{vv}(\omega)$ ([8] pp.9-14). The variance of velocity spectrum written as a function of the circle-frequency ω is defined as $\sigma_v^2=\int_0^\infty S_{vv}(\omega)\,d\omega$ and the variance of velocity spectrum written as a function of the frequency f is defined as $\sigma_v^2=\int_0^\infty S_{vv}(f)\,df$.

The relation between both velocity spectrums is:

$$S_{vv}(\omega)d\omega=S_{vv}(f)df\rightarrow S_{vv}(\omega)=S_{vv}\left(f=\frac{\omega}{2\pi}\right)\frac{df}{d\omega}=S_{vv}\left(f=\frac{\omega}{2\pi}\right)\frac{1}{2\pi}$$

By substituting the Davenport spectrum into the formula above the variance spectrum of velocities as a function of cyclic-frequency ω is obtained:

$$S_{vv}(\omega)=\frac{1}{2\pi}\frac{2}{3}\frac{\left(\frac{\omega L}{2\pi\bar{v}(10)}\right)^2}{\left(1+\left(\frac{\omega L}{2\pi\bar{v}(10)}\right)^2\right)^{4/3}}\frac{2\pi\sigma_v{}^2}{\omega}=\frac{\left(\frac{\omega L}{2\pi\bar{v}(10)}\right)^2}{\left(1+\left(\frac{\omega L}{2\pi\bar{v}(10)}\right)^2\right)^{4/3}}\frac{2\sigma_v{}^2}{3\omega}$$

The variance spectrum of the fluctuating wind speed as a function of cyclic frequency ω is shown below in (**Figure 6.8**)

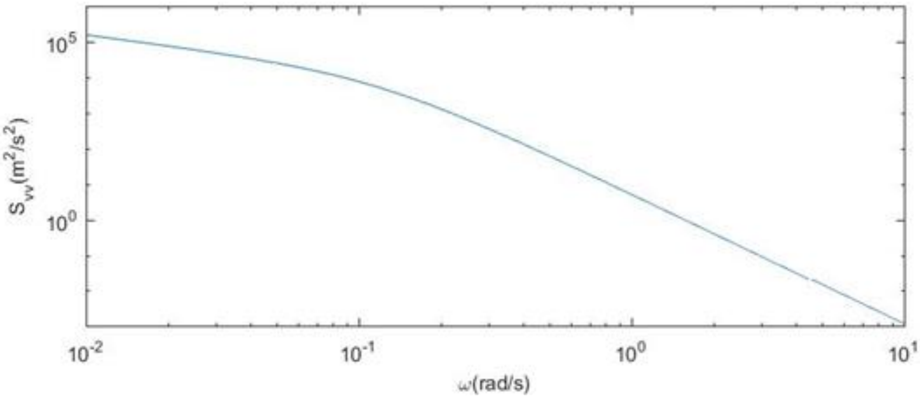


Figure 6.8: Spectrum of the fluctuating wind speed

A realization of the wind in the time domain can now be calculated by dividing the cyclic frequency ω domain into a finite number of points ω_k at equal distance $\Delta\omega$. The fluctuating part of the wind speed is now described by:

$$v = \sum_{k=1}^{N} a_k \sin\left(\omega_k t + \varphi_k\right)$$

with:

$$a_k = \sqrt{2S_{vv}\Delta\omega_k}$$

φ_k Random number between 0 and 2π

A possible realization of the wind velocity on basis of this model is shown in **Figure 6.9**.

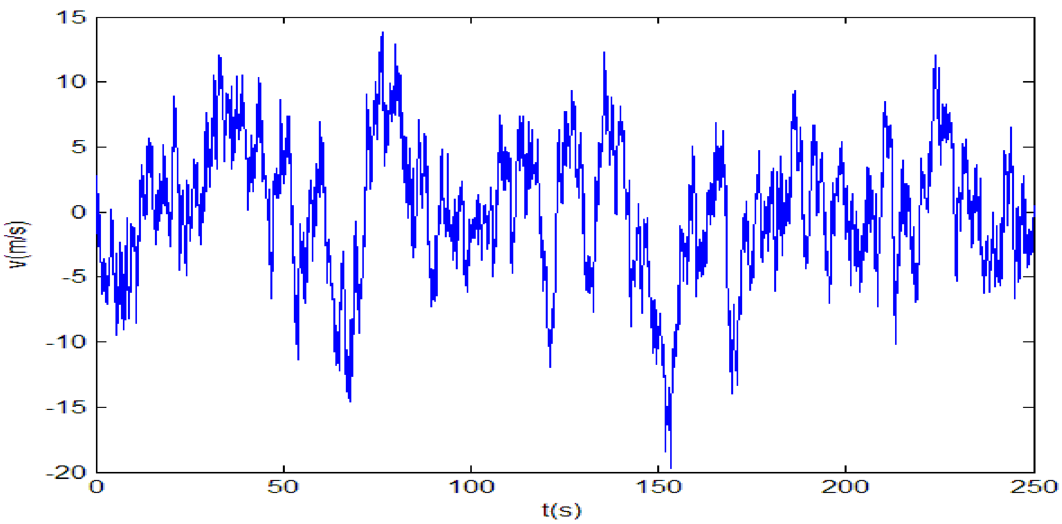


Figure 6.9: A realization of the fluctuating wind speed

6.4. Hourly-averaged part of the wind speed for return period of one year

The formulas are given to determine the mean and standard deviation for maximum hourly-averaged wind speed with the return period of 1 year in ([1] Ch.6 p.7). The formulas are given below:

$$\mu(v_1) = u_1 + \frac{0.577}{a} \qquad \sigma(v_1) = \frac{\pi}{\left(a\sqrt{6}\right)}$$

Ihe values are given for a and u_1 for area II in the Netherlands In table 6.3 ([1] Ch.6 p.7). These values are $a = 0.55\frac{s}{m}$ and $u_1 = 20.4\frac{m}{s}$. Out of which we can determine that $\mu(v_1) = 20.4 + \frac{0.577}{0.55} = 21.45\frac{m}{s}$ and $\sigma(v_1) = \frac{\pi}{\left(0.55 * \sqrt{6}\right)} = 2.33\frac{m}{s}$.

The hourly-averaged wind speed described by a logarithmic function is given ([1] Ch.6 p.2). ([3])

$$\bar{v}(z) = \frac{u_*}{\kappa}\ln\left(\frac{z-d}{z_0}\right)$$

For the Juffertoren we know that some of the values are given and known (6.2) for the situation $\bar{v}(10)$ and only the u_* friction velocity has to be determined.

κ	Von Karman constant [1];	$= 0.4$	[-]
z	height above the surface of the earth [m]	$= 10$	[m]
d	displacement height [m]	$= 0$	[m]
z_0	roughness length [m]	$= 0.2$	[m]

With the values above $u_* = 2.1932\frac{m}{s}$.

For the Harris and Deaves and Eurocode profile the unknown friction velocity (u_*) andonalterd reference speed at height of 10 m ($u_{ref,0}$) are determined in the same method as above.

The NEN 6702 describes the hourly-averaged wind speed which varies with the height as: ([3])

$$\bar{v}(z) = \frac{u_*}{\kappa} \ln\left(\frac{z - d_w}{z_0}\right)$$

In which:

u_*	Friction velocity [m/s]	= 2.1932	[m/s]
κ	Von Karman constant [1];	= 0.4	[-]
z	height above the surface of the earth [m]	= 144	[m]
d_w	displacement height [m]	= 0	[m]
z_0	roughness length [m]	= 0.2	[m]

From the values in Table 6.1 we acquired the profile for the extreme hourly wind speed according to the NEN 6702 ([3])

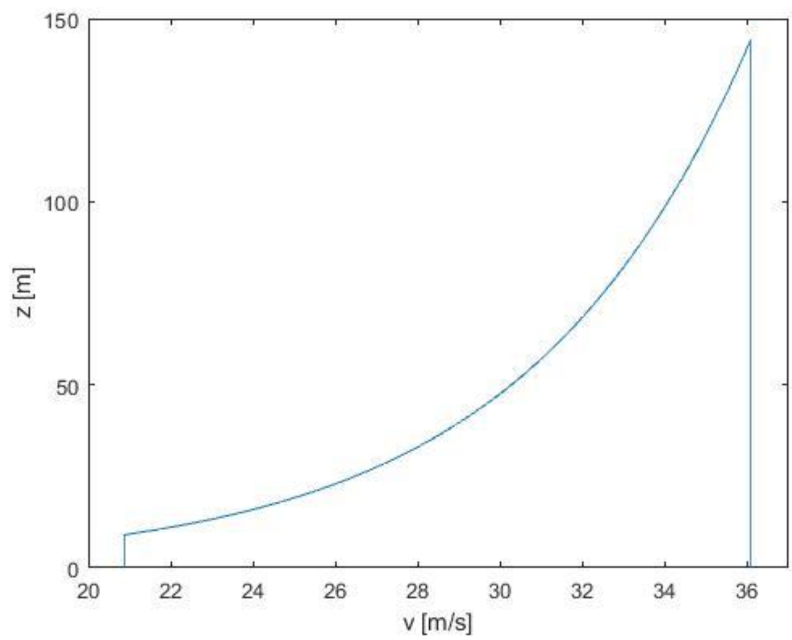


Figure 6.10: Extreme hourly-averaged wind speed profile for return period of one year in Rotterdam

Harris and Deaves describes the hourly-averaged wind speed which varies the height: [13]

$$\bar{v}(z) = \frac{u_*}{\kappa}\left(\ln\left(\frac{z-d}{z_0}\right) + 5.75a - 1.88a^2 - 1.33a^3 + 0.25a^4\right)$$

With: $a = \left(\frac{z-d}{z_g}\right)$, $z_g = \frac{u_*}{6f_c}$ and $f_c = 2\Omega \sin\lambda$

In which:

u_*	Friction velocity [m/s]	$= 2.18$	[m/s]
κ	Von Karman constant [1];	$= 0.4$	[-]
z	height above the surface of the earth [m]	$= 144$	[m]
d	displacement height [m]	$= 0$	[m]
z_0	roughness length [m]	$= 0.2$	[m]
f_c	Coriolisparamenter [s^{-1}]		
Ω	rotation speed of the earth [rad/s]	$\left(2\pi/24*60*60 = 7.2722E^{-5}\ rad/s\right)$	
λ	width degree [$^\circ$]	$\lambda = 51.75^\circ$	(Rotterdam)

From the values above, we acquired the wind profile for the extreem hourly wind speed of Harris and Deaves.

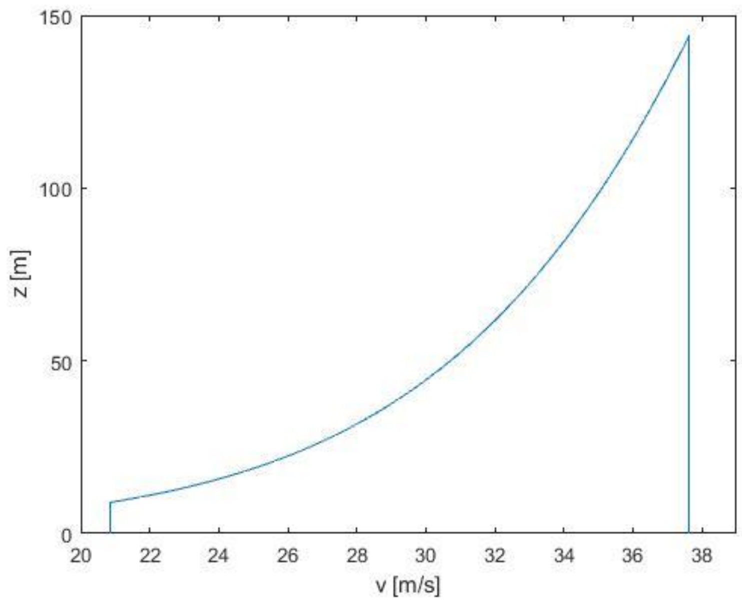


Figure 6.11: Extreme hourly-averaged wind speed profile for return period of one year in Rotterdam

Eurocode describes the hourly-averaged wind speed which varies the height: [13]

$$\bar{v}(z) = u_{ref} k_t \ln\left(\frac{z}{z_0}\right)$$

With: $u_{ref} = c_{dir} * c_{temp} * c_{alt} * u_{ref,0}$, $k_t = 0.19 * \left(\frac{z_0}{z_{0,\text{II}}}\right)^{0.0706}$

In which:

$\bar{v}(z)$	mean wind speed at height z [m/s]	=	[m/s]
u_{ref}	reference speed at height of 10 m [m/s]	$= 26.17$	[m/s]
k_t	terrain factor [-]	$= 0.21$	[-]
z	height above the face of the earth [m]	$= 144$	[m]
z_0	measure for the roughness of the terrain [m]	$= 0.2$	[m]
c_{dir}	direction factor [-]	$= 1$	[-]
c_{temp}	season factor [-]	$= 1$	[-]
c_{alt}	height factor [-]	$= 1$	[-]
$u_{ref,0}$	onalterd reference speed at height of 10 m [m/s]	$= 26.17$	[m/s]
$z_{0,\text{II}}$	roughness length	$= 0.05$	[m]

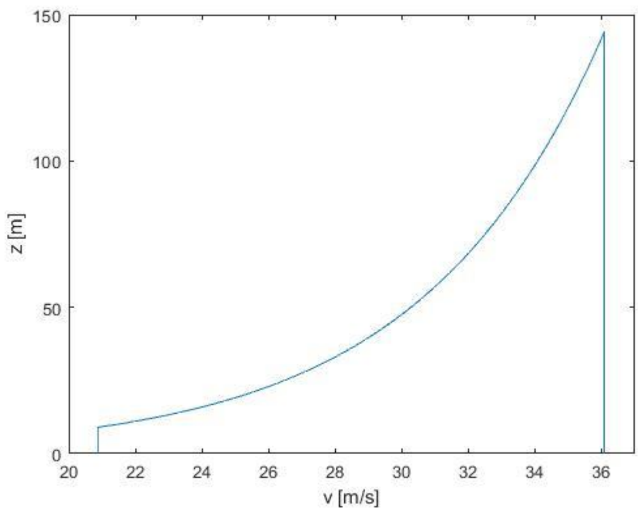


Figure 6.12: Extreme hourly-averaged wind speed for eurocode for return period of one year in Rotterdam

For this thesis the extreme hourly-averaged wind of Harris and Deaves will be used from now on in Matlab. This wind profile would lead to a larger overturning moment, stresses and strains in the structure. Harris and Deaves is more precise.

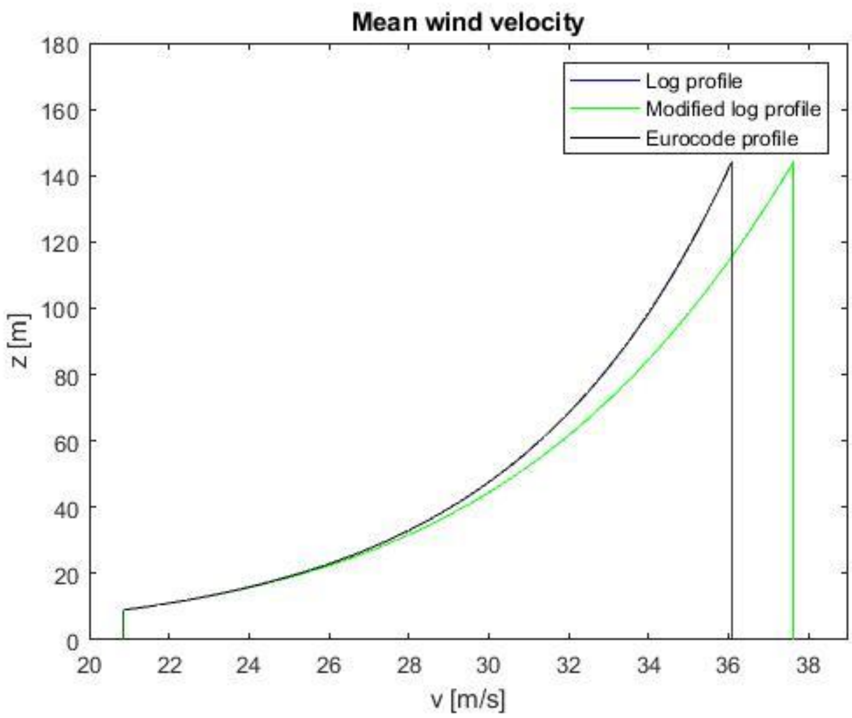


Figure 6.13: Extreme hourly-averaged wind speed for NEN, Harris and Deaves and Eurocode for return period of one year in Rotterdam

6.5. Fluctuating part of wind speed for return period of one year

For the Juffertoren:

$$\sigma_v = 2.33 \text{ m/s} \quad (6.4)$$

The standard deviation of the wind speed varies per height. The standard deviation of the wind speed on the top of the building is determined with formula:

$$\sigma_v(z) = \sigma_v(h_0)\left(\frac{z-d}{h_0}\right)^{\delta} \quad \text{([1] Ch.6 p.10).}$$

In which:

$\sigma_v(h_0)$	Standard deviation of wind speed at reference height	= 2.33 [m/s]
h_0	reference height	= 10 [-]
z	height above the surface of the earth	= 144 [m]
d	displacement height	= 0 [m]
δ	power	= 0.03 [-]

For the Juffertoren the values for the wind speed, standard deviation of wind speed and turbulence intensity are shown for the top of the building and reference height of 10 meters. The wind speed at reference height and standard deviation of wind speed on the top of building are put into Matlab in the wind generator. (Appendix 2)

Height m	V Harris & Deaves m/s	σ_v m/s	I -
144	37.62	2.52	0.07
10	21.45	2.33	0.11

[13] (p. 84)

Table 5: Mean wind speed, standard deviation and turbulence intensity

6.6. Wind loads on the structure

In this thesis the focus is on the fluctuating pressures on the facade which cause torsional moments.
The acceleration due to vortex shedding are not taken into account in the Matlab code.

The wind load of a horizontal row will be lumped as forces on the nodes, as can be seen in **Figure 6.14**. The structure is divided into a grid of areas on the façade.

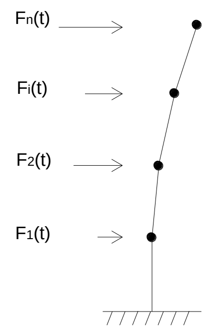


Figure 6.14: Forces on a row of horizontal nodes as a function of time

The Juffertoren is divided into 2 * 48 height segments (1.5 meters) perpendicular to the width of the building. Each height segment is also divided into 10 width segments (2.6 meters) in width direction. And illustration of a width row of horizontal areas can be seen below in **Figure 6.15**.

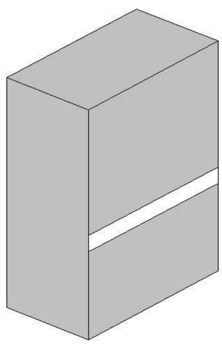


Figure 6.15: Arbitrary chosen row of horizontal nodes

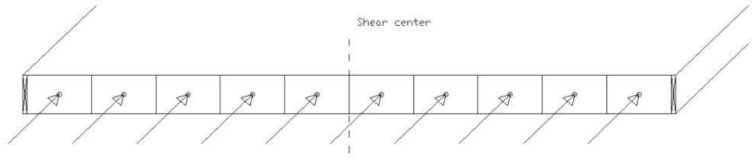


Figure 6.16: Row of horizontal areas with arbitrary chosen forces

The force on a node is equal to:

$$F_i(t) = AC_h q_w$$

with:

$$q_w = \frac{1}{2}\rho\left(\bar{v} + v\right)^2$$

$F_i(t)$ Force of horizontal row of areas loaded by wind.

i Node number

A The area loaded by wind; $A = 20 * 1,5 * 26 = 3 * 26 = 78$ m^2

ρ Density of air; ρ = 1,25 kg/m^3

C_h Thrust and suction shape factor; $C_h = 0,8 + 0,4 = 1,2$ (NEN 6702 figure A.4)

The summation of the forces on a row of horizontal nodes can be represented as a vector where each element represents a nodal force. The nodal force in the vector is the summation of all the forces on that specific height. By discretization in time, these nodal forces can be calculated for every time step, which makes a matrix of the nodal forces as shown in equation below. The summations of nodal forces on a horizontal row are numbered $1, i, .. n$ and the time steps are numbered $1, k, .. m$.

The nodal forces on a horizontal row can be written as:

$$\mathbf{F}(t) = \begin{bmatrix} F_1(t) \\ F_i(t) \\ . \\ F_n(t) \end{bmatrix} \xrightarrow{\text{discretization in place}} = \begin{bmatrix} F_{11} & F_{1j} & . & F_{1l} \\ F_{i1} & F_{ij} & . & . \\ . & . & . & . \\ F_{n1} & . & . & F_{nl} \end{bmatrix}$$

The summations of nodal forces on a horizontal row are numbered $1, i, .. n$ and the time steps are numbered $1, k, .. m$.

$$\mathbf{F}(t) = \begin{bmatrix} F_1(t) \\ F_i(t) \\ . \\ F_n(t) \end{bmatrix} \underrightarrow{\text{discretization in time}} = \begin{bmatrix} F_{111} & F_{1jk} & . & F_{1lm} \\ F_{i11} & F_{ijk} & . & . \\ . & . & . & . \\ F_{n11} & . & . & F_{nlm} \end{bmatrix}$$

The nodal force matrix can be calculated by making use of the two equations in the beginning of the section.

The mean wind speed $\bar{v}$ (Harris and Deaves logarithmic wind profile) is only fluctuating in place and the fluctuating wind speed v is fluctuating in place and in time (random wind generator), which gives:

$$\mathbf{F} = \begin{bmatrix} F_{111} & F_{1jk} & . & F_{1lm} \\ F_{i11} & F_{ijk} & . & . \\ . & . & . & . \\ F_{n11} & . & . & F_{nlm} \end{bmatrix} = \frac{1}{2} A C_h \rho \begin{bmatrix} \left(\bar{v}_{11} + v_{111}\right)^2 & \left(\bar{v}_1 + v_{1jk}\right)^2 & . & \left(\bar{v}_1 + v_{1lm}\right)^2 \\ \left(\bar{v}_{ij} + v_{i11}\right)^2 & \left(\bar{v}_{ij} + v_{ijk}\right)^2 & . & . \\ . & . & . & . \\ \left(\overline{v_{n1}} + v_{n11}\right)^2 & . & . & \left(\bar{v}_n + v_{nlm}\right)^2 \end{bmatrix}$$

The NEN 6702 assumes that the wind load on a structure for $(z < 9m)$ height is equal to the wind load for $(z = 9m)$, this is done to account for the wind pressure what has to flow around the base of the building, accounting for a higher mean wind speed. In this Matlab model we will also assume this for the mean wind speed $\bar{v}(z \leq 9 \text{ m}) = \bar{v}(z = 9 \text{ m})$ and for the fluctuating wind speed $v(y, z \leq 9 \text{ m}, t) = v(y, z = 9 \text{ m}, t)$.

6.7. Wind moments on a structure

The wind moments on the horizontal row will be lumped as couples on the nodes as can be seen in **Figure 6.17**. The moment on a node is equal to:

$$M_j(t) = \sum_{j}^{n} AC_h q_w a_j$$

with:

$$q_w = \frac{1}{2}\rho\left(\bar{v} + v\right)^2$$

a_j	Horizontal distance from node to the shear center
j	Lateral Node number
A	The area loaded by wind; A = 20*1,5*26 = 3*26 = 78 m^2
ρ	Density of air; ρ = 1,25 kg/m^3
C_h	Thrust and suction shape factor; C_h = 0,8 + 0,4 = 1,2 (NEN 6702 figure A.4)

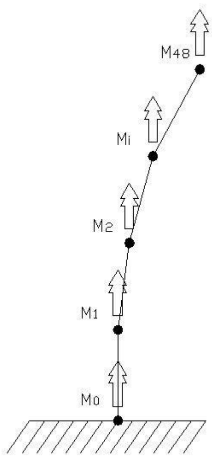


Figure 6.17: Moments on the nodes as a function of time

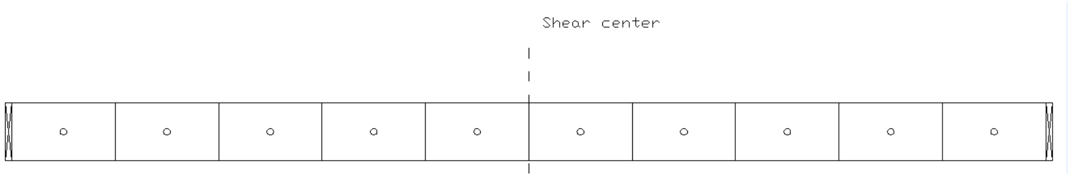


Figure 6.18: Row of horizontal areas

The moments on the nodes can be presented as a vector where each element represents a nodal moment. Which is a summation of all the nodal moments of a horizontal row of nodes. The moment of each node is calculated by multiplying the fluctuating wind force by the distance from the node to the shear center. By discretization in time, these nodal moments can be calculated for every time step, which makes a matrix of the nodal moments as shown in the equation below. The nodes are numbered $1, i, ..n$, the place steps are numbered $1, j, ..l$ and the time steps are numbered $1, k, ..m$. The horizontal distance from the node to the shear center of building a_p number $1, p, ..r$.

$$M(t) = \begin{bmatrix} M_1(t) \\ M_i(t) \\ . \\ M_n(t) \end{bmatrix} \xrightarrow{\text{discretization in time}} = \begin{bmatrix} F_{111}a_1 & F_{1jk}a_p & . & F_{1lm}a_r \\ F_{i11}a_1 & F_{ijk}a_p & . & . \\ . & . & . & . \\ F_{n11}a_1 & . & . & F_{nlm}a_r \end{bmatrix}$$

The moment on an area is the force of a wind load (fluctuating and mean) on a considered area multiplied by the distance from the node to shear center.

The nodal moment matrix can be calculated by making use of the two equation in the beginning of the section. The mean wind speed $\bar{v}$ (Harris and Deaves logarithmic wind profile) is only fluctuating in place and fluctuating wind speed v is fluctuating in place and in time (random wind generator), which gives:

$$M = \begin{bmatrix} F_{111}a_1 & F_{1jk}a_p & . & F_{1lm}a_r \\ F_{i11}a_1 & F_{ijk}a_p & . & . \\ . & . & . & . \\ F_{n11}a_1 & . & . & F_{nlm}a_r \end{bmatrix} = \frac{1}{2}AC_h\rho \begin{bmatrix} \left(\bar{v}_{11} + v_{111}\right)^2 a_1 & \left(\bar{v}_1 + v_{1jk}\right)^2 a_p & . & \left(\bar{v}_1 + v_{1lm}\right)^2 a_r \\ \left(\bar{v}_{ij} + v_{i11}\right)^2 a_1 & \left(\bar{v}_{ij} + v_{ijk}\right)^2 a_p & . & . \\ . & . & . & . \\ \left(\bar{v}_{n1} + v_{n11}\right)^2 a_1 & . & . & \left(\bar{v}_n + v_{nlm}\right)^2 a_r \end{bmatrix}$$

In which a_p is the horizontal distance from the node to the shear center of building number $1, p, ..r$.

For all the nodes left of the shear center, a negative value will be given to the moments of each node. The values of the moments on the right of the shear center will be held positive. This is done because in reality these moments in different directions would equalize each other. The resultant of these moments left and right would work on the structure.

6.8. Correlation of the wind speed at different locations

The fluctuating part of the wind speed varies in time and in place, due to wind gusts. Because of this, the wind speed on the façade at different locations will not be the same at each time instant. If there is a peak in the wind speed at one part of the building on the other, a peak gust may not occur. In the NEN 6702 area reduction is present. Also now in EN1991-1-4 via the factor $c_s c_d$. If the fluctuating wind speed at the different places of the façade (nodes of the program) is totally independent of each other in lateral and longitudinal direction, we call this **uncorrelated.** If the fluctuating wind speed at the different places of the façade are the same, we call this **fully correlated**.

The coherence function is a frequency dependant measure of the amount of correlation between the wind speeds at two points in space [15]

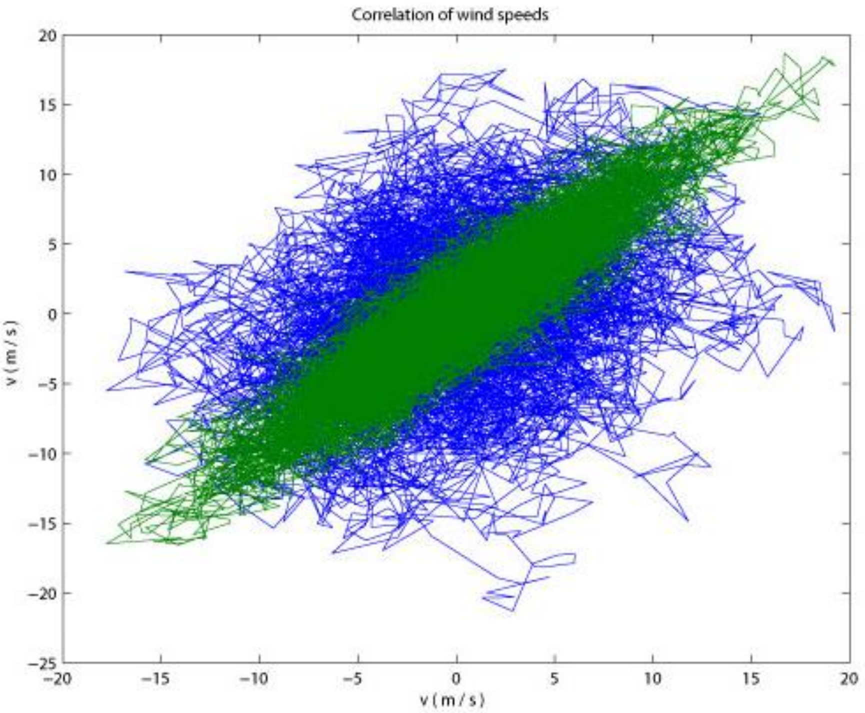


Figure 6.19: Correlation of Wind speed between yr node 1, 91 (blue) and node 1 ,2 (green)

In the figure above the correlation is shown for: Point 1 (yr=1.3 m, zr=9 m) and Point 91 (yr=1.3 m, zr=144 m) in blue and for Point 1 (yr=1.3 m, zr=9 m) and Point 2 (yr=1.3 m, zr=10.5 m) in green. The figure shows that there is more correlation between points what are situated closer to one an other (green) than points situated far from each other (blue).

The correlation between the wind speeds at different points can be described with cross-spectra:

$$\left|S_{v_1v_2}(f)\right|^2 = S_{v_1v_1}(f)\,S_{v_2v_2}(f)\,coh^2_{v_1v_2}(f)$$

with:

$S_{v_1v_2}(f)$ the cross-spectrum of the wind speeds in points 1 and 2

$S_{v_iv_i}(f)$ auto-spectrum of the wind in point i

$coh_{v_1v_2}(f)$ coherence between the wind speeds in point 1 and 2 ([16] p.34])

The coherence can be written in many ways [16]. If the coherence has to be compared or calculated one must recognize which coherence formula's is used, to correctly compare two coherence samples.

Examples of this are, the coherence formulas:

$$coh_{v_1v_2}(f) = \exp\left(-\frac{f\sqrt{C_z^2(z_1-z_2)^2 + C_y^2(y_1-y_2)^2}}{u_{m,av}}\right)$$

$$u_{m,av} = \frac{u_m(z_i) + u_m(z_j)}{2}$$

with:

y, z lateral and vertical coordinate, respectively;

C_z, C_y coherence constant in lateral and longitudinal-direction

$u_{m,av}$ average of mean wind speed of the wind speed at points z_i, z_j

$$coh_{v_1v_2}(f) = \exp\left(-\frac{f\sqrt{C_z^2(z_1-z_2)^2 + C_y^2(y_1-y_2)^2}}{U_{m\_10}}\right)$$

with:

y, z lateral and vertical coordinate, respectively;

C_z, C_y coherence constant in lateral and longitudinal-direction.

u_{m_10} mean wind speed at reference height 10 meters.

The coherence formula above, does not account for the reference height dependence of the coherence. This assumption is not a truth proven by experiments. [16]

The coherence can be expressed with the Sandia method as (Frost): [15].

$$Coh_{jk}(f) = \exp\left(-\frac{C\Delta r_{jk} f}{U}\right)$$

with:

Δr_{jk}	the distance between point's j and k
U	the velocity at reference height 10 meters
f	frequency
$Coh_{jk}(f)$	coherence between the wind speeds in point j and k

The coherence can be expressed with the Solari method as:

$$C_{jk} = b\left(-\frac{\Delta r_{jk}}{z_m}\right)^{0.25}$$

$$b = 12 + 5\mu_b$$

$$Coh_{jk}(f) = \exp\left(-\frac{C_{jk}\Delta r_{jk} f}{U}\right)$$

$$z_m = (z_j + z_k)/2$$

with:

Δr_{jk}	the distance between point's j and k.
U	the velocity at reference height 10 meters.
f	frequency
z_m	value of the lateral and vertical coordinate divided by 2
z_j	lateral coordinate
z_k	vertical coordinate
$C_{jk}(f)$	Coherence decrement
$Coh_{jk}(f)$	coherence between the wind speeds in point j and k

For this thesis the coherence will be calculated with the formula without height dependence which is more conservative than the formula with height dependence. The height dependant coherence formula defines the coherence better.

An other example of differences in coherence formula's is that the coherence is squared in the one formula and the others not, when determining the cross spectrum of the wind speeds in 2 points. (Begin Section 6.8)

A written Matlab program will be used to determine the correlation on the façade [4]. Before the correlation program can be ran, the mean wind speed at each longitudinal node has to be calculated. These values have to be used in the written Matlab program.

The Autopower spectrum is determined in Matlab program [4] .

This program (wind0) calculates the fluctuating wind speed for every predefined point with coordinates (y,z) as a discrete function of the time. Before running the file, the standard deviation, the mean wind speed and the coherence have to be defined. The Matlab program (wind0) can be viewed in ([4], Appendix 2).

The façade of the building has to be divided in equal areas with in the centre of each area a point for which the wind speed as a function of time will be calculated. By knowing the wind speed in a point, the wind load on the considered area can be calculated. (Section 5.2 and 6.6). The wind loads (fluctuating and mean) on a horizontal row of areas will be lumped in the 48 nodes and are put in vector.

The moment on the building is the force of a wind load (fluctuating and mean) on a considered area multiplied by the distance to shear center.

The mesh of points on the façade has to be fine enough so that the coherence can be taken into account. On the façade 910 points were taken ([5] p.21]). This led to a fine enough mesh. The bending, torsional and bending and torsional acceleration and location will be shown afters this.

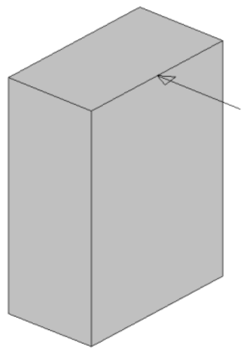
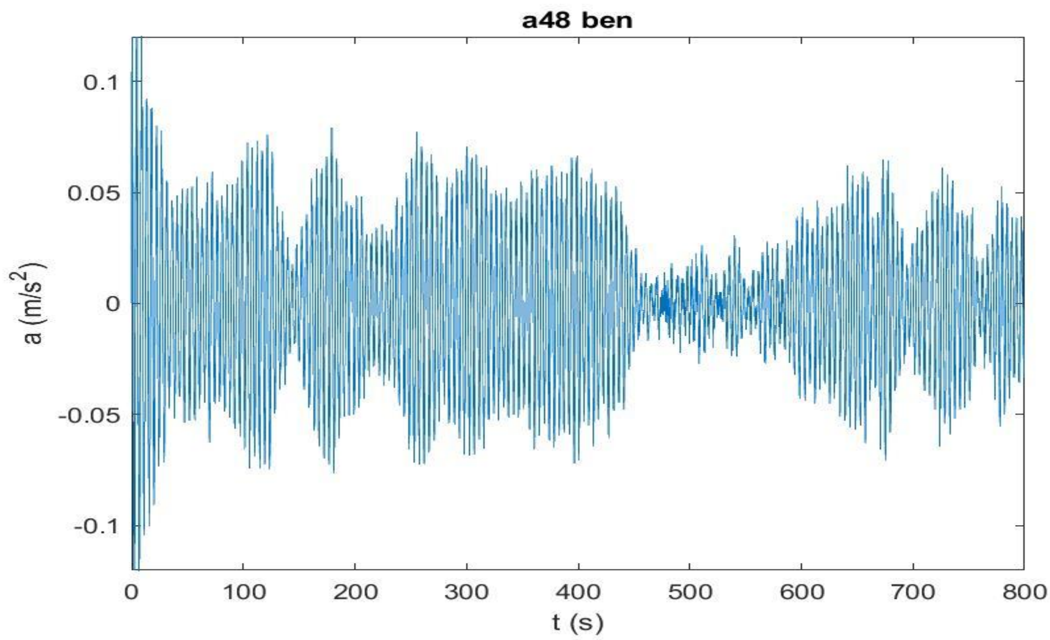


Figure 6.20: Location of determined bending accelerations



+

Figure 6.21: Wind speed calculation with 910 points: bending acceleration of the top

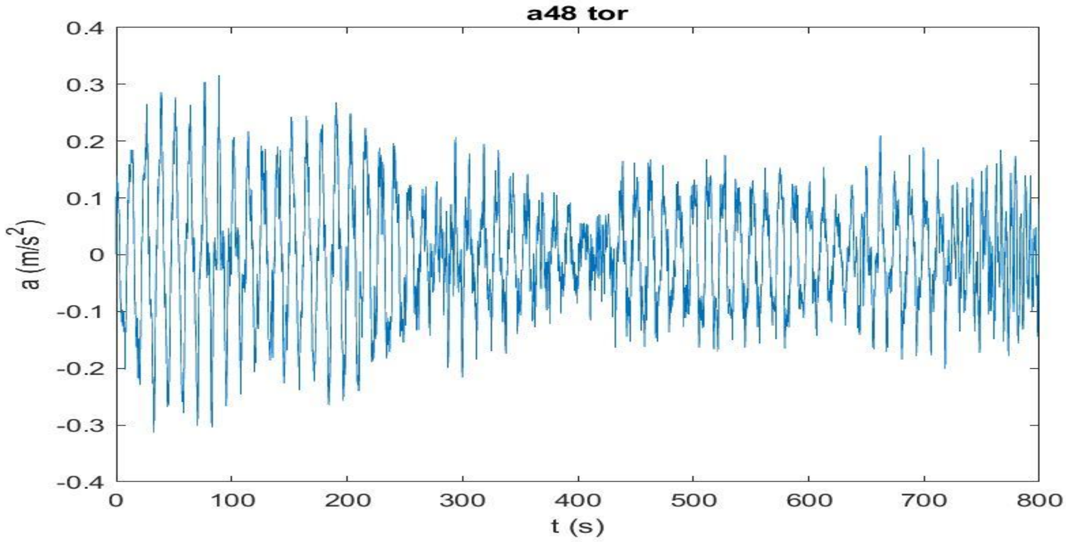


Figure 6.22: Wind speed calculation with 910 points: torsional acceleration of the top corner building

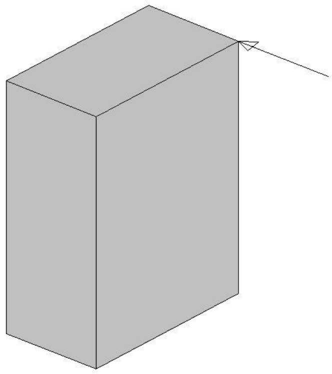


Figure 6.23: Location of determined bending and torsional accelerations

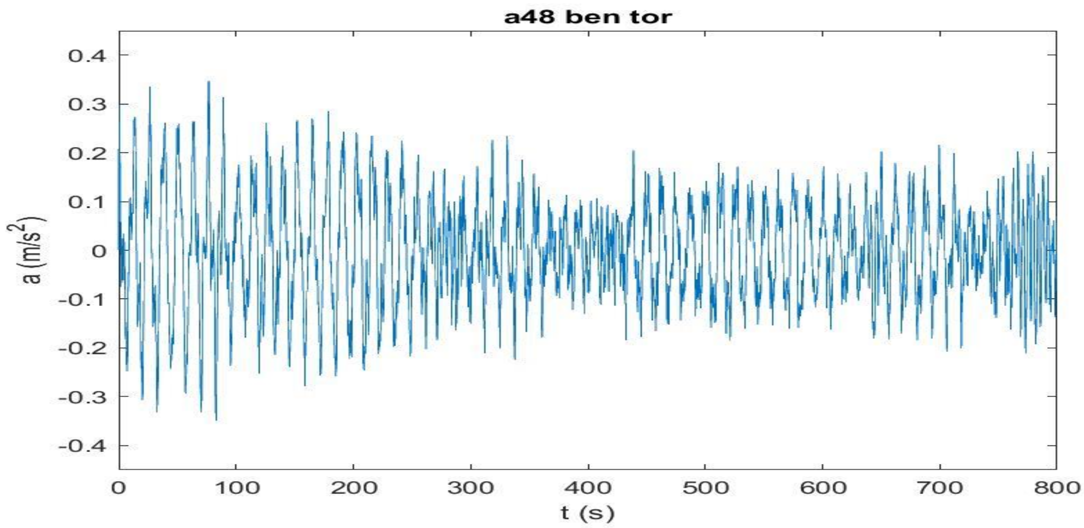


Figure 6.24: Wind speed calculation with 910 points: bending and torsional acceleration of the top corner building

6.9. Autospectrum density function estimate

The reason why an autospectrum density function estimate (**Figure 6.25**) has to be done , is to check if the turbulence of the wind (**Figure 6.9**) has been generated correctly.

For the Juffertoren, the values for wind speed at the hub height ($v = 21.45\frac{m}{s}$) and standard deviation at the top of the building ($\sigma_v = 2.52\frac{m}{s}$) were inputted. (Appendix 2: Matlab file: specest_test_run.m)

In the figure below we see the controle line in blue (average) and the line with generated turbulence in orange, we observe that they both fit each other.

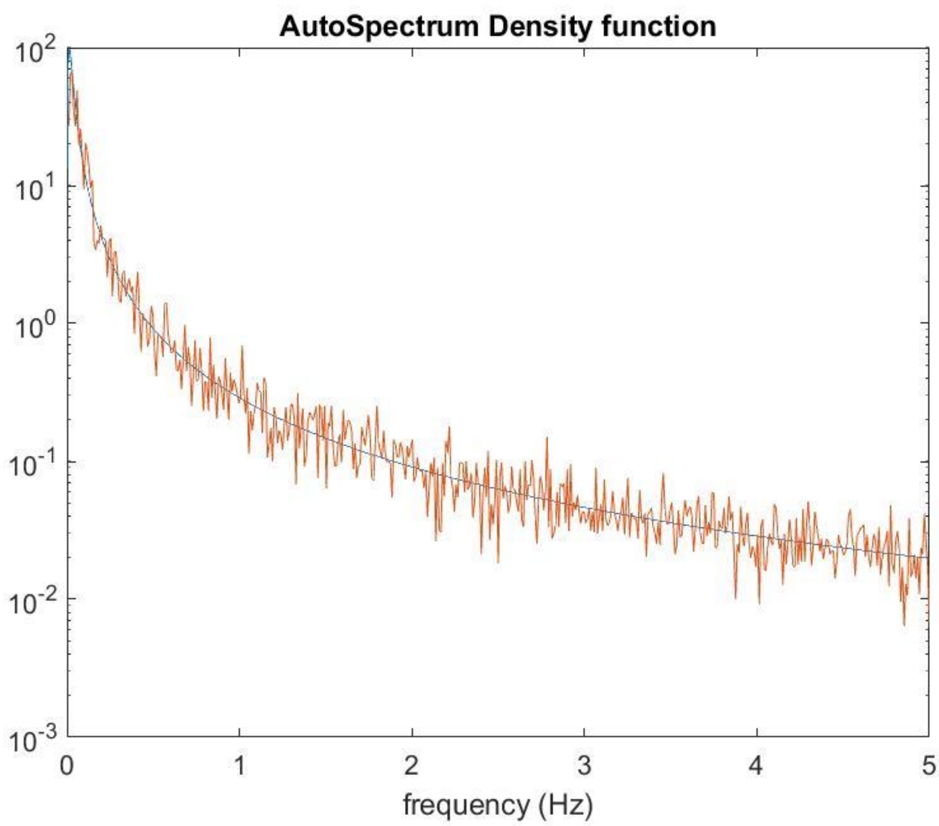


Figure 6.25: Autospectrum density function estimate of one data sequence

7. Calculation results Juffertoren

In this chapter the modeled calculation results of the Juffertoren (bending and torsional acceleration) are presented and compared to the known maximum allowed bending acceleration. In this chapter the summary values of a 100 simulation in time-domain are also presented.

7.1. Matlab

With the use of a Simulink model the maximum acceleration due to bending and torsion added together is calculated for a time span of 10 minutes. For the calculation in this paragraph we take the values form one of the 100 simulations, with mean velocity $V_{10} = 21.45$ m/s and return period of one year. (6.5)

7.1.1. Bending and torsion added together

The acceleration of the top of the building (Bending and Torsion motion added together)

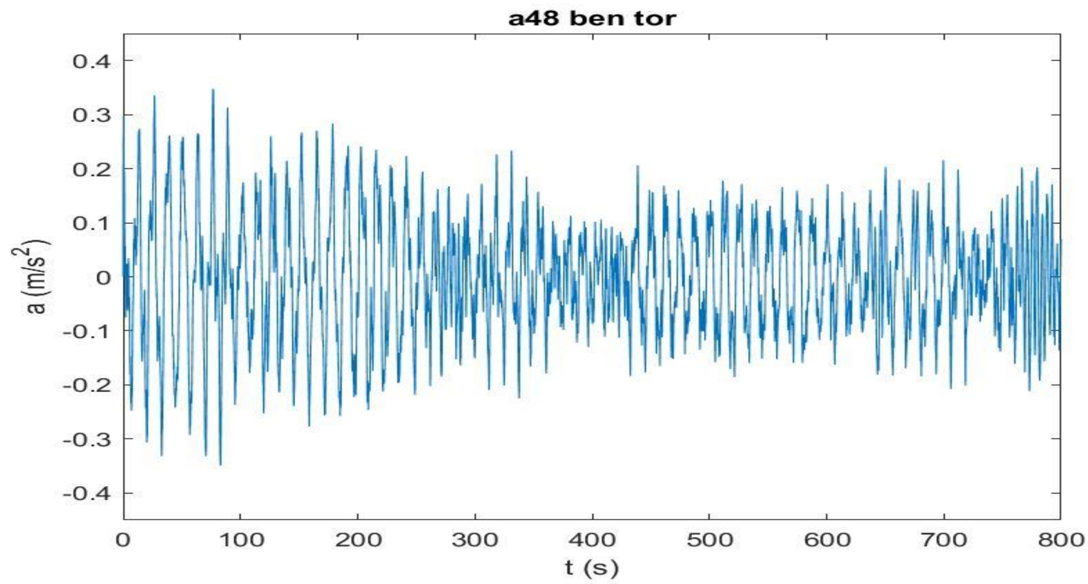


Figure 7.1: Acceleration of the top of the building (bending and torsion motion added together)

The frequency belonging to this acceleration equals the lowest eigenfrequency of the structure $f_e = T_e^{-1} = 0.224Hz$ (section 5.4.1). The peak acceleration of the top of the building for bending and torsion added together is $a_{48;\max} = 0.3479 m/s^2$, which is higher than the maximal acceptable annual acceleration of $0.1 m/s^2$ according to NEN. In which the number 48 is the node number at the top of the building.

We may assume that a_{48} is a normally distributed signal. The standard deviation of a_{48} can be determined from:

$$\sigma_{a;48} = \left(\frac{1}{n} \sum_{i=1}^{n} \left(a_{48;i} - \mu_{a;48} \right)^2 \right)^{\frac{1}{2}}$$

Where:

n the number of discrete time points for which a_{48} have been calculated

$a_{48;i}$ the acceleration of the top of the building at time point i

$\mu_{a;48}$ the mean value of a_{48}

With $\sigma_{a;48} = 0.1030$ m/s^2 and $\mu_{a;48} = 0 m/s^2$ calculated with Matlab, the expected peak value of the acceleration for a given time range follows from:

$$a_{48;peak} = \mu_{a;48} + \sigma_{a;48} \sqrt{2 \ln(N)} \quad \text{([1] eq. 3.38)}$$

Note: The formula above assumes that all peaks are independent of each other which is not true in reality, so this formula will give a value larger than the real peak acceleration.

Where N is the number of draws which follows from the number of local peaks in the total time range:

$$N = T_s f_e$$

Where:

T_s time range of the signal; $T_s = 600$ s

f_e natural frequency which comes out of the mathlab model; $f_e = 0.224 Hz$

Filling in peak value of acceleration gives:

$$a_{48;peak;600} = 0.1030 \sqrt{2 \ln(600 * 0.224)} = 0.322 \text{ m/s}^2$$

The difference between the expected peak value and the actual value is $(1-(0.3479/0.322))*100=-8.0\%=8.0\%$ which is reasonably accurate. The expected peak value for a storm of 6 hours 21600 s can be calculated from:

$$a_{48;peak;21600}=0.1030\sqrt{2\ln(21600*0.224)}=0.424\text{ m/s}^2$$

The expected peak value for a storm of 1 hour (3600s) can be calculated from:

$$a_{48;peak;3600}=0.1030\sqrt{2\ln(3600*0.224)}=0.377\text{ m/s}^2$$

The expected peak value for an hour long storm derived from the Simulink simulation values can be compared with the maximum acceleration according to NEN (8.2.1).

$$\frac{a_{\max;NEN}}{a_{48;peak;3600}}=\frac{0.113}{0.377}=0.300$$

The maximum acceleration due to **bending and torsion motion added together** is exceeded by a factor of 3.3. The acceleration of the building is unacceptable.

7.1.2. Bending

In this section the maximum acceleration will be calculated for bending acceleration only because this part of the model can be compared to known literature.

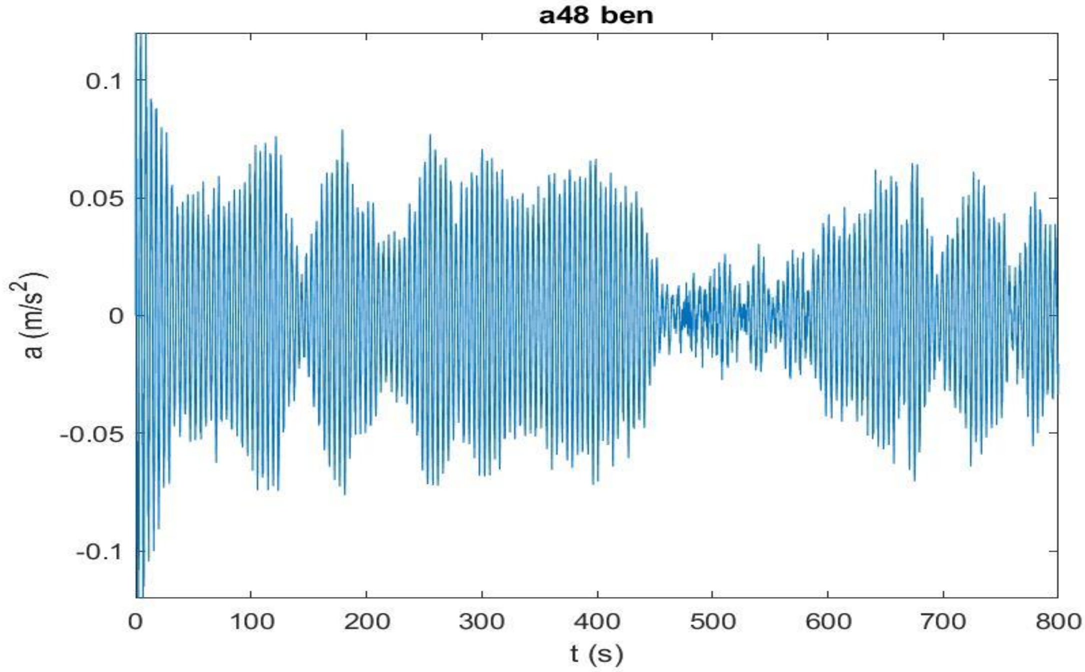


Figure 7.2: Acceleration of the top of the building (bending)

The frequency belonging to this acceleration equals the lowest eigenfrequency of the structure $f_e = T_e^{-1} = 0.224Hz$ (section 5.4.1) The peak acceleration of the top of the building equals: $a_{48;\max} = 0.0790 m/s^2$, which is higher than the maximal acceptable acceleration. We may assume that a_{48} is a normally distributed signal. View p.78 for the formula of the standard deviation of a_{48}.

With $\sigma_{a;48} = 0.0319$ m/s^2 and $\mu_{a;48} = 0$ m/s^2.

The expected peak value for a hour long storm derived from the Simulink simulation values can be compared with the maximum acceleration according to NEN (8.2.1).

$$\frac{a_{\max;NEN}}{a_{48;peak;3600}} = \frac{0.113}{0.117} = 0.966\cdot$$

The maximum acceleration due to **bending** is exceeded by about 4%. The acceleration of the building is slightly unacceptable.

7.2. 100 simulations in Time Domain

In the table below the summary values of a 100 simulations of the Juffertoren are given. The average values and standard deviation for bending, torsion and bending and torsion are given. The maximum and minimum of the 100 simulations are also given. We can conclude that there is a big spread between the maximum and minimum value. For the values out of the simulations view Appendix 3. The time of 1 simulation is about 18 minutes.

Average	0.0890	0.2585	0.2803		0.0282	0.0855	0.0904
	a_ben	a_tor	a_ben_tor		sigma_a_ben	sigma_a_tor	sigma_a_ben_tor
Max	0.1468	0.3473	0.3671		0.0422	0.1289	0.1322
Min	0.0671	0.1707	0.2028		0.0198	0.0570	0.0640
Max - Ave	0.0578	0.0889	0.0868		0.0140	0.0434	0.0418
Average -	0.0220	0.0878	0.0775		0.0085	0.0286	0.0264
Percentag	64.90%	34.39%	30.98%		49.71%	50.74%	46.27%
Percentag	24.69%	33.97%	27.65%		30.01%	33.40%	29.16%

Table 6: Summary values of a 100 simulations of the Juffertoren in Time Domain

7.3. 10 simualtions in the Time Domain

In the table below the values and summary values of 10 simulations of the Juffertoren are given. The average values and standard deviation for bending, torsion and bending and torsion are given.

	$a_{\_ben}$ m/s^2	$a_{\_tor}$ m/s^2	$a_{\_ben\_tor}$ m/s^2		σ_{a_ben} m/s^2	σ_{a_tor} m/s^2	$\sigma_{a\_ben\_tor}$ m/s^2
Simulation 1	0.079	0.315	0.348		0.032	0.098	0.103
Simulation 2	0.095	0.229	0.226		0.028	0.067	0.072
Simulation 3	0.089	0.231	0.234		0.027	0.078	0.084
Simulation 4	0.106	0.203	0.271		0.025	0.067	0.072
Simulation 5	0.074	0.202	0.247		0.026	0.081	0.086
Simulation 6	0.081	0.234	0.269		0.027	0.070	0.076
Simulation 7	0.085	0.249	0.265		0.029	0.083	0.089
Simulation 8	0.072	0.258	0.241		0.020	0.094	0.096
Simulation 9	0.131	0.240	0.236		0.030	0.060	0.068
Simulation 10	0.081	0.227	0.242		0.026	0.077	0.081
Summation	**0.0893**	**0.2388**	**0.2578**		**0.0270**	**0.0775**	**0.0827**

Table 7: Values of 10 simulations of the Juffertoren in Time Domain

We notice that there is a difference between the average values of a 100 and 10 simulations. How much simulations must be run to get a good average is not known in advance.

8. Comparison to design formulas Juffertoren

The essence of this chapter is to show that the different documentation gives different natural frequencies ω. There is also a considerable difference between the natural frequency determined analytically and natural frequency determined in Mathlab. If we determine the maximum acceptable deflection of the top of the building ($u_{top} = h/500$) we can determine the maximum acceleration of the buildings for each different formula.

In this chapter three cases will be distinguished.

1. The difference in acceleration between the different formulas of the NEN, Scheuller, Euro code, Woudenberg, Dicke/Nijsse and proposed model in Simulink. (without the torsional acceleration).
   Most of these Formulas/Norms only look at along wind response while the across wind response due to vortex shedding can be dominant for rectangular buildings ([17] p.61]).

2. The difference in acceleration between the different formulas without the use of the proposed model in Simulink, Woudenberg *empirical* (with the torsional acceleration and only alongwind response) .

3. The difference in acceleration between the different formulas, and total acceleration due to bending and torsional movement in the proposed model in Simulink. The acrosswind acceleration also will be determined by use of National Building Code of Canada (NBCC).

8.1. Maximum occuring accelerations for return period once in 1 year

In the table below the values of the comfort requirement is given for some norms and rules of thumb. The average value (Matlab) for a 100 simulations is also given for bending, torsion and bending and torsion.

Formula	Natural frequency		Along wind			Across wind		
	Hz	rad/s	Max bending acceleration m/s^2	Max torsional acceleration m/s^2	Max total acceleration m/s^2	Max bending acceleration m/s^2	Max torsional acceleration m/s^2	Max total acceleration m/s^2
NEN	0.231	1.451	0.113		0.113			
Eurocode	0.134	0.843	1.965		1.965	1.066		1.066
NBCC	0.200	1.258				0.179		0.179
Woudenberg (emp)	0.319	2.007	0.580	0.039	0.620			
Woudenberg	0.231	1.451	0.059		0.059			
Schueller	0.186	1.167	0.196		0.196			
Dicke/Nijsse	0.102	0.640	0.074		0.074			

Table 8: Resulting annual maxima

Matlab	0.224	1.407	0.089	0.259	0.280			

Table 9: The average out of a 100 simulations occurring acceleration in Simulink for the Juffertoren

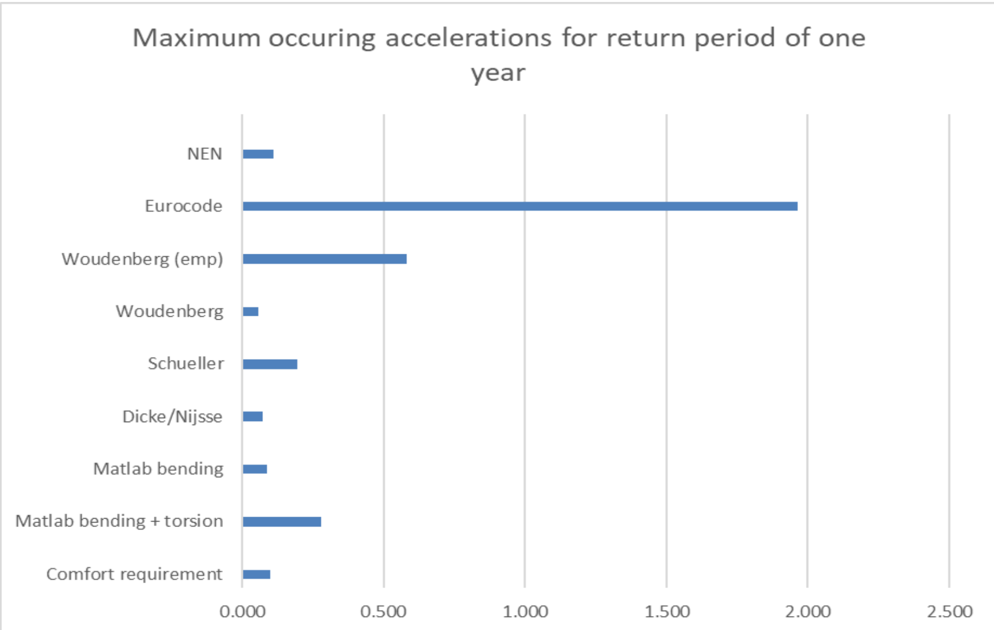


Figure 8.1: Maximum occurring accelerations for return period of one year for the Juffertoren

It can be concluded that the average value out of a 100 simulations for bending, out of Matlab ($0.089\frac{m^2}{s}$) is smaller the value for the NEN, Eurocode, Woudenberg (emp) and Schueller. For bending and torsion added together we see that the average value out of a 100 simulations for bending and torsion added together ($0.280\frac{m^2}{s}$) out of Matlab is larger than, most Norms and rules of thumb. The Eurocode value is so over conservative for bending, that it still meets the comfort requirement for bending and torsion which is not logical.

The acceleration above does not take into account the effect of shear lag, the second order effect and the reduction of effective area due to openings (windows) in structural elements. The fact that the crosswind acceleration can be larger than the along wind acceleration in many cases is neglected by many of the used formulas above.

The maximum occurring acceleration is the superposition of the alongwind, acrosswind and torsional acceleration. In most of these formulas only one component of the acceleration is taken into account, which makes these formula non-conservative. The actual acceleration felt by a person in the building will be probably larger than the outcome of any formula above.

The characteristic values of the simulations are given below: bending, torsion and bending and torsion. The standard deviations in the table below has been taken out of the average of 10 simulations (**Table 7**).

Juffertoren R = 1 year	k -	σ_a m/s²	a_{kar} m/s²
Bending	2.91	0.027	0.08
Torsion	2.91	0.078	0.23
Bending +Torsion	2.91	0.083	0.24

Table 10: characteristic values of the 10 acceleration simulations

k is related to about 1 promille of the time that the value exceeds a-kar and the maximum expected value in one hour is about 3.5 times the standard deviation.

σ_a = mean of standard deviation of 10 simulations

$a_{kar} = k * \sigma_a$ = characteristic vaue of acceleration for 10 simulations

The reason for determining a_{kar} is because by only taking the outcome of one simulation, you may be above or below the average of large numbers. In simple words, you maybe below or above the average of a 10000 simulations. Because of time constraints, we use the 95% reliability threshold to determine the characteristic mean of the 10 simualtions

8.2. Formulas to determine the maximum acceptable acceleration

8.2.1. NEN

According to [3] vibrations are annoying when the acceleration exceeds a value depending on the frequency (**Figure 8.2).**

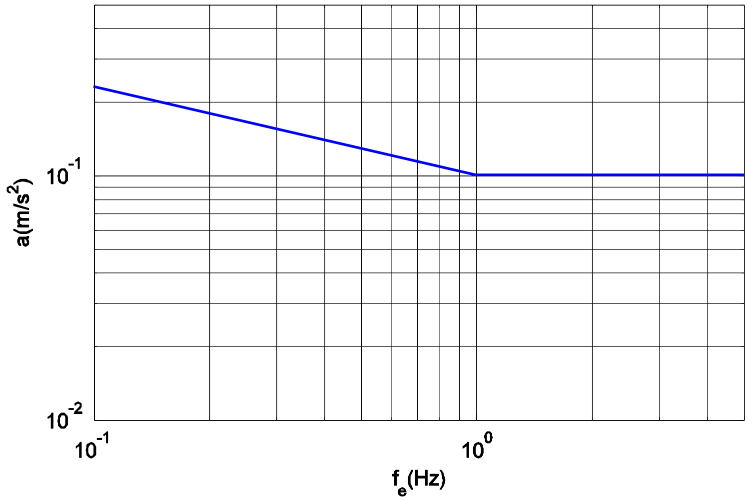


Figure 8.2: Limitation demand for the acceleration according to NEN 6702 figure 21

The [3] which gives the limitation demand for the peak value of the acceleration follows from:

$$a_{\max} = 1.6 \frac{\phi_2 \tilde{p}_{w;1} C_t b_m}{\rho_1} < a$$

Where:

a limitation demand of the acceleration for once per year a = 0.10 m/s^2 ([3])

ρ_1 mass of the building per meter height ρ_1 = 2.68*10^5 kg/m

b_m width of the building perpendicular to the wind direction b_m = 26.3 m

C_t summation of the shape factor C_t = 0.8 + 0.4 = 1.2 ([3])

$\tilde{p}_{w;1}$ value for the varying part of the wind pressure $\tilde{p}_{w;1} = 100\ln(h/0.2) = 660$ N/m^2

ϕ_2 value depending on the dimensions, the eigenfrequency and the damping

$$\phi_2 = \sqrt{\frac{0.0344 * f_e^{-2/3}}{\varsigma\left(1+0.12 f_e h\right)\left(1+0.2 f_e b_m\right)}} = \sqrt{\frac{0.0344 * 0.231^{-2/3}}{0.01 * \left(1+0.12 * 0.199 * 144\right)\left(1+0.2 * 0.199 * 26.3\right)}} = 0.909$$

Where:

f_e eigenfrequency $f_e = T^{-1} = 0.224$ Hz (From Mathlab)

ς damping ratio $\varsigma = 0.01$ (Section 4.8)

f_e eigenfrequency $f_e = \sqrt{\frac{a}{\delta}} = \sqrt{\frac{0.384}{7.2}} = 0.231$ Hz (NEN 6702 [3])

a acceleration $a = 0.384\,\text{m/s}^2$

δ displacement $\delta = \frac{ql^4}{8EI} = \frac{2.68E^6 * 144^4}{8 * 3.00E^{10} * 688} = 7.2$ m

In which:

l = The building length:	144	m	
Q = The building weight:	386,5	MN	(From Mathlab)
q = dead weight of the structure:	2.68	MN/m	(From Mathlab)
g = The gravitational acceleration:	9.81	m/s^2	
E = Young's modulus:	3.00E10	N/m^2	
I = The bending stiffness:	688	m^4	

The deflection is determined by using the equation which is given in [14]. In this equation, variable q is the total deadweight of the structure. The total deadweight of the structure is the summation of the deadweight of the skyscraper's floors, load-bearing elements and façade.

Floors:

The representative deadweight of the floor is $q_{g;rep}=0.25*24=6.0\text{kN/m}^2$ .The design value is $q_d=6.0*1.2=7.2\text{kN/m}^2$. The total area of the building is $26.3*15.4 \approx 405\ \text{m}^2$ The total deadweight of a storey of the building is: $405*7.2 = 2916\ \text{kN}$. The storey height is 3.0 metres, so it follows that deadweight perimeter height due to dead weight is $2916:3 = 972\ \text{kN/m}$

Vertical load-bearing elements:

The area of the concrete walls is $40.14\ \text{m}^2/\text{m}$. The deadweight of the concrete walls is $q_{g;rep} = 40.14 * 24 \approx 964\ \text{kN/m}^2$.

Facade:

The deadweight of the facade elements is $1.2\ \text{kN/m}$. The perimeter of the building is $2*26.3+2*15.4=83.4\ \text{m}$. This means that the total deadweight of the facade elements is: $83*1.2 \approx 100\ \text{kN/m}$.

Total:

The total deadweight per meter building height is: $972+964+100=2036\ \text{kN/m}$

This gives:

$$\phi_2 = \sqrt{\frac{0.0344*0.231^{-2/3}}{0.01*(1+0.12*0.199*144)(1+0.2*0.199*26.3)}} = 0.909$$

Filling in the formula for limitation demand for peak acceleration gives:

$$1.6\frac{0.909*660*1.2*26.3}{2.68*10^5} = 0.113\ \text{m/s}^2 > 0.10\ \text{m/s}^2$$

8.2.2. Matlab

For tekst view paragraph 7.1

9. Idealisation of the Voorhof building

In this chapter, the student building Voorhof is idealized, the structural characteristics, natural frequencies, damping ratios, wind characteristics, mean wind speed and fluctuating wind spead are determined, before and after renovation. The accelerations (bending, torsion and bending and torsion) are determind with the new model for 4 time instances, 2 of which are recorded mean wind velocities ([17] Appendix I) and 2 according to the NEN (SLS) and compared with earlier determined values.

9.1. Pictures and idealisation

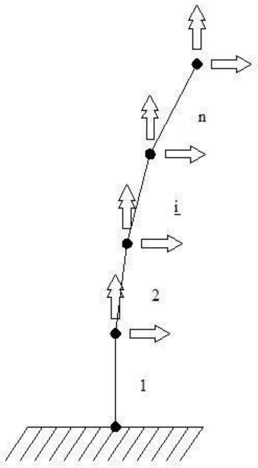
For the Student dormitory building "Voorhof" a **bending and torsional** model was made. The Student building "Voorhof" has a height of 51.30 m, width of 80.81 m and depth of 14.20 m. [17]. The structure has been built at the E. du Perronlaan in Delft in the Netherlands (**Figure 9.1**).

Figure 9.1: Picture of the Voorhof Student Building.

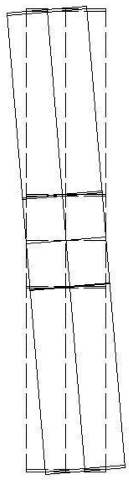
The structure is built up out of load-bearing light weight steel frames.

Figure 9.2: Picture of the Voorhof Student Building.

The building (Voorhof) is modeled by 19 elements. The elements are connected at nodes, each node has two degrees of freedom, which represent the displacement and the rotation of a floor (**Figure 9.2**).



Model a



Model b

Figure 9.3: a) Displacements and rotations of the floors b) Rotated of a representative cross-section.

9.2. Structural Characteristics

9.2.1. Cross-section properties

Centre of gravity

The centre of gravity of the cross-section is assumed to be located in the neutral-axis of the cross-section in x and y direction of the Voorhof building structure.

The bending stiffness in y direction before renovation

The bending stiffness in y direction of the Voorhof building before renovation

$EI_y = 2.53 * 10^{12} Nm^2$ ([17]). The bending stiffness had to be calibrated to fit the recorded natural frequency, $EI_{y\_calibrated} = 2.085 * 10^{12} Nm^2$. The calibarion is explained later in 9.2.2.

The bending stiffness in x direction before renovation

The bending stiffness in x direction of the Voorhof building was determined by multiplying the Young modulus of steel $E_s = 2.1E^5$ N/mm^2 $= 2.10E^{11}$ N/m^2 with the second moment of inertia in the x-direction.

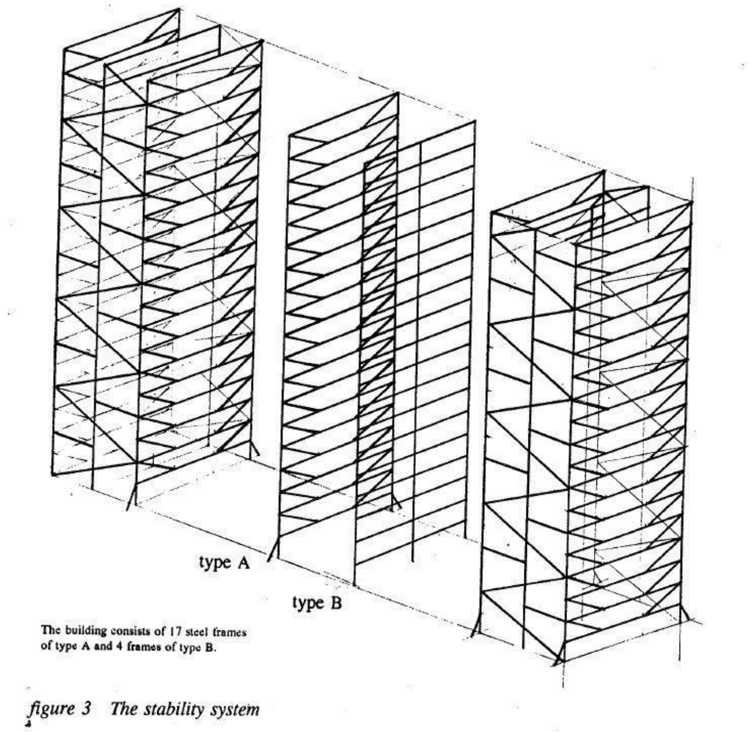


Figure 9.4: Schematization stability system Voorhof [17]

The second moment of inertia of the Voorhof building in the x-direction was calculated with the formula:

$$I_{x;ss} = \sum_{i=1}^{n} s_i^2 A_i$$

where:

$I_{x;ss}$	The second moment of inertia of the steel structure of the Voorhof building before renovation.
n	The number of steel cross-sections that are located in the building structure.
s_i	The perpendicular distance between the reference line and the centre of gravity of the considered steel cross-section. [17]
A_i	The area of the considered steel cross-section. [17]

The second moment of inertia in x direction was calculated in Matlab the outcome is $I_{x;ss} = 629.76 \text{ m}^4$.

The first summation term $\sum_{i=1}^{n} \frac{b_i h_i^3}{12}$ in the complete formula $I_{x;ss} = \sum_{i=1}^{n} \frac{h_i b_i^3}{12} + \sum_{i=1}^{n} s_i^2 A_i$ can be neglected because this term is a lot smaller than the second summation term $\sum_{i=1}^{n} s_i^2 A_i$.

The second moments of the cross-section <u>before renovation</u>

The second moments of the cross-section can be determined, once both bending stiffnesses are known in x and y direction. The known bending stiffness is divided by the Young modulus.

$$I_{x;b} = \frac{EI_x}{E} \qquad I_{y;b} = \frac{EI_y}{E}$$

where:

EI_x, EI_y	Bending stiffness in x and y direction.

E Young modulus.

$$I_{x;b} = \frac{1.32E^{14}}{2.1E^{11}} = 629.76 \text{ m}^4 \qquad I_{y;b} = \frac{2.53E^{12}}{2.1E^{11}} = 12.076 \text{ m}^4$$

The second moments of the cross-section after renovation are

$$I_x = I_{x;b} + I_{x;w} \qquad I_y = I_{y;b} + I_{y;w}$$

$$I_{x;w} = \sum_{i=1}^{n} \frac{b_i h_i^3}{12} + \sum_{i=1}^{n} s_i^2 A_i \qquad I_{y;w} = \sum_{i=1}^{n} \frac{h_i b_i^3}{12} + \sum_{i=1}^{n} s_i^2 A_i$$

where:

I_x, I_y The second moment of inertia of the Voorhof building after renovation.

$I_{x;b}, I_{y;b}$ The second moment of inertia of the Voorhof building before renovation.

$I_{x;w}, I_{y;w}$ The second moment of inertia of the added concrete walls.

n The number of added concrete walls in the cross-section of the building.

b The width of added concrete walls.

h The length of added concrete walls.

s_i The perpendicular distance between the reference line and the centre of gravity of the considered wall. [17]

A_i The area of the considered wall. [17]

The second moment of inertia in the x and y direction were calculated in Matlab. The outcome is $I_{x;w} = 8403.25 \text{ m}^4$ and $I_{y;w} = 85.74 \text{ m}^4$.

The second moment of inertia the x and y direction of the building after renovation are:

$$I_x = 629.76 + 8403.25 = 9033.01 \text{ m}^4 \qquad I_y = 12.076 + 85.74 = 97.82 \text{ m}^4$$

$$EI_x = 1.90E^{15} \text{ Nm}^2 \qquad EI_y = 2.05E^{13} \text{ Nm}^2$$

The value of the bending stiffness determined in a previous study [17] was put into Matlab, $EI_y = 5.17E^{12} \text{ Nm}^2$ and was later calibrated, $EI_{y\_calibrated} = 4.26E^{12} \text{ Nm}^2$ to match the recorded frequency, view 9.2.2.

The shear modulus before renovation is defined as

$$G_{ss} = G = \frac{E}{2(1+v_s)}$$

We assume $v_s = 0.25$ for the structural steel. The Young's modulus for structural steel is assumed $E = 210000$ N/mm^2 $= 2.1E^{11}$ N/m^2 [25], which lead to a shear modulus of $E/2.5 =$ N/mm^2 $= 8.4 * 10^9$ N/m^2 .

The shear modulus after renovation is defined as

For the steel structure:

$$G_{ss} = G = \frac{E}{2(1+v_s)} = 8.4 * 10^9 \text{ N/m}^2$$

For the added concrete walls:

$$G_w = \frac{E_c}{2(1+v_c)}$$

We assume $v_c = 0.15$ for the uncracked concrete. The concrete used for the additional structural walls is B22.5 The Young's modulus given for B22.5 $E'_b = 27875$ N/mm^2 $= 2.7872E^{10}$ N/m^2 [17] which lead to a shear modulus of $E/2.3 = 11150$ N/mm^2 $= 1.115 * 10^{10}$ N/m^2 .

$$G_w = 1.115E^{10} \text{ N/m}^2$$

Torsion Stiffness before renovation

The torsional stiffness of the cross section was determined by placing 4 floors of the steel structure of the Voorhof building [17] in Matrix frame CA out of which the

$$GJ = 4.53E^{10}\ \text{Nm}^2$$

The torsion stiffness had to be determined in this manner because no values could be found for the torsion stiffness in the available documentation.

The torsion stiffness had to be calibrated to fit the recorded natural frequency (view 9.2.3).

$$GJ_{calibrated} = 1.863E^{12}\ \text{Nm}^2$$

Calculation of the torsion stiffness

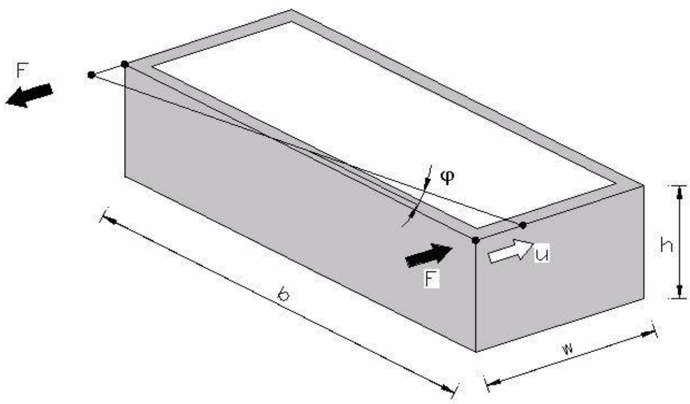


Figure 9.5: Calculation torsion stiffness

$$\left.\begin{aligned} M &= GJ\frac{\varphi}{h} \\ \varphi &= \frac{u}{\frac{1}{2}b} \\ M &= Fb \end{aligned}\right\} GJ = \frac{Fhb^2}{2u}$$

F	Force on structure in 3D model	[N]
u	Displacement calculated by 3D program	[m]
h	Height of 3D model	[m]
b	The length of 3D model, distance between both applied forces	[m]

$$GJ = \frac{Fhb^2}{2u} = \frac{200000 * 10.6 * 80^2}{2 * ((0.1529 + 0.1460)/2)} = 4.53E^{10} N/m^2$$

Calculation of the diagonal which replaces a floor

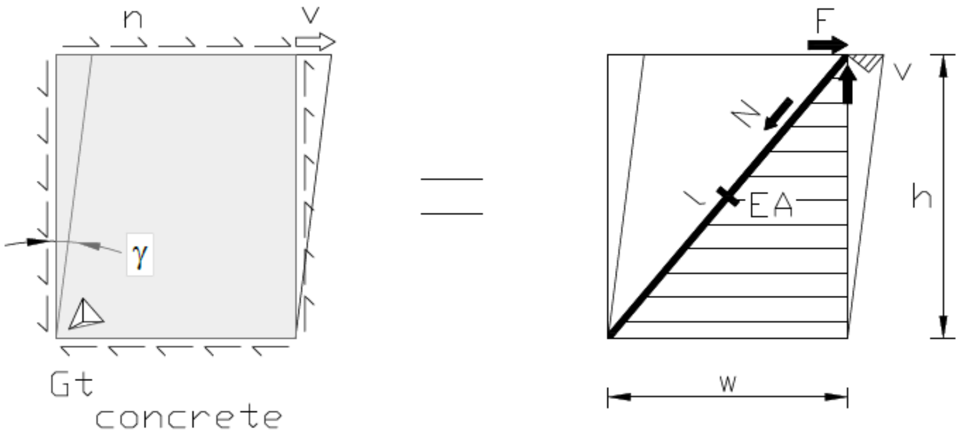


Figure 9.6: Diagonal which replaces a floor

$$\left.\begin{array}{l} n = Gt\gamma \\ \gamma = \frac{v}{h} \quad l^2 = w^2 + h^2 \quad \frac{\Delta l}{v} = \frac{w}{l} \\ F = nw \quad N = EA\frac{\Delta l}{l} \quad \frac{l}{w} = \frac{N}{F} \end{array}\right\} EA = Gt\frac{l^3}{wh}$$

G	shear modulus of concrete	[N/m^2]
t	thickness of the concrete floor	[m]
l	length of the diagonal	[m]
w, h	width and height of the concrete floor	[m]

$$EA = 1.212E^{10} * 0.13 * \frac{11.33^3}{10.6 * 4} = 5.40E^{10} \text{ N}$$

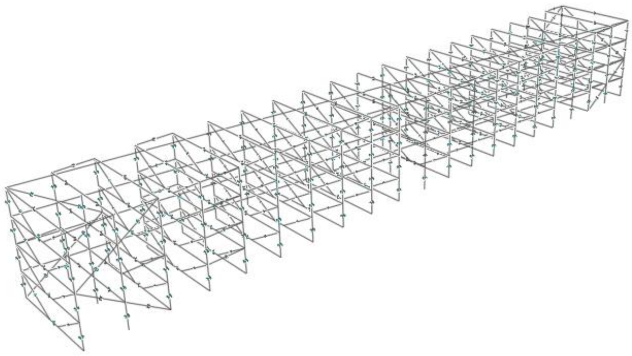


Figure 9.7: Voorhof structure in matrix

Torsion Stiffness after renovation

The torsional stiffness after renovation is the torsion stiffness before renovation plus the torsional stiffness of the additional 4 concrete walls. Because the building is modeled as a rectangular tube profile, the torsion stiffness of the added walls cannot be simply added to the torsion stiffness before renovation. The added concrete walls are added to the Matrix frame CAE model in a similar manner as the floor, a diagonal with calculated axial rigidity.

The torsional stiffness of the cross section after renovation was determined by placing 4 floors of the steel structure of the Voorhof building [17] in Matrix frame CA.

$$GJ = 1.3046E^{12} \text{ Nm}^2$$

The torsion stiffness had to be determined is this manner because no values could be found for the torsion stiffness in the available documentation.

$$GJ_{calibrated} = 3.187E^{12} \text{ Nm}^2 \text{ (view 9.2.3)}$$

Calculation of torsion Stiffness

$$GJ = \frac{Fhb^2}{2u} = \frac{200000 * 10.6 * 80^2}{2 * ((0.0037 + 0.0067)/2)} = 1.3046\text{E}^{12}\text{N/m}^2$$

Diagonal which replaces a floor

$$EA = 1.212E^{10} * 0.25 * \frac{10.97^3}{10.1 * 2.65} = 1.49E^{11} \text{ N}$$

9.2.2. Story Mass

Before renovation

The mass of the floor and the walls of one storey can be calculated with the formula:

$$M = \rho_c \sum_{i=1}^{n} h_i A_i + \rho_c h_{fl} A_{fl} + p_{vb} A_{fl}$$

where:

ρ_c specific gravity of reinforced concrete, $\rho_c = 2400$ kg/m^3 ;

h_i height of the walls, $h = 2.65$ m

A_i area of the walls

h_{fl} thickness of the floor, $h = 0.15$ m

A_{fl} area of the floor

p_{vb} The variable load on the floor , $p_{vb} = 58.3$ kg/m^2 $= 0.583$ kN/m^2 ;

Mass of the furniture, decorations and inhabitants are estimated at
$p_{vb} = 58.3$ kg/m^3 $= 0.583$ kN/m^2 [17]

The determining of the weight of the building ($10.943E^6$ kg) was done in a previous study [17], from which the mass of the floor can be approximately determined, $10.943E^6/19 = 0.576E^6$ kg . When the mass of $M_{1-18} = 580514$ kg , $M_{19} = 493753$ kg and $EI_y = 2.53\text{E}^{12} Nm^2$ was put into Matlab, the frequency of the first mode was $f_e = 0.69$ Hz . The bending modal parameters had to be calibrated ($M_{1-18} = 580514$ kg $M_{19} = 493753$ kg) and $EI_y = 2.085\text{E}^{12} Nm^2$) to match the cyclic frequency $f_e = 0.624$ Hz , which also was determined in a previous study [17].

After renovation

The determining of the weight of the building ($12.090E^{6}$ kg) was done in a previous study ([17] Appendix F p.6), from which the mass of the floor can be approximately determined, $12.090E^{6}/19 = 0.636E^{6}$ kg . When the mass of $M_{1-18} = 642499$ kg , $M_{19} = 525106$ kg and $EI_{y} = 5.17E^{12}\,Nm^{2}$ was put into Matlab, the frequency of the first mode which was calculated was $f_{e} = 0.94$ Hz . The bending modal parameters had to be calibrated ($M_{1-18} = 642499$ kg , $M_{19} = 525106$ kg and $EI_{y} = 4.26E^{12}\,Nm^{2}$) to match the cyclic frequency $f_{e} = 0.85$ Hz was determined in a previous study ([17] p.146 Appendix E p.2).

9.2.3. Polar Moment of Inertia

The polar moment of inertia of the wall cross section is

$$I_{p\_wall} = \int_{A} r^{2} dA = I_{x} + I_{y}$$

The polar moment of inertia of the floor cross section is

$$I_{p\_floor} = \int_{A} r^{2} dA = I_{x} + I_{y}$$

Before renovation

The structure is modeled as discontinuous, this entails that the polar moment of inertia of the structure is lumped in the nodes. The polar moment of inertia of each element is put on the diagonal in the polar moment of inertia matrix (before renovation).

$$I_{p\_floor} = \left(\frac{1}{12}BH^{3} + \frac{1}{12}HB^{3}\right) = 643738.7 \text{ m}^{4}$$

$$I_{p\_wall} = I_{x} + I_{y} = 641.8 \text{ m}^{4}$$

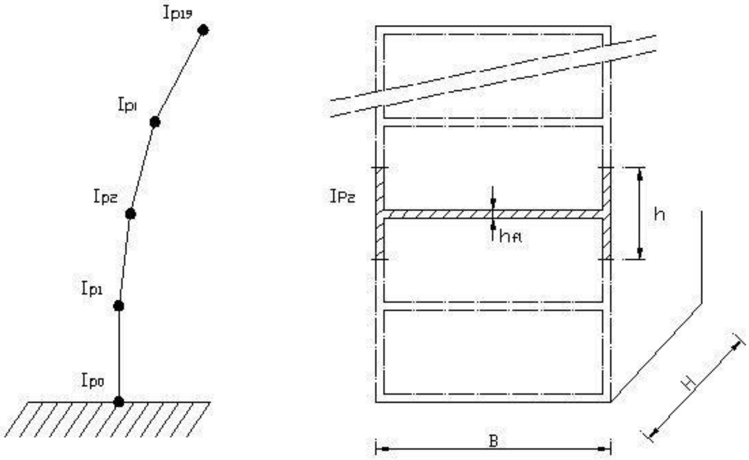


Figure 9.8: Lumping the polar moment of inertia in nodes

$$I_p = I_{p\_wall} h\rho + I_{p\_floor}\left(h_{fl}\rho + m_{vb}\right) = 2.827E^8 \ \text{kgm}^2$$

story height	$e = h = 2.65$ m	mean density	$\rho = 2400$ kg/m^3
floor height	$h_{fl} = 0.15$ m	width building	$B = 80.81$ m
depth building	$H = 14.20$ m	variable mass	$m_{vb} = 58.3$ kg/m^2

$$I_{p19} = I_{p\_wall}\frac{h}{2}\rho + I_{p\_floor}\left(h_{fl}\rho + m_{vb}\right) = 2.760E^8 \ \text{kgm}^2$$

The polar moment of inertia which are shown above were determined in Matlab (Appendix 6) by use of the given values ([17] Appendix F). The Torsion stiffness is $4.53E^{10}$ Nm^2 and was determined with the formulas in (Section 9.2.1). If both parameters were put into Mathlab the natural frequency of the torsional motion of the first mode calculated was $f_{e\_tor} = 0.10$ Hz. The natural frequency of the torsional motion recorded was $f_{e\_tor} = 0.64 - 0.75$ Hz ([8]) ([17] p.139). The torsional modal parameters had to be calibrated ($GJ = 1.863E^{12}$ Nm^2) to match the recorded cyclic frequency $f_{e\_tor} = 0.64$ Hz, lower bound of the measured torsional frequency. ([17] p.139 Appendix E p.2).

After renovation

The structure is modeled as discontinuous, this entails that the polar moment of inertia of the structure is lumped in the nodes. The polar moment of inertia of each element is put on the diagonal in the polar moment of inertia matrix.

$$I_{p\_floor} = \left(\frac{1}{12} BH^3 + \frac{1}{12} HB^3 \right) = 643738.7 \text{ m}^4$$

$$I_{p\_wall} = I_x + I_y = 9130.82 \text{ m}^4$$

$$I_p = I_{p\_wall} h\rho + I_{p\_floor} \left(h_{fl}\rho + m_{vb} \right) = 3.367E^8 \text{ kgm}^2$$

story height	$e = h = 2.65$ m	mean density	$\rho = 2400$ kg/m^3
floor height	$h_{fl} = 0.15$ m	width building	$B = 80.81$ m
depth building	$H = 14.20$ m	variable mass	$m_{vb} = 58.3$ kg/m^2

$$I_{p19} = I_{p\_wall} \frac{h}{2} \rho + I_{p\_floor} \left(h_{fl}\rho + m_{vb} \right) = 3.030E^8 \text{ kgm}^2$$

The polar moment of inertia which are shown above were determined in Matlab (Appendix 7) by use of the given values ([17] Appendix F). The torsion stiffness is $1.3046E^{12}$ Nm^2 and was determined with formulas (Section 9.2.1). When both parameters were put into Matlab the natural frequency of the torsional motion of the first mode calculated was $f_{e\_tor} = 0.49$ Hz. The natural frequency of the torsion recorded was $f_{e\_tor} = 0.77$ Hz ([17] p.139). The torsional modal parameters had to be calibrated ($GJ = 3.187E^{12}$ Nm^2) to match the cyclic frequency $f_{e\_tor} = 0.77$ Hz, lower bound of the measured recorded torsional frequency ([17] p.139 Appendix E p.2).

9.2.4. Natural frequencies

Before renovation

A study [17] was done by a previous student in which the frequencies for the bending and torsional motion are determined, f_{e_ben} = 0.625 Hz = 3.927 $\frac{rad}{s}$ and

f_{e_tor} = 0.64 - 0.75 Hz = 4.021 - 4.712 $\frac{rad}{s}$ ([8]) ([17] p.139).

The natural frequencies of the Voorhof building for bending and torsion before renovation have been calibrated and are shown in the tables below:

Natural frequencies Bending [rad/s]							
3.922	24.524	68.251	132.358	215.601	315.968	430.990	557.781
693.107	833.478	975.234	1114.654	1248.055	1371.898	1482.885	1578.054
		1654.857	1711.237	1745.682			

Table 11: Natural frequencies bending before renovation

Natural frequencies Torsion [rad/s]							
4.021	12.037	19.974	27.781	35.407	42.801	49.917	56.707
63.128	69.138	74.698	79.774	84.331	88.341	91.778	94.620
		96.850	98.453	99.419			

Table 12: Natural frequencies torsion before renovation

The recorded first natural frequency for bending ($3.927\ \frac{rad}{s}$) corresponds with the first natural frequency for bending in the model ($3.922\ \frac{rad}{s}$) (**Table 11**), this means that the bending motions (acceleration,velocity and displacement) in the model will be the same as reality.

The recorded first natural frequency for torison ($4.021 - 4.712 \frac{rad}{s}$)corresponds with the first natural frequency for torsion in the model ($4.021 \frac{rad}{s}$) (**Table 12**), this means that the torsional motion (acceleration,velocity and displacement)) in the model will be the same as reality.

After renovation

The given frequencies for the bending and torsional motion were determined by a previous student and are f_{e_ben} = 0.85 Hz = $5.341 \frac{rad}{s}$ and f_{e_tor} = 0.77 Hz = $4.838 \frac{rad}{s}$ ([8]) ([17] p.146).

The natural frequencies of the Voorhof building for bending and torsion after renovation have been calibrated and are shown in the tables below:

Natural frequencies Bending [rad/s]							
5.346	33.415	92.963	180.222	293.479	429.977	586.352	758.675
942.553	1133.243	1325.783	1515.123	1696.269	1864.420	2015.105	2144.305
		2248.569	2325.105	2371.863			

Table 13: Natural frequencies bending after renovation

Natural frequencies Torsion [rad/s]							
4.838	14.482	24.030	33.418	42.584	51.468	60.011	68.157
75.853	83.050	89.701	95.765	101.203	105.983	110.075	113.456
		116.105	118.008	119.155			

Table 14: Natural frequencies torsion after renovation

The recorded first natural frequency for bending ($5.341\ \frac{rad}{s}$) corresponds with the first natural frequency for bending in the model ($5.346\ \frac{rad}{s}$) (**Table 13**), this means that the bending motions (acceleration,velocity and displacement) in the model will be the same as reality.

The recorded first natural frequency for torison ($4.838\ \frac{rad}{s}$)corresponds with the first natural frequency for torsion in the model ($4.838\ \frac{rad}{s}$) (**Table 14**). This means that the torsional motion (acceleration,velocity and displacement) in the model will be the same as reality.

9.2.5. Damping ratios before and after renovation

Before renovation

The damping ratios of mode **1** and mode **2** are chosen to be specified.

For the Voorhof Building before renovation the damping ratio is selected for mode 1 $\varsigma_1 = 0.0108$ and for mode 2 is $\varsigma_2 = 0.0108$ ([17] Appendix I).

The Raleigh damping coefficients for bending are $a_{0ben} = 0.0730$ and $a_{1ben} = 0.0008$ for mode 1 and mode 2. When interpreting the values of the Raleigh damping constants for bending, one can see that the damping acts totally on the Modal mass matrix (low frequencies) $\omega_{e\_ben\_matlab} = 3.92 rad/s$.

The Raleigh damping coefficients for torsion are $a_{0tor} = 0.0651$ and $a_{1tor} = 0.0013$ for mode 1 and mode 2. When interpreting the values of the Raleigh damping constants for torsion, one can see that the damping acts nearly totally on the Modal polar moment of inertia matrix (low frequencies) $\omega_{e\_tor\_matlab} = 4.02 rad/s$.

After renovation

For the Voorhof Building after renovation the damping ratio is selected for mode 1 $\varsigma_1 = 0.0146$ and for mode 2 is $\varsigma_2 = 0.0146$. ([17] Appendix I)

The Raleigh damping coefficients for bending are $a_{0ben} = 0.1346$ and $a_{1ben} = 0.0008$ for mode 1 and mode 2. When interpreting the values of the Raleigh damping constants for bending, one can see that the damping acts totally on the Modal mass matrix (low frequencies) $\omega_{e\_ben\_matlab} = 5.35 rad/s$.

The Raleigh damping coefficients for torsion are $a_{0tor} = 0.1059$ and $a_{1tor} = 0.0015$ for mode 1 and mode 2. When interpreting the values of the Raleigh damping constants for torsion, one can see that the damping acts totally on the Modal polar moment of inertia matrix (low frequencies) $\omega_{e\_tor\_matlab} = 4.84 rad/s$.

9.3. Wind parameters for the Student building "Voorhof" according to NEN

The Dutch norm NEN 6702 [3] indicates that the Voorhof student building is located in **Urban area II**. We will use the values for **Urban area II** of the NEN 6702 [3]. The coefficients are shown in the table below.

	Rural	**Urban**
	II	II
$u_{*12,5}$	2.30 m/s	2.82 m/s
z_0	0.2 m	0.7 m
d	0 m	3.5 m
k	1.0 -	0.9 -

Table 15: Parameters for hourly averaged wind speed, NEN 6702 page 128 Table 6.1

The stability system of the **Voorhof Building** has 2 main directions; because of this the use of the reference period of 12.5 years is valid.

9.4. Wind parameters for the Student building "Voorhof" for the measured wind speeds

The parameters for the average wind speed before and after renovation for the Voorhof student building were determined in ([17] pp.131-138, Appendix E).

	Before	After
$u_{*12,5}$	2.25 m/s	1.18 m/s
z_0	1.0 m	1.0 m
d	10.0 m	10.0 m
k	1.0 -	1.0 -

Table 16: Parameters for hourly averaged wind speed, ([17] Appendix E)

The manner in which the wind velocity is transformed from Rotterdam Airport to the Voorhof Student Building in Delft is not correct ([17] pp.131-138, Appendix E). The wrong transformation formula without the temperature as a coefficient is used. The wind velocity transformation which was done is applicable to determine the average wind speed in a year and not for short periods of 10 minutes. The temperature is a very important factor for wind velocity transformations with short periods, making it not possible to determine the wind velocity accurately without it.

Because no better values are available, the values ([17] pp.131-138, Appendix E) will be used even thought it is known that these values are not 100% correct.

9.5. Hourly-averaged part of the wind speed for return period of 12.5 years

The NEN 6702 describes the hourly-averaged wind speed which varies with the height as: ([3])

$$\bar{v}(z) = \frac{u_*}{\kappa} \ln\left(\frac{z - d_w}{z_0}\right)$$

In which:

u_*	Friction velocity [m/s]	= 2.82	[m/s]
κ	Von Karman constant [1];	= 0.4	[-]
z	height above the surface of the earth [m]	= 51.3	[m]
d_w	displacement height [m]	= 3.5	[m]
z_0	roughness length [m]	= 0.7	[m] ([17] Appendix E)

From the values in Table 6.1 we acquired the profile for the extreme hourly wind speed according to the NEN 6702 ([3])

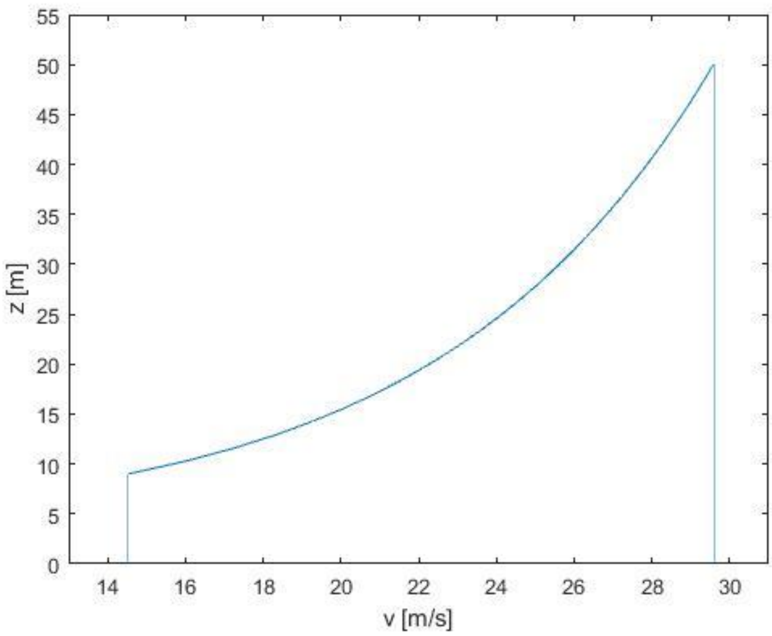


Figure 9.9: Extreme hourly-averaged wind speed profile for return period of 12.5 years or for return period of 50 years under unfavorable direction in Delft

Harris and Deaves describes the hourly-averaged wind speed which varies the height: [13]

$$\bar{v}(z) = \frac{u_*}{\kappa}\left(\ln\left(\frac{z-d}{z_0}\right) + 5.75a - 1.88a^2 - 1.33a^3 + 0.25a^4\right)$$

With: $a = \left(\frac{z-d}{z_g}\right)$, $z_g = \frac{u_*}{6f_c}$ and $f_c = 2\Omega \sin\lambda$

In which:

u_*	Friction velocity [m/s]	= 2.82	[m/s]
κ	Von Karman constant [1];	$= 0.4$	[-]
z	height above the surface of the earth [m]	$= 51.3$	[m]
d	displacement height [m]	$= 3.5$	[m]
z_0	roughness length [m]	$= 0.7$	[m]
f_c	Coriolisparamenter [s^{-1}]		
Ω	rotation speed of the earth [rad/s]	$\left(2\pi/24*60*60 = 7.2722E^{-5}\ rad/s\right)$	
λ	width degree [$^\circ$]	$\lambda = 52.00^\circ$ (Delft)	

From the values in Table 6.1 we acquired the wind profile for the extreem hourly wind speed of Harris and Deaves.

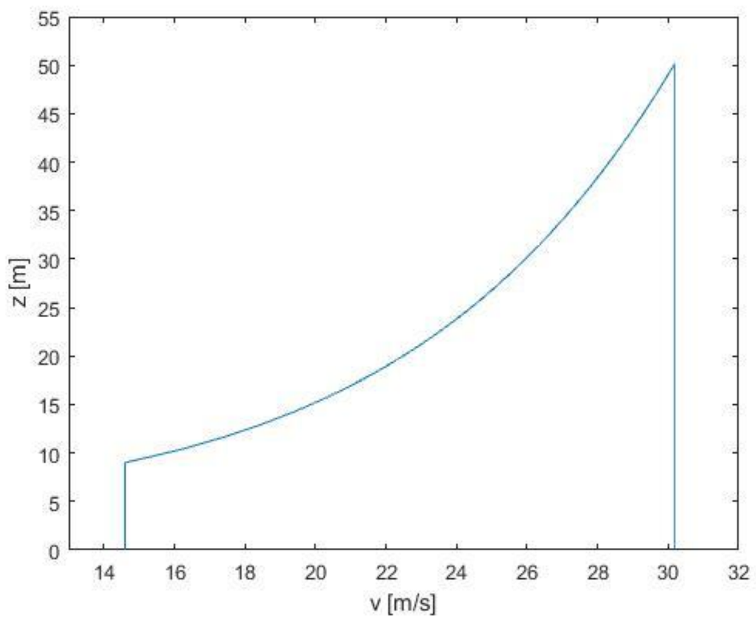


Figure 9.10: Extreme hourly-averaged wind speed profile for return period of 12.5 years or for return period of 50 years under unfavorable direction in Delft

Eurocode describes the hourly-averaged wind speed which varies the height: [13]

$$\bar{v}(z) = u_{ref} k_t \ln\left(\frac{z}{z_0}\right)$$

With: $u_{ref} = c_{dir} * c_{temp} * c_{alt} * u_{ref,0}$, $k_t = 0.19 * \left(\frac{z_0}{z_{0,\text{II}}}\right)^{0.0706}$

In which:

$\bar{v}(z)$	mean wind speed at height z [m/s]	=	[m/s]
u_{ref}	reference speed at height of 10 m [m/s]	$= 25$	[m/s]
k_t	terrain factor [-]	$= 0.21$	[-]
z	height above the face of the earth [m]	$= 51.3$	[m]
z_0	measure for the roughness of the terrain [m]	$= 0.2$	[m]
c_{dir}	direction factor [-]	$= 1$	[-]
c_{temp}	season factor [-]	$= 1$	[-]
c_{alt}	height factor [-]	$= 1$	[-]
$u_{ref,0}$	onalterd reference speed at height of 10 m [m/s]	$= 25$	[m/s]
$z_{0,\text{II}}$	roughness length	$= 0.05$	[m]

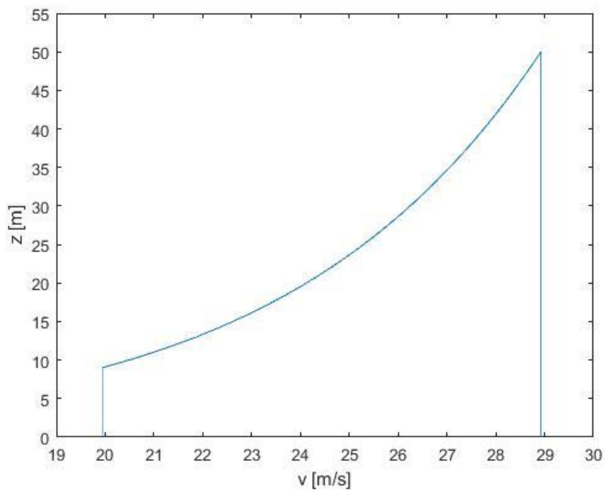


Figure 9.11: Extreme hourly-averaged wind speed for eurocode for return period of 12.5 years or return period of 50 years under unfavorable direction in Delft

9.6. Fluctuating part of wind speed for return period of 12.5 years

For the Voorhof the standard deviation of wind speed at reference height 10 m is determined below:

$$\sigma_v = \frac{ku_*}{\kappa} = \frac{0.9 * 2.82}{0.4} = 6.345 \text{ m/s}$$

In which:

u_*	Friction velocity	= 2.82	[m/s]
κ	Von Karman constant	$= 0.4$	[-]
k	Shape factor	$= 0.9$	[-]

The standard deviation of the wind speed varies per height. The standard deviation of the wind speed at reference height has been determined above. The standard deviation of the wind speed on the top of the building is determined with formula:

$$\sigma_v(z) = \sigma_v(h_0)\left(\frac{z-d}{h_0}\right)^{\delta}$$ ([1] Ch.6 p.10).

In which:

$\sigma_v(h_0)$	Standard deviation of wind speed at reference height	= 6.345	[m/s]
h_0	reference height	= 10	[-]
z	height above the surface of the earth	= 51.3	[m]
d	displacement height	= 3.5	[m]
δ	power	= 0.03	[-]

For the Voorhof the values for the wind speed, standard deviation of wind speed and turbulence intensity are shown for the top of the building and reference height of 10 meters. The wind speed at reference height and standard deviation of wind speed on the top of building are put into Matlab in the wind generator, if the ultimate state load has to be determined.

Height m	$V_{\text{Harris \& Deaves}}$ m/s	σ_V m/s	I -
51.3	30.17	6.65	0.22
10	15.79	6.35	0.40

[13] (p. 84)

Table 17: Mean wind speed, standard deviation and turbulence intensity

9.7. Hourly-averaged part of the wind speed for return period of one year

The formulas are given to determine the mean and standard deviation for maximum hourly-averaged wind speed with the return period of 1 year in ([1] Ch.6 p.7). The formulas are given below:

$$\mu(v_1) = u_1 + \frac{0.577}{a} \qquad \sigma(v_1) = \frac{\pi}{\left(a\sqrt{6}\right)}$$

Ihe values are given for a and u_1 for area II in the Netherlands In table 6.3 ([1] Ch.6 p.7). These values are $a = 0.55\frac{s}{m}$ and $u_1 = 20.4\frac{m}{s}$. Out of which we can determine that $\mu(v_1) = 20.4 + \frac{0.577}{0.55} = 21.45\frac{m}{s}$ and $\sigma(v_1) = \frac{\pi}{\left(0.55 * \sqrt{6}\right)} = 2.33\frac{m}{s}$.

The hourly-averaged wind speed described by a logarithmic function is given ([1] Ch.6 p.2). ([3])

$$\bar{v}(z) = \frac{u_*}{\kappa}\ln\left(\frac{z-d}{z_0}\right)$$

For the Juffertoren we know that some of the values are given and known (6.2) for the situation $\bar{v}(10)$ and only the u_* friction velocity has to be determined.

κ	Von Karman constant [1];	$= 0.4$	[-]
z	height above the surface of the earth [m]	$= 10$	[m]
d	displacement height [m]	$= 3.5$	[m]
z_0	roughness length [m]	$= 0.7$	[m]

With the values above $u_* = 3.851\frac{m}{s}$.

For the Harris and Deaves and Eurocode profile the unknown friction velocity (u_*) andonalterd reference speed at height of 10 m ($u_{ref,0}$) are determined in the same method as above.

The NEN 6702 describes the hourly-averaged wind speed which varies with the height as: ([3])

$$\bar{v}(z) = \frac{u_*}{\kappa} \ln\left(\frac{z - d_w}{z_0}\right)$$

In which:

u_*	Friction velocity [m/s]	$= 3.851$	[m/s]
κ	Von Karman constant [1];	$= 0.4$	[-]
z	height above the surface of the earth [m]	$= 51.3$	[m]
d_w	displacement height [m]	$= 3.5$	[m]
z_0	roughness length [m]	$= 0.7$	[m]

From the values in Table 6.1 we acquired the profile for the extreme hourly wind speed according to the NEN 6702 ([3])

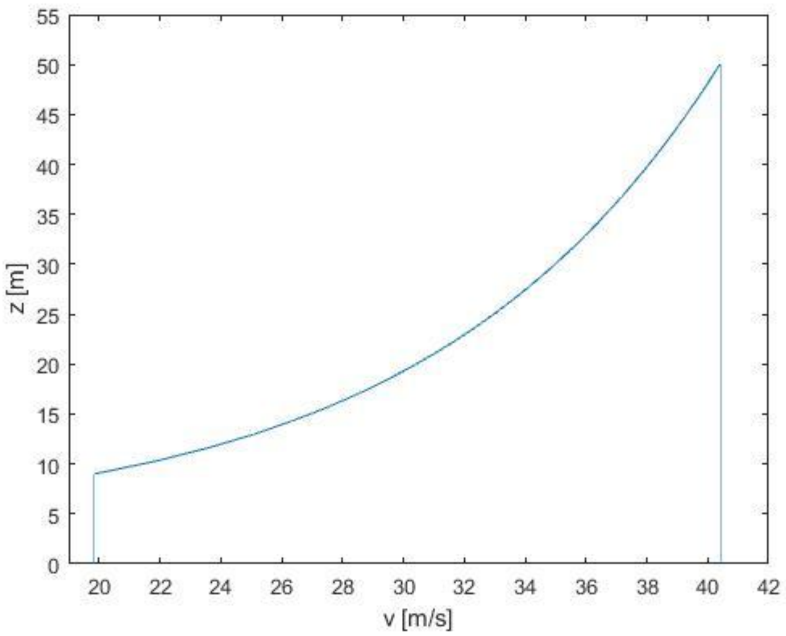


Figure 9.12: Extreme hourly-averaged wind speed profile for return period of one year in Delft

Harris and Deaves describes the hourly-averaged wind speed which varies the height: [13]

$$\bar{v}(z) = \frac{u_*}{\kappa}\left(\ln\left(\frac{z-d}{z_0}\right) + 5.75a - 1.88a^2 - 1.33a^3 + 0.25a^4\right)$$

With: $a = \left(\frac{z-d}{z_g}\right)$, $z_g = \frac{u_*}{6f_c}$ and $f_c = 2\Omega \sin\lambda$

In which:

u_*	Friction velocity [m/s]	= 3.835 [m/s]
κ	Von Karman constant [1];	$= 0.4$ [-]
z	height above the surface of the earth [m]	$= 51.3$ [m]
d	displacement height [m]	$= 3.5$ [m]
z_0	roughness length [m]	$= 0.7$ [m]
f_c	Coriolisparamenter [s^{-1}]	
Ω	rotation speed of the earth [rad/s]	$(2\pi/24*60*60 = 7.2722E^{-5}\ rad/s)$
λ	width degree [$^\circ$]	$\lambda = 52.00^\circ$ (Delft)

From the values above, we acquired the wind profile for the extreem hourly wind speed of Harris and Deaves.

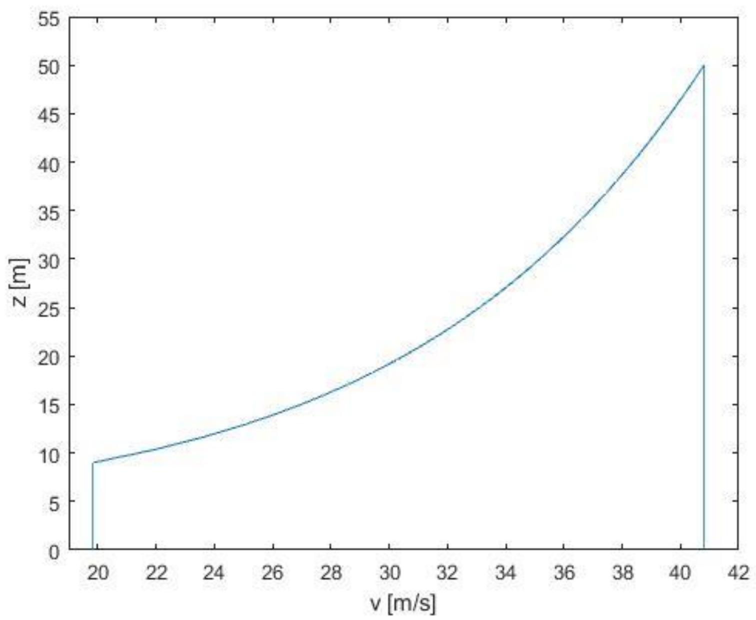


Figure 9.13: Extreme hourly-averaged wind speed profile for return period of one year in Delft

Eurocode describes the hourly-averaged wind speed which varies the height: [13]

$$\bar{v}(z) = u_{ref} k_t \ln\left(\frac{z}{z_0}\right)$$

With: $u_{ref} = c_{dir} * c_{temp} * c_{alt} * u_{ref,0}$, $k_t = 0.19 * \left(\frac{z_0}{z_{0,\text{II}}}\right)^{0.0706}$

In which:

$\bar{v}(z)$	mean wind speed at height z [m/s]	=	[m/s]
u_{ref}	reference speed at height of 10 m [m/s]	$= 26.17$	[m/s]
k_t	terrain factor [-]	$= 0.21$	[-]
z	height above the face of the earth [m]	$= 51.3$	[m]
z_0	measure for the roughness of the terrain [m]	$= 0.2$	[m]
c_{dir}	direction factor [-]	$= 1$	[-]
c_{temp}	season factor [-]	$= 1$	[-]
c_{alt}	height factor [-]	$= 1$	[-]
$u_{ref,0}$	onalterd reference speed at height of 10 m [m/s]	$= 26.17$	[m/s]
$z_{0,\text{II}}$	roughness length	$= 0.05$	[m]

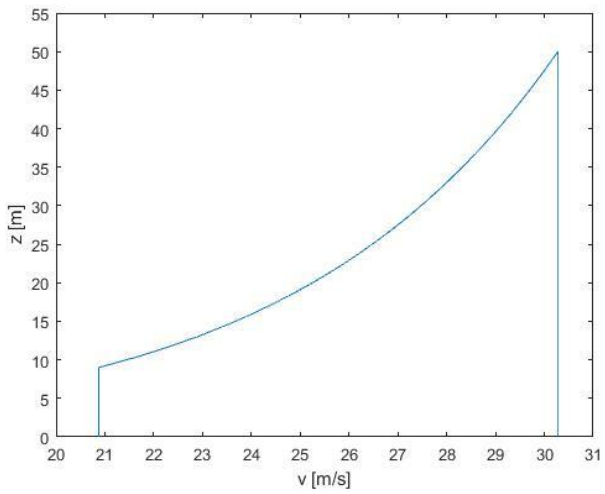


Figure 9.14: Extreme hourly-averaged wind speed for eurocode for return period of one year in Delft

9.8. Fluctuating part of wind speed for return period of one year

For the Voorhof:

$$\sigma_v = 2.33 \text{ m/s} \quad (0)$$

The standard deviation of the wind speed varies per height. The standard deviation of the wind speed on the top of the building is determined with formula:

$$\sigma_v(z) = \sigma_v(h_0)\left(\frac{z-d}{h_0}\right)^{\delta} \quad \text{([1] Ch.6 p.10).}$$

In which:

$\sigma_v(h_0)$	Standard deviation of wind speed at reference height	= 2.33	[m/s]
h_0	reference height	= 10	[-]
z	height above the surface of the earth	= 51.3	[m]
d	displacement height	= 3.5	[m]
δ	power	= 0.03	[-]

For the Juffertoren the values for the wind speed, standard deviation of wind speed and turbulence intensity are shown for the top of the building and reference height of 10 meters. The wind speed at reference height and standard deviation of wind speed on the top of building are put into Matlab in the wind generator. (Appendix 6)

Height m	V Harris & Deaves m/s	σ_v m/s	I -
51.3	40.83	2.44	0.06
10	21.45	2.33	0.11

[13] (p. 84)

Table 18: Mean wind speed, standard deviation and turbulence intensity

9.9. Hourly-average part of recorded wind speeds

In this paragraph the recorded hourly-average wind speeds are presented.

9.9.1. Recorded hourly-average wind velocity before renovation 18.4 m/s

In a previos study ([17] Appendix E) the wind speed profile before renovation was recorded on 14 January 1986.

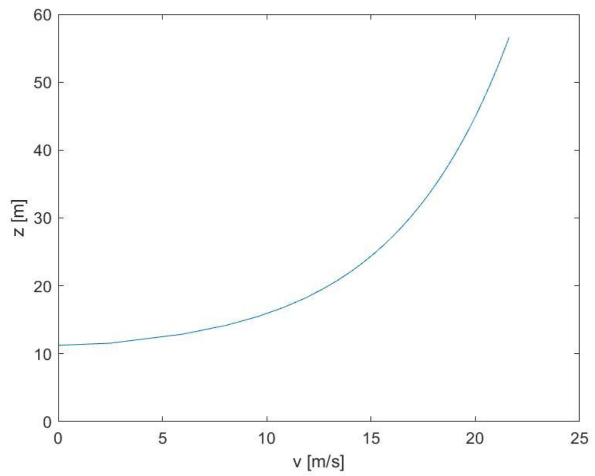


Figure 9.15: transformed hourly-averaged wind speed profile recorded 14 January 1986 unmodified in Delft

The hourly-mean wind speed before renovation is shown below, which is $18.4m/s$ at 10 m height, because this profile is only valid above .

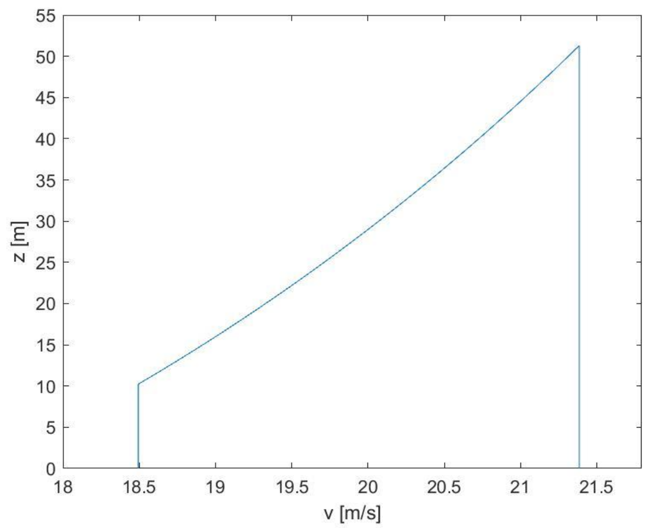


Figure 9.16: transformed hourly-averaged wind speed profile recorded 14 January 1986 modified in Delft

9.9.2. Recorded hourly-average wind velocity after renovation 9.2 m/s

In a previos study ([17] p.139) the wind speed profile after renovation was recorded on 31 Oktober 1994.

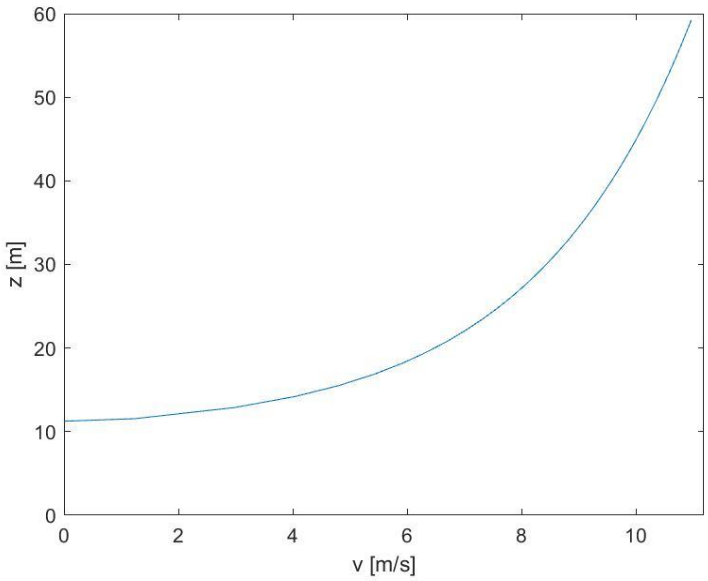


Figure 9.17: hourly-averaged wind speed profile recorded 31 October 1994 unmodified in Delft

The hourly-mean wind speed after renovation was determined below, which is $9.2m/s$ at 10 meters height, because this profile is only valid above $9.2m/s$.

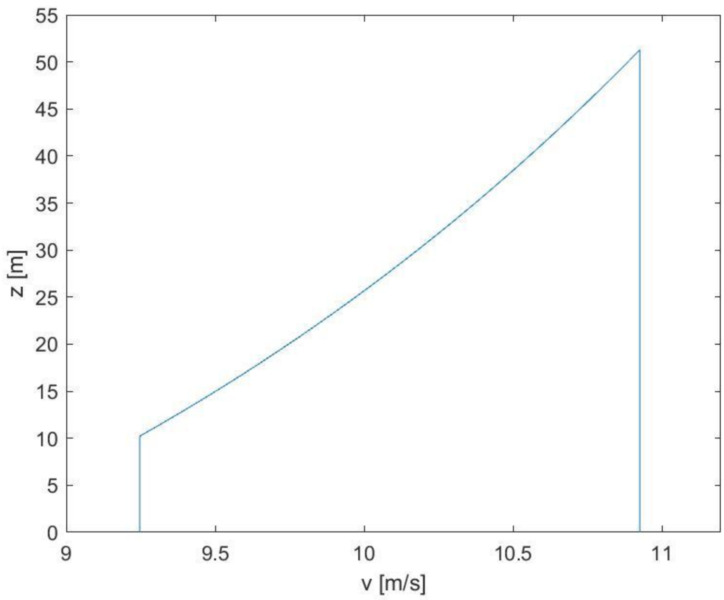


Figure 9.18: hourly-averaged wind speed profile recorded 31 October 1994 modified in Delft

9.10. Fluctuating part of recorded wind speed [17]

Fluctuating wind velocity before renovation

$$I(z) = \frac{k_c}{\ln\left(\frac{z - d_w}{z_0}\right)}$$ Turbulence intensity factor;

For the Voorhof building:

$k_c = 1.0$ is a factor

$d_w = 10$ ([17] Appendix E p.2)

$z_0 = 1.0$ ([17] Appendix E p.2)

$$I(z) = \frac{1}{\ln\left(\frac{z - d}{z_0}\right)}$$

$$I(51.3) = \frac{1}{\ln\left(\frac{51.3 - 10}{1}\right)} = 0.269$$

$$I(10) = \frac{\sigma_v}{\bar{V}} = \frac{5.60}{18.4} = 0.304$$

Using the mean wind speed $\bar{V}$ from section 6.3 the formula for the standard deviation $\sigma = \bar{V} * I$ results in:

$u_* = 2.25$ ([17] Appendix E p.3)

$$\sigma_v = \frac{ku_*}{\kappa} = \frac{1.0 * 2.25}{0.4} = 5.6 \text{ m/s}$$

The standard deviation of the wind speed varies per height. This value corresponds to the value at the 10 m height of the modeled building.

For the 14 of January 1986 the recorded mean wind speed, standard deviation and turbulence intensity are given in the table below.

Height m	V $_{Mean}$ m/s	σ_v m/s	I -
51.3	21	5.64	0.269
10	18.4	5.60	0.304

Table 19: Recorded mean wind speed, standard deviation and turbulence intensity before renovation

Fluctuating wind velocity after renovation

For the 31 of Oktober 1994 the recorded mean wind speed, standard deviation and turbulence intensity are given in the table below.

Height m	V $_{Mean}$ m/s	σ_v m/s	I -
51.3	11	2.96	0.269
10	9.2	2.95	0.321

Table 20: Recorded mean wind speed, standard deviation and turbulence intensity after renovation

10. Calculation results Student building "Voorhof"

In this chapter the modeled calculation results of the student building "Voorhof" (bending and torsional acceleration) before and after renovation are presented for the comfort requirement and the recorded wind speeds and accelerations. The summary values of a 100 simulations in time-domain are also presented. The Simulink outcome for acceleration before and after renovation is presented and evaluated. For all the realisations the first 50 seconds of the modeled acceleration is not taken into account when determining the maximum acceleration and standard deviation.

10.1. Before renovation with calibration

10.1.1. 100 simulations in Time Domain

In the table below the summary values of a 100 simulations of the Student building "Voorhof" before renovation are given. The average values and standard deviation for bending, torsion and bending and torsion added together are given. The maximum and minimum of the 100 simulations are also given. We can conclude that there is a big spread between the maximum and minimum value. For the values out of the simulations view Appendix 8. The time of 1 simulation is about 16 minutes.

Average	0.0984	0.2607	0.2968		0.0285	0.0745	0.0746
	a_ben	a_tor	a_ben_tor		sigma_a_ben	sigma_a_tor	sigma_a_ben_tor
Max	0.1520	0.4306	0.4847		0.0386	0.0909	0.0944
Min	0.0720	0.2073	0.2302		0.0237	0.0632	0.0620
Max - Average	0.0536	0.1699	0.1879		0.0101	0.0164	0.0198
Average - Min	0.0264	0.0534	0.0666		0.0048	0.0114	0.0126
Percentage Max	54.49%	65.20%	63.31%		35.61%	21.96%	26.53%
Percentage Min	26.88%	20.49%	22.44%		16.90%	15.26%	16.90%

Table 21: Summary values of a 100 simulations of the Voorhof building before renovation for return period of one year

10.1.2. 10 simulations in Tine Domain

In the table below the values and summary values of 10 simulations of the Student building "Voorhof" before renovation are given. The average values and standard deviation for bending, torsion and bending and torsion added together are given.

	a_{ben} m/s²	a_{tor} m/s²	a_{ben_tor} m/s²		σ_{a_ben} m/s²	σ_{a_tor} m/s²	$\sigma_{a\_ben\_tor}$ m/s²
Simulation 1	0.070	0.232	0.243		0.021	0.070	0.073
Simulation 2	0.095	0.332	0.336		0.027	0.075	0.076
Simulation 3	0.080	0.319	0.324		0.027	0.088	0.086
Simulation 4	0.089	0.239	0.273		0.025	0.064	0.064
Simulation 5	0.108	0.216	0.243		0.029	0.069	0.069
Simulation 6	0.104	0.272	0.301		0.034	0.080	0.085
Simulation 7	0.140	0.311	0.328		0.031	0.076	0.076
Simulation 8	0.113	0.291	0.317		0.028	0.073	0.073
Simulation 9	0.101	0.269	0.285		0.029	0.069	0.072
Simulation 10	0.095	0.271	0.308		0.028	0.078	0.077
Summation	**0.0993**	**0.2753**	**0.2958**		**0.0279**	**0.0740**	**0.0749**

Table 22: Values of 10 simulations of the Voorhof building before renovation for return period of one year

We notice that there is a difference between the average values of a 100 and 10 simulations. How much simulations must be run to get a good average.

10.2. After renovation with calibration

10.2.1. 100 simulations in Time Domain

In the table below the summary values of a 100 simulations of the Student building "Voorhof" after renovation are given. The average values and standard deviation for bending, torsion and bending and torsion added together are given. The maximum and minimum of the 100 simulations are also given. We can conclude that there is a big spread between the maximum and minimum value. For the values out of the simulations view Appendix 9. The time of 1 simulation is about 16 minutes.

Average	0.0522	0.1554	0.1696		0.0150	0.0435	0.0456
	a_ben	a_tor	a_ben_tor		sigma_a_ben	sigma_a_tor	sigma_a_ben_tor
Max	0.0650	0.1971	0.2201		0.0178	0.0505	0.0529
Min	0.0427	0.1209	0.1292		0.0128	0.0370	0.0397
Max - Average	0.0129	0.0417	0.0506		0.0028	0.0070	0.0072
Average - Min	0.0094	0.0345	0.0404		0.0022	0.0065	0.0059
Percentage Max	24.66%	26.84%	29.83%		18.71%	16.09%	15.84%
Percentage Min	18.07%	22.20%	23.81%		14.76%	14.96%	13.04%

Table 23: Summary values of a 100 simulations of the Voorhof building after renovation for return period of one year

10.2.2. 10 simulations in Time Domain

In the table below the values and summary values of 10 simulations of the the Student building "Voorhof" after renovation are given. The average values and standard deviation for bending, torsion and bending and torsion added together are given.

	$a_{\_ben}$ m/s^2	$a_{\_tor}$ m/s^2	$a_{\_ben\_tor}$ m/s^2	σ_{a_ben} m/s^2	σ_{a_tor} m/s^2	$\sigma_{a\_ben\_tor}$ m/s^2
Simulation 1	0.060	0.151	0.158	0.016	0.045	0.047
Simulation 2	0.057	0.145	0.142	0.016	0.045	0.047
Simulation 3	0.051	0.165	0.169	0.014	0.046	0.048
Simulation 4	0.051	0.121	0.129	0.014	0.039	0.042
Simulation 5	0.050	0.159	0.177	0.016	0.046	0.049
Simulation 6	0.052	0.148	0.147	0.015	0.043	0.046
Simulation 7	0.050	0.144	0.157	0.015	0.041	0.043
Simulation 8	0.058	0.156	0.168	0.016	0.039	0.042
Simulation 9	0.058	0.142	0.170	0.016	0.043	0.045
Simulation 10	0.047	0.164	0.172	0.014	0.047	0.048
Summation	**0.0534**	**0.1496**	**0.1589**	**0.0152**	**0.0434**	**0.0459**

Table 24: Values of 10 simulations of the Voorhof building after renovation for return period of one year

We notice that there is a slight difference between the average values of a 100 and 10 simulations.

10.3. The acceleration realizations at 4 different point in time

In the following paragraph, the accelerations (bending, torsion and bending and torsion added together) are determined and shown for 4 point in time, 2 of which are recorded mean wind velocities [17] (Appendix I) and 2 according to the NEN and compared with earlier determined values.

The reason for doing this is to show that with 1 simulation it can be proven that a building (Voorhof) may not meet the comfort requirement ($< 0.1\frac{m}{s^2}$) and that it is possible to model the motions of a building in a storm accurately.

10.3.1. Before renovation u_p = 18.4 m/s (Recorded in reality [17] Appendix E)

For the recorded time instance u_p = 18.4 m/s and $\sigma_v = 5.64$ m/s, the modeled accelerations are presented below, bending, torsion and bending and torsion added together.

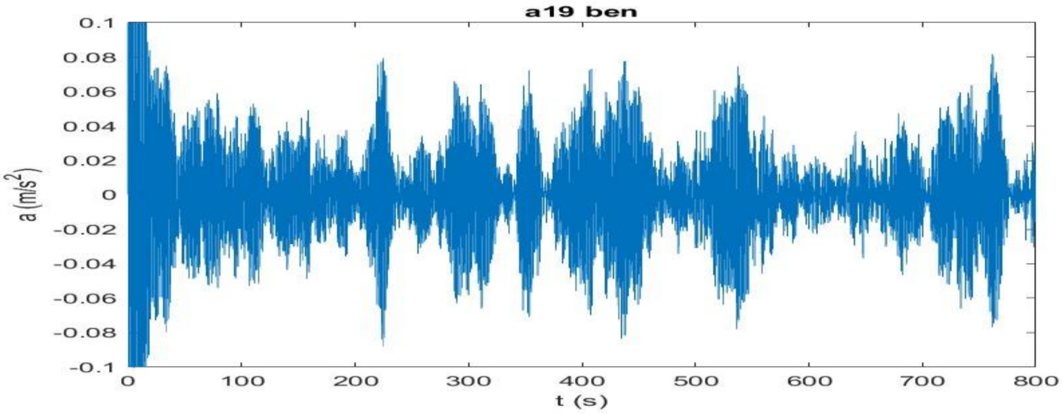


Figure 10.1: Wind speed calculation: bending acceleration of the top

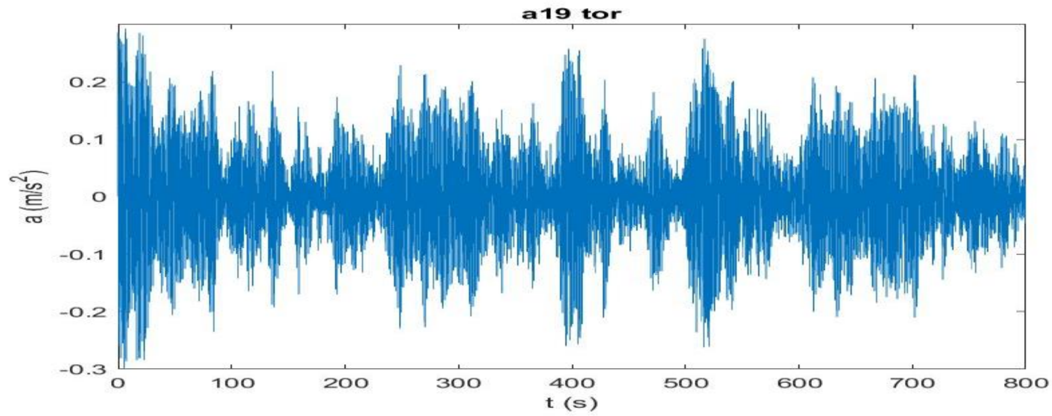


Figure 10.2: Wind speed calculation: torsional acceleration of the top

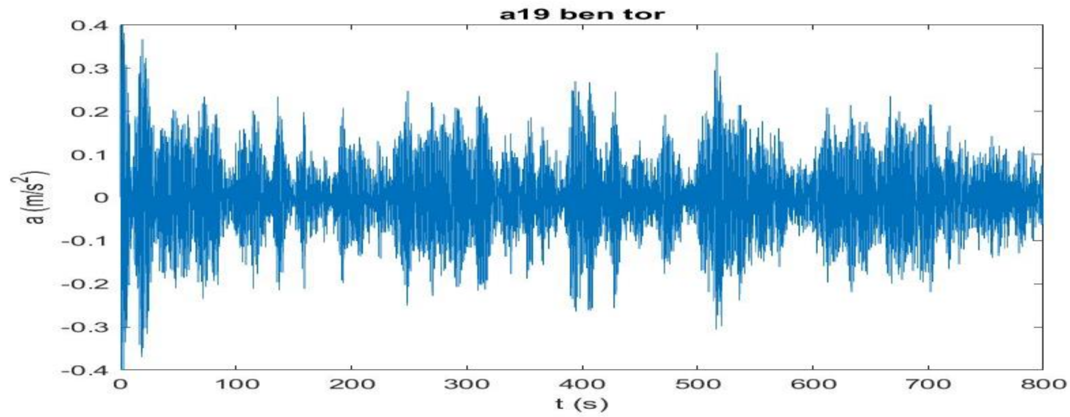


Figure 10.3: Wind speed calculation: bending and torsional acceleration of the top

For the recorded time instance u_p = 18.4 m/s and $\sigma_v = 5.64$ m/s, we can conclude for this realization that the acceleration for torsion and bending and torsion added together does not meet the comfort requirement in this storm.

The maximum recorded acceleration at the top and far end of the building before renovation [17] (Appendix E) was $0.0767\frac{m}{s^2}$ with $u(51.3m) = 21.0\frac{m}{s}$, $u_p(10m) = 18.4\frac{m}{s}$ and $u_* = 2.25\frac{m}{s}$.

The average bending acceleration (new method) before renovation was $0.1067\frac{m}{s^2}$ and the average bending and torsional acceleration added together was $0.3508\frac{m}{s^2}$ (Appendix 10). These accelerations values (new method) where taking out of the average of 50 simulations.

It can be concluded that this model did not fit the recorded measurements. One option is that the recorded value by company Van Dorsser was incorrect. The model after renovation fits perfectly with the measured data.

10.3.2. Before renovation u_p = 21.45 m/s (Return period once in 1 year)

For the NEN u_p = 21.45 m/s and $\sigma_v = 2.44$ m/s, the modeled accelerations are presented below, bending, torsion and bending and torsion added together.

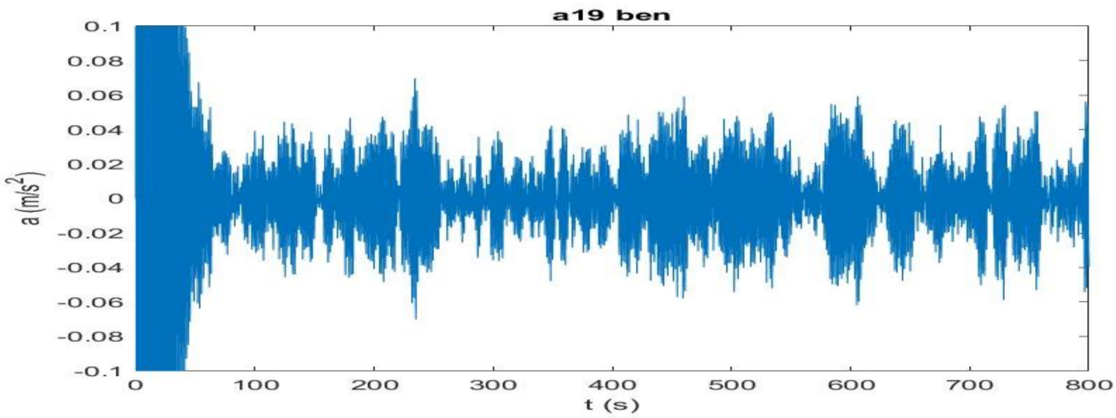


Figure 10.4: Wind speed calculation: bending acceleration of the top

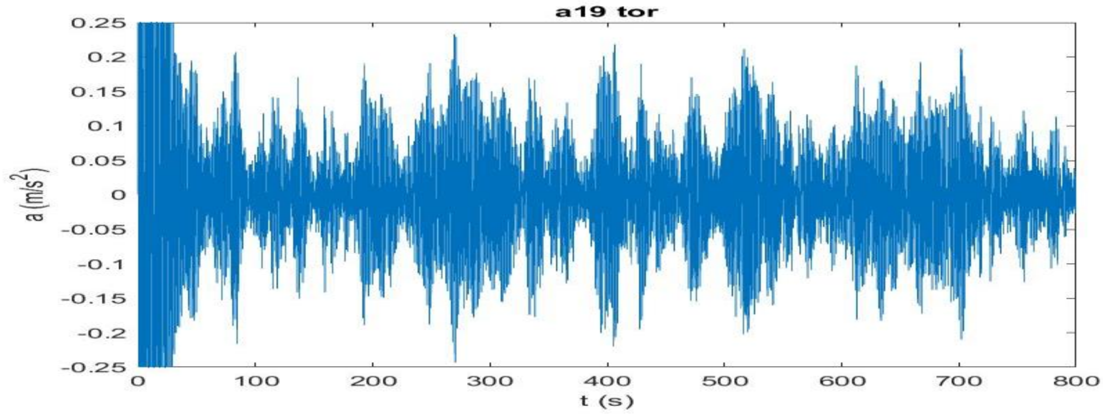


Figure 10.5: Wind speed calculation: torsional acceleration of the top

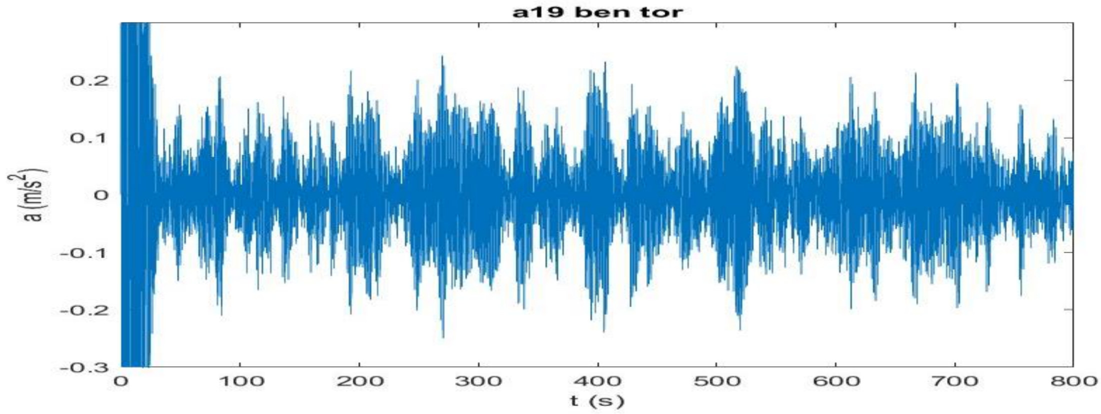


Figure 10.6: Wind speed calculation: bending and torsional acceleration of the top

For the recorded time instance u_p = 21.45 m/s and $\sigma_v = 2.44$ m/s , we can conclude for this realization that the acceleration for torsion and bending and torsion added together does not meet the comfort requirement.

The average bending acceleration according to NEN (new method) before renovation was $0.0984\frac{m}{s^2}$ and the average bending and torsional acceleration added together was $0.2968\frac{m}{s^2}$.

These accelerations values above (new method) where taking out of the average of a 100 simulations.

10.3.3. After renovation u_p = 9.2 m/s (Recorded in reality [17] Appendix E)

For the recorded time instance u_p = 9.2 m/s and $\sigma_v = 2.96$ m/s, the modeled accelerations are presented below, bending, torsion and bending and torsion added together.

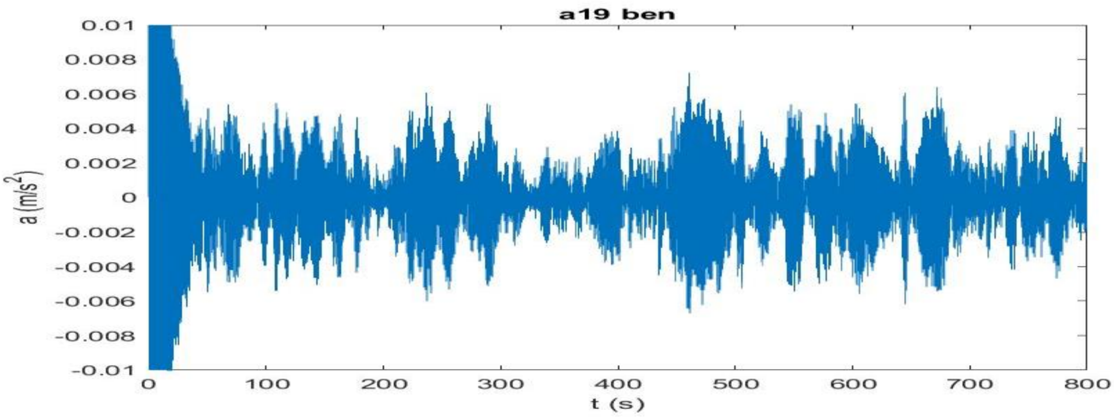


Figure 10.7: Wind speed calculation: bending acceleration of the top

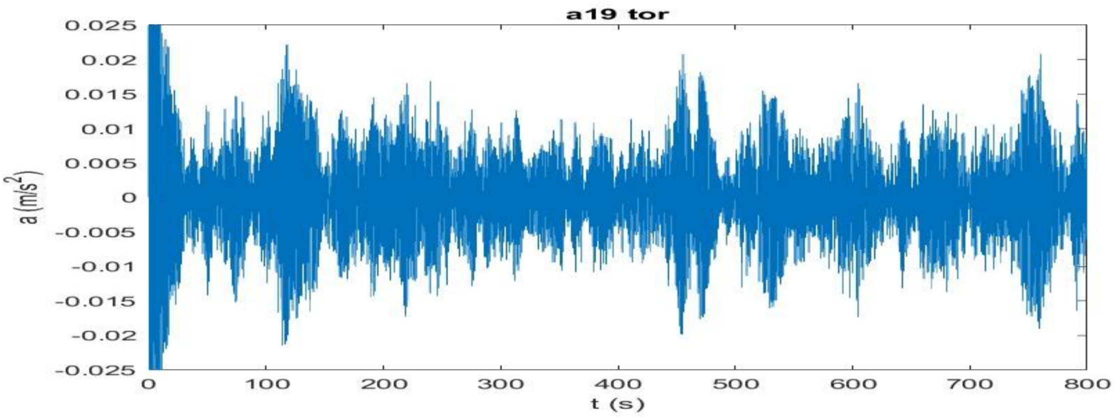


Figure 10.8: Wind speed calculation: torsional acceleration of the top

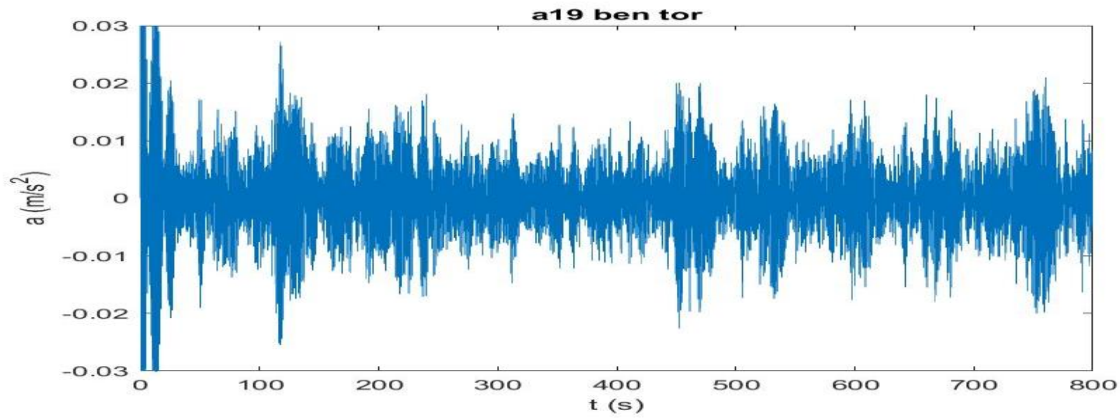


Figure 10.9: Wind speed calculation: bending and torsional acceleration of the top

For the recorded time instance u_p = 9.2 m/s and $\sigma_v = 2.96$ m/s , we can conclude for this realization that the acceleration for bending, torsion and bending and torsion added together meets the comfort requirement in the storm.

The maximum recorded acceleration at the top and far end of the building after renovation [17] (Appendix J) was $0.0377\frac{m}{s^2}$ with $u(51.3m) = 11.0\frac{m}{s}$, $u_p(10m) = 9.2\frac{m}{s}$, $u_* = 1.18\frac{m}{s}$.

The average bending acceleration (new method) after renovation was $0.0076\frac{m}{s^2}$ and the average bending and torsional acceleration added together was $0.0258\frac{m}{s^2}$ (Appendix 11). These accelerations values (new method) where taking out of the average of 50 simulations.

Comparing the accelerations above of $0.0377\frac{m}{s^2}$ to $0.0258\frac{m}{s^2}$, it can be concluded that this model did fit the recorded measurements.

10.3.4. After renovation u_p = 21.45 m/s (Return period once in 1 year)

For the NEN u_p = 21.45 m/s and $\sigma_v = 2.44$ m/s , the modeled accelerations are presented below, bending, torsion and bending and torsion added together.

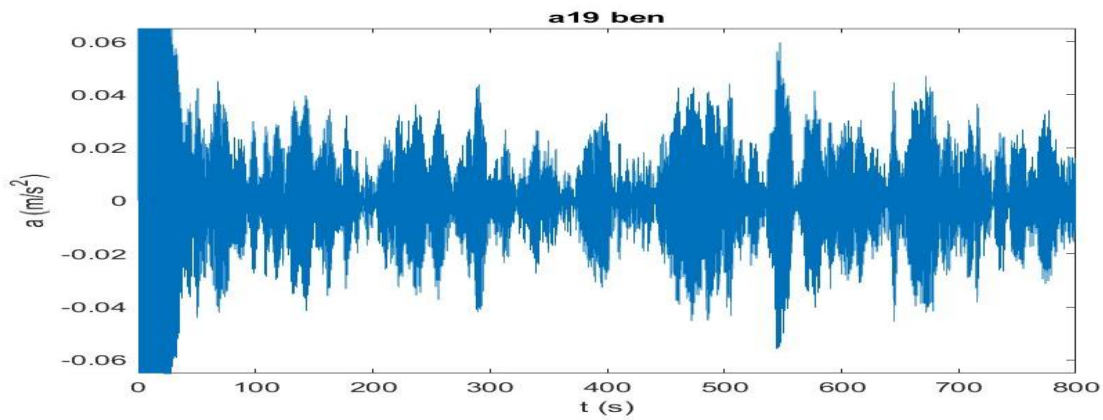


Figure 10.10: Wind speed calculation: bending acceleration of the top

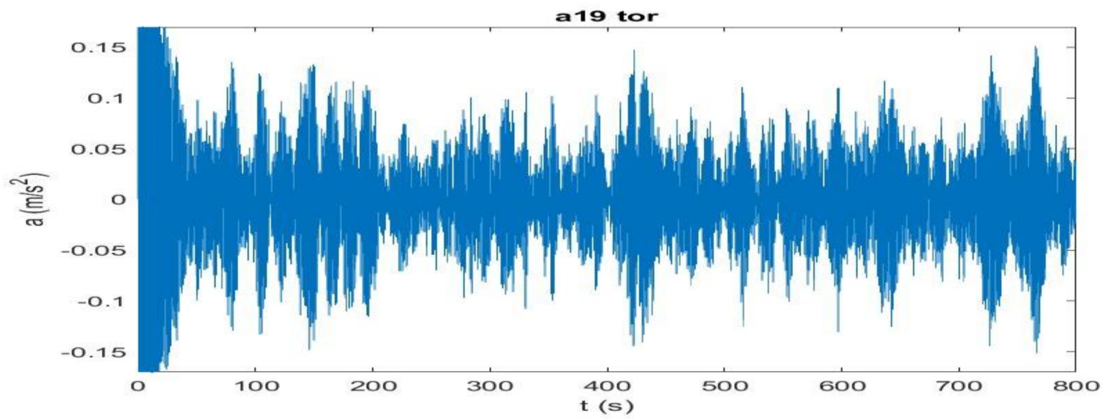


Figure 10.11: Wind speed calculation: torsional acceleration of the top

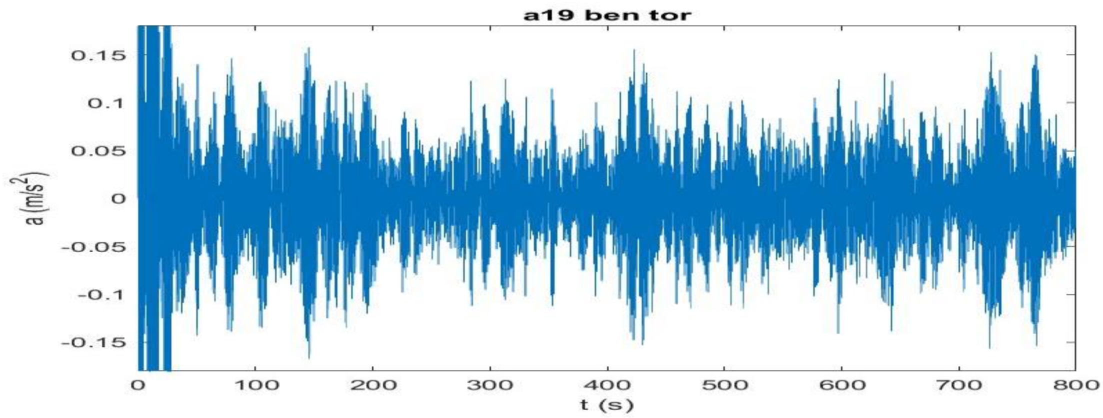


Figure 10.12: Wind speed calculation: bending and torsional acceleration of the top

For the recorded time instance u_p = 21.45 m/s and $\sigma_v = 2.44$ m/s, we can conclude for this realization that the acceleration for bending meets the comfort requirement. The acceleration for torsion and bending and torsion added together does not meet the comfort requirement.

The average bending acceleration according to NEN (new method) after renovation was $0.0522 \frac{m}{s^2}$ and the average bending and torsional acceleration added together was $0.1696 \frac{m}{s^2}$.

These accelerations values above (new method) where taking out of the average of a 100 simulations.

It is possible to determine the bending, torsional and total (bending and torsional added together) acceleration of a building when the structural characteristics are known. We can now see how a refurbished office building will act in a storm. Will this building comply with the comfort requirement.

11. Comparison to design formulas for the Student building "Voorhof"

In this chapter, the acceleration out of the new model (bending, torsion and bending and torsion added together) are compared to design formulas and rules of thumb.

11.1. Maximum occuring accelerations for return period of one year before renovation

In the table below the values of the comfort requirement is given for some norms and rules of thumb. The average value (Matlab) for a 100 simulations is also given for bending, torsion and bending and torsion added together.

Formula	Natural frequency		Along wind			Across wind		
			Max bending acceleration m/s^2	Max torsional acceleration m/s^2	Max total acceleration m/s^2	Max bending acceleration m/s^2	Max torsional acceleration m/s^2	Max total acceleration m/s^2
	Hz	rad/s						
NEN	0.733	4.605	0.099		0.099			
Eurocode	0.897	5.634	1.373		1.373	NVT		NVT
NBCC	0.733	4.605				0.020		0.020
Woudenberg (emp)	0.897	5.634	1.628	0.521	2.149			
Woudenberg	0.741	4.655	0.157		0.157			
Schueller	0.590	3.707	0.705		0.705			
Dicke/Nijsse	0.076	0.479	0.026		0.026			

Table 25: Resulting annual maxima before renovation

Matlab	0.624	3.922	0.098	0.261	0.297			

Table 26: The average out of a 100 simulations occurring acceleration in Simulink for the Voorhof before renovation for return period of one year

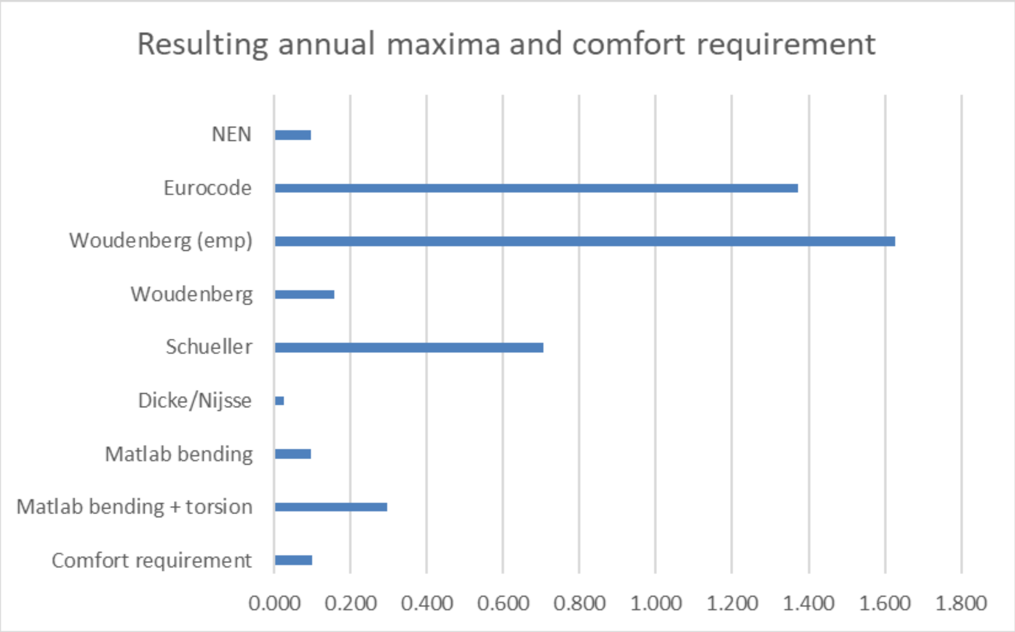


Figure 11.1: Resulting annual maxima and comfort requirement before renovation

It can be concluded that the average value for bending out of Matlab ($0.098\frac{m^2}{s}$) is smaller the value for the NEN, Eurocode, Woudenberg (emp), Woudenberg and Schueller. For bending and torsion added together we see that the average value out of a 100 simulations for bending and torsion added together ($0.297\frac{m^2}{s}$) out of Matlab is larger than NEN, Woudenberg, Dicke/Nijsse. The Eurocode, Woudenberg(emp) and Schueller value are so over conservative for bending, that it still meets the comfort requirement for bending and torsion which is not logical.

The acceleration above does not take into account the effect of shear lag, the second order effect and the reduction of effective area due to (window) openings in structural elements. The fact that the crosswind acceleration which can be larger than the along wind acceleration in many cases is also neglected by many of the used formulas above.

The maximum acceleration is the superposition of the alongwind, acrosswind and torsional acceleration. In the most of these formulas only one component of the acceleration is taken into account, which makes these formula non-conservative. The actual acceleration felt by a person dwelling in the building will be larger than the outcome of any formula above.

The characteristic values of the simulations are given below: bending, torsion and bending and torsion added together. The standard deviations in the table below has been taken out of the average of 10 simulations (**Table 22**).

Voorhof before renovation R = 1 year	k -	σ_a m/s²	a_{kar} m/s²
Bending	2.91	0.028	0.08
Torsion	2.91	0.074	0.22
Bending +Torsion	2.91	0.075	0.22

Table 27: characteristic values of the 10 acceleration simualtions before renovation

View page 85 for explination of k, σ_a a_{kar} .

11.2. Maximum occuring accelerations for return period of one year after renovation

In the table below the values of the comfort requirement is given for some norms and rules of thumb. The average value (Matlab) for a 100 simulations is also given for bending, torsion and bending and torsion added together.

Formula	Natural frequency		Along wind			Across wind		
	Hz	rad/s	Max bending acceleration m/s^2	Max torsional acceleration m/s^2	Max total acceleration m/s^2	Max bending acceleration m/s^2	Max torsional acceleration m/s^2	Max total acceleration m/s^2
NEN	0.996	6.258	0.061		0.061			
Eurocode	0.897	5.634	0.852		0.852	NVT		NVT
NBCC	0.996	6.259				0.013		0.013
Woudenberg (emp)	0.897	5.634	1.628	0.267	1.895			
Woudenberg	0.698	4.388	0.211		0.211			
Schueller	0.801	5.034	1.300		1.300			
Dicke/Nijsse	0.130	0.816	0.049		0.049			

Table 28: Resulting annual maxima after renovation

Matlab	0.851	5.346	0.052	0.155	0.170			

Table 29: The average out of a 100 simulations occurring acceleration in Simulink for the Voorhof after renovation for return period of one year

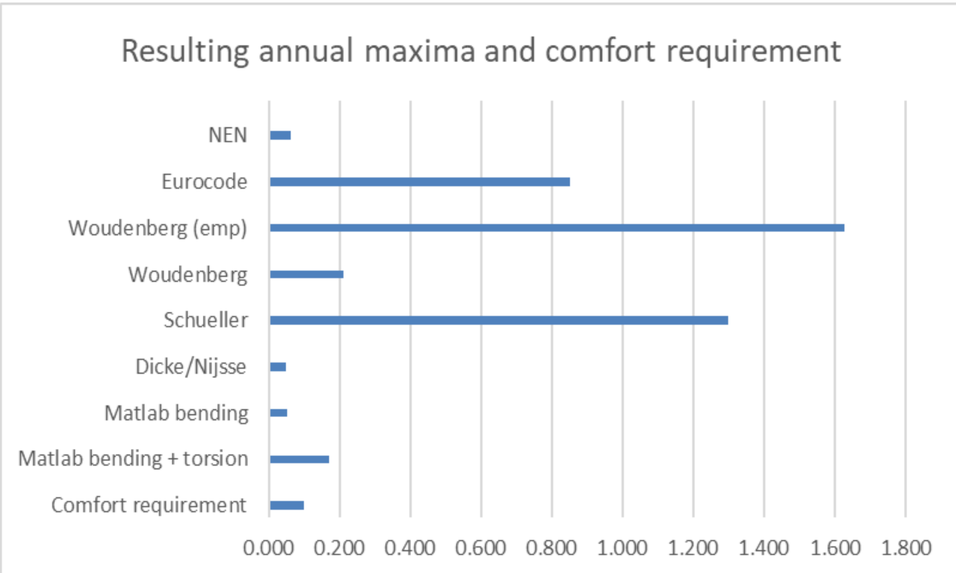


Figure 11.2: Resulting annual maxima and comfort requirement after renovation

It can be concluded that the average value for bending out of Matlab ($0.052\frac{m^2}{s}$) is smaller the value for the NEN, Eurocode, Woudenberg (emp), Woudenberg and Schueller. For bending and torsion we see that the average value out of a 100 simulations for bending and torsion ($0.170\frac{m^2}{s}$) out of Matlab is smaller than most Norms and rules of thumb. The Eurocode, Woudenberg (emp) and Schueller value is so over conservative for bending, that it still meets the comfort requirement for bending and torsion which is not logical.

The acceleration above does not take into account the effect of shear lag, the second order effect and the reduction of effective area due to (window) openings in structural elements. The fact that the crosswind acceleration which can be larger than the along wind acceleration in many cases is also neglected by many of the used formulas above.

The maximum acceleration is the superposition of the alongwind, acrosswind and torsional acceleration. In the most of these formulas only one component of the acceleration is taken into account, which makes these formula non conservative. The actual acceleration felt by a person dwelling in the building will be larger than the outcome of any formula above.

The characteristic values of the simulations are given below: bending, torsion and bending and torsion added together. The standard deviations in the table below has been taken out of the average of 10 simulations (**Table 24**).

Voorhof after renovation R = 1 year	k -	σ_a m/s²	a_{kar} m/s²
Bending	2.91	0.015	0.04
Torsion	2.91	0.043	0.13
Bending +Torsion	2.91	0.046	0.13

Table 30: characteristic values of the 10 acceleration simulations after renovation

View page 85 for explination of k , σ_a a_{kar} .

12. Frequency domain analysis

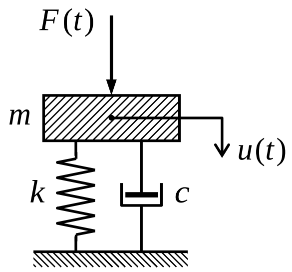
In this chapter, the method for the frequency domain analysis is presented. The method is explained for the case of a force spectra with an arbitrary loading for a single damped degree of freedom system (SDOF) and the case of a force spectrum with an arbitrary loading for a multi degree of freedom system (NDOF). The spectra of accelerations for bending are also presented for the Juffertoren and Student building "Voorhof" before and after renovation. The spectra of accelerations for torsion and bending and torsion added together was not done.

12.1. Introduction of a single damped degree of freedom system under stochastic loading

Note: Most of the following text and figures are taken directly from the lecture notes of Random Vibration ([1] ch. 2-3), the reason for giving it, is because if left away, one does not understand how the calculated values are built up and how the calculated values are reached. First the method is explained for a single damped degree of freedom system with harmonic loading after which the single damped degree of freedom system with stochastic loading.

12.1.1. Single damped degree of freedom system with harmonic loading

A single damped degree of freedom system under harmonic loading ([1] ch.3, [11] ch. 2-4) can be described by a mass m, spring constant k and damping constant c, which is loaded by a time dependant load $F(t)$. The response is given as $u(t)$.



$$\omega_e = \sqrt{\frac{k}{m}} \quad ; \quad \zeta = \frac{c}{2\sqrt{k\,m}}$$

Figure 12.1: Single damped degree of freedom system under harmonic loading ([1] ch.3)

The force F is input and the response $u(t)$ output. The response of a harmonic load is given below.

$$F = \hat{F}\sin(\omega t + \varphi) = \mathrm{Im}\left(\hat{F}e^{i(\omega t+\varphi)}\right) \text{ (input)}$$

$$u = |H|\hat{F}\sin(\omega t + \varphi + \psi) = |H|\,\mathrm{Im}\left(\hat{F}e^{i(\omega t+\varphi+\psi)}\right) \text{ (output)}$$

With:

$$|H| = |H(\omega)| = \frac{1}{k\left[\left\{1-(\omega/\omega_e)^2\right\}^2 + \left\{2\zeta(\omega/\omega_e)^2\right\}\right]^{1/2}}$$

$$\psi = \arctan\left(\frac{2\zeta(\omega/\omega_e)}{1-(\omega/\omega_e)^2}\right)$$

ω = frequency ω_e = natural frequency; $\omega_e = \sqrt{k/m}$

ζ = damping factor; $\zeta = c/2\sqrt{km}$

If we have a single damped degree of freedom system with harmonic load with parameters:

$\hat{F} = 100N$	$\varphi = 0rad$	$t = 0-50s$
$m = 1000kg$	$c = 173.25N/(m/s)$	$k = 3000N/m$
$\omega = 1rad/s$	$\omega_e = 1.73rad/s$	$\zeta = 0.05[-]$

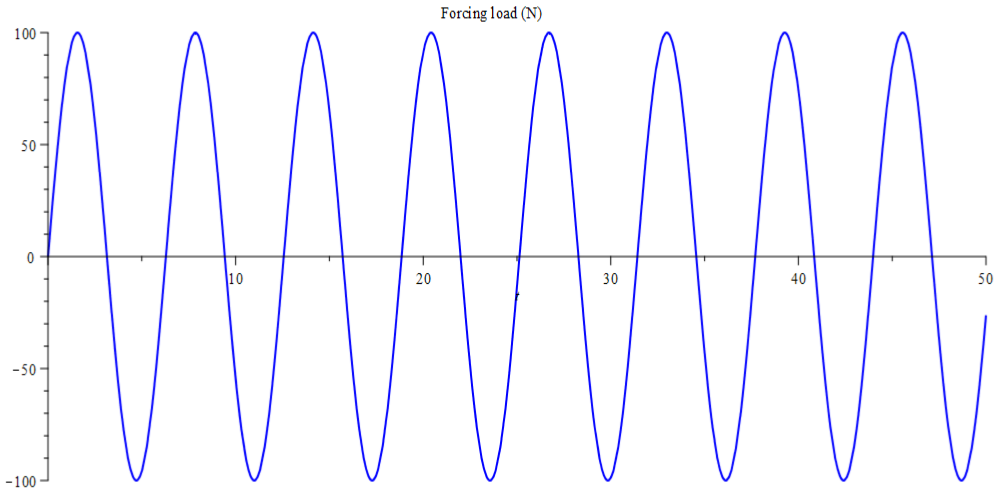


Figure 12.2: Forcing load of the single damped degree of freedom system with harmonic loading

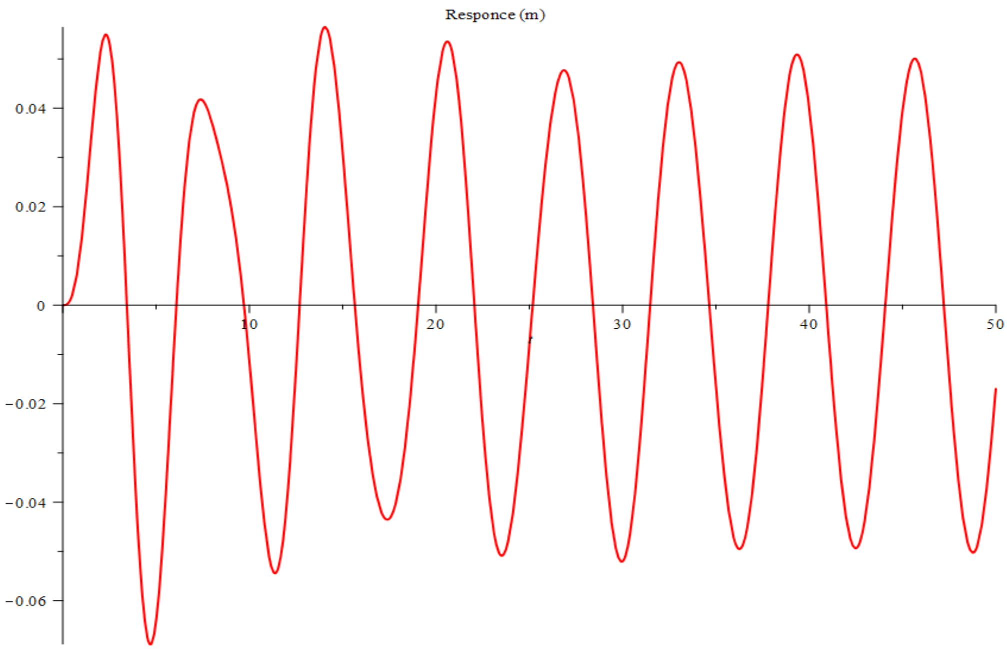


Figure 12.3: Response of the single damped degree of freedom system with harmonic loading

12.1.2. Single damped degree of freedom system with stochastic loading with μ is zero

The only difference between a single damped degree of freedom system with harmonic loading and stochastic loading with $\mu = 0$ is that load $\left(F = \hat{F}\sin\left(\omega t + \varphi\right)\right)$ is made up of a infinite amount of sinusses with an infinite amount of cyclic frequencies and random phases ([1] ch.2.2). An illustration of this is given below.

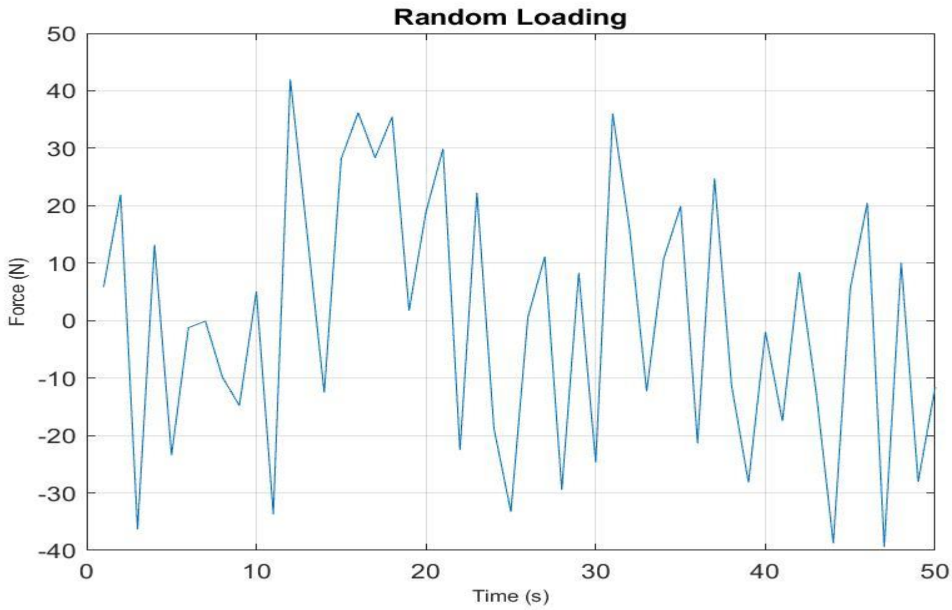


Figure 12.4: Single degree of freedom system under random loading

To explain the random loading, we take a forcing term made up of 2 harmonic loads $\left(F = \hat{F}_1\sin\left(\omega_1 t + \varphi_1\right) + \hat{F}_2\sin\left(\omega_2 t + \varphi_2\right)\right)$.

with values: (just a possible realization)

$\hat{F}_1 = 50N$	$\omega_1 = 0.5 rad/s$	$\varphi_1 = 4.119 rad$
$\hat{F}_2 = 50N$	$\omega_2 = 0.1 rad/s$	$\varphi_2 = 1.076 rad$

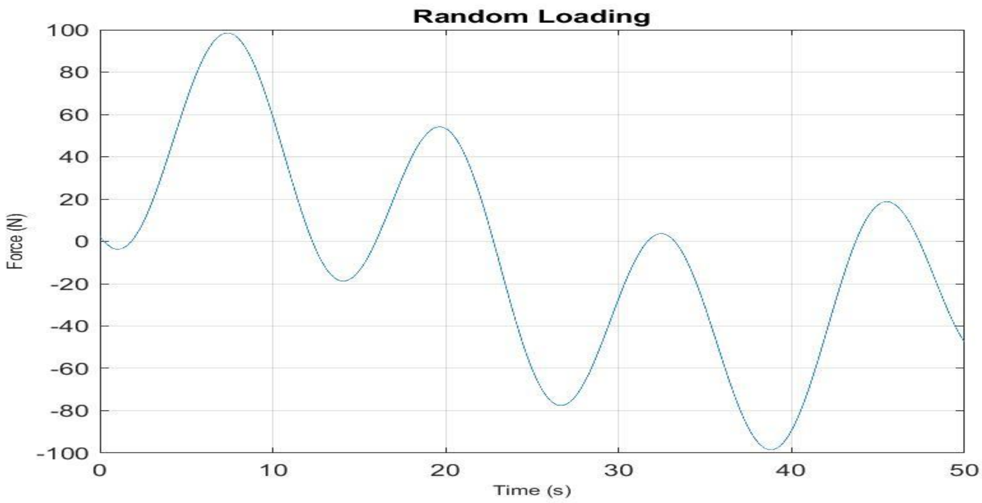


Figure 12.5: Single damped degree of freedom system under random loading made up of 2 terms

The output of the system will be a summation of the ouput of the single terms, view 12.1.1 for the output of a single term.

If we have a single damped degree of freedom system with system parameters (below) and forcing parameters as the previous page, will get a response as given below.

$m = 1000kg$ $\quad c = 34.64N/(m/s)$ $\quad k = 3000N/m$

$\omega_e = 1.73rad/s$ $\quad \zeta = 0.01[-]$

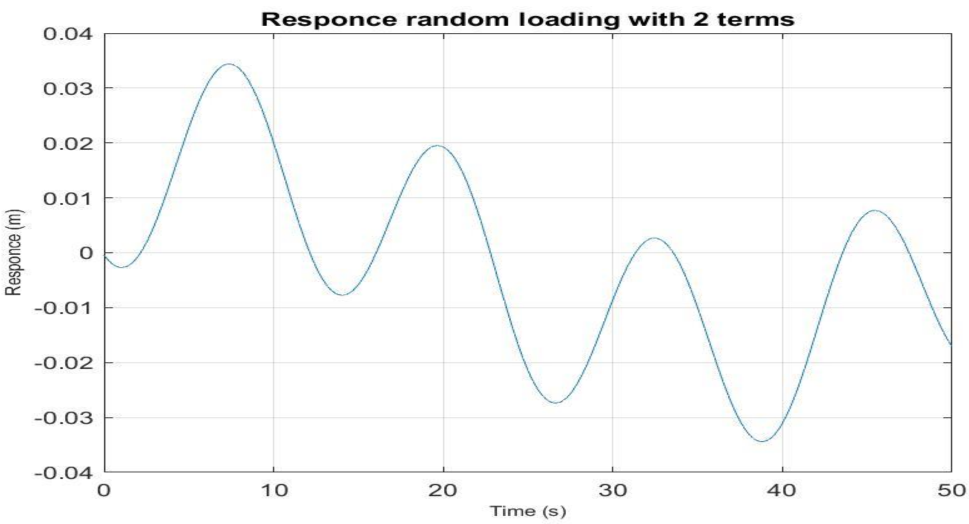


Figure 12.6: Single damped degree of freedom system under random loading made up of 2 harmonic terms

If the random loading $\left(F(t) = \sum_{k=1}^{N} F_k \sin(\omega_k + \varphi_k) \right)$ (Figure **12.4)** is given and the amplitudes of force (F_k)**,** cyclic frequencies (ω_k) and random phases (φ_k) of the infinite amount of sinusses are unknown, then a Fourier series ([1] ch.2 p.3) is used to determine variance (σ^2) of the random loading.

12.2. Random wind loading for a single damped degree of freedom system

12.2.1. Theory

The only difference between a single damped degree of freedom system with stochastic loading with mean of the force spectra zero $(\mu_F = 0)$ and random wind loading is that mean of the force spectra in not zero $(\mu_F \neq 0)$. When the load is a Gaussian process with mean (μ_F) and force spectra (S_{FF}) then it can be said according to ([1] ch. 2.4):

$$\mu_u = \frac{1}{k}\mu_F$$

$$S_{uu}(\omega) = |H(\omega)|^2 S_{FF}$$

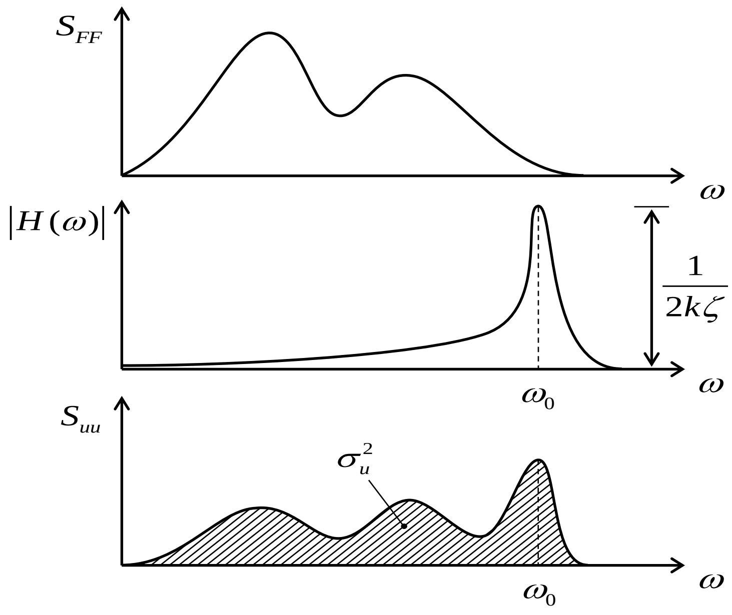


Figure 12.7: Analysis of the single mass spring system ([1] ch. 3.1)

The variance of the response is given by

$$\sigma_u^2 = \int_0^\infty S_{uu}(\omega)\,d\omega$$

For the comfort requirement we are not instrested in the variance of response but the variance of acceleration.

The variance of the acceleration is given by

$$\sigma_a^2 = \int_0^\infty S_{aa}(\omega)\,d\omega = \int_0^\infty \left(\omega^2 S_{uu}(\omega)\right)d\omega$$

An approximation of the arbitrary load is given for the spectrum of responce ([1] ch. 3.3 p.5 eq. 3.23). This approximation is used later in this report (12.6-12.8) to determine the variance of response and ultimately the variance of acceleration for the Juffertoren, Voorhof before and after renovation for return period of 12.5 years and for return period of one year.

$$S_{uu}(\omega) = \frac{S_{FF}(\omega)}{k^2} + \frac{S_{FF}(\omega_e)}{k^2\left\{\left(1-\omega^2/\omega_e^2\right)^2 + \left(2\zeta\omega/\omega_e\right)^2\right\}}$$

With:

S_{uu} = response spectrum ω = frequency S_{FF} = force spectrum

k = spring stiffness ω_e = natural frequency ζ = damping factor;

12.2.2. Matlab

Before the approximation formula can be used , the velocity spectra (S_{vv}) of the windload on the building must be determined, after which the force spectra (S_{FF}) can be determined.

The velocity spectra (S_{vv}) is taken at reference height of the building.

For the Juffertoren: $v_{hub;ULS} = v_{10;ULS} = 22.62 m/s$ (6.3 **Table 4**)

$\sigma_{v;Top;ULS} = \sigma_{v;144;ULS} = 6.23 m/s$

$v_{hub;SLS} = v_{10;SLS} = 21.45 m/s$ (6.5 **Table 5**)

$\sigma_{v;Top;SLS} = \sigma_{v;144;SLS} = 2.52 m/s$

For the Voorhof: $v_{hub;ULS} = v_{10;ULS} = 15.79 m/s$ (9.6 **Table 17**)

$\sigma_{v;Top;ULS} = \sigma_{v;51.3;ULS} = 6.65 m/s$

$v_{hub;SLS} = v_{10;SLS} = 21.45 m/s$ (9.8 **Table 18**)

$\sigma_{v;Top;SLS} = \sigma_{v;144;SLS} = 2.44 m/s$

First the velocity spectra (S_{vv}) is determined in Matlab (Appendix 14 S_FF_we.m). The characteristic length $(L = 1200m)$, windspeed at the hub height (v_{10}) and standard deviation at top of building (σ_v) are inputted in the formula for fluctuating wind velocities for the Davenport spectrum.

$$S_{vv}(f) = \frac{F_D \sigma_v^{\ 2}}{f}$$

with:

$F_D = \frac{2}{3}\frac{x^2}{(1+x^2)^{4/3}}$ Davenport spectrum

σ_v standard deviation of the wind speed

$x = \frac{fL_{gust}}{\bar{v}_{10}}$ dimensionless frequency

The force spectra (S_{FF}) is determined by multiplying the velocity spectra with the air density $(\rho_{air} = 1.25 kg/m^3)$, Area $(A[m^2])$ of one floor of the building parallel to wind flow direction, the summation of drag and suction coefficient $(Ch = 1.2[-])$ and the mean wind speed at height h $(\bar{v}_h)$. The force spectra (S_{FF}) is determined for each floor of the building of the building after which the force spectra is selected for the top floor.

$S_{FF} = (\rho_{air} * A * Ch * \bar{v}_h)^2 * S_{vv}$ ([1] ch.6 eq. 6.50)

with:

For the Juffertoren: $v_{Top;ULS} = v_{144;ULS} = 39.6 m/s$ (6.3 **Table 4**)

$v_{Top;SLS} = v_{144;SLS} = 37.62 m/s$ (6.5 **Table 5**)

For the Voorhof: $v_{Top;ULS} = v_{51.3;ULS} = 30.17 m/s$ (9.6 **Table 17**)

$v_{Top;SLS} = v_{51.3;SLS} = 40.83 m/s$ (9.8 **Table 18**)

Now the response spectra (S_{uu}) can be determined with the approximation formula $S_{uu}(\omega) = \frac{S_{FF}(\omega)}{k^2} + \frac{S_{FF}(\omega_e)}{k^2\left\{(1-\omega^2/\omega_e^2)^2 + (2\zeta\omega/\omega_e)^2\right\}}$, after which the acceleration spectra (S_{aa}) out of formula $(S_{aa}(\omega) = \omega^2 S_{uu}(\omega))$ can be determind.

Last the standard deviation is taken from the acceleration spectra (S_{aa}). No aerodynamic admittance and coherence calculated for the single damped degree of freedom system.

12.3. Multi damped degree of freedom system with harmonic loading

The steady state response of a multi degree of freedom system with harmonic loading in the frequency domain can be simply solved ([2] pp. 44-59) ([5] pp. 52-53). Modal analysis can be applied to the multi degree of freedom system.

The equation of motion is given below:

$$\underline{\underline{M}}\ddot{\underline{u}} + \underline{\underline{C}}\dot{\underline{u}} + \underline{\underline{K}}\underline{u} = \underline{F}$$

With $\underline{\underline{M}}$, $\underline{\underline{C}}$, $\underline{\underline{K}}$ which are (n*n) matrices and $\underline{F}$ and $\underline{u}$ which are vectors with length (n*1), n being the being the number of degres of freedom.

The forcing and response vector can be written as:

$$\underline{F}(t) = \underline{F}(\omega)e^{i\omega t}$$
$$\underline{u}(t) = \underline{u}(\omega)e^{i\omega t}$$

The transfer function or dynamic flexibility coefficient can be written as:

$$\underline{\underline{H}}_{uF} = \left(\underline{\underline{K}} - \omega^2\underline{\underline{M}} + i\omega\underline{\underline{C}}\right)^{-1}$$

The absolute of the complex response can be written as:

$$\left|\underline{u}(\omega)\right| = \left|\underline{\underline{H}}_{uF}\right|\left|\underline{F}(\omega)\right|$$

For the comfort requirement the absolute of the complex acceleration is required and can be written as:

$$\left|\underline{a}(\omega)\right| = \omega^2\left|\underline{u}(\omega)\right| = \omega^2\left|\underline{\underline{H}}_{uF}\right|\left|\underline{F}(\omega)\right|$$

12.3.1. Juffertoren

For the Juffertoren a (48*48) matrix is found. The transfer function of acceleration for the 48 node (top of building) is given below. The eigenfrequencies corresponding to the natural bending frequencies found in time domain are given $\omega_{ben\_1;2;3;4;5} = 1.41;\ 8.82;\ 24.67;\ 48.24;\ 79.52$ rad/s (p. 31) and the natural torsional frequencies found in time domain are given $\omega_{tor\_1;2;3;4;5} = 0.50;\ 1.50;\ 2.51;\ 3.50;\ 4.50$ rad/s (p. 40).

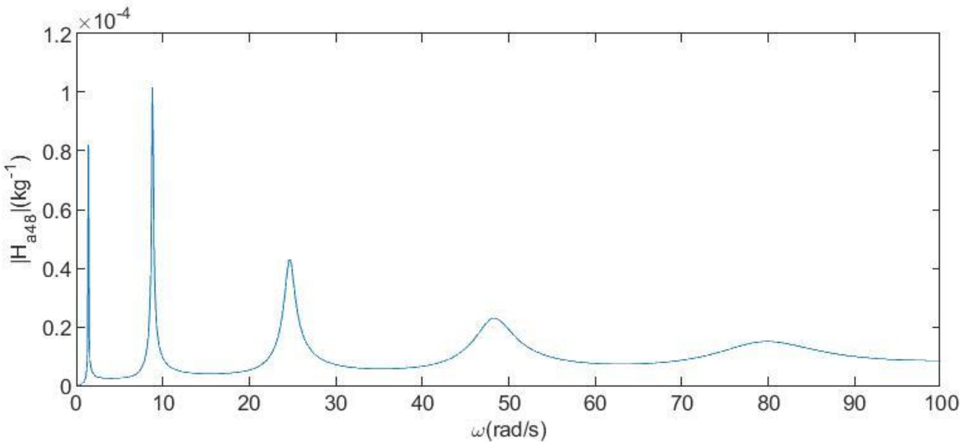


Figure 12.8: The transfer function of acceleration of the Juffertoren building for the 48 node for return period of one year

12.3.2. Voorhof before renovation

For the Voorhof before renovation a (19*19) matrix is found. The transfer function of acceleration for the 19 node (top of building) is given below. The eigenfrequencies corresponding to the natural frequencies found in time domain are given $\omega_{ben\_1;2;3} = 3.92;\ 24.52;\ 68.25$ rad/s (p. 105) and the natural torsional frequencies found in time domain are given $\omega_{tor\_1;2;3} = 4.02;\ 12.04;\ 19.98$ rad/s (p. 105).

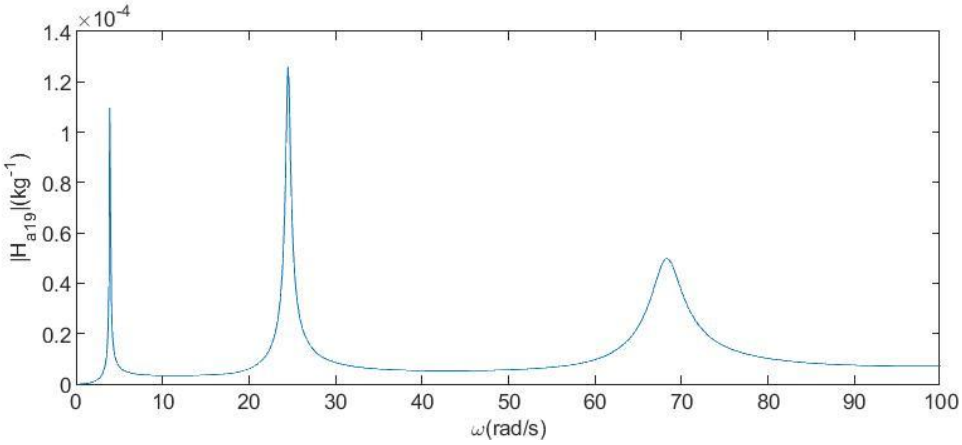


Figure 12.9: The transfer function of acceleration of the Voorhof before renovation for the 19 node for return period of one year

12.3.3. Voorhof after renovation

For the Voorhof after renovation a (19*19) matrix is found. The transfer function of acceleration for the 19 node (top of building) is given below. The eigenfrequencies corresponding to the natural frequencies found in time domain are given $\omega_{ben\_1;2;3} = 5.35;\ 33.42;\ 92.96$ rad/s (p. 106) and the natural torsional frequencies found in time domain are given $\omega_{tor\_1;2;3} = 4.84;\ 14.48;\ 24.03$ rad/s (p. 106).

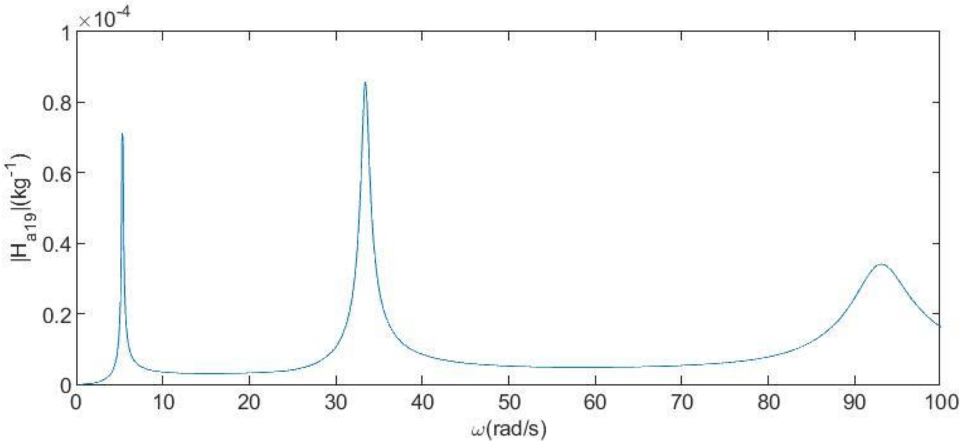


Figure 12.10: The transfer function of acceleration of the Voorhof after renovation for the 19 node for return period of one year

12.4. Random wind loading for a multi damped degree of freedom system (NDOF) with stochastic loading with full coherence

12.4.1. Theory

The only difference between a multi damped degree of freedom system with harmonic loading and a multi damped degree of freedom system with stochastic loading is that the loading term is not harmonic but random (stochastic). To be able to determine the response, we use discrete Fourier transform to turn the arbitrarily loaded masses in time domain to force spectra in frequency domain, now the force spectra of load $\left(S_{FF}\right)$ is known. With the force spectra known, we go from force spectra to displacement spectra $\left(S_{uu}\right)$ and then to the acceleration spectra $\left(S_{aa}=\omega^2 S_{uu}\right)$.

When determing the displacement spectra we use the formula $S_{uu}\left(\omega\right)=\left|H\left(\omega\right)\right|^2 S_{FF}$ to determine the displacement spectra.

12.4.2. Matlab

No separate code is given for the multi damped degree of freedom system with random wind loading with full coherence. The matlab code in Appendix 13 can be modified, just omit the aerodynamic admittance and coherence terms.

12.5. Random wind loading for a multi damped degree of freedom system (NDOF) with stochastic loading with coherence and admittance

12.5.1. Theory

The only difference between a multi damped degree of freedom system with stochastic loading with full coherence (previous paragraph) and with coherence and admittance are those terms. The procedure with admittance and coherence is explained in the Mathlab section below.

12.5.2. Matlab

In Appendix 13, the Matlab code for the multi damped degree of freedom system is given for the Juffertoren for return period once in 1 year. First the velocity spectra (S_{vv}) is determined with the Davenport Spectrum (6.3) in Matlab (Appendix 13 Spectrum_acceleration.m). The characteristic length (L), windspeed at the reference height of the building (v_{10}) and standard deviation at top of building (σ_v) are inputted in the formula for fluctuating wind velocities for the Davenport spectrum.

$$S_{vv}(f) = \frac{F_D \sigma_v^{\ 2}}{f}$$

with:

$F_D = \frac{2}{3}\frac{x^2}{\left(1+x^2\right)^{4/3}}$	Davenport spectrum
σ_v	standard deviation of the wind speed
$x = \frac{fL_{gust}}{\bar{v}_{10}}$	dimensionless frequency
f	frequency

The values for the velocity spectra can be found in the Matlab file.

One must remember that the velocity spectra for Davenport is independent of height. In simple words all spectras on the nodes are the same.

In the program the mean wind velocity on the 96 half floors are made into the velocity on each floor, 48 nodes. These velocities are needed to calculate the force spectrum $\left(S_{FF}\right)$.

With coherence :

$$S_{\tilde{v}_1\tilde{v}_2} = \left(Coh_{\tilde{v}_1\tilde{v}_2} * \sqrt{\left(S_{\tilde{v}_1\tilde{v}_1} * S_{\tilde{v}_2\tilde{v}2}\right)}\right)$$ ([1] ch. 6 p.25)

In the matlab code, the velocity spectra terms on the diagonal are written as aa or bb in which aa is equal to bb, these are the auto spectra terms.

$$S\_vv(aa,bb,n) = \left(Cohm(aa,bb) * \sqrt{\left(S\_vv(aa,bb,n) * S\_vv(aa,bb,n)\right)}\right)$$

In the matlab code, the velocity spectra terms off the diagonal are written as aa or bb in which aa is unequal to bb, these are the cross spectra terms.

$$S\_vv(aa,bb,n) = \left(Cohm(aa,bb) * \sqrt{\left(S\_vv(aa,aa,n) * S\_vv(bb,bb,n)\right)}\right)$$

With:

aa and bb from 1 to 48 degrees of freedom

In the coherence file (Appendix 13 coherence_2_run.m), the coherency is determined for the different points.

$$\Delta_{12} = \sqrt{\left(C_z^2\left(z_1 - z_2\right)^2 + C_y^2\left(y_1 - y_2\right)^2\right)}$$ ([1] ch. 6 p.26)

$$coh = \exp\left(\frac{-f\Delta_{12}}{\bar{v}_{10}}\right)$$

With:

$C_z = 7$	londitudinal coherence	[-]
$C_y = 10$	lateral coherence	[-]
z_1	height node 1	[m]
z_2	height node 2	[m]
y_1	lateral place node 1	[m]
y_2	lateral place node 2	[m]
f	frequency	[Hz]
$\bar{v}_{10}$	mean wind speed at 10 meters	[Hz]

The force spectra (S_{FF}) is determined for each height. For each height a different force spectra (S_{FF})

$$S_{FF} = (\rho A C_h \bar{v}_h)^2 \sigma_{\tilde{v}_h}^2 \chi^2 \frac{F_D}{f}$$ ([1] ch. 6 p.26 eq. 6.51.a)

With:

ρ	air density	[kg/m^3]
A	area of one story perpendicular to wind direction	[m^2]
C_h	suction and drag coefficient of wind	[-]
$\bar{v}_h$	mean wind speed at height h	[m/s]

$\sigma_{\tilde{v}_h}$ standard deviation of the fluctuating wind speed at height h [m/s]

F_D Davenport spectrum (6.3)

f frequency [Hz]

χ^2 aerodynamic admittance

$$\chi(f) = \frac{1}{1 + \left(\frac{2 f \Delta_{12}}{\bar{v}_{10}} \right)}$$

With:

In the admittance file (Appendix 13 admittance_2_run.m), the admittance is determined for the different points.

$$\Delta_{12} = \sqrt{\left(C_z^2 (z_1 - z_2)^2 + C_y^2 (y_1 - y_2)^2 \right)}$$ ([1] ch. 6 p.26)

After that the displacement spectra (S_{uu}) is determind. The displacement spectra is built up of a quasi-static part and a dynamic part and then added together. The formulas for both are given below.

Quasi-static part:

$$S_{u_l u_l}(f) = \sum \sum S_{F_i F_j}(f) H_{u_l F_i}(0) H^*_{u_l F_j}(0)$$ ([1] ch. 6 eq. 6.58)

$$S_{u_i u_j} = \sum_p \sum_q \sum_k \sum_l \frac{u_{k,i} u_{k,p}}{k_k} \frac{u_{l,j} u_{l,q}}{k_l} S_{F_p F_q}$$ ([1] ch. 4 eq. 4.32) (with $\omega = 0$)

k_k = k^{th} generalized spring constant $\underline{u}_k^T [K] \underline{u}_k$

k_l = l^{th} generalized spring constant $\underline{u}_l^T [K] \underline{u}_l$

$\underline{u}_k$ = k^{th} eigenvector

$\underline{u}_{k,i}$ = i^{th} characteristic number of the k^{th} eigenvector

$\underline{u}_{k,p}$ = p^{th} characteristic number of the k^{th} eigenvector

$\underline{u}_l$ = l^{th} eigenvector

$\underline{u}_{l,j}$ = j^{th} characteristic number of the l^{th} eigenvector

$\underline{u}_{l,q}$ = q^{th} characteristic number of the l^{th} eigenvector

Dynamic part:

$$S_{u_i u_j} = \sum_p \sum_q \sum_k \sum_l \frac{\underline{u}_{k,i}\underline{u}_{k,p}}{k_k - m_k\omega^2 + ic_k\omega} \frac{\underline{u}_{l,j}\underline{u}_{l,q}}{k_l - m_l\omega^2 - ic_l\omega} S_{F_p F_q} \quad \text{([1] ch. 4 eq. 4.32)}$$

with:

m_k = k^{th} generalized mass $\underline{u}_k^T [M] \underline{u}_k$

c_k = k^{th} generalized damping constant $\underline{u}_k^T [C] \underline{u}_k$

k_k = k^{th} generalized spring constant $\underline{u}_k^T [K] \underline{u}_k$

m_l = l^{th} generalized mass $\underline{u}_l^T [M] \underline{u}_l$

c_l = l^{th} generalized damping constant $\underline{u}_l^T [C] \underline{u}_l$

k_l = l^{th} generalized spring constant $\underline{u}_l^T [K] \underline{u}_l$

After which the acceleration spectra $\left(S_{aa}\right)$ is determind.

Then the standard deviation $\left(\sigma_a\right)$ is taken out of the acceleration spectra $\left(S_{aa} = \omega^2 S_{uu}\right)$.

12.6. Juffertoren

The formula for the spectra of the acceleration of the top of the Juffertoren building will be given below. In the computer simulation 48 modes were used.

$$S_{a_{48}a_{48}}(\omega) = \sum_{i=1}^{48}\sum_{j=1}^{48} H_{a_{48}F_i}(\omega) H^*_{a_{48}F_j}(\omega) S_{F_iF_j}(\omega)$$

Where $H_{a_{48}F_i}$ is the complex transfer function of the acceleration of the 48th node for a harmonic force on node i and $H^*_{a_{48}F_j}$ is the complex conjugate of $H_{a_{48}F_j}$. For $i=j$, $S_{F_iF_j}(\omega)$ is the auto-spectra of the force and for $i \neq j$, $S_{F_iF_j}(\omega)$ is the cross-spectra of the force.

12.6.1. Return period of 12.5 years

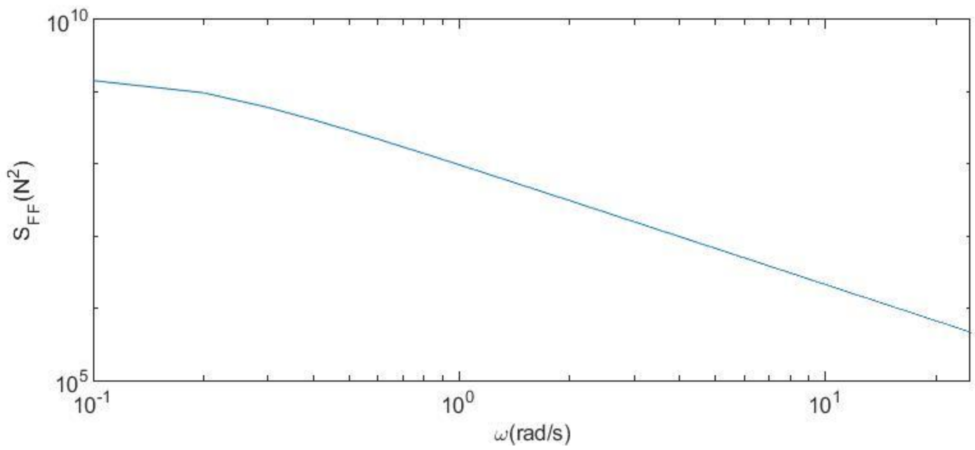


Figure 12.11: Spectra of force of node 48 of the system for return period of 12.5 years

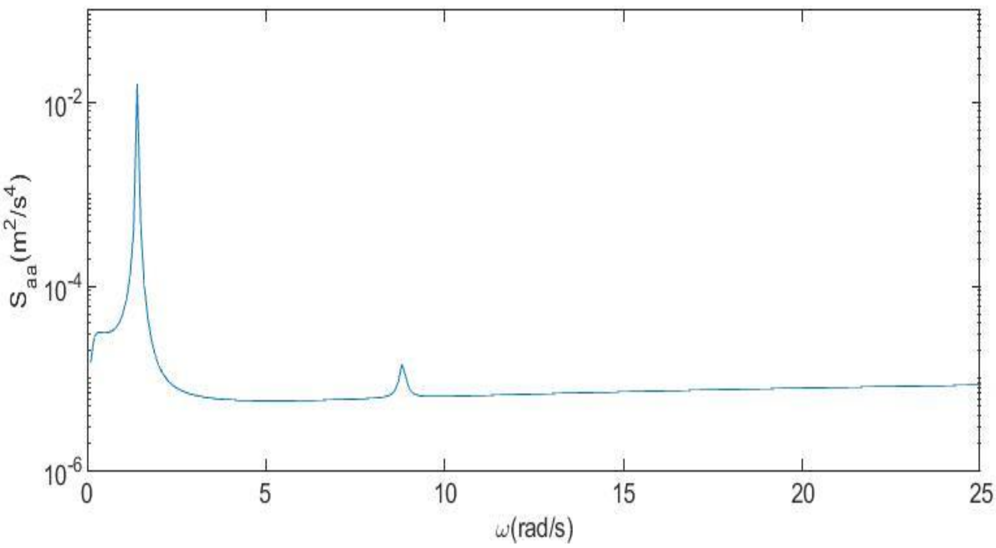


Figure 12.12: Spectra of acceleration of node 48 of the system for return period of 12.5 years

From the spectra of the accelerations the variance of the acceleration can be determined by:

$$\sigma^2_{a;48} = \int_0^\infty S_{aa}(\omega)\,d\omega$$

And the standard deviation follows from:

$$\sigma_{a;48} = \sqrt{\sigma^2_{a;48}}$$

The decisive standard deviation for the Juffertoren for return period of 12.5 years equals: $\sigma_{a;48} = 0.061 m/s^2$. The expected peak value can be calculated from:

$$a^*_{48;peak;expected} = \sigma_{a;48}\sqrt{2\ln(T_s f_e)}$$ ([1] eq. 3.38)

With $T_s = 800s$ and $f_e = 0.22Hz$ the decisive expected peak value for the Juffertoren for return period of 12.5 years equals: $a^*_{48;peak} = 0.196 m/s^2$. In the table below, the peak values and standard deviations are given for a single damped degree of freedom and a 48 degree of freedom system for different ranges of the frequency.

2 different types of Spectral analysis were run in the Frequency Domain:

S Dof	Omega: 0 to 25		Time = 2 s			S Dof	Omega: 0 to 165		Time = 2 s	
	a_star (a_ben_max)		sigma_a_ben				a_star (a_ben_max)		sigma_a_ben	
UPPERBOUND	0.026	**Average**	0.008			**UPPERBOUND**	0.053	**Average**	0.017	
(t=800)			step 0.01			(t=800)			step 0.01	
S Dof	Omega: 0 to 677		Time = 7 s			48 Dof	Omega 0 to 2		Time = 222 s	
	a_star (a_ben_max)		sigma_a_ben				a_star (a_ben_max)		sigma_a_ben	
UPPERBOUND	0.127	**Average**	0.040			**UPPERBOUND**	0.133	**Average**	0.041	
(t=800)			step 0.01			(t=800)			step 0.1	
48 Dof	Omega 0 to 25		Time = 2506 s			48 Dof	Omega 0 to 165		Time = 16379 s	
	a_star (a_ben_max)		sigma_a_ben				a_star (a_ben_max)		sigma_a_ben	
UPPERBOUND	0.140	**Average**	0.043			**UPPERBOUND**	0.196	**Average**	0.061	
(t=800)			step 0.1			(t=800)			step 0.1	
w_e =	1.4085	rad/s								
f_0 =	0.22	Hz								
T=	800	s								

$$a^* = \mu_a + \sigma_a \sqrt{2 \ln\left(T f_0\right)}$$

Table 31: Values of 2 different spectral analysis for the Juffertoren

12.6.2. Return period of one year

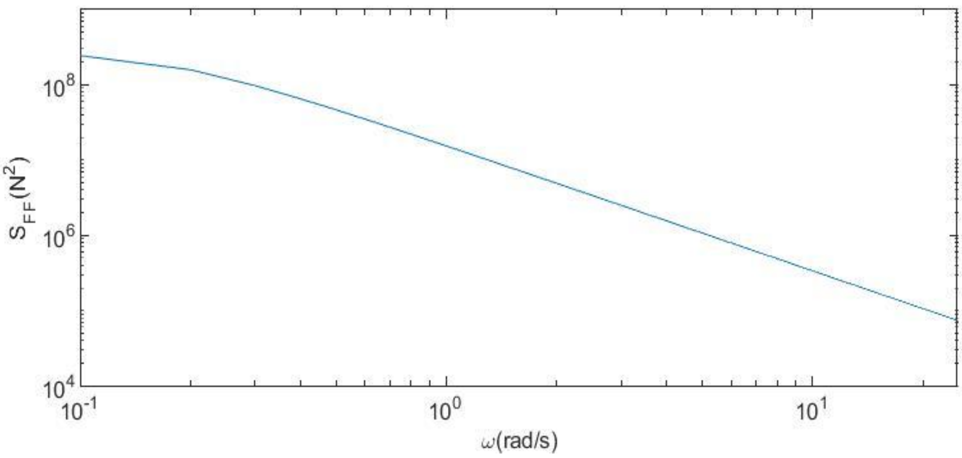


Figure 12.13: Spectra of force of node 48 of the system for return period of one year

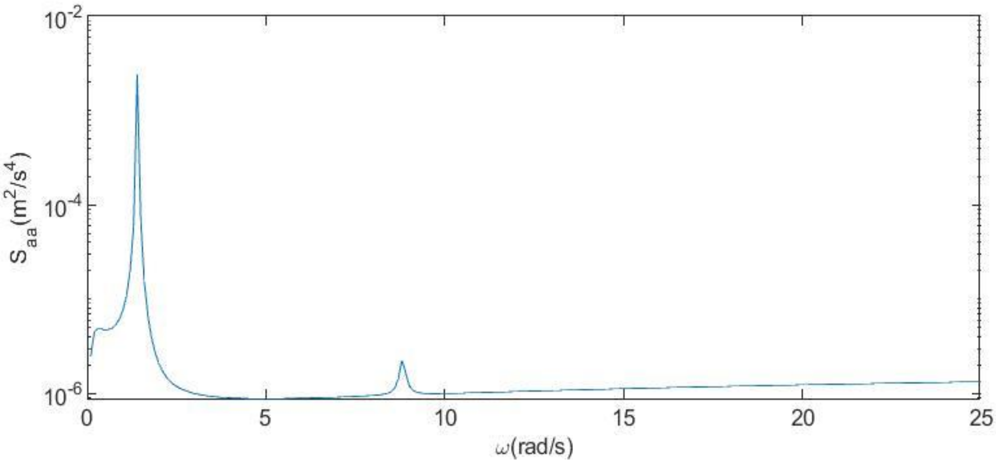


Figure 12.14: Spectra of acceleration of node 48 of the system for return period of one year

From the spectra of the accelerations the variance of the acceleration can be determined by:

$$\sigma_{a;48}^2 = \int_0^\infty S_{aa}(\omega)\,d\omega$$

And the standard deviation follows from:

$$\sigma_{a;48} = \sqrt{\sigma^2_{a;48}}$$

The decisive standard deviation for the Juffertoren for return period of one year equals: $\sigma_{a;48} = 0.024 m/s^2$. The expected peak value can be calculated from:

$$a^*_{48;peak;expected} = \sigma_{a;48}\sqrt{2\ln(T_s f_e)} \quad \text{([1] eq. 3.38)}$$

With $T_s = 800s$ and $f_e = 0.22Hz$ the decisive expected peak value for the Juffertoren for return period of one year equals: $a^*_{48;peak} = 0.077 m/s^2$. In the table below, the peak values and standard deviations are given for a single damped degree of freedom and a 48 degree of freedom system for different ranges of the frequency.

2 different types of Spectral analysis were run in the Frequency Domain:

S Dof	Omega: 0 to 25		Time = 2 s		S Dof	Omega: 0 to 165		Time = 2 s
	a_star (a_ben_max)		sigma_a_ben			a_star (a_ben_max)		sigma_a_ben
UPPERBOUND		0.010 **Average**	0.003		**UPPERBOUND**		0.020 **Average**	0.006
(t=800)			step 0.01		(t=800)			step 0.01
S Dof	Omega: 0 to 677		Time = 2 s		48 Dof	Omega 0 to 2		Time = 219 s
	a_star (a_ben_max)		sigma_a_ben			a_star (a_ben_max)		sigma_a_ben
UPPERBOUND		0.048 **Average**	0.015		**UPPERBOUND**		0.052 **Average**	0.016
(t=800)			step 0.01		(t=800)			step 0.1
48 Dof	Omega 0 to 25		Time = 2882 s		48 Dof	Omega 0 to 165		Time = 17396 s
	a_star (a_ben_max)		sigma_a_ben			a_star (a_ben_max)		sigma_a_ben
UPPERBOUND		0.054 **Average**	0.017		**UPPERBOUND**		0.077 **Average**	0.024
(t=800)			step 0.1		(t=800)			step 0.1
w_e =	1.4085	rad/s						
f_0 =	0.22	Hz						
T=	800	s						
$a^* = \mu_a + \sigma_a\sqrt{2\ln(Tf_0)}$								

Table 32: Values of 2 different spectral analysis for the Juffertoren

12.7. Voorhof before renovation

The formula for the spectra of the acceleration of the top of the Juffertoren building will be given below. In the computer simulation 19 modes were used.

$$S_{a_{19}a_{19}}(\omega) = \sum_{i=1}^{19}\sum_{j=1}^{19} H_{a_{19}F_i}(\omega) H^*_{a_{19}F_j}(\omega) S_{F_iF_j}(\omega)$$

Where $H_{a_{19}F_i}$ is the complex transfer function of the acceleration of the 19th node for a harmonic force on node i and $H^*_{a_{19}F_j}$ is the complex conjugate of $H_{a_{19}F_j}$. For $i = j$, $S_{F_iF_j}(\omega)$ is the auto-spectra of the force and for $i \neq j$, $S_{F_iF_j}(\omega)$ is the cross-spectra of the force.

12.7.1. Return period of 12.5 years

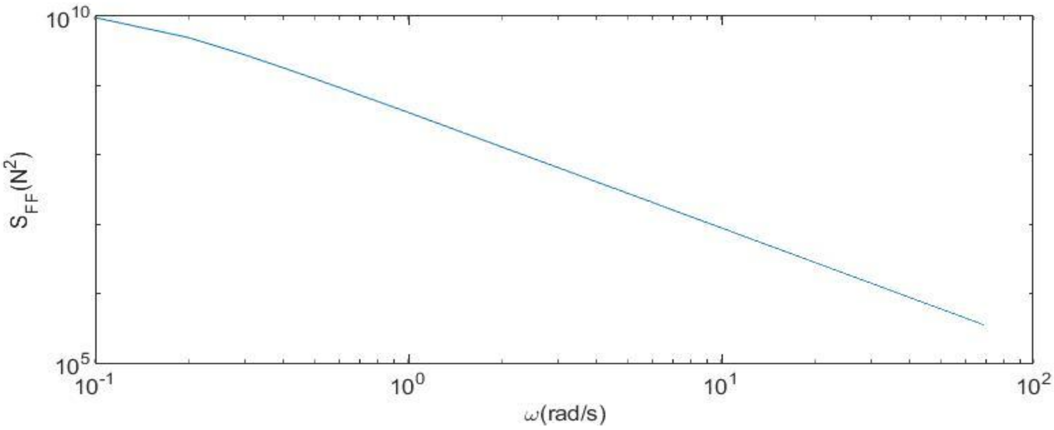


Figure 12.15: Spectra of force of node 19 of the system for return period of 12.5 years before renovation

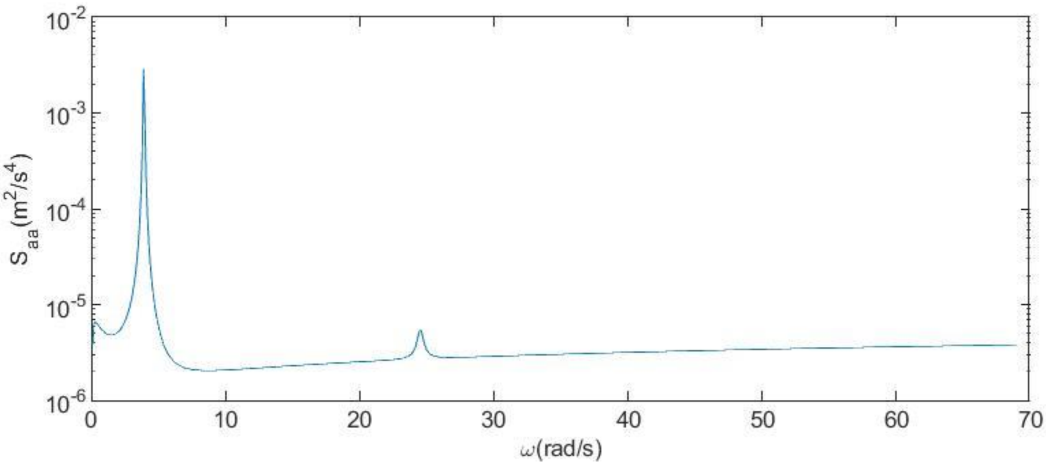


Figure 12.16: Spectra of acceleration of node 19 of the system for return period of 12.5 years before renovation

From the spectra of the accelerations the variance of the acceleration can be determined by:

$$\sigma_{a;19}^2 = \int_0^\infty S_{aa}(\omega)\, d\omega$$

And the standard deviation follows from:

$$\sigma_{a;19} = \sqrt{\sigma_{a;19}^2}$$

The decisive standard deviation for the Voorhof before renovation for return period of 12.5 years equals: $\sigma_{a;19} = 0.053 m/s^2$. The expected peak value can be calculated from:

$$a_{19;peak;expected}^* = \sigma_{a;19}\sqrt{2\ln(T_s f_e)} \quad \text{([1] eq. 3.38)}$$

With $T_s = 800s$ and $f_e = 0.625Hz$ the decisive expected peak value for the Voorhof before renovation for return period of 12.5 years equals: $a_{19;peak}^* = 0.185 m/s^2$. In the table below, the peak values and standard deviations are given for a single damped degree of freedom and a 19 degree of freedom system for a range of the frequency.

2 different types of Spectral analysis were run in the Frequency Domain:

S Dof	Omega: 0 to 69		Time = 2 s		S Dof	Omega: 0 to 431		Time = 2 s
	a_star (a_ben_max)		sigma_a_ben			a_star (a_ben_max)		sigma_a_ben
UPPERBOUND		0.051 **Average**	0.015		**UPPERBOUND**		0.097 **Average**	0.028
(t=800)			step 0.01		(t=800)			step 0.01
S Dof	Omega: 0 to 1372		Time = 2 s		19 Dof	Omega 0 to 69		Time = 161 s
	a_star (a_ben_max)		sigma_a_ben			a_star (a_ben_max)		sigma_a_ben
UPPERBOUND		0.192 **Average**	0.054		**UPPERBOUND**		0.092 **Average**	0.026
(t=800)			step 0.01		(t=800)			step 0.1
19 Dof	Omega 0 to 431		Time = 1011 s					
	a_star (a_ben_max)		sigma_a_ben					
UPPERBOUND		0.185 **Average**	0.053					
(t=800)			step 0.1					
w_e =	3.922	rad/s						
f_0 =	0.62	Hz						
T =	800	s						

$$a^{*} = \mu_{a} + \sigma_{a}\sqrt{2\ln\left(Tf_{0}\right)}$$

Table 33: Values of 2 different spectral analysis for the Voorhof before renovation

12.7.2. Return period of one year

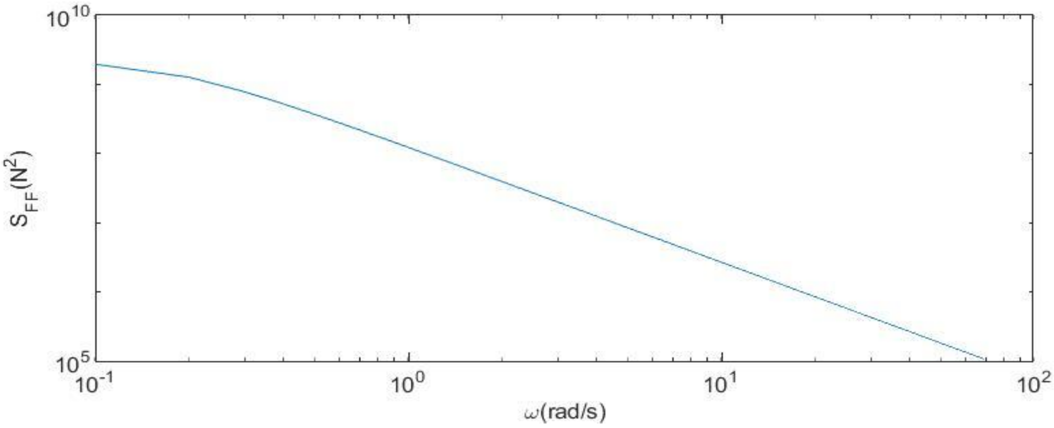


Figure 12.17: Spectra of force of node 19 of the system for return period of one year before renovation

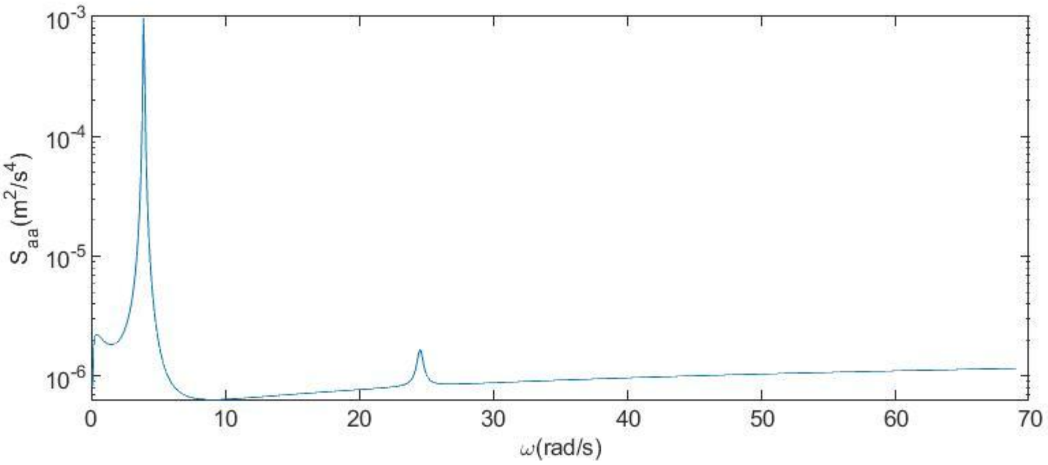


Figure 12.18: Spectra of acceleration of node 19 of the system for return period of one year before renovation

From the spectra of the accelerations the variance of the acceleration can be determined by:

$$\sigma_{a;19}^{2} = \int_{0}^{\infty} S_{aa}(\omega)\, d\omega$$

And the standard deviation follows from:

$$\sigma_{a;19} = \sqrt{\sigma^2_{a;19}}$$

The decisive standard deviation for the Voorhof before renovation for return period of one year equals r: $\sigma_{a;19} = 0.0293 m/s^2$. The expected peak value can be calculated from:

$$a^*_{19;peak;expected} = \sigma_{a;19}\sqrt{2\ln(T_s f_e)} \quad \text{([1] eq. 3.38)}$$

With $T_s = 800s$ and $f_e = 0.625Hz$ the decisive expected peak value for the Voorhof before renovation for return period of one year equals: $a^*_{19;peak} = 0.103 m/s^2$. In the table below, the peak values and standard deviations are given for a single damped degree of freedom and a 19 degree of freedom system for a range of the frequency.

2 different types of Spectral analysis were run in the Frequency Domain:

S Dof	Omega: 0 to 69		Time = 2 s		S Dof	Omega: 0 to 431		Time = 2 s
	a_star (a_ben_max)		sigma_a_ben			a_star (a_ben_max)		sigma_a_ben
UPPERBOUND		0.028 **Average**	0.008		**UPPERBOUND**		0.053 **Average**	0.015
(t=800)			step 0.01		(t=800)			step 0.01
S Dof	Omega: 0 to 1371		Time = 2 s		19 Dof	Omega 0 to 69		Time = 160 s
	a_star (a_ben_max)		sigma_a_ben			a_star (a_ben_max)		sigma_a_ben
UPPERBOUND		0.105 **Average**	0.030		**UPPERBOUND**		0.053 **Average**	0.015
(t=800)			step 0.01		(t=800)			step 0.1
19 Dof	Omega 0 to 431		Time = 1003 s					
	a_star (a_ben_max)		sigma_a_ben					
UPPERBOUND		0.103 **Average**	0.029					
(t=800)			step 0.1					
w_e =		3.922 rad/s						
f_0 =		0.62 Hz						
T =		800 s						

$$a^* = \mu_a + \sigma_a\sqrt{2\ln(Tf_0)}$$

Table 34: Values of 2 different spectral analysis for the Voorhot before renovation

12.8. Voorhof after renovation

The formula for the spectra of the acceleration of the top of the Juffertoren building will be given below. In the computer simulation 19 modes were used.

$$S_{a_{19}a_{19}}(\omega) = \sum_{i=1}^{19}\sum_{j=1}^{19} H_{a_{19}F_i}(\omega) H^*_{a_{19}F_j}(\omega) S_{F_iF_j}(\omega)$$

Where $H_{a_{19}F_i}$ is the complex transfer function of the acceleration of the 19th node for a harmonic force on node i and $H^*_{a_{19}F_j}$ is the complex conjugate of $H_{a_{19}F_j}$. For $i = j$, $S_{F_iF_j}(\omega)$ is the auto-spectra of the force and for $i \neq j$, $S_{F_iF_j}(\omega)$ is the cross-spectra of the force.

12.8.1. Return period of 12.5 years

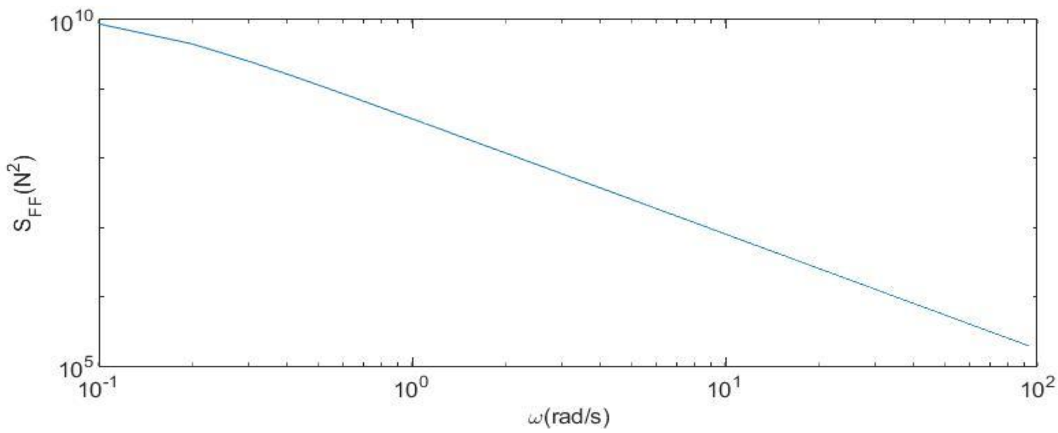


Figure 12.19: Spectra of force of node 19 of the system for return period of 12.5 years after renovation

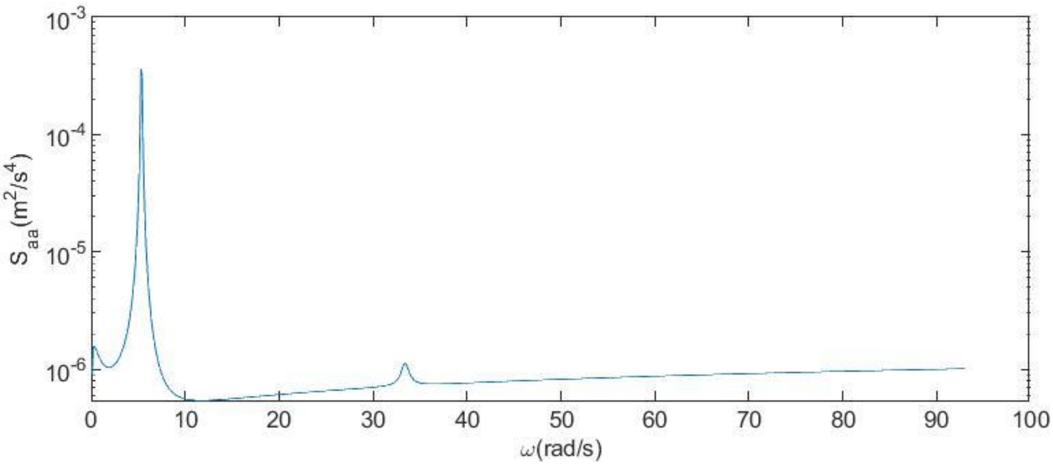


Figure 12.20: Spectra of acceleration of node 19 of the system for return period of 12.5 years after renovation

From the spectra of the accelerations the variance of the acceleration can be determined by:

$$\sigma_{a;19}^2 = \int_0^\infty S_{aa}(\omega)\, d\omega$$

And the standard deviation follows from:

$$\sigma_{a;19} = \sqrt{\sigma_{a;19}^2}$$

The decisive standard deviation for the Voorhof after renovation for return period of 12.5 years equals: $\sigma_{a;19} = 0.031 m/s^2$. The expected peak value can be calculated from:

$$a^*_{19;peak;expected} = \sigma_{a;48}\sqrt{2\ln\left(T_s f_e\right)} \quad \text{([1] eq. 3.38)}$$

With $T_s = 800s$ and $f_e = 0.85Hz$ the decisive expected peak value for the Voorhof after renovation for return period of 12.5 years equals: $a^*_{19;peak} = 0.111 m/s^2$. In the table below, the peak values and standard deviations are given for a single damped degree of freedom and a 19 degree of freedom system for a range of the frequency.

2 different types of Spectral analysis were run in the Frequency Domain:

S Dof	Omega: 0 to 93		Time = 2 s
	a_star (a_ben_max)		sigma_a_ben
UPPERBOUND	0.033	**Average**	0.009
(t=800)			step 0.01

S Dof	Omega: 0 to 587		Time = 2 s
	a_star (a_ben_max)		sigma_a_ben
UPPERBOUND	0.061	**Average**	0.017
(t=800)			step 0.01

S Dof	Omega: 0 to 1865		Time = 2 s
	a_star (a_ben_max)		sigma_a_ben
UPPERBOUND	0.118	**Average**	0.033
(t=800)			step 0.01

19 Dof	Omega 0 to 93		Time = 237 s
	a_star (a_ben_max)		sigma_a_ben
UPPERBOUND	0.049	**Average**	0.014
(t=800)			step 0.1

19 Dof	Omega 0 to 587		Time = 1419 s
	a_star (a_ben_max)		sigma_a_ben
UPPERBOUND	0.111	**Average**	0.031
(t=800)			step 0.1

w_e =	5.3456	rad/s
f_0 =	0.85	Hz
T =	800	s

$$a^* = \mu_a + \sigma_a \sqrt{2 \ln \left(T f_0 \right)}$$

Table 35: Values of 2 different spectral analysis for the Voorhof before renovation

12.8.2. Return period of one 1 year

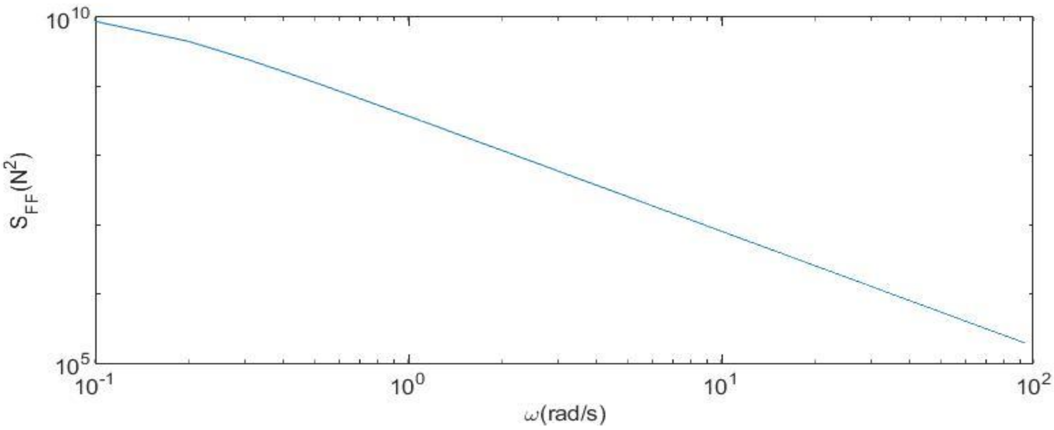


Figure 12.21: Spectra of force of node 19 of the system for return period of one year after renovation

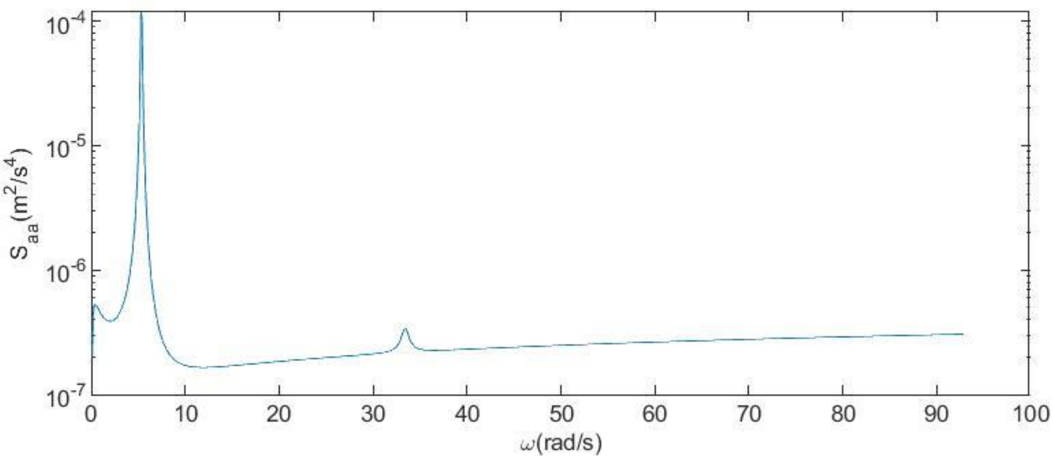


Figure 12.22: Spectra of acceleration of node 19 of the system for return period of one year after renovation

From the spectra of the accelerations the variance of the acceleration can be determined by:

$$\sigma^2_{a;19} = \int_0^\infty S_{aa}(\omega)\, d\omega$$

And the standard deviation follows from:

$$\sigma_{a;19} = \sqrt{\sigma_{a;19}^2}$$

The decisive standard deviation for the Voorhof after renovation for return period of one year equals: $\sigma_{a;19} = 0.017 m/s^2$. The expected peak value can be calculated from:

$$a^*_{19;peak;expected} = \sigma_{a;19}\sqrt{2\ln(T_s f_e)} \quad \text{([1] eq. 3.38)}$$

With $T_s = 800s$ and $f_e = 0.85Hz$ the decisive expected peak value for the Voorhof after renovation for return period of one year equals: $a^*_{19;peak} = 0.061 m/s^2$. In the table below, the peak values and standard deviations are given for a single damped degree of freedom and a 19 degree of freedom system for a range of the frequency.

2 different types of Spectral analysis were run in the Frequency Domain:

S Dof	Omega: 0 to 93		Time = 2 s			S Dof	Omega: 0 to 587		Time = 2 s
	a_star (a_ben_max)		sigma_a_ben				a_star (a_ben_max)		sigma_a_ben
UPPERBOUND	0.018	**Average**	0.005			**UPPERBOUND**	0.033	**Average**	0.009
(t=800)			step 0.01			(t=800)			step 0.01
S Dof	Omega: 0 to 1865		Time = 2 s			19 Dof	Omega 0 to 93		Time = 253 s
	a_star (a_ben_max)		sigma_a_ben				a_star (a_ben_max)		sigma_a_ben
UPPERBOUND	0.065	**Average**	0.018			**UPPERBOUND**	0.028	**Average**	0.008
(t=800)			step 0.01			(t=800)			step 0.1
19 Dof	Omega 0 to 587		Time = 1468 s						
	a_star (a_ben_max)		sigma_a_ben						
UPPERBOUND	0.061	**Average**	0.017						
(t=800)			step 0.1						
w_e =	5.3456	rad/s							
f_0 =	0.85	Hz							
T =	800	s							

$$a^* = \mu_a + \sigma_a\sqrt{2\ln\left(Tf_0\right)}$$

Table 36: Values of 2 different spectral analysis for the Voorhof after renovation

13. Comparisson Time Domain Frequency Domain

In this chapter the bending accelerations of the Juffertoren computed in the time domain (p.81) are compared to the bending accelerations computed in the frequency domain (pp.161-165). The bending accelerations of the Voorhof before and after renovation computed in the time domain are also given (p.125 and p.126) and compared to the bending accelerations computed in the frequency domain (pp.166-170 and pp.171-175). The conclusions are also given for the comparision of each situation and all situations. The comparison of time domain and frequency domain was requested by the graduation committee. For torsion and bending and torsion no comparisson will be given, because this was not done.

13.1. Juffertoren

13.1.1. Return period of 12.5 years

The peak bending acceleration value of the Juffertoren for return period of 12.5 years in time domain is $a_{peak} = 0.164 m/s^2$ (Appendix 17).

The peak bending acceleration value of the Juffertoren for return period of 12.5 years in frequency domain with random wind loading with a single damped degree of freedom system is $a^*_{peak} = 0.127 m/s^2$ (Range omega is 0-677 rad/s) (12.6.1).

The peak bending acceleration value of the Juffertoren for return period of 12.5 years in frequency domain with random wind loading with a multi damped degree of freedom system is $a^*_{peak} = 0.196 m/s^2$ (Range omega is 0-165 rad/s) (12.6.1).

The peak bending acceleration of the spectral analysis does not agree with the peak bending acceleration of the simulations in time-domain.

13.1.2. Return period of one year

The peak bending acceleration value of the Juffertoren for return period of one year in time domain is $a_{peak} = 0.147 m/s^2$.

The peak bending acceleration value of the Juffertoren for return period of one year in frequency domain with random wind loading with a single damped degree of freedom system is $a^*_{peak} = 0.048 m/s^2$ (Range omega is 0-677 rad/s) (12.6.2).

The peak bending acceleration value of the Juffertoren for return period of one year in frequency domain with random wind loading with a multi damped degree of freedom system is $a^*_{peak} = 0.077 m/s^2$ (Range omega is 0-165 rad/s) (12.6.2).

The peak bending acceleration of the spectral analysis does not agree with the peak bending acceleration of the simulations in time-domain.

13.1.3. Conclusion comparison for Juffertoren

For return period of 12.5 years and for return period of one year the comparison time domain frequency domain does not fit, the reason for this was not found.

13.2. Voorhof before renovation

13.2.1. Return period of 12.5 years

The peak bending acceleration value of the Juffertoren for return period of 12.5 years in time domain is $a_{peak} = 0.211 m/s^2$ (Appendix 18).

The peak bending acceleration value of the Juffertoren for return period of 12.5 years in frequency domain with random wind loading with a single damped degree of freedom system is $a^*_{peak} = 0.192 m/s^2$ (Range omega is 0-1372 rad/s) (12.7.1)

The peak bending acceleration value of the Juffertoren for return period of 12.5 years in frequency domain with random wind loading with a multi damped degree of freedom system is $a^*_{peak} = 0.185 m/s^2$ (Range omega is 0-431 rad/s) (12.7.1).

The peak bending acceleration of the spectral analysis does fit with the peak bending acceleration of the simulations in time-domain.

13.2.2. Return period of one year

The peak bending acceleration value of the Juffertoren for return period of one year in time domain is $a_{peak} = 0.152 m/s^2$ (10.1.1)

The peak bending acceleration value of the Juffertoren for return period of one year in frequency domain with random wind loading with a single damped degree of freedom system is $a^*_{peak} = 0.105 m/s^2$ (Range omega is 0-1372 rad/s) (12.7.2).

The peak bending acceleration value of the Juffertoren for return period of one year in frequency domain with random wind loading with a multi damped degree of freedom system is $a^*_{peak} = 0.103 m/s^2$ (Range omega is 0-431 rad/s) (12.7.2).

The peak bending acceleration of the spectral analysis does not fit with the peak bending acceleration of the simulations in time-domain.

13.2.3. Conclusion comparison for Voorhof before renovation

For return period of 12.5 years the comparison time domain frequency domain does fit but for return period of one year it does not fit, the reason for this was not found.

13.3. Voorhof after renovation

13.3.1. Return period of 12.5 years

The peak bending acceleration value of the Juffertoren for return period of 12.5 years in time domain is $a_{peak} = 0.109 m/s^2$ (Appendix 19).

The peak bending acceleration value of the Juffertoren for return period of 12.5 years in frequency domain with random wind loading with a single damped degree of freedom system is $a^*_{peak} = 0.118 m/s^2$ (Range omega is 0-1865 rad/s) (12.8.1).

The peak bending acceleration value of the Juffertoren for return period of 12.5 years in frequency domain with random wind loading with a multi damped degree of freedom system is $a^*_{peak} = 0.111 m/s^2$ (Range omega is 0-587 rad/s) (12.8.1).

The peak bending acceleration of the spectral analysis fits with the peak bending acceleration of the simulations in time-domain.

13.3.2. Return period of one 1 year

The peak bending acceleration value of the Juffertoren for return period of one year in time domain is $a_{peak} = 0.065 m/s^2$ (10.2.1).

The peak bending acceleration value of the Juffertoren for return period of one year in frequency domain with random wind loading with a single damped degree of freedom system is $a^*_{peak} = 0.065 m/s^2$ (Range omega is 0-1865 rad/s) (12.8.2).

The peak bending acceleration value of the Juffertoren for return period of one year in frequency domain with random wind loading with a multi damped degree of freedom system is $a^*_{peak} = 0.061 m/s^2$ (Range omega is 0-587 rad/s) (12.8.2).

The peak bending acceleration of the spectral analysis fits with the peak bending acceleration of the simulations in time-domain.

13.3.3. Conclusion comparison for Voorhof after renovation

For return period of 12.5 years and for return period of one year the comparison time domain frequency domain fits.

13.4. Conclusion of comparisons

Building Return Period = R	Accelerations		
	a_{peak} m/s^2	a_{peak} m/s^2	
	Time domain simulation	Frequency domain	
		SDOF approximation	NDOF
Juffertoren (R = 12.5 years)	0.164	0.127	0.196
Juffertoren (R = 1 year)	0.147	0.048	0.077
Voorhof before renovation (R = 12.5 years)	0.211	0.192	0.185
Voorhof before renovation (R = 1 year)	0.152	0.105	0.103
Voorhof after renovation (R = 12.5 years)	0.109	0.118	0.111
Voorhof after renovation (R = 1 year)	0.065	0.065	0.061

Table 37: Values comparisson time domain frequency domain

For the Juffertoren for return period of 12.5 years and for return period of one year the comparison time domain frequency domain does not fit.

For the Voorhof before renovation for return period of 12.5 years the comparison time domain frequency domain partially fits.

For the the Voorhof before renovation for return period of one year, the comparison time domain frequency domain does not fit.

For the Voorhof after renovation for return period of 12.5 years and for return period of one year the comparison time domain frequency domain fits.

The reason why the models do not fit has not been found.

14. Conclusions and Recomendations

In this chapter the conclusions and recommendations are given for the research done.

14.1. Conclusions

- It is possible to make a computational model in which the bending and torsional acceleration can be determined accurately, without the use of the Finite Element Method and wind tunnels models which saves time and costs in early design fase. This was done for the Juffertoren (pp.7-82) and it has been shown that the acceleration due to torsional motion can be very substantial $0.280 m/s^2$ (p.84). A model was also made for the Voorhof building before and after renovation (pp.91-124), and it is also shown that the acceleration due to torsional motion can be very substantial respectively $0.297 m/s^2$ and $0.170 m/s^2$ (p.135) (p.138).

- The Juffertoren, as it was planned to be built, would not have satisfied the comfort criterion of the NEN 6702 (Dutch Norm) (pp.77-81). This does not agree with Breen's conclusion [5].

- The student building Voorhof does not comply with the comfort criterion when looking at bending, torsional or bending and torsional accelerations added together in the along wind direction, even after structurally strengthening. (p.138)

- Rules of thumb are not applicable in all situations. In the case of light buildings these formulas can become invalid or result in large over estimations. (p.138) The rules of thumb that are not applicable in all situations are Woudenberg (empirical), Sheuller, Dicke/Nijsse.

14.2. Recommendations.

- A study must be conducted to see if windcomfort norm for pedestrians (NEN 8100) is satisfied. This is if, a person can walk outside a building, without been blown away. I did not do this.

- Seeing that the Dutch NEN norm does not look at the across wind acceleration, which is normally much larger than the along wind acceleration. But is this really true ? When we add the acceleration due to torsional motion on the along wind acceleration, then this along wind acceleration will once again be the dominant one. We have to check if the alongwind acceleration due to bending and torsion is dominant compared to the alongwind acceleration in concrete structures and lightweight steel structures.

- It is recommended to develop new rules of thumb.

- In this study the Saint-Venant torsion theory has been used. It is recommended to repeat the study uving the Vlasov torsion theory because restrained warping at the foundation cn have a significant effect on the torsion stiffness and on the torsion natural frequencies. In addition, the rotation stiffness of the foundation can be included in future studies.

- One limitation of this model is that the wind field is generated as if there are no other buildings in the surrounding area. This is not true in reality. Buildings in the immediate area can make the wind velocity on the building increase substantially.

- Make a model in which coupling of the modes is incorporated.

- Modeling of irregular shapes and determining how the torsional acceleration will be affected by this.

- Moving the shear center and looking at the effect on the total acceleration of the building.

- Determining the natural frequency with a FEM model, from which the maximum acceleration can be calculated and see how this corresponds to this model.

- When reviewing literature, the National Building Code of Canada (NBCC), the acrosswind acceleration is larger than the alongwind acceleration if $(bd)^{0.5}/h < 0.33$ for a rectangular shape building. (In which the b is the wind in the along wind direction.) This acrosswind acceleration is dominant for the dimensions of the Juffertoren building. ([17] p.61]). (Richting afhankelijke respons ([13] p.56]) A model would have to be made in which the vortex-shedding is taken into account to determine the across wind acceleration.

- A spectral analysis for torsion should be made. Does the spectral analysis for bending and torsion added together, equal the acellerations for bending and torsion added together in the time domain ?

References

[1] A.C.W.M. Vrouwenvelder, "Lecture notes Ct5145: Random vibrations", 2004, TU Delft, Delft.

[2] J.M.J. Spijkers, A.C.W.M. Vrouwenvelder, E.C. Klaver "Lecture notes Ct4140: Structural dynamics part1", 2005, TU Delft, Delft.

[3] Nederlands Normalisatie Instituut, "NEN 6702", 2001, Delft.

[4] Wim Bierbooms, " Simulation of stochastic wind fields which encompass measured wind speed series enabling time domain comparison of simulated and measured wind turbine loads" , European Wind Energy Confrence'04, London, 2005.

[5] Tall stories: "Active controle of wind impact on high-rise buildings", 2007, TU Delft, Delft.

[6] P.J.C. Hoogenboom, "1TMS_aantekeningen", 2007, TU Delft, Delft.

[7] J. Blauwendraad, "Theory of Elasticity, Direct Methods", 2002, Lecture Book, TU Delft, Delft.

[8] G.P.C. van Oosterhout, "Wind-induced Dynamic Behaviour of Tall buildings", 1996, TU Delft, Delft.

[9] R.D.M.N. Steenbergen "Icaros", "Super elements under stochastic wind loading", 1997, dissertation TU Delft, Delft.

[10] Nederlands Normalisatie Instituut, "Euro code 1: Actions on structures – General actions – Part1-4: Wind actions", 2005, NEN, Delft.

[11] Prof. dr. Ir. J. Blauwendraad, "Dynamica van systemen", 2003, TU Delft, Delft.

[12] A.V. Metrikine, "Dynamics, Slender structures and an introduction to Continuum Mechanics", TU Delft, Delft.

[13] I. Woudenberg, "Windbelasting en de hoogbouwdraagconstructie", master thesis TU Delft, 2002.

[14] A. Dijkstra, "Limitations to the Benchmark Skyscraper", 2008, master thesis TU Delft, Delft.

[15] Paul S. Veers, "Three-Dimensional Wind Simulation", Sandia National Laboratories.

[16] C.P.W. Geurts, "Wind-Induced Pressure Fluctuations on Buildings Facades", 1997, dissertation TU Eindhoven, Eindhoven.

[17] M.J.A. Pols, "The relation between the stiffness and serviceability of tall buildings", 1997, master thesis TU Delft, Delft.

[18] Wolfgang Schueller, Book "The Vertical Building Structure", 1990, Van Nostrand Reinhold, New York, Usa.

[19] Ivar Woudenberg, Cement 1 (2006) No.1, pp. 28-36.

[20] R. Nijsse, " Onze beroep heeft een zekere glamour", Bouwen met staal 203, pp. 19-24.

[21] The University of Western Ontario, Faculty of Engineering London, Ontario, Canada url: http://www.blwtl.uwo.ca/Public/ForceBalanceMethod.aspx

[22] The University of Western Ontario, Faculty of Engineering London, Ontario, Canada url: http://www.blwtl.uwo.ca/Public/PressureIntegrationMethod.aspx

[23] The University of Western Ontario, Faculty of Engineering London, Ontario, Canada url: http://www.blwtl.uwo.ca/Public/DeterminationofStructuralLoadsandResponses.aspx

[24] T. Kijewski, A. Kareem, " Dynamic Wind Effects: A Comparative Study of Provisions in Codes And Standards with Wind Tunnel Data" , NatHaz Modeling Laboratory, Department of Civil Engineering And Geological Sciences, University of NotreDame, Notre Dame, IN, 46556.

[25] W.C. Vis, R. Sagel, "Constructief ontwerpen in beton", 1991, Stichting Beton Prisma, Den Bosch.

[26] A.K. Chopra, "Dynamics of Structures", 2014, Pearson Education Limited.

[27] S.S. Rao, "Mechanical Vibrations", 2010, Prentice Hall.

[28] F. Bijlaard, A. Gresnigt, "Staalconstructie 2", 2004, Lecture Book, TU Delft, Delft.

List of figures

List of tables

Appendix 1 Flow diagrams for Matlab code

In this appendix we give the flow diagrams for the Matlab programs in Time Domain and Frequency Domain.

Time Domain

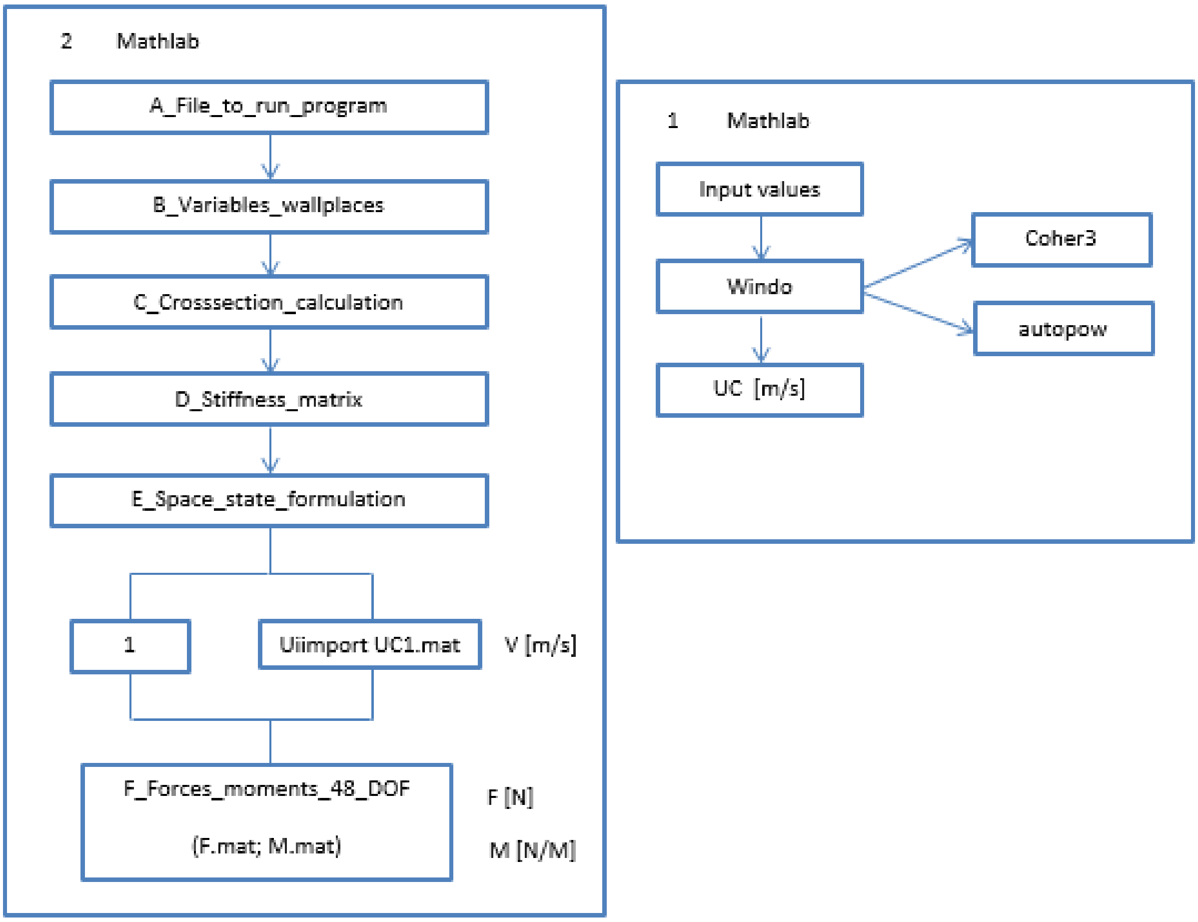


Figure 14.1: A flow diagram for Matlab in Time Domain

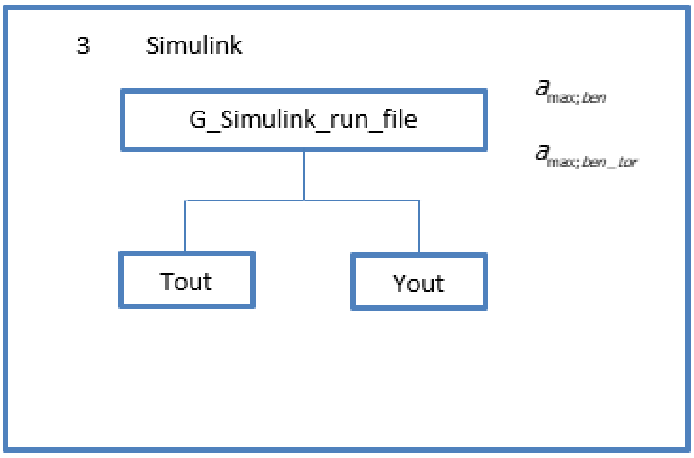


Figure 14.2: Flow diagram for Simulink in Time Domain

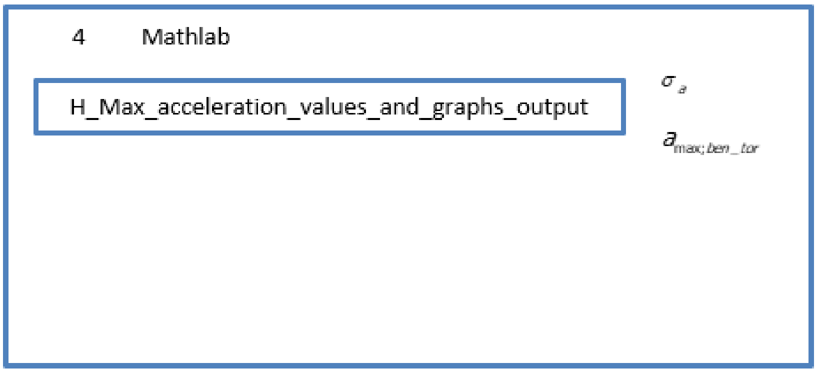


Figure 14.3: A flow diagram for Matlab in Time Domain

Frequency Domain

NDOF

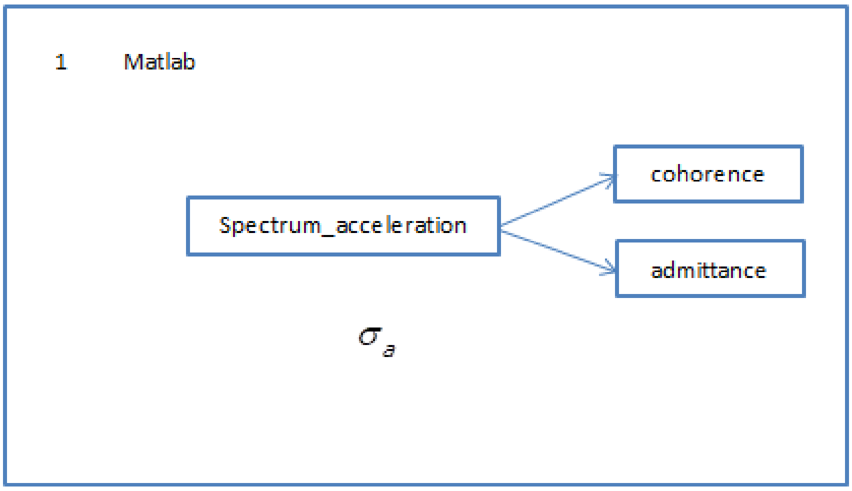


Figure 14.4: Flow diagram for Matlab in Frequency Domain (NDOF)

SDOF

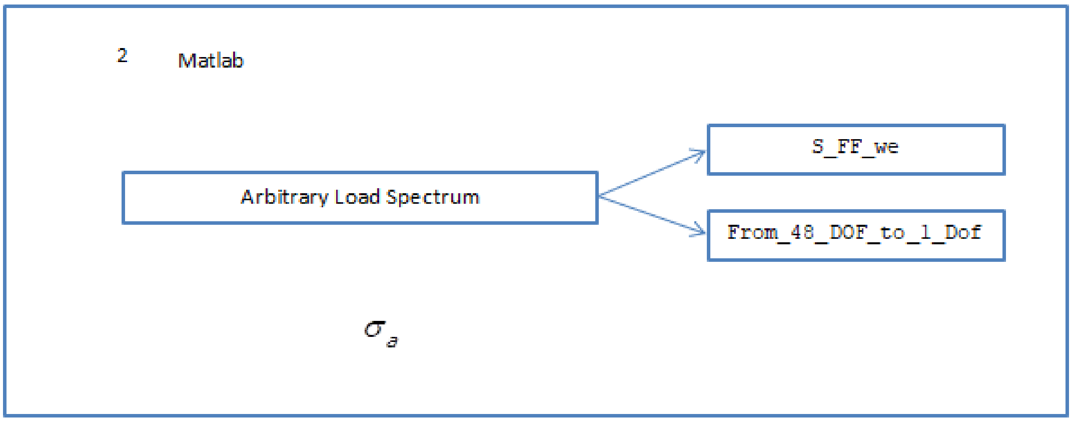


Figure 14.5: Flow diagram for Matlab in Frequency Domain (SDOF)

Appendix 2 Matlab code Juffertoren (time domain analysis)

Below the Matlab code is given for the Juffertoren for return period of one year, the only difference with the Matlab code for return period of 12.5 years is that the wind parameters for Harris and Deaves (6.2) are taken and inputted in Matlab file: A_File_to_run_programs.m and $v_{10} = 22.62 m/s$ and $\sigma_v = 6.23 m/s$ (6.3) are inputted in Matlab file: Inputvalues.m

A_File_to_run_programs.m

```
% Program to run the different modules
%
%
% Author: H.A.O.Richardson (Anthony)
% Thesis: Torsion motions of high-rise buildings due to wind loading.
%
%--------------------------------------------------------------------------
%


disp('----------------------------------------------------')
disp('Thesis: Torsion motions of high-rise buildings')
disp('----------------------------------------------------')

% Clear memory

clear
clear all
clc
close all

%--------------------------------------------------------------------------
% Building input
%--------------------------------------------------------------------------


disp('-Reading problem data')

%inputting_variables_of_the_buidling_construction

B_Variabels_Wallplaces;

% Determining of the mass, bendingstiffness polar moments and
% torsional stiffness of the cross section.


load matrices
```

```
[Area,Awalltot,Dist,DistX,DistY,xref,yref,Jtot,G,TorSTif,Ix,Iy,Ixtot,Iytot,
EI,Ipwall,Ipfloor,Ip1,Ip,Ip48,massfloor,mass1,mass48,mass,Q_gebouw,rhob,rho
11]=CCrossectioncalcualtion(L,B,H,rhoconcrete,h,l,hwall,g,nbfloors,hfl,Pvb,
qfacade,E,vu,v,xi1,xi2,tend,timestep,nwallsegm,P);

save matrices Area Awalltot Dist DistX DistY xref yref Jtot G TorSTif Ix Iy
Ixtot Iytot EI Ipwall Ipfloor Ip Ip48 massfloor mass1 mass48 mass Q_gebouw
rhob rho11 -append

%inputting_variables_of_the_buidling_construction

%buidling_construction



% Determining maximum wind profile

u_star=2.18;        % Friction velocity
kappa=0.4;          % Terrain Roughness
%d=3.5;             % average height of buildings (m)
d=0;                % average height of buildings (m) ( Ivar Woudenberg)
%z_0=2;             % Terrain Roughness according NEN 6702
z_0=0.2;            % Terrain Roughness according to Ivar Woudenberg.
step_z=1.5;         %


% DETERMING V_10

step_z=1.0;

[v_mean]=velocitymeanlogmodified(L,u_star,kappa,d,z_0,step_z);

Vhub=v_mean(1,2);


% meanwind velocity according to Harris and Deaves

step_z=1.5;

[v_mean]=velocitymeanlogmodified(L,u_star,kappa,d,z_0,step_z);

% meanwind velocity according to NEN 6702

%[v_mean]=velocitymeanlog(L,u_star,kappa,d,z_0,step_z);

vh=max(v_mean);

save matrices vh Vhub v_mean -append


%-----------------------------------------------------------------------------
% Simulink input
%-----------------------------------------------------------------------------
```

```
disp('-Input Simulink data')


% Printing Without graphs

[k,K_ben,M_ben,omega_eig_ben,a_ben,Cd_ben,phsi_ben,ktor,K_tor,M_tor,omega_e
ig_tor,a_tor,Cd_tor,Cd,Cr]=D_Stiffness_matrix(v,EI,L,l,Cd,massfloor,mass1,m
ass48,G,Jtot,Ip,Ip48,xi1,xi2);


save matrices K_ben M_ben omega_eig_ben Cd_ben a_ben v k Cr -append;
save matrices K_tor M_tor omega_eig_tor Cd_tor a_tor ktor -append;


%--------------------------------------------------------------------------
% Determining of space state formulations for simulink
%--------------------------------------------------------------------------


f=186300;
l_crossc=B;

% Run Without graphs

[A_ben,B_ben,C_ben,D_ben,fe,fe_tor,A_tor,B_tor,C_tor,D_tor]=E_Space_state_f
ormulation(f,v,K_ben,M_ben,Cd_ben,omega_eig_ben,EI,L,l_crossc,K_tor,M_tor,C
d_tor,omega_eig_tor,l,G,Jtot);


save matrices A_ben B_ben C_ben D_ben v -append;

save matrices fe fe_tor A_tor B_tor C_tor D_tor -append;


%-------------------- -----------------------------------------------------
-
% Fluctuatiing wind
%--------------------------------------------------------------------------


%--------------------------------------------------------------------------
% Simulink Run
%--------------------------------------------------------------------------


% load matrices

  uiimport UC1.mat %loading the fluctuating velocities


 Inputvalues;  %undo

 % for running windo unindent text below
```

```
 % tic
 % [t,UC]=wind0(yr,zr,U,sigma_v,N,deltat,fmax);
 % toc
 %
 % save ('UC')
 % save ('UC1','UC')  % orignal modify

%save matrices;                First time when run UC and UC1 not present
in map
%save matrices UC1 -append


Are=(yr(2)-yr(1))*(zr(2)-zr(1));% area of one node which is loaded by wind


[F,M,t,FvoorWoudenberg,F_woud]=F_Forces_moments_48_DOF(v_mean,N,Ch,Are,rho,
B,deltat,L);

save ('F','F','t') % Saving Force of each time step
save ('M','M','t') % Saving moment of each time step

save matrices  FvoorWoudenberg F_woud -append



disp('-Running Simulink ')
```

B_Variabels_Wallplaces.m

```
% Inputting varibles of the construction
%
% Author: H.A.O.Richardson (Anthony)
% Thesis: Torsion motions of high-rise buildings due to wind loading.
%
%---------------------------------------------------------------------------

% The outer dimensions of the total building

    L=144;                  % [m] length of the building
    B=26.34;                % [m] width of the building
    H=15.44;                % [m] height of the building
    rhoconcrete= 2400;      % [kg/m3]specific density concrete
    rho=1.29;               % [kg/m3]specific density air
    Ch=1.2;                 % [-] Thrust and Suction shape factor

    h=3.00;                 % [m] height of story of the building
    l=h;                    % [m] lenght of one element

    hfl=0.25;               % [m] thickness of floor of the building

    hwall=h-hfl;            % [m] height of the walls of the building

    nbfloors=L/h;           % [-] amount of floors of the building

    Pvb=70;                 % [kg/m2] weight of load on the floor (0,7kN/m2)


    qfacade=120;            % [kg/m2] weight of facade elements per m2
(1,2kN/m2)
                            % Limitation to the benchmark Skyscraper page 111)


    E=3.0e10;            % [N/m2] Youngsmodulus of the concrete C35/B65

    vu=0.15;                 % [-] Poisson ratio

    g=9.81;                 % [m/s2] Gravitational acceleration

 %--------------------------------------------------------------------------

    v= 48;                  % number of elements



    dof= 2;                 % degrees of freedom per node

 % units (meters and kN)

    xi1=0.01;              % damping ratio of the first eigenmode
    xi2=0.01;              % damping ratio of the second eigenmode
    deltat=0.1;            % delta t
```

```
% nwallseg is the amount wall segments that the walls of the cross section
is
% divided into.

    nwallsegm=16;

% Declaring matrix for the wall segments.

    P=zeros(nwallsegm,4);


% The cross-section is divided into 16 wall segments for calculation.

%  The origin of the cross sectiion of the building is taken in the left
%  bottom corner.

%    Length x          Length y          Distance x midpoint to  Distance y
midpoint to
%                                             Startpoint O
Startpoint O

P(1,1)=15.440;  P(1,2)=0.500;     P(1,3)=15.440/2;
P(1,4)=26.340-0.5/2;
P(2,1)=6.920;   P(2,2)=0.600;     P(2,3)=6.920/2;
P(2,4)=0.5+4.7+0.6+14.74+0.6/2;
P(3,1)=1.000;   P(3,2)=0.300;     P(3,3)=6.620+0.3+1.1+1.0/2;
P(3,4)=0.5+4.7+0.6+14.74+0.3/2;
P(4,1)=0.300;   P(4,2)=3.720;     P(4,3)=6.620+0.3+1.1+1.0+0.3/2;
P(4,4)=0.5+4.7+0.6+14.74+0.3-2.72/2;
P(5,1)=6.120;   P(5,2)=0.300;     P(5,3)=15.44-6.120/2;
P(5,4)=0.5+4.7+0.6+14.74-2.1-0.3/2;
P(6,1)=3.100;   P(6,2)=0.300;   P(6,3)=15.44-3.02-0.3-3.1/2;
P(6,4)=0.5+4.7+0.6+14.74-2.1-0.3-1.02-0.3/2;
P(7,1)=3.100;   P(7,2)=0.200;   P(7,3)=15.44-3.02-0.3-3.1/2;
P(7,4)=0.5+4.7+0.6+14.74-2.1-0.3-1.02-0.3-2.3-0.2/2;
P(8,1)=0.300;   P(8,2)=14.740;  P(8,3)=6.620+0.3/2;
P(8,4)=0.5+4.7+0.6+14.74/2;
P(9,1)=0.300;   P(9,2)=7.900;   P(9,3)=15.44-3.02-0.3/2;
P(9,4)=0.5+4.7+0.6+14.74/2;
P(10,1)=3.100;  P(10,2)=0.200;  P(10,3)=15.44-3.02-0.3-3.1/2;
P(10,4)=0.5+4.7+0.6+2.1+0.3+1.02+0.3+2.3+0.2/2;

P(11,1)=3.100;  P(11,2)=0.300;  P(11,3)=15.44-3.02-0.3-3.1/2;
P(11,4)=0.5+4.7+0.6+2.1+0.3+1.02+0.3/2;
P(12,1)=6.120;  P(12,2)=0.300;  P(12,3)=15.44-6.120/2;
P(12,4)=0.5+4.7+0.6+2.1+0.3/2;
P(13,1)=0.300;  P(13,2)=3.720;  P(13,3)=6.620+0.3+1.1+1.0+0.3/2;
P(13,4)=0.5+5.0+2.72/2;
P(14,1)=6.920;  P(14,2)=0.600;  P(14,3)=6.920/2;
P(14,4)=0.5+4.7+0.6/2;
P(15,1)=1.000;  P(15,2)=0.300;  P(15,3)=6.620+0.3+1.1+1.0/2;
P(15,4)=0.5+5.0+0.3/2;

P(16,1)=15.440; P(16,2)=0.500;  P(16,3)=15.440/2;
P(16,4)=0.5/2;

%save matrices
```

```
save matrices L B H rhoconcrete rho Ch h l hwall g nbfloors hfl Pvb qfacade
E vu v xi1 xi2 tend timestep deltat nwallsegm  P -append
```

CCrossectioncalcualtion.m

```
function[Area,Awalltot,Dist,DistX,DistY,xref,yref,Jtot,G,TorSTif,Ix,Iy,Ixto
t,Iytot,EI,Ipwall,Ipfloor,Ip1,Ip,Ip48,massfloor,mass1,mass48,mass,Q_gebouw,
rhob,rho11]=CCrossectioncalcualtion(L,B,H,rhoconcrete,h,l,hwall,g,nbfloors,
hfl,Pvb,qfacade,E,vu,v,xi1,xi2,tend,timestep,nwallsegm,P)
%
Syntax:[Area,Awalltot,Dist,DistX,DistY,xref,yref,Jtot,G,TorSTif,Ix,Iy,Ixtot
,Iyto
% t,EI,Ipwall,Ipfloor,Ip1,Ip,Ip48,massfloor,mass1,mass48,mass,Q_gebouw,rhob
%
]=C_Crossection_calcualtion(L,B,H,rhoconcrete,h,l,hwall,g,nbfloors,hfl,Pvb,
qfacade,E,vu,v,xi1,xi2,tend,timestep,nwallsegm,P)
%
% Calculating the variables of the cross-section
%
% Author: H.A.O.Richardson (Anthony)
% Thesis: Torsion motions of high-rise buildings due to wind loading.
%
%---------------------------------------------------------------------------


%load matrices
% Area of the wall or columbs


        for j=1:nwallsegm
            Area(j)=P(j,1)*P(j,2);
        end

        Awalltot = sum(Area);


 % Perpendicular distance between midpoint of the walls and Point O.


        for j=1:nwallsegm
            Dist(j,1)=P(j,3); Dist(j,2)=P(j,4);
        end


% Perpendicular distance between midpoint of the walls and the centre of
gravity.

        xref= (15.440/2); % xref= (H/2);
        yref= (26.34/2);  % yref= (B/2);

        for j=1:nwallsegm

            DistX(j)=(P(j,3)-xref);
            DistY(j)=(P(j,4)-yref);
        end


% Distance between the reference line and the neutral axis.
```

```
        Zxa=dot(Area,DistY)/(sum(Area));
        Zya=dot(Area,DistX)/(sum(Area));


% For sipliciy we first assume Distance between the reference line and the
neutral axis.

        Zxa=0;
        Zya=0;


% Torsion constant of each wall.

        for j=1:nwallsegm
            if(P(j,1)>P(j,2))
            J(j)=(P(j,1)*(P(j,2))^3)/3;
            else
            J(j)=(P(j,2)*(P(j,1))^3)/3;
            end
        end


% Torsion constant of all walls.

        Jtot=sum(J);

% Distance from middle of walls to centre of gravity.

        for j=1:nwallsegm
            r(j)=((DistX(j)-Zxa)^2+(DistY(j)-Zya)^2)^.5;
        end

% Shearmodulus of the concrete.( Ridigidy modulus)

        G=E/(2*(1+vu));

% The torsional stiffness of one floor of the structure.

        TorSTif=G*Jtot;

 % Second moment of inertia and steiners rule

        for j=1:nwallsegm
            Ix(j)=(P(j,1)*(P(j,2))^3)/12+((DistY(j)^2)*Area(j));
            Iy(j)=(P(j,2)*(P(j,1))^3)/12+((DistX(j)^2)*Area(j));
        end

        Ixtot=sum(Ix);
        Iytot=sum(Iy);

 % Bending stiffness in X-direction

        EI=E*Iytot;
```

```
 % Polar inertia of the walls Ix and Iy of a floor

        Ipwall=Ixtot+Iytot;

 % Polar inertia of the floor Ix and Iy

        Ipfloor=((1/12)*H*B^3)+((1/12)*B*H^3);


   % Polar inertia of the bottom floor of the building

        Ip1=Ipwall*(3/2)*h*rhoconcrete+Ipfloor*(hfl*rhoconcrete+Pvb);


   % Polar inertia of the floor segments node 2 to node 47

        Ip=Ipwall*h*rhoconcrete+Ipfloor*(hfl*rhoconcrete+Pvb);

   % Polar inertia of the top floor of the building

        Ip48=Ipwall*(h/2)*rhoconcrete+Ipfloor*(hfl*rhoconcrete+Pvb);


% Mass of bottom floor of the building node 1 extra length wall
% Loading of floor gekozen( 70 kg/m2 ) qfacade= ( 120 kg/m2 )

        mass1=(Awalltot+(h/2))*hwall*rhoconcrete+H*B*hfl*rhoconcrete+(H*B-
Awalltot)*Pvb+qfacade*2*B*H*h;


% The mass of one storey existing of walls, floor and loading on the floor
% node 2- 47
% Loading of floor gekozen( 70 kg/m2 ) qfacade= ( 120 kg/m2 )

        massfloor=Awalltot*hwall*rhoconcrete+H*B*hfl*rhoconcrete+(H*B-
Awalltot)*Pvb+qfacade*2*B*H*h;


% Mass of top floor of the building


        % No VB on roof

        mass48=sum(Area)*(h/2)*rhoconcrete+(H*B-
Awalltot)*hfl*rhoconcrete+qfacade*2*B*H*(h/2);


% Total mass of the buidling

        % mass building of 48 floors walls + top floor

        mass=mass1+(v-2)*massfloor+mass48;
```

```
% Total weigth of the building

        Q_gebouw=mass*g;           % Gewicht building

% Spefic density of building              [kg/m3]

        rhob=mass/(B*H*L);

% mass per meter building height          [kg/m1]

        rho11=mass/(L);
```

D_Stiffness_matrix.m

```
function[k,K_ben,M_ben,omega_eig_ben,a_ben,Cd_ben,phsi_ben,ktor,K_tor,M_tor
,omega_eig_tor,a_tor,Cd_tor,Cd,Cr]=D_Stiffness_matrix(v,EI,L,l,Cd,massfloor
,mass1,mass48,G,Jtot,Ip,Ip48,xi1,xi2)
%
% Determining stiffness and torsion matrix
%
%INPUT
%
%   v:              number of elements                  [-]
%   EI:             bending stiffness                   [Nm2]
%   l:              lenght of element                   [m]
%   massfloor:      mass of a floor  1-47 floor         [m]
%   mass1:          mass of bottom floor                [m]
%   mass48:         mass of top floor                   [m]
%   G:              Shear modulus                       [N/mm2]
%   Jtot:           Torsion constant                    [mm4]
%   Ip:             Second moment of inertia 1-47 floor [mm4]
%   Ip48:           Second moment of inertia 48 floor   [mm4]
%   xi1:            damping ratio of first eigenmode    [-]
%   xi2:            damping ratio of second eigenmode   [-]
%
%OUTPUT
%
%   k:              Bending stiffness field elements
%   K_ben:          Bending stiffness matrix
%   M_ben:          Mass stiffness matrix
%   omega_eig_ben:  Eigen frequency bending matrix MDF
%   a_ben:          Damping martix first and second term
%   Cd_ben:         Damping matrix bending MDF
%   phsi_ben:       Damping ratios matrix bending
%   ktor:           Torsional stiffness field elements
%   K_tor:          Torsion stiffness matrix
%   M_tor           Mass torsion matrix
%   omega_eig_tor:  Eigen frequency torsion matrix MDF
%   a_tor:          Damping martix first and second term
%   cd_tor:         Torsion damping matrix MDF
%
%   Cd:             Damping matrix contoles system bending MDF
%   omega_eig:      Eigen frequency contolled bending matrix MDF
%
% Thesis: Torsion motions of high-rise buildings due to wind loading.
% Author: H.A.O.Richardson (Anthony)

%---------------------------------------------------------------------------



%-------------------------------------------------------------------------
% Assembling the stiffness matrix
%-------------------------------------------------------------------------
% Bending Field elements


    k=EI/l^3*[1     -2     1;
             -2     4    -2;
              1     -2     1];% bending element stifnessmatrix
```

```
    K_ben=zeros(v+4,v+4); % Total systemmatrix bending
    for o=0:1:(v-1)%v-1
        for n=1:1:3
            for m=1:1:3
                K_ben(o+n,o+m)=K_ben(o+n,o+m)+k(n,m);
            end
        end
    end



 %Rearanging bending matrix for simulink test

    K_ben(:,1)=[];
    K_ben(1,:)=[];

    K_ben(:,1)=[];
    K_ben(1,:)=[];


% Top element
% Delting the unwanted and coloum k-matrix for half element.

    K_ben(:,50)=[];         % K_ben(:,((Size(K_ben,2))+2)=[]
    K_ben(50,:)=[];

    K_ben(:,49)=[];
    K_ben(49,:)=[];


%---------------------------------------------------------------------------
% Assembling the mass matrix
%---------------------------------------------------------------------------

    M_ben=zeros(v,v);

    for n=1:1:v
        M_ben(n,n)=massfloor;

    end

    % Making the first and last term of mass matrix correct

    %M_ben(1,1)=mass1;
    M_ben(n,n)=mass48;


%---------------------------------------------------------------------------
% Determing the eigen frequency (MDF)
%---------------------------------------------------------------------------


    % E is the modal matrix; omegakw is modal K*E = M*E*OMEGAKW

    [E_ben,omegakw_ben] = eig(K_ben,M_ben);
    for n=1:1:v
```

```
        omega_eig_ben(n)=sqrt(omegakw_ben(n,n));
    end


%-----------------------------------------------------------------------------
% Assembling of the damping matrix
%-----------------------------------------------------------------------------

% Damping on mode 1 and mode 2

    a_ben=2*(inv([1/(omega_eig_ben(1)) (omega_eig_ben(1));
1/omega_eig_ben(2) omega_eig_ben(2)])*[xi1;xi2]);

 % Damping on mode 1 and mode 10

    %a_ben=2*(inv([1/(omega_eig_ben(1)) (omega_eig_ben(1));
1/omega_eig_ben(10) omega_eig_ben(10)])*[xi1;xi2]);

    Cd_ben=a_ben(1,1)*M_ben+a_ben(2,1)*K_ben;


%-----------------------------------------------------------------------------

%-----------------------------------------------------------------------------
% plotting damped motion
%-----------------------------------------------------------------------------

% damping ratio of the eigenmodes of the structure

    for n=1:1:v;
phsi_ben(n)=a_ben(1,1)/(2*omega_eig_ben(n))+a_ben(2,1)/2*omega_eig_ben(n);
    end



%---------------------------------------------------------------------------
% Torsional
%---------------------------------------------------------------------------



%---------------------------------------------------------------------------
% Assembling the stiffness matrices
%---------------------------------------------------------------------------

    % Torsional Field elements


    ktor=(G*Jtot)/l*[1  -1
                    -1   1];   % 1 floor height is 3 meters


    K_tor=zeros(v+2,v+2); % Total systemmatrix bending
```

```
    for o=0:1:(v-1)
      for n=1:1:2
        for m=1:1:2
            K_tor(o+n,o+m)=K_tor(o+n,o+m)+ktor(n,m);
        end
      end
    end


% Bottom element
% Delting the unwanted row and coloum k-matrix for half element.

    K_tor(:,1)=[];
    K_tor(1,:)=[];


% Top element
% Delting the unwanted and coloum k-matrix for half element.

    K_tor(:,49)=[];
    K_tor(49,:)=[];



%---------------------------------------------------------------------------
% Assembling the mass matrix
%---------------------------------------------------------------------------

    M_tor=zeros(v,v);

    for n=1:1:v
        M_tor(n,n)=Ip;
    end

% Making the first and last term of the torsional matrix correct.

    M_tor(n,n)=Ip48;


%---------------------------------------------------------------------------
% Determing the eigen frequency (MDF)
%---------------------------------------------------------------------------


    % E is the modal matrix; omegakw is modal K*E = M*E*OMEGAKW

    [E_tor,omegakw_tor] = eig(K_tor,M_tor);
    for n=1:1:v
        omega_eig_tor(n)=sqrt(omegakw_tor(n,n));
    end


%---------------------------------------------------------------------------
% Assembling of the damping matrix torsion
%---------------------------------------------------------------------------
```

```
    % Damping on mode 1 and mode 2

    a_tor=2*(inv([1/(omega_eig_tor(1)) (omega_eig_tor(1));
1/omega_eig_tor(2) omega_eig_tor(2)])*[xi1;xi2]);


    % Damping on mode 1 and mode 10

    %a_tor=2*(inv([1/(omega_eig_tor(1)) (omega_eig_tor(1));
1/omega_eig_tor(10) omega_eig_tor(10)])*[xi1;xi2]);


    Cd_tor=a_tor(1,1)*M_tor+a_tor(2,1)*K_tor;


%---------------------------------------------------------------------------

%---------------------------------------------------------------------------
% plotting damped motion
%---------------------------------------------------------------------------

% damping ratio of the eigenmodes of the structure

    for n=1:1:v;

phsi_tor(n)=a_tor(1,1)/(2*omega_eig_tor(n))+a_tor(2,1)/2*omega_eig_tor(n);
    end
```

E_Space_state_formulation.m

```
function[A_ben,B_ben,C_ben,D_ben,fe,fe_tor,A_tor,B_tor,C_tor,D_tor]=E_Space
_state_formulation(f,v,K_ben,M_ben,Cd_ben,omega_eig_ben,EI,L,l_crossc,K_tor
,M_tor,Cd_tor,omega_eig_tor,l,G,Jtot)
%
% Determining Space state formulation bending and torsion,Plotting
% displacement of building bending and torsion,Space stateformulaton
% bending and torsion,Validation of dynamic behaviour bending and torsion.
%
%INPUT
%
%   f:              Total force of horizontal row of nodes   [N]
%   v:              Number of elements                       [-]
%   K_ben:          Bending stiffness matrix
%   M_ben:          Mass stiffness matrix
%   Cd_ben:         Damping matrix bending MDF
%   omega_eig_ben:  Eigen frequency bending matrix MDF       [rad]
%   EI:             bending stiffness                        [Nm2]
%   L:              Height building                          [m]
%   l_crossc:       Width building                           [m]
%   K_tor:          Torsion stiffness matrix
%   M_tor:          Mass torsion matrix
%   Cd_tor:         Damping matrix bending MDF
%   omega_eig_tor:  Eigen frequency torsion matrix MDF       [rad]
%   l:              Height building                          [m]
%   G:              Shear modulus                            [N/m2]
%   Jtot:           Torsion constant                         [mm4]
%
%OUTPUT
%
%   A_ben           State space bending input matrix for simulink
%   B_ben           State space bending input matrix for simulink
%   C_ben           State space bending input matrix for simulink
%   D_ben           State space bending input matrix for simulink
%   fe              Bending Frequency                        [Hz]
%   fe_tor          Torsion Frequency                        [Hz]
%   A_tor           State space torsion input matrix for simulink
%   B_tor           State space torsion input matrix for simulink
%   C_tor           State space torsion input matrix for simulink
%   D_tor           State space torsion input matrix for simulink
%
%
% Thesis: Torsion motions of high-rise buildings due to wind loading.
% Author: H.A.O.Richardson (Anthony)




%--------------------------------------------------------------------------
% Space state formulation
% Changing the matrices into formulation readable for simulink
%--------------------------------------------------------------------------



%--------------------------------------------------------------------------


% Determining static diplacement
```

```
%-----------------------------------------------------------------------------


    %f=186300; %  Force on each node in N

    F= zeros(v,1);

        for  lr=1:1:v
             F(lr,1)=f;
        end

% Bending Displacement


    u_ben=(-inv(K_ben))*F;

% The z cordinates of the nodes in vertical direction

    Z= zeros(v,1);

    zzz= 1.5;

    for  zrn=1:1:v

        Z(zrn,1)=(zzz+(zrn-1)*3);
    end

%-----------------------------------------------------------------------------

%Space State Formulation.

%-----------------------------------------------------------------------------

%{%}
% Placing space state formlation Matrix A

    A2_ben=eye(v,v);
    A3_ben=-inv(M_ben)*K_ben;
    A4_ben=-inv(M_ben)*Cd_ben;

% Placing space state formlation Matrix B

    B1_ben=inv(M_ben);
    C1_ben=eye(v,v);

    A_ben(1:v,v+1:1:2*v)=A2_ben;
    A_ben(v+1:1:2*v,1:v)=A3_ben;
    A_ben(v+1:1:2*v,v+1:1:2*v)=A4_ben;
    B_ben(v+1:1:2*v,1:v)=B1_ben;

    C_ben=eye(2*v,2*v);
    D_ben=zeros(2*v,v);
```

```
%--------------------------------------------------------------------------

% Torsional movement

%--------------------------------------------------------------------------


%--------------------------------------------------------------------------

% Determining static diplacement Torsion

%--------------------------------------------------------------------------

% over on each side 0.17
    over=0.17;
% horizondat division length
    Yrd=2.6/2;
    Arm=l_crossc/2-over-Yrd;
    %f=186300; %  Force on each node in N
    f=62100; %  Force per meter on each node in N
    m=f*Arm*30; % *30 voor groter moment 21797100 Nm


% To validate the static torsional displacement we will put a positive
moentt and then a negative monet on each consecative node.


    M= zeros(v,1);

    for  lr=1:2:v
         M(lr,1)=m;
         M(lr+1,1)=-m;
    end

u_tor=-inv(K_tor)*M;

% The z cordinates of the nodes in vertical direction

Zy_tor= zeros(v,1);

    zzz= 1.5;

for  zrn=1:1:v

      Zy_tor(zrn,1)=(zzz+(zrn-1)*3);
end


%--------------------------------------------------------------------------

%Space State Formulation.

%--------------------------------------------------------------------------
```

```
%{%}
% Placing space state formlation Matrix A

    A2_tor=eye(v,v);
    A3_tor=-inv(M_tor)*K_tor;
    A4_tor=-inv(M_tor)*Cd_tor;

% Placing space state formlation Matrix B

    B1_tor=inv(M_tor);
    C1_tor=eye(v,v);

    A_tor(1:v,v+1:1:2*v)=A2_tor;
    A_tor(v+1:1:2*v,1:v)=A3_tor;
    A_tor(v+1:1:2*v,v+1:1:2*v)=A4_tor;
    B_tor(v+1:1:2*v,1:v)=B1_tor;
    C_tor=eye(2*v,2*v);
    D_tor=zeros(2*v,v);
```

autopow.m

```
function S=autopow(f,U,sigma)
% syntax: function S=autopow(f,U,sigma_v)
% Autopower spectral density function of turbulence
% Input:
%   f: frequency (Hz)
%   U: the (10 minute) mean wind speed (m/s)
%   sigma_v: standard deviation (m/s)
% Output:
%   S: autopower spectral density (m^2/s)
%
% Thesis: Torsion motions of high-rise buildings due to wind loading
% Author: H.A.O. Richardson (Anthony)

%---------------------------------------------------------------------------
% Reader CT 5145 Chapter 6 Page 12.
%---------------------------------------------------------------------------

L=1200;

%---------------------------------------------------------------------------
% Dimesionless frequency
%---------------------------------------------------------------------------

df=f.*L./U;       % Normally written as caracter x


%---------------------------------------------------------------------------
% Davenport spectrum
%---------------------------------------------------------------------------


S=(2/3)*((f.*L./U).^2./((1+(f.*L./(U)).^2).^(4/3))).*(sigma.^2./(f));
```

Coher3.m

```
function Coh=Coher3(f,Yrtot,Zrtot,U,option)
% syntax: function Coh=coher(f,Yrtot,Zrtot,Vhub,zhub,option)
% Coherency function of longitudinal wind velocity fluctuations
% Implemented options:
%   %   1. Coherence of longitudinal and lateral wind velocity fluctuations
%
% Input:
%   f:      frequency
[Hz]
%   Yrtot:  lateral distance (in projection of rotor plane)              [m]
%   Zrtot:  longditudinal distance (in projection of rotor plane)        [m]
%   Vhub:   the 10 minute average wind speed at hub height [m/s}
%   zhub: the hub height of the wind turbine (m)
% Output:
%   Coh: coherency (-)
%
% Thesis: Torsion motions of high-rise buildings due to wind loading
% Author: H.A.O. Richardson (Anthony)
%
%--------------------------------------------------------------------------


Cz=7;                    % Londitudinal  Coherence
Cy=10;                   % Lateral       Coherence




if (option==1)

% Coherency according to NEN 6702
% Turbulent scale parameter

    x=f.*((sqrt((Cz.^2).*((Zrtot).^2)+(Cy.^2).*((Yrtot).^2)))./U);
    Coh=exp(-1.*x);

else
  error 'option not implemented in COHER'
end
```

wind0.m

```
function [t,UC]=wind0(yr,zr,U,sigma_v,N,deltat,fmax)
% simulation of a turbulent wind field
%
% INPUT:
%     yr, zr:     specification of coordinates on the facade of the structure
%     U:          mean wind velocity at 10 m above the surface of the earth
(m/s)
%     sigma_v:    standard deviation of the fluctuating part of the wind
speed (m/s)
%     N:          number of time points (including zero); N must be a power
of 2
%     deltat:     time step (s)
%     fmax:       maximum frequentie spectrum (Hz)
%
% OUTPUT:
%     UC:         constrained turbulent wind velocities (m/s)
%
% Thesis: Torsion motions of high-rise buildings due to wind loading
% Modified by: H.A.O. Richardson (Anthony)

%yr=1.3:2.6:24.7;
%zr=9:1.5:144;
%v_10=16.36;
%sigma=6.345;
%N=10;
%deltat=.1;
%fmax=5;

% number of points in rotor plane
Ny=length(yr);
Nz=length(zr);
Np=Ny*Nz;

% y and z coordinates of all rotor points in one column vector
Yr=reshape(yr'*ones(1,Nz),Np,1);
Zr=reshape(ones(Ny,1)*zr,Np,1);



Yrtot=zeros(Np,Np);
for i=1:Np
   for j=i+1:Np
      % distances between points
      Yrtot(i,j)=Yr(i)-Yr(j);
      Yrtot(j,i)=Yrtot(i,j);
   end
end



Zrtot=zeros(Np,Np);
for i=1:Np
   for j=i+1:Np
      % distances between points
      Zrtot(i,j)=(Zr(i)-Zr(j));
      Zrtot(j,i)=Zrtot(i,j);
   end
```

```
end


% time vector
t=[0:N-1]'*deltat;
% period
T=N*deltat;
% frequency step
deltaf=1/T;
% discretized frequencies
k=[1:N/2-1]';
f=k.*deltaf;
% autopower spectral density (one-sided)
Sa=autopow(f,U,sigma_v);
% spectrum is cut-off above fmax by application of window
Index=find(f>fmax);
if ~isempty(Index)
  Nw=Index(1);
  w=zeros(N/2-1,1);
  W=window('hann',2*Nw+1);w(1:Nw+1)=W(Nw+1:2*Nw+1);
  Sa=w.*Sa;
end
% renormalize Sa to variance
Sa=sigma_v^2/(sum(Sa)/T)*Sa;

% Fouriercoefficients points in rotor plane
ak=zeros(Np,N/2-1);
bk=zeros(Np,N/2-1);
for k=1:N/2-1
   Coh=coher3(f(k),Yrtot,Zrtot,U,1);
   % Choleski decomposition
   L=sqrt(Sa(k)/T)*chol(Coh)';
   % vector of unit variance normal random numbers
   ran=randn(Np,1);
   ak(:,k)=L*ran;
   ran=randn(Np,1);
   bk(:,k)=L*ran;
end

% complex notation
i=sqrt(-1);
UC=zeros(N,Np);
for j=1:Np
   C=ak(j,:)'-i*bk(j,:)';
   C=1/2*[0;C;0;rot90(C')];
   % inverse FFT
   uc=N*ifft(C);
   if any(abs(imag(uc)) >= 1e-7*abs(uc) & abs(imag(uc)) >= 1e-12)
     max(abs(uc))
     max(imag(uc))
     error('imag too large uc')
   end
   UC(:,j)=real(uc);
end
% reshape UC: separate indices for y and z
UC=reshape(UC,N,Ny,Nz);
```

F_Forces_moments_48_DOF.m

```
function[F,M,t,FvoorWoudenberg,F_woud]=F_Forces_moments_48_DOF(v_mean,N,Ch,
Are,rho,B,deltat,L)
% Syntax: function[F,M]=Force_mod(v_mean)
%
% Summation of mean wind speed and flucuating wind speed determining the
% Nodal forces and Nodal moments for the grided area.
%
%INPUT
%   vmean:       Mean wind speead at height
[kg/m]
%   UC1:         Fluctuating velocity from the random generator
[m/s]
%
%OUTPUT
%   F            Force martix                                          [N]
%   M            Moment matrix
[Nm]
%
%
% Thesis: Torsion motions of high-rise buildings due to wind loading.
% Author: H.A.O.Richardson (Anthony)
%-----------------------------------------------------------------------------

load matrices

load UC1                %Importing the fluctuating velocity from the random
generator.
Uf(:,:,3:1:93)=UC;      %places matrix UC in matrix Uf from position 3 to
93
Uf(:,:,1)=Uf(:,:,3);    %places the fluctuating values of Uf(:,:,3) in
Uf(:,:,2)and Uf(:,:,1)
Uf(:,:,2)=Uf(:,:,3);

N=8192;

% Mean velocity according to NEN 6702
%v_mean(1,1,3:1:93)=u_star/kappa*log(z-d/z_0);   % mean wind speed at
reference height
%for n=1:

v_mean2(1,1,3:1:93)=v_mean;
v_mean2(1,1,1)=v_mean2(1,1,3);                      %velocity out of vmean(1,1,3)
is given to vmean(1,1,1 and 2)
v_mean2(1,1,2)=v_mean2(1,1,3);

v_mean=v_mean2;

% Mean velocity is determined for each time step
% There is 10 timesteps in a second this is the 1 to 10 in vmean
% v_mean(:,1,:)or N
% v_mean(:,1,:) nodes in the width direction which are 10  for this
% buidling


v_mean(:,1,:)=v_mean(1,1,:);
```

```
v_mean(:,2,:)=v_mean(1,1,:);
v_mean(:,3,:)=v_mean(1,1,:);
v_mean(:,4,:)=v_mean(1,1,:);
v_mean(:,5,:)=v_mean(1,1,:);
v_mean(:,6,:)=v_mean(1,1,:);
v_mean(:,7,:)=v_mean(1,1,:);
v_mean(:,8,:)=v_mean(1,1,:);
v_mean(:,9,:)=v_mean(1,1,:);
v_mean(:,10,:)=v_mean(1,1,:);


v_mean=repmat(v_mean,[N 1 1]);

U=Uf+v_mean;         %Sommation of the fluctuation velocity and the mean
velocity in matrix U
U(:,:,94)=0;         % U(:,:,94)=0;This term in needed for the active
damping system ?? on even number at top

F=1/2*Are*Ch*rho*(U).^2; % turrning velocities on area into forces

%--------------------------------------------------------------------------

 %Moment code

%--------------------------------------------------------------------------


M=zeros(size(F));    % making Moment matrix (M)the same size as F matrix


for n=1:1:94         % Reading 94 half heights of 1,5 meters into 45

           % make 10 a variable
    for p=1:1:10            % horizontal distance is divided in 10 pieces Yr

    if p<6
        M(:,p,n)=F(:,p,n)*(-1)*((B/2-(p-0.5)*2.6)-0.17);        % B/2-over-
Yrd;
    else
        M(:,p,n)=F(:,p,n)*(1)*((B/2+((p-5.5)*2.6)-0.17));

    end
    end

end




M=sum(M,2);              % Summation of the moments of each row in the
matrix. 6 horizontal places
M=squeeze(M);            % remove the singleton variable.


m(1:1:N,1:1:2)=2*M(1:1:N,1:1:2);    %onderste 2 nodes
```

```
for n=3:1:48                          % Summating the system 90 half heights
of 1,5 meters into 45
    m(:,n)=M(:,2*n-3)+M(:,2*n-2);     % into 45 spaces of 3 meters
    m(:,n)=M(:,2*n-3)+M(:,2*n-2);     % into 45 spaces of 3 meters
end
M=m;

%the torsion stiffess was taken in the stifnessmatix per meter height here
it is per 3 meters height so i have to diide the moents by 3 to get the
same height.


%--------------------------------------------------------------------------

%Ending of moment code

%--------------------------------------------------------------------------




F=sum(F,2);              % Summation of the forces of each row in the
matrix. 6 horizontal places
F=squeeze(F);            % remove the singleton variable.

f(1:1:N,1:1:2)=2*F(1:1:N,1:1:2);      % onderste 2 nodes
for n=3:1:48                          % Summating the system 90 half heights
of 1,5 meters into 45
    f(:,n)=F(:,2*n-3)+F(:,2*n-2);     % into 45 spaces of 3 meters
end
F=f;


t=[deltat:deltat:(N*deltat)]';


FvoorWoudenberg=sum(F,2);
F_woud=(FvoorWoudenberg(1))/L;
```

H_Max_acceleration_values_and graphs_output.m

```
%
% Thesis: Torsion motions of high-rise buildings due to wind loading.
% Author: H.A.O.Richardson (Anthony)
%
%--------------------------------------------------------------------------
%

close all


load matrices

%--------------------------------------------------------------------------
% Determining the maximum bending acceleration in yout
%--------------------------------------------------------------------------

we=omega_eig_ben(1,1);


% Determining of the highest acceleration, mean and standard deviation of
% the realization.

    REALI_max_a=max(yout(50000:end,3));
    REALI_mean_a=mean(yout(50000:end,3));
    REALI_std=std(yout(50000:end,3),1);


%Determining the mean and standard deviation of the realization of time
%50 sec to 800 sec

Reali_a_48_peak_750_ben=REALI_mean_a+REALI_std*(2*log(21600*(we/2*pi)))^0.5
;

     save matrices REALI_max_a REALI_mean_a REALI_std
Reali_a_48_peak_750_ben -append


%--------------------------------------------------------------------------
% Determining the maximum torsional acceleration in yout (Right)
%--------------------------------------------------------------------------


% Determining of the highest acceleration, mean and standard deviation of
% the realization.

    REALI_max_a_tor_R=max(yout(50000:end,6));
    REALI_mean_a_tor_R=mean(yout(50000:end,6));
    REALI_std_tor_R=std(yout(50000:end,6),1);


%Determining the mean and standard deviation of the realsisation of time
%50 sec to 800 sec
```

```
Reali_a_48_peak_750_tor_R=REALI_mean_a_tor_R+REALI_std_tor_R*(2*log(21600*(
we/2*pi)))^0.5;


save matrices REALI_max_a_tor_R REALI_mean_a_tor_R REALI_std_tor_R
Reali_a_48_peak_750_tor_R -append

%-------------------------------------------------------------------------
% Determining the maximum torsional acceleration in yout (Left)
%-------------------------------------------------------------------------



% Determining of the highest acceleration, mean and standard deviation of
% the realization.

    REALI_max_a_tor_L=max(yout(50000:end,10));
    REALI_mean_a_tor_L=mean(yout(50000:end,10));
    REALI_std_tor_L=std(yout(50000:end,10),1);


%Determining the mean and standard deviation of the realsisation of time
%200 sec to 800 sec


Reali_a_48_peak_750_tor_L=REALI_mean_a_tor_L+REALI_std_tor_L*(2*log(21600*(
we/2*pi)))^0.5;


save matrices REALI_max_a_tor_L REALI_mean_a_tor_L REALI_std_tor_L
Reali_a_48_peak_750_tor_L -append


%-------------------------------------------------------------------------
% Determining the maximum acceleration bending and torsion in yout (right)
%-------------------------------------------------------------------------

    REALI_max_a_ben_tor_R=max(yout(50000:end,7));
    REALI_mean_a_ben_tor_R=mean(yout(50000:end,7));
    REALI_std_ben_tor_R=std(yout(50000:end,7),1);



%-------------------------------------------------------------------------
% Determining the maximum acceleration bending and torsion in yout (left)
%-------------------------------------------------------------------------

    REALI_max_a_ben_tor_L=max(yout(50000:end,11));
    REALI_mean_a_ben_tor_L=mean(yout(50000:end,11));
    REALI_std_ben_tor_L=std(yout(50000:end,11),1);

Reali_a_48_peak_750_ben_tor_L=REALI_mean_a_ben_tor_L+REALI_std_ben_tor_L*(2
*log(21600*(we/2*pi)))^0.5;
```

```
save matrices REALI_max_a_ben_tor_R REALI_max_a_ben_tor_L
REALI_mean_a_ben_tor_R REALI_mean_a_ben_tor_L REALI_std_ben_tor_R
REALI_std_ben_tor_L Reali_a_48_peak_750_ben_tor_L -append


%-----------------------------------------------------------------------------
% Plotting the accelerations of yout
%-----------------------------------------------------------------------------



figure
plot(tout, yout(:,3));
hold on
axis([0 800 -0.12 0.12])
    xlabel('t (s)')
    ylabel('a (m/s^2)')
    title('a48 ben')

hold off

saveas(gcf, 'a48_ben.jpeg', 'jpeg');



figure
plot(tout, yout(:,6));
hold on
axis([0 800 -0.4 0.4])
    xlabel('t (s)')
    ylabel('a (m/s^2)')
    title('a48 tor')

hold off

saveas(gcf, 'a48_tor.jpeg', 'jpeg');



figure
plot(tout, yout(:,7));

hold on
axis([0 800 -0.45 0.45])
    xlabel('t (s)')
    ylabel('a (m/s^2)')
    title('a48 ben tor')
hold off

saveas(gcf, 'a48_ben_tor.jpeg', 'jpeg');
```

velocitymeanlogmodified.m

```
function [v_mean]=velocitymeanlogmodified(L,u_star,kappa,d,z_0,step_z)
%syntax: function velocitymeanlog
% The corrected log profile Harris and Deaves
%
%
% INPUT:          (Windbelasting en de hoogbouwdraagconstructie page 84)
%
%   L:            length of the building (m)
%   u_star:       shear velocity (m/s)
%   kappa:        Von Karman constant = 0.4
%   d:            average height of buildings (m)
%   z:            height above the face of the earth (m)
%   z_0:          measure for the roughness of the terrain (roughness length)
%   step_z:       height of grid in which the building is devided in z
%                 direction (m)
%
% Calcualted input
%   a:            dimensionless argument (-)
%   z_g           gradient height (m)
%   f_c           Coriolisparameter (s -1)
%   omega_a       angular velocity of the earth (rad/s)
%   lambda        width degree(degree)
% OUTPUT:
%   v_mean:        mean wind speed ( in x-directions ) at height z (m/s)
%
% Thesis: Torsion motions of high-rise buildings due to wind loading.
% Author: H.A.O. Richardson (Anthony)



%---------------------------------------------------------------------------
% Inputing variables
%---------------------------------------------------------------------------


%L=144;                   % Height of the buidling
%u_star=2.82;             % Friction velocity
%kappa=0.4;               % Terrain Roughness
%d=3.5;                  % average height of buildings (m)
%d=0;                     % average height of buildings (m) ( Ivar Woudenberg)
%z_0=2;                   % Terrain Roughness according NEN 6702
%z_0=0.2;                 % Terrain Roughness according to Ivar Woudenberg.
%step_z=1.5;              %
lambda=51.75;            % Width degree of rotterdam: 51,75 degrees



%---------------------------------------------------------------------------
% Begin calcualtion
%---------------------------------------------------------------------------



z=(9:step_z:L);         % Grid on which the height is divided for 9 meters.
%z=(9:1.5:144);

%---------------------------------------------------------------------------
% Extra formulas Harris and Deaves
%---------------------------------------------------------------------------
```

```
omega_a=(2*pi)/(24*60*60);  % Cyclic frequency of the earth 7,2722e-5
(rad/s))

f_c=2*omega_a*sin(lambda);  %Coriolisparameter (s -1)

z_g=u_star/(6*f_c);         % Gradient height

a=((z-d)/z_g);              % dimensionless argument


%-------------------------------------------------------------------------
% Logarithmic function Mean velocity according to Harris and Deaves
%-------------------------------------------------------------------------



%v_mean=(u_star/kappa)*((log(z-d/z_0)+5.75*a)); until 200 meters
v_mean=u_star/kappa.*(log(z-d/z_0)+5.75.*a-1.88.*a.^2-
1.33.*a.^3+0.25.*a.^4);
% mean wind speed at reference height Harris and Deaves.

%-------------------------------------------------------------------------
% Plotting mean wind speed logarithmic function with boundaries
%-------------------------------------------------------------------------

figure(1)
plot(v_mean,z);

xlabel('v  [m/s^2]')
ylabel('z  [m]')



%-------------------------------------------------------------------------
% Plotting mean wind speed logarithmic function with boundaries
%-------------------------------------------------------------------------

% rearanging values in matrices z and v_mean

r=zeros(size(z,1),(size(z,2)+2));

for X=1:size(z,2);
    r(1,X+1)=z(1,X);
end

v_mean2(1,(size(z,1)+1):1:(size(z,2)+1))=u_star/kappa*(log(z-
d/z_0)+5.75.*a-1.88.*a.^2-1.33.*a.^3+0.25.*a.^4);

v_mean2(1,1)=v_mean2(1,2);
v_mean2(1,(size(z,2)+2))=v_mean2(1,(size(z,2)+1));


figure(2)
plot(v_mean2,r);

xlabel('v  [m/s^2]')
ylabel('z  [m]')
```

velocitymeaneurocode.m

```
function
[v_mean]=velocitymeaneurocode(L,u_ref_0,C_DIR,C_TEMP,C_ALT,z_o,step_z)
%syntax: function
%[v_mean]=velocitymeaneurocode1(L,u_ref,u_ref_0,C_DIR,C_TEMP,C_ALT,z_o,step
_z)
% Exponential powerfunction Eurocode 1 (ENV 1991-2-4)
%
%
% INPUT:          (Belasting en de hoogbouwdraag constructie page 86 )
%
%   L:          length of the building (m)
%   u_ref:      reference speed at height of 10 m  (m/s)
%   u_ref_0:    onalterd reference speed at height of 10 m  (m/s)
%   C_DIR:      direction factor (-)
%   C_TEMP:     season factor (-)
%   C_ALT:      height factor (-)
%   K_t:        terrain factor (-)
%   z:          height above the face of the earth (m)
%   z_o:        measure for the roughness of the terrain (roughness length)
%   step_z:     height of grid in which the building is devided in z
%               direction (m)
% OUTPUT:
%   vmean:      mean wind speed ( in x-directions ) at height z (m/s)
%
% Thesis: Torsion motions of high-rise buildings due to wind loading.
% Author: H.A.O. Richardson (Anthony)


%L=144;                   % Height of the buidling
%u_star=2.82;             % Friction velocity
%u_ref_0=27.5;            % onalterd reference speed at height of 10 m  (m/s)
%C_DIR=1;                 % direction factor (-)
%C_TEMP=1;                % season factor (-)
%C_ALT=1;                 % height factor (-)
%z_0=0.2;                 % measure for the roughness of the terrain (roughness
length)
%z_0_2=0.05;              % roughness length area 2
%step_z=1.5;              %height of grid in which the building is devided in z
direction (m)



%-----------------------------------------------------------------------------
% approximation of the wind speed profile with a log profile
%-----------------------------------------------------------------------------



z=(9:step_z:L);      % Grid on which the height is divided for 9 meters.


 k_T =0.19*(z_0/z_0_2)^0.0706;

 u_ref=C_DIR*C_TEMP*C_ALT*u_ref_0;

 v_mean=u_ref*k_T*log(z/z_0);
```

```
%-------------------------------------------------------------------------
% Plotting mean wind speed logarithmic function without boundaries
%-------------------------------------------------------------------------


figure(1)
plot(v_mean,z);

xlabel('Vmean [m/s^2]')
ylabel('L [m]')


%-------------------------------------------------------------------------
% Plotting mean wind speed logarithmic function with boundaries
%-------------------------------------------------------------------------

%rearanging values in matrices z and v_mean

r=zeros(size(z,1),(size(z,2)+2));

for X=1:size(z,2);
    r(1,X+1)=z(1,X);
end

v_mean2(1,(size(z,1)+1):1:(size(z,2)+1))=u_ref*k_T*log(z/z_0);

v_mean2(1,1)=v_mean2(1,2);
v_mean2(1,(size(z,2)+2))=v_mean2(1,(size(z,2)+1));


figure(2)
plot(v_mean2,r);


xlabel('Vmean [m/s^2]')
ylabel('L [m]')
```

Inputvalues.m

```
% Input values for simulation of a turbulent wind field
%
% INPUT:
%     yr, zr: specification of coordinates on the facade of the structure
%     v_10: mean wind velocity at 10 m above the surface of the earth (m/s)
%     sigma: standard deviation of the fluctuating part of the wind speed
(m/s)
%     N: number of time points (including zero); N must be a power of 2
%     deltat: time step (s)
%     fmax: maximum frequentie spectrum (Hz)
% OUTPUT:
%     UC: constrained turbulent wind velocities (m/s)
%
% Thesis: Torsion motions of high-rise buildings due to wind loading
% Author: H.A.O. Richardson (Anthony)


yr=1.3:2.6:24.7;
zr=9:1.5:144;

v_10=21.45;
U=v_10;

sigma=2.52;
sigma_v=sigma;

N=8192;
deltat=.1;
fmax=5;
timestep=0.1;


save matrices yr zr U sigma_v N deltat timestep fmax -append
```

specest.m

```
function [Pxx,freq] = specest(x,m,samplf,noverlap)

% Syntax : [Pxx,freq] = specest(x,m,samplf,noverlap)
%
% AutoPower Spectral Density function estimate of one data sequence.
% [Pxx,freq] = specest(x,m,samplf) performs FFT analysis of
% the sequence using the Welch method of power spectrum estimation.
% The x sequence of n points is divided into k sections of
% m points each (m must be a power of two). Using an m-point FFT,
% successive sections are windowed, FFT'd and accumulated. The choise
% of a certain window, can be made within the listing of this file,
% specest.m.
%
% The unit of the power spectrum Pxx is such, that integration of the
% values of Pxx over the frequencies yields the standard deviation.
%
% Input :
%   x : data sequence
%   m : length of window
%   samplf : sampling frequency (Hz) of signal x, equal to
%            1/(sampling time)
%   noverlap : m-point sections should overlap 'noverlap' points,
%              is optional, doesn't have to be specified
%
% Output :
%   Pxx : Autopower Spectral Density function estimate of x, only for
%         postive frequencies, compensation in the positive frequencies
%         for the negative frequencies
%   freq : corresponding frequency vector for Pxx, only positive
%          frequencies, upto Nyquist frequency (=sampling frequency/2)

%==========================================================================
% Author : A.H.J. Winnemuller
% Date : 04-20-1995
% Revised :
%==========================================================================

% Check if noverlap is specified, if not : make it zero
 if (nargin == 3),  noverlap = 0; end

% Make sure x is a column vector
 x = x(:);

% Number of data points
 n = max(size(x));

% Number of windows (k=fix(n/m) for noverlap=0)
 k = fix((n-noverlap)/(m-noverlap));

% Window specification; choose from :
%  bartlett, blackman, boxcar, chebwin, hamming, hanning,
%  kaiser or triang
% For a good representation of the stochastic process the
% signal belongs to, use : e.g. blackman (but not boxcar).
% If no window is to be applied, use boxcar.
```

```
index = 1:m;
w = boxcar(m);

% Normalizing scale factor KMU
W=fft(w);
WW=W.*conj(W);
KMU=sum(WW)*k*(1/m)*(samplf);

% Calculate power spectral density of signal x, where trend is removed
% from the sections, to prevent distortion of the spectrum
Pxx = zeros(m,1);
for i=1:k
   xw = w.*detrend(x(index));
   index = index + (m - noverlap);
   Xx = abs(fft(xw)).^2;
   Pxx = Pxx + Xx;
end

% Normalize the two-sided spectrum
Pxx=Pxx/KMU;

% Nyquist frequency occurs at point m/2+1 of the m-point section.
% Remove the spectral estimates corresponding to negative frequencies
% ans compensate for them in the positive frequencies
Pxx(m/2+2:m)=[];
Pxx(2:m/2)=2*Pxx(2:m/2);

% Creation of frequency vector, running from 0 Hz to the Nyquist
% frequency
freq=(0:(m/2))/(m/2)*samplf/2;
```

specest_test_run.m

```
% Checking  the Spectral density function
%
% Thesis: Torsion motions of high-rise buildings due to wind loading
% Author : H.A.o.Richardson (Anthony)

clear
close all

% Loading the generated wind velocities

load UC1

% Selecting a node and generating a time vector

u=UC(:,1,1);
t=(1:1:8192)*0.1;

% Plotting figure time versus the generated wind velocities

figure
plot(t,u);

% Calling the program specest.m (different ranges to choose from)

[Pxx,freq] = specest(u,1024,10);
[Pxx,freq] = specest(u,512,10);
[Pxx,freq] = specest(u,256,10);


% Plotting output specest

figure
plot(freq,Pxx);

figure
semilogy(freq,Pxx);

% Making a frequency time line and inputting the inputted values for
windspeed at hub height and standard deviation of wind speed in the
autopower spectrum

f=(0.001:0.001:5);
S=autopow(f,21.45,2.52);

% Plotting frequency versus autopower spectrum

figure
semilogy(f,S);

% Plotting frequency versus autopower spectrum and generated wind
velocities

figure
semilogy(f,S,freq(2:end),Pxx(2:end));
```

Appendix 3 The Juffertoren output of the 100 simulations for return period of one year

	max_a_ben	max_a_tor	max_a_ben_tor		sigma_v_a_ben	sigma_v_a_tor	sigma_v_a_ben_tor
1	0.07901997	0.3154349	0.347868574		0.031884807	0.097929211	0.102993374
2	0.09502604	0.2287908	0.225850924		0.028115223	0.067419462	0.072179545
3	0.08879392	0.2312267	0.2336136		0.026846394	0.077938953	0.083766051
4	0.10615966	0.2031683	0.270922504		0.0251388	0.066920912	0.07188217
5	0.0736211	0.2015533	0.247111679		0.026224231	0.080844871	0.0855873
6	0.08078583	0.2341519	0.268814781		0.026592566	0.070060124	0.075708769
7	0.08541438	0.249131	0.265200283		0.029473679	0.083269218	0.089176754
8	0.07210333	0.2579831	0.241273924		0.019752246	0.093835705	0.096105797
9	0.13054919	0.2398426	0.235520817		0.030073112	0.060388326	0.068064975
10	0.08138736	0.2269369	0.241830173		0.025732798	0.076559066	0.081261858
11	0.09919487	0.2680968	0.317176085		0.03039881	0.090701335	0.096010367
12	0.10400851	0.2415562	0.278614354		0.029040682	0.075157222	0.08106332
13	0.07506417	0.227017	0.254067133		0.027530949	0.088724806	0.093777656
14	0.09112941	0.3118245	0.357956698		0.02548089	0.095829132	0.099183317
15	0.08084645	0.2374471	0.254215733		0.02471953	0.067617723	0.071942393
16	0.08101202	0.2334264	0.224069929		0.024796893	0.06944557	0.073362182
17	0.07679948	0.2721974	0.275561386		0.027328724	0.088819358	0.094256668
18	0.09665245	0.250642	0.282281823		0.027810513	0.098101193	0.101829621
19	0.08839106	0.2431482	0.245513462		0.0278197	0.080150922	0.085250782
20	0.10854	0.2707078	0.30502623		0.031819829	0.110032915	0.114261271
21	0.07837437	0.2071293	0.266883724		0.022751603	0.062801089	0.065724809
22	0.09778053	0.2570626	0.243423482		0.02504102	0.082809933	0.086483211
23	0.09505646	0.2273086	0.227212401		0.029416008	0.069109222	0.076017137
24	0.08856752	0.204231	0.25276404		0.030100928	0.064249038	0.071356932
25	0.07834746	0.3206495	0.348140414		0.030072151	0.126312435	0.129295129
26	0.08914042	0.2502305	0.292177947		0.030658104	0.070394482	0.078406003
27	0.08900862	0.2059822	0.231338263		0.025037439	0.066827051	0.071495308
28	0.09113444	0.2773968	0.300049901		0.023573273	0.102236978	0.103935674
29	0.09694114	0.194166	0.262098859		0.032457306	0.056953225	0.064016696
30	0.14682591	0.2146753	0.318700152		0.03361227	0.076898999	0.083250098
31	0.09421751	0.2712761	0.272828404		0.031457569	0.077414355	0.082692316
32	0.07695353	0.3351562	0.353238403		0.027332071	0.112661957	0.116399174
33	0.08367757	0.3312502	0.290940187		0.029931995	0.115752231	0.119598381
34	0.09421759	0.2728489	0.225796349		0.031673278	0.07713853	0.085205891
35	0.08154371	0.2314921	0.258869055		0.031584988	0.084135747	0.090691919
36	0.1155729	0.2667198	0.325697711		0.031441818	0.074342546	0.080823537
37	0.0804843	0.196095	0.215927765		0.026585369	0.078399722	0.082615485
38	0.06993247	0.2227549	0.214069599		0.025415821	0.065959439	0.071919526
39	0.12859407	0.2479667	0.321107747		0.033112213	0.089871517	0.095594086
40	0.10089678	0.2555727	0.305778185		0.03020859	0.08070116	0.085343493
41	0.07104905	0.209015	0.222081497		0.022916168	0.066224983	0.070737656
42	0.09715038	0.1932312	0.202770808		0.03311036	0.073444956	0.081073318
43	0.09401802	0.2428575	0.290574477		0.025615311	0.081211157	0.085516732
44	0.07130808	0.2742605	0.306073536		0.029068044	0.087484177	0.091663874
45	0.10649363	0.3009411	0.336744989		0.027140327	0.101589876	0.105994995
46	0.08179491	0.2400594	0.253401477		0.027596213	0.068729826	0.073191005
47	0.08239944	0.2618932	0.255315624		0.026949153	0.09156893	0.095856063
48	0.09343369	0.2692511	0.307915902		0.036771507	0.095312997	0.101861375
49	0.09845596	0.2728585	0.29025337		0.028753191	0.088335183	0.093319901
50	0.08412185	0.2703046	0.290280601		0.023639648	0.10862627	0.110818164

50	0.08412185	0.2703046	0.290280601		0.023639648	0.10862627	0.110818164
51	0.07757371	0.2408457	0.249179007		0.027674879	0.076548667	0.081859227
52	0.08524064	0.2275992	0.26608179		0.022325849	0.091220343	0.093831935
53	0.09738446	0.268816	0.280123701		0.032560162	0.093529436	0.100062341
54	0.08704866	0.2287985	0.260372811		0.031450677	0.093413447	0.097739543
55	0.08698624	0.27792	0.290317349		0.030526951	0.073664855	0.080888252
56	0.07884446	0.313843	0.327210731		0.024894461	0.102307881	0.105978544
57	0.08623691	0.2995805	0.294183097		0.023215837	0.102809614	0.106135784
58	0.06947999	0.2719755	0.25935968		0.026427083	0.101268979	0.105217104
59	0.09358635	0.1706568	0.212125945		0.029660752	0.05983622	0.067299691
60	0.09212752	0.2591273	0.274868804		0.027422612	0.087991725	0.092057406
61	0.08199818	0.3158783	0.339245102		0.027068803	0.128905333	0.132189224
62	0.09527037	0.2713995	0.286476756		0.033568575	0.083852398	0.089943952
63	0.09812317	0.3473477	0.351292021		0.02754538	0.10948055	0.112792408
64	0.07391004	0.2407578	0.232606726		0.022893411	0.07538152	0.078314601
65	0.07119504	0.2263872	0.273643925		0.023026724	0.087625033	0.091573059
66	0.09013005	0.292834	0.318489913		0.026868967	0.097863106	0.10241843
67	0.09310774	0.2727566	0.305832959		0.032845032	0.102269849	0.107687308
68	0.09912049	0.2612763	0.269895485		0.029103781	0.080287329	0.084555794
69	0.09760287	0.2587812	0.286304882		0.033875528	0.079054052	0.083976231
70	0.07419547	0.2990337	0.306723114		0.023075607	0.098203667	0.100454213
71	0.07624585	0.2312022	0.26083605		0.023797771	0.081636732	0.085785605
72	0.06705839	0.2530859	0.254394652		0.024483663	0.071932967	0.076072072
73	0.08852208	0.2745583	0.27554601		0.02764329	0.071856698	0.075769768
74	0.08760393	0.2730636	0.295432966		0.032179728	0.085560979	0.091520915
75	0.08339804	0.2857885	0.291488836		0.02930333	0.084931794	0.089144172
76	0.06792917	0.1979205	0.217703377		0.022535256	0.078949515	0.082945319
77	0.08407408	0.2994298	0.307316015		0.033765918	0.098522841	0.103670046
78	0.10738459	0.2443405	0.267945388		0.031696823	0.08347872	0.088317221
79	0.06846484	0.2231908	0.238895539		0.023611531	0.076111416	0.080320097
80	0.11170106	0.3074799	0.331554775		0.034383715	0.09505573	0.100782984
81	0.10842811	0.1971032	0.234713847		0.036041491	0.071335391	0.081142271
82	0.07893048	0.2726799	0.280179963		0.026414594	0.088925232	0.094385945
83	0.13492696	0.2426756	0.296094743		0.037171804	0.080594688	0.088603472
84	0.07787353	0.2597489	0.27426618		0.02557191	0.100593297	0.103853019
85	0.09812013	0.2659874	0.303037628		0.024636878	0.08141364	0.084322415
86	0.09323893	0.29038	0.308445665		0.031062577	0.088509935	0.09509343
87	0.07698314	0.2501046	0.276608006		0.025723913	0.080182305	0.083584801
88	0.07885024	0.2708426	0.328016825		0.026798197	0.080291469	0.085373717
89	0.07875916	0.3378433	0.342113176		0.022621681	0.125466468	0.127974392
90	0.09667609	0.3275849	0.367087398		0.026730192	0.097159731	0.100739464
91	0.07428866	0.1918223	0.218785754		0.028769533	0.061062749	0.065828115
92	0.0999251	0.3441147	0.351481911		0.028265931	0.105735321	0.109267861
93	0.08467466	0.3473279	0.366065206		0.024819995	0.097631654	0.101408988
94	0.07615857	0.2139997	0.246478605		0.025440583	0.073022348	0.076841441
95	0.06722977	0.266699	0.260068328		0.022860946	0.082241712	0.085264018
96	0.09471137	0.2079731	0.267910045		0.032242381	0.073597462	0.079429955
97	0.11237251	0.3008401	0.322325274		0.042248094	0.100937033	0.109828299
98	0.07993901	0.3120294	0.307897054		0.027270763	0.097586129	0.101943831
99	0.09029062	0.2539699	0.290262274		0.031911913	0.091324537	0.096527497
100	0.07821876	0.2886099	0.304810918		0.026384818	0.083215684	0.088117416

Appendix 4 Formulas to determine the maximum acceptable acceleration for Juffertoren

Eurocode

The wind force acting on a structure or structural component is: [10]

$$F_w = c_s c_d * c_f * q_p(z_e) * A_{ref}$$

$$F_w = 0.93 * 1.44 * q_p(z_e) * A_{ref}$$

With:

$c_s c_d$:	structural factor
c_f :	force coefficient
$q_p(z_e)$:	peak velocity pressure at reference height z_e
A_{ref} :	reference area of the structure

Force coefficient. [10] p.67

$$c_f = 1.44$$

Structural factor [10]

$$c_s c_d = \frac{1 + 2 * k_p * I_v(z_e) * \sqrt{B^2 + R^2}}{1 + 7 * I_v(z_e)}$$

$$c_s c_d = \frac{\left(1 + 2 * 3.17 * 0.16 * \sqrt{0.45 + 0.487}\right)}{(1 + 7 * 0.16)} = 0.93$$

With:

z_e :	reference height or height of structure.
k_p :	peak factor.
I_v :	turbulence intensity

B^2 :	background factor.
R^2 :	resonance response factor.

Background factor [10] **Eurocode procedure 2**

$$B^2 = \frac{1}{1+\frac{3}{2}\sqrt{\left(\frac{b}{L(z_e)}\right)^2+\left(\frac{h}{L(z_e)}\right)^2+\left(\frac{b}{L(z_e)}*\frac{h}{L(z_e)}\right)^2}}$$

$$B^2 = \frac{1}{1+\frac{3}{2}\sqrt{\left(\frac{26.3}{182.9}\right)^2+\left(\frac{144}{182.9}\right)^2+\left(\frac{26.3}{182.9}*\frac{144}{182.9}\right)^2}} = 0.45$$

Wind Turbulence [10]

$$L(z_e) = L_t\left(\frac{z_e}{z_t}\right)^{\alpha} = 300\left(\frac{86.4}{200}\right)^{0.59} = 182.9$$

$$\alpha = 0.67 + 0.05 * \ln(z_0) = 0.67 + 0.05 * \ln(0.2) = 0.59 \quad [13]\ \text{pp.86-87}$$

With:

b, h :	width and height of structure
$L(z_e)$:	turbulence length scale. It is on the safe side to use $B^2 = 1$

$$z_e = 0.6 * h \geq z_{\min} = 0.6 * 144 = 86.4 \quad [10]$$

$$k_p = \sqrt{2 * \ln(v * T)} + \frac{0.6}{\sqrt{2 * \ln(v * T)}}$$

$$k_p = \sqrt{2 * \ln(0.13 * 600)} + \frac{0.6}{\sqrt{2 * \ln(0.13 * 600)}} = 3.17$$

Resonance response factor ([10] p.110])

$$R^2 = \frac{\pi^2}{2*\delta} * S_L\left(z_e, n_{1,x}\right) * K_s\left(n_{1,x}\right)$$

$$R^2 = \frac{\pi^2}{2*\delta} * S_L\left(z_e, n_{1,x}\right) * K_s\left(n_{1,x}\right)$$

$$R^2 = \frac{\pi^2}{2*0.48} * 0.143 * 0.3319 = 0.487$$

With:

δ	The total logarithmic decrement of damping	[10]
S_L	wind power spectral density function given	B. 1 (2)
$n_{1,x}$	natural frequency of the structure	[10]
K_s	size reduction function	[10]

The total logarithmic decrement of damping

$$\delta = \delta_s + \delta_a + \delta_d = 0.10 + 0.38 + 0 = 0.48$$

The total logarithmic decrement of structural damping

$$\delta_s = 0.10$$

The total logarithmic decrement of aerodynamic damping

$$\delta_a = 0.38$$

The total logarithmic decrement of damping due to special devices

$$\delta_d = 0$$

wind power spectral density function given [10]

$$S_L\left(z_e,n_{1,x}\right)=\frac{n*S_v\left(z_e,n\right)}{\sigma_v^2}=\frac{6.8*f_L\left(z_e,n\right)}{\left(1+10.2*f_L\left(z_e,n\right)\right)^{5/3}}=\frac{6.8*0.716}{\left(1+10.2*0.716\right)^{5/3}}=0.143$$

$$f_L\left(z_e,n\right)=\frac{n*L\left(z_e\right)}{v_m\left(z_e\right)}=f_L\left(86.4,0.13\right)=\frac{0.13*182.9}{34.3}=0.716$$

$$L\left(z_e\right)=L_t\left(\frac{z_e}{z_t}\right)^{\alpha}=300\left(\frac{86.4}{200}\right)^{0.59}=182.9$$

$$\alpha=0.67+0.05*\ln\left(z_0\right)=0.67+0.05*\ln\left(0.2\right)=0.59$$ [13] pp.86-87

natural frequency of the structure [10]

$$\delta=\frac{ql^4}{8EI}=\frac{2.68E^6*144^4}{8*3.00E^{10}*688}=6.98\text{ m}$$

$$n=n_{1,x}==\frac{1}{2\pi}\sqrt{\left(\frac{g}{x_1}\right)}=\frac{1}{2\pi}\sqrt{\left(\frac{9.81}{6.98}\right)}=0.13\text{ Hz}$$ [10] p.185

size reduction function [10]

$$K_s\left(n\right)=\frac{1}{1+\sqrt{\left(G_y*\phi_y\right)^2+\left(G_z*\phi_z\right)^2+\left(\frac{2}{\pi}*G_y*\phi_y*G_z*\phi_z\right)^2}}$$

$$K_s\left(n\right)=\frac{1}{1+\sqrt{\left(0.5*1.18\right)^2+\left(0.278*6.48\right)^2+\left(\frac{2}{\pi}*0.5*1.18*0.278*6.48\right)^2}}=0.3319$$

$$\phi_y = \frac{c_y * b * n}{v_m(z_e)} = \frac{11.5 * 26.3 * 0.13}{34.3} = 1.18$$

$$\phi_z = \frac{c_y * h * n}{v_m(z_e)} = \frac{11.5 * 144 * 0.13}{34.3} = 6.48$$

Decay constants

$$c_y = c_z = 11.5$$

The peak velocity pressure is calculated using $q_p(z) = 1 + 7 * I_v(z) * \frac{1}{2} * \rho * v_m^2(z)$

With:

$I_v(z)$	turbulence intensity
ρ	air density
$v_m(z)$	mean wind velocity

Turbulence intensity ([10] p.21])

$$I_v(z) = \frac{\sigma_v}{v_m(z)} = \frac{k_I}{c_0(z) * \ln(z/z_0)} = \frac{1}{1 * \ln(144/0.2)} = 0.15$$ [13] pp. 86-87

$$I_v(z_e) = \frac{\sigma_v}{v_m(z_e)} = \frac{k_I}{c_0(z_e) * \ln(z_e/z_0)} = \frac{1}{1 * \ln(86.4/0.2)} = 0.16$$

Terrain roughness [10] p.21

$$k_r(z) = 0.19 * \ln\left(\frac{z_0}{z_{0,II}}\right)^{0.07}$$ (Terrain category III)

$$k_r(z) = 0.19 * \ln\left(\frac{0.2}{0.05}\right)^{0.07} = 0.21 \text{ [-]}$$

$$c_r(z) = k_r(z) * \ln\left(\frac{z}{z_0}\right)$$

$$c_r(z) = 0.21 * \ln\left(\frac{144}{0.2}\right) = 1.38 \text{ [-]}$$ (Terrain category III)

$$c_r(z_e) = 0.21 * \ln\left(\frac{86.4}{0.2}\right) = 1.27 \text{ [-]}$$

Mean wind velocity [10] p.19

$$v_m(z) = c_r(z) * c_0(z) * v_b = 1.38 * 1 * 27 = 37.2 \text{ m/s}^2$$

$$v_m(z_e) = c_r(z_e) * c_0(z_e) * v_b = 1.27 * 1 * 27 = 34.3 \text{ m/s}^2$$

With:

$c_r(z)$	roughness factor
$c_0(z)$	orography factor
v_b	basic wind velocity

Basic wind velocity [10] p.18

$$v_b(z) = c_{dir} * c_{season} * v_{b,0} = 1 * 1 * 27 = 27 \text{ m/s}^2$$

With:

c_{dir}	directional factor
c_{season}	season factor
$v_{b,0}$	fundamental value of basic wind velocity

For the Netherlands, the euro code states that the fundamental value of basic wind velocity is equal to 27 m/s.

Acceleration for serviceability assessments [10] pp.111-112

$$\sigma_{a,x}(y,z) = c_f * \rho * I_v(z_e) * v_m^2(z_e) * R * \frac{K_y * K_z * \Phi(y,z)}{\mu_{ref} * \Phi_{max}}$$

$$\sigma_{a,x}(y,z) = 1.44 * 1.29 * 0.16 * (34.3)^2 * 0.698 * \frac{1 * 5/3 * 1}{675 * 1} = 0.621 \text{ m/s}^2$$

The standard deviation $\sigma_{a,x}$ of the characteristic along-wind acceleration of the structural point with coordinates (*y*,*z*) is approximately given by Expression [10].

where:

c_f	force coefficient
ρ	air density
$I_v(z_e)$	turbulence intensity at height $z = z_e$ above ground
$v_m(z_e)$	characteristic mean wind velocity at height z_e
R	square root of the resonant response
K_y, K_z	constants given in C.2 (6)
μ_{ref}	the reference mass per unit area
$\Phi(y,z)$	the mode shape
Φ_{max}	mode shape value at the point with maximum amplitude

The characteristic peak accelerations are obtained by multiplying the standard deviation in by the peak factor in B. 2 (3) using the natural frequency as upcrossing frequency, i.e. $v = n_{1,x}$.

$$a_{max} = k_p \sigma_{a,x}(y,z) = 3.17 * 0.621 = 1.965 \text{ m/s}^2$$

This value is totally unrealistic. The found value for only bending is about 17 times bigger than the found yearly annual maxima for NEN of $0.113 m/s^2$.

Woudenberg empirical formula's

The formula to calculate the frequency in this article is [19]

$$f_{e\_Woudenburg} = \frac{46}{h}$$

In which:

height building: 144 m

Substituting the variables into the frequency formula gives:

$$f_{e\_Woudenburg} = \frac{46}{144} \approx 0.319 \text{ Hz}$$

The cyclic frequency: $\omega_{e\_Woudenburg} = 2 * \pi * f_e \approx 2.01\text{Hz}$.

The maximum acceptable deflection of the top of the building is $u_{top} = H / 500$. When the height is substituted we get a deflection of $144 / 500 = 0.288m$. The amplitude A of our building is the half of the maximum acceptable deflection of the top of the building, this is $0.288 / 2 = 0.144m$.

The formula for the natural vibration can be written as $u(t) = A\sin(\omega t + \varphi)$. The acceleration is the second derivative of the natural vibration formula $a(t) = \ddot{u}(t) = A\omega^2 \sin(\omega t + \varphi)$. Out of which the maximum bending acceleration is $a = A * \omega^2$. Substituting the variables into the formula gives $a = 0.144 * 2.01^2 = 0.580 \text{ m/s}^2$.

This maximum occurring acceleration is **far above** the maximum acceptable acceleration of $a = 0.1 m/s^2$.

Determining the torsional acceleration with the empirical formula in Woudenberg article [19]

$$f_{e\_Woudenburg} = \frac{72}{h}$$

In which:

h = height building: 144 m

Substituting the variables into the frequency formula gives:

$$f_{e\_Woudenburg} = \frac{72}{144} = 0.5 \text{ Hz}$$

The angular cyclic frequency is $\omega_{e\_Woudenburg} = 2 * \pi * f_e \approx 3.14\text{Hz}$.

The maximum acceptable deflection of one floor of the building is $u_{top} = H / 500$. When the height is substituted we get a deflection of $144/500 = 0.288$ m. The maximum acceptable bending deflection per story is $0.288/48 = 6E^{-3}$ m. To determine the torsion on a floor we have to look at the interstory drift of a building. The maximum acceptable displacement due to interstory drift is $u_{story} = h_{story} / 300$. Substitution gives $u_{story} = 3/300 = 1E^{-2} m$. The displacement left for torsion motion is $0.01 - 0.006 = 0.004$ m.

We assume in our model that the bending and torsional motions are uncoupled. The maximum amplitude due to torsional motion is 0.004 m. This maximum amplitude for torsion should not be disregarded for the maximum deflection of the total building.

$$u_{floor} = \frac{h_{story}}{300} - \frac{h}{500 * n_{floors}} = \frac{3}{300} - \frac{144}{500 * 48} = 0.01 - 0.006 = 0.004$$

The formula for the natural vibration can be written as $u(t) = A\sin(\omega t + \varphi)$. The acceleration is the second derivative of the natural vibration formula $a(t) = \ddot{u}(t) = A\omega^2 \sin(\omega t + \varphi)$. Substituting the variables into the formula gives $a = 0.004 * 3.14^2 = 0.039\,\text{m/s}^2$ which is the maximum torsional acceleration.

This maximum occurring acceleration due to bending and torsion motion is $a = 0.580 + 0.039 = 0.619 \text{ m/s}^2$, far above the maximum acceptable acceleration of $a = 0.1 m/s^2$.

Woudenberg formulas

The formulas to calculate the frequency in this article "Windbelasting en het hoogbouwontwerp" [19]

Formulas

Bending displacement at top of building due to wind load

$$\partial_{EI} = \left(\frac{q_w * h^4}{8EI}\right) = \left(\frac{16160 * 144^4}{8 * 2.003\text{E}^{13}}\right) = 4.33\text{E}^{-2} \text{ m}$$

Shear displacement at top of building due to wind load

$$\partial_{GA} = \left(\frac{q_w * h^2}{2GA}\right) = \left(\frac{16160 * 144^2}{2 * 1.25\text{E}^{10} * 40.14}\right) = 3.34\text{E}^{-4} \text{ m}$$

displacement at top of building due to foundation rotation

$$C_f = 20 * EI / L = 2.78E^{12} \text{ Nm} \qquad [5]$$

$$\partial_{C_f} = \left(\frac{h^3}{2C_f}\right) = \left(\frac{144^3}{2 * 2.78E^{12}}\right) = 8.67\text{E}^{-3} \text{ m}$$

Total displacement at top of the building

$$\partial_{Top} = \partial_{EI} + \partial_{GA} + \partial_{C_f} = 5.23\text{E}^{-2} \text{ m}$$

Critical buckling force

$$Q_k = \left(\frac{q_w * h^2}{\partial_{Top}}\right) = \left(\frac{16160 * 144^2}{5.23\text{E}^{-2}}\right) = 6401798934 \text{ N}$$

Second order effect

$$\text{second order} = \frac{n}{n-1} = \frac{16.56}{16.56-1} = 1.06$$

$$n = \frac{Q_k}{Q_{optr}} = \frac{1}{q_m} \frac{1}{\left(\frac{h^3}{8EI} + \frac{h}{2GA} + \frac{h^2}{2C_f}\right)} = \frac{1}{2.68E^6} \frac{1}{\left(\frac{144^3}{8*2.003E^{13}} + \frac{144}{2*1.25E^{10}*40.14} + \frac{144^2}{2*2.78E^{12}}\right)} = 16.56$$

$m_{buidling} = 39399240$ kg (Mathlab) $\quad q_m = \frac{m_{buidling} * g}{h} = \frac{39399240*9.81}{144} = 2.68E^6$ N/m

Total displacement due to wind load and foundation rotation (including second order effect)

$$\partial_{Top;sec} = \frac{n}{n-1}\partial_{Top} = 1.06 * 5.23E^{-2} = 5.54E^{-2} \text{ m}$$

Amplitude

$$A = 0.0554/2 = 2.77E^{-2} \text{ m}$$

Natural frequency (Section 5.4.2)

$$\omega_n = C\sqrt{\left(EI / \rho A l^4\right)} = 3.52\sqrt{\left(\left(3.0E^{10}*667\right)/\left(675.5366*405.02*144^4\right)\right)} = 1.451 \text{ rad/s}$$

$$\omega_{e\_Woudenberg} = \omega_n \approx 1.451 \text{ rad/s}$$

The formula for the natural vibration can be written as $u(t) = A\sin(\omega t + \varphi)$. The acceleration is the second derivative of the natural vibration formula. $a(t) = \ddot{u}(t) = A\omega^2 \sin(\omega t + \varphi)$, Of which the maximum acceleration is $a = A*\omega^2$. Substituting the variables into the formula gives $a = 0.0277*1.451^2 = 0.059 \text{ m/s}^2$.

This maximum occurring acceleration is **below** the maximum acceptable acceleration of $a = 0.1 m/s^2$.

Schueller

The eigenfrequency of the structure must be known to determine the acceleration what occurs. [18] .

The formula to calculate the vibration time is:

$$T = 2 * \pi * \sqrt{((Q * H \wedge 3) / (8 * EI * g))}$$

In which:

H =	the building height:	144	m	
Q =	the building weight:	386.5	MN	(Mathlab)
g =	the gravitational acceleration:	9.81	m/s^2	
E =	Young's modulus:	3.00E10	N/m^2	
I =	the moment of inertia:	688	m^4	

Substituting the variables into the formula gives:

$$T = 2 * \pi * \sqrt{((Q * H \wedge 3) / (8 * EI * g))} = 5.38 \text{ s}$$

The eigenfrequency $f = 1/T = 1/5.38 \approx 0.186 \text{ Hz}$. The cyclic frequency is $\omega = 2 * \pi * f = 1.167 \text{ rad}$.

The maximum acceptable deflection of the top of the building is u_{top}=H/500. When the height is substituted we get a deflection of 144/500 = 0.288 m. The amplitude A of our building is the half of the maximum acceptable deflection of the top of the building, this is 0.288/2 = 0.144 m.

The formula for the natural vibration can be written as $u(t) = A\sin(\omega t + \varphi)$. The acceleration is the second derivative of the natural vibration formula $a(t) = \ddot{u}(t) = A\omega^2 \sin(\omega t + \varphi)$, of which the maximum acceleration is $a = A\omega^2$. Substituting the variables into the formula gives
$a = 0.144*1.167^2 = 0.196$ m/s .

This maximum occurring acceleration is **far above** the maximum acceptable acceleration of
$a = 0.1 m/s$.

Dicke/Nijsse

The formula to calculate the frequency in this article is [20] :

The values of c by different types of fixations

c=1	spring
c=0.83	Cantilever
c=0.67	**on piles**
c=0.64	on rock ground

The variables of the building loaded by wind

L	Length of building	[m]
m	Mass of the building	[kg]
I	Second Moment of Area	[m^4]

The gravitational effect of the upside-down pendulum

K_e = Effective stiffness of the system	[N/m]
K_b = Spring stiffness	[N/m]
K_p = Negative stiffness of the upside-down pendulum	[N/m]

Mathematic Pendulum

$T=2*\pi*\sqrt{\frac{l}{g}}$ (Independent of Mass) [S]

$T=\frac{2*\pi}{\omega}$ (From vibration time to cyclic frequency) [S]

summation of $\omega^2=\frac{g}{l}$

$\omega^2 = \sqrt{\frac{k}{m}}$ (Single degree of freedom system) $\omega^2=\frac{K_p}{m}$

Combination of $\omega^2 = \frac{K_p}{m}$ and $\omega^2 = \frac{g}{l}$ gives us $K_p = \frac{mg}{l}$

with $G = m * g$

Upside-down pendulum

l_p	pendulum length
c	values of cantilevers.
l	building length

$l_p = c * l$

Calculations

Weight building

$$G = m * g = 39399240 * 9.81 = 396506544 \text{ N}$$

Pendulum length

$$l_p = 0.67 * 144 = 96.48 \text{ m}$$

Negative stiffness of the upside-down pendulum

$$K_p = \frac{m*g}{l_p} = \frac{39399240*9.81}{96.48} = 40060800 \text{ N/m}$$

Spring stiffness

$$K_b = \frac{3EI}{l^3} = \frac{3*2.003E^{13}}{144^3} = 20134066 \text{ N/m}$$

Effective stiffness of the system

$$K_e = K_b - K_p = 16127987 \text{ N/m}$$

$$\omega_p = \sqrt{\frac{K_p}{m}} = \sqrt{\frac{40060800}{39399240}} = 0.319 \text{ rad/s}$$

$$\omega_b = \sqrt{\frac{K_b}{m}} = \sqrt{\frac{20134066}{39399240}} = 0.715 \text{ rad/s}$$

$$\omega_e = \sqrt{\frac{K_e}{m}} = \sqrt{\frac{16127987}{39399240}} = 0.640 \text{ rad/s}$$

Frequency of Professor Dicke

$$f = \frac{\omega_e}{2\pi} = \frac{0.640}{2\pi} = 0.102 \text{ Hz}$$

Vibration time of Professor Dicke

$$T = \frac{1}{f_e} = \frac{1}{0.102} = 9.821 \text{sec}$$

Critical buckling factor

$$n = \frac{\omega_b^2}{\omega_p^2} = \frac{0.715^2}{0.640^2} = 5.026$$

Second order effect

$$n_{sec} = \frac{n}{n\text{-}1} = \frac{5.026}{4.026} = 1.25$$

Amplitude

$$A = \frac{h}{500}\Big/2 = 0.288/2 = 0.144 \text{ m}$$

Maximum acceleration

The maximum acceleration is $a = A * \omega^2 * n_{\sec}$. Substituting the variables into the formula gives $a = 0.144 * 0.640^2 * 1.25 = 0.074\ \text{m/s}^2$

This maximum occurring acceleration is **below** the maximum acceptable acceleration of $a = 0.1 m/s^2$.

NBCC Acrosswind acceleration

The acrosswind acceleration will be larger than the alongwind acceleration if $(bd)^{0.5} < 0.33$. [17]. This is the case in most buildings with a rectangular cross section, so the acrosswind acceleration will be dominant.

The National Building Code of Canada (NBCC) will be used to determine the across wind acceleration of the building [17]

$$a_y = \frac{f_e g_p a_r \sqrt{bd}}{\rho_b g \sqrt{\beta_w}}$$

With:

f_e	frequency of the building	Hz
g_p	peak factor	[-]
b_m	width of the building	m
ρ	average density of the building	kg/m3
g	gravitational acceleration	m/s2
β_w	lift damping ratio	[-]
b	width of the building	m
d	depth of the building	m

a_r is calculated by the following equation:

$$a_r = 0.0785\left(\frac{v_h}{f_e\sqrt{b_m * b_m}}\right)^{3.3}$$

With:

f_e	frequency of the building	Hz
b_m	width of the building	m
v_h	mean wind speed at top of the building	m/s2

$$a_y = \frac{f_e g_p a_r \sqrt{bd}}{\rho_b g \sqrt{\beta_w}} = \frac{0.201 * 3.5 * 47.40 * \sqrt{26.3 * 16.3}}{2.8E^5 * 9.81 * \sqrt{0.01}} = 0.179 \text{ m/s}^2$$

a_r is calculated by the following equation:

$$a_r = 0.0785 \left(\frac{v_h}{f_e \sqrt{b_m * b_m}} \right)^{3.3} = 0.0785 \left(\frac{36.8}{0.201 \sqrt{26.3 * 26.3}} \right)^{3.3} = 47.40 \, [-]$$

Eurocode acrosswind acceleration

E.1.2 Criteria for vortex shedding [10]

(1) The effect of vortex shedding should be investigated when the ratio of the largest to the smallest crosswind dimension of the structure, both taken in the plane perpendicular to the wind, exceed 6.

(2) The effect of vortex shedding need not be investigated if

$$v_{crit,i} > 1.25 * v_m \qquad \text{(E.1)}$$

$$25.84 > 1.25 * 37.19$$

where:

$v_{crit,i}$ is the critical wind velocity for mode I, as defined in E.1.3.1

v_m is the characteristic 10 minutes mean wind velocity specified in 4.3.1 (1) at the cross section where vortex shedding occurs. (Figure E.3)

E.1.3 Basic parameters for vortex shedding

E.1.3.1 Critical wind velocity $v_{crit,i}$

The critical wind velocity for bending vibration mode *i* is defined as the wind velocity at which the frequency of vortex shedding equals a natural frequency of the structure or a structural element and is given in Expression (E.2).

$$v_{crit,i} = \frac{b * n_{i,y}}{St} \qquad \text{(E.2)}$$

$$v_{crit,i} = \frac{15.4 * 0.13}{0.08} = 25.84 \text{ m/s}^2$$

where:

b is the reference width of the cross-section at which resonant vortex shedding occurs and where the modal deflection is maximum for the structure or structural part considered; for circular cylinders the reference width is the outer diameter.

$n_{i,y}$ is the natural frequency of the considered flexural mode *i* of cross-wind vibration; approximations for $n_{1,y}$ are given in F.2

St Strouhal number as defined in E.1.3.2.

.

E.1.3.2 Strouhal number *St*

The Strouhal number St for different cross-sections may be taken from Table E.1

$$d/b = 26.34/15.44 = 1.71 \text{ [-]}$$

$$St = 0.08 \text{ [-]}$$

E.1.3.3 Scruton number *Sc* ([10] p.119)

The susceptibility of vibrations depends on the structural damping and the ratio of structural mass to fluid mass.

This is expressed by the Scruton number *Sc*, which is given in Expression (E.4).

$$Sc = \frac{2 * \delta_s * m_{i,e}}{\rho * b^2} \quad \text{(E.4)}$$

$$Sc = \frac{2 * 0.1 * 2.11E^5}{1.29 * 15.4^2} = 137.8$$

where:

δ_s is the structural damping expressed by the logarithmic decrement.

ρ is the air density under vortex shedding conditions.

$m_{i,e}$ is the equivalent mass m_e per unit length for mode *i* as defined in F.4 (1)

b is the reference width of the cross-section at which resonant vortex shedding occurs

E.1.5 Calculation of the cross wind amplitude

E.1.5.1 General

(1) Two different approaches for calculating the vortex excited cross-wind amplitudes are given in E.1.5.2 and E.1.5.3.

(2) The approach given in E.1.5.2 can be used for various kind of structures and mode shapes. It includes turbulence and roughness effects and it may be used for normal climatic conditions.

(3) The approach given in E.1.5.3 may be used to calculate the response for vibrations in the first mode of cantilevered structures with a regular distribution of cross wind dimensions along the main axis of the structure.

Typically structures covered are chimneys or masts. It cannot be applied for grouped or in-line arrangements and for coupled cylinders. This approach allows for the consideration of different turbulence intensities, which may differ due to meteorological conditions. For regions where it is likely that it may become very cold and stratified flow conditions may occur (e.g. in coastal areas in Northern Europe), approach E.1.5.3 may be used.

E.1.5.2 Approach 1 for the calculation of the cross wind amplitudes

E.1.5.2.1 Calculation of displacements [10]

The largest displacement $y_{F,max}$ can be calculated using Expression (E.7).

$$\frac{y_{F,\max}}{b} = \frac{1}{St^2} * \frac{1}{Sc} * K * K_w * c_{lat} \qquad \text{(E.7)}$$

$$y_{F,\max} = \frac{1}{0.08^2} * \frac{1}{137.8} * 0.13 * 0.60 * 1.1 * 15.4 = 1.499 \text{ m}$$

where:

St is the Strouhal number given in Table E.1

Sc is the Scruton number given in E.1.3.3

K_w is the effective correlation length factor given in E.1.5.2.4

K is the mode shape factor given in E.1.5.2.5

c_{lat} is the lateral force coefficient given in Table E.2

E.1.5.2.2 Lateral force coefficient c_{lat}

The basic value, $c_{lat,0}$, of the lateral force coefficient is given in Table E.2.

$$c_{lat,0} = 1.1$$

E.1.5.2.3 Correlation length L [10] pp 124-126

The correlation length L_j, should be positioned in the range of antinodes. Examples are given in Figure E.3. For guyed masts and continuous multispan bridges special advice is necessary.

$$K_w = 3\frac{L_j/b}{\lambda} * \left[1 - \frac{L_j/b}{\lambda} + \frac{1}{2} * \left(\frac{L_j/b}{\lambda}\right)^2\right]$$

$$K_w = 0.6 \text{ [-]}$$

E.1.5.2.5 Mode shape factor [10]

The mode shape factor K is given in Expression (E.9).

$$K = \frac{\sum_{j=1}^{m} \left| \int_{l_j} \Phi_{i,y}(s) \right| ds}{4 * \pi * \sum_{j=1}^{m} \int_{l_j} \Phi_{i,y}^2(s) ds}$$

$$K = 0.13 \qquad \text{(E.9)}$$

where:

m is defined in E.1.5.2.4 (1)

$\Phi_{i,y}(s)$ is the cross-wind mode shape i (see F.3)

l_j is the length of the structure between two nodes (see Figure E.3)

The maximum acceleration is:

$$a_{\max} = y_{F,\max} * \left(2*\pi*f_e\right)^2 = 1.499*\left(2*\pi*0.13\right)^2 = 1.066 \text{ m/s}^2 .$$

Approach 2, for the calculation of the cross wind amplitudes

The characteristic maximum displacement at the point with the largest movement is given in Expression (E.13).

$$y_{max} = \sigma_y * k_p \qquad \text{(E.13)}$$

$$0.746 = 0.528 * 1.415$$

where:

σ_y is the standard deviation of the displacement

k_p is the peak factor

The standard deviation σ_y of the displacement related to the width *b* at the point with the largest deflection (Φ =1) can be calculated by using Expression (E.14).

$$\frac{\sigma_y}{b} = \frac{1}{St^2} * \frac{C_c}{\sqrt{\frac{Sc}{4*\pi} - K_a * \left(1 - \left(\frac{\sigma_y}{b*a_L}\right)^2\right)}} * \sqrt{\frac{\rho * b^2}{m^e}} * \sqrt{\frac{b}{h}} \qquad \text{(E.14)}$$

where:

C_c is the aerodynamic constant dependent on the cross-sectional shape, and for a circular cylinder also dependent on the Reynolds number Re as defined in E.1.3.4 (1); given in Table E.6.

K_a is the aerodynamic damping parameter as given in E.1.5.3 (4)

a_L is the normalised limiting amplitude giving the deflection of structures with very low damping; given in TableE.6

St is the Strouhal number given in E.1.6.2

ρ is the air density under vortex shedding conditions, see Note 1

m_e is the effective mass per unit length; given in F.4 (1)

h, b is the height and width of structure. For structures with varying width, the width at the point with largest displacements is used.

(3) The solution to Expression (E.14) is given in Expression (E.15).

$$\left(\frac{\sigma_y}{b}\right)^2 = c_1 + \sqrt{c_1^2 + c_2} \qquad \text{(E.15)}$$

$$\left(\frac{\sigma_y}{15.4}\right)^2 = (-0.1) + \sqrt{(-0.1)^2 + 0}$$

$$\sigma_y = 0.528 \text{ m}$$

where the constants c_1 and c_2 are given by:

$$c_1 = \frac{a_L^2}{2} * \left(1 - \frac{Sc}{4 * \pi * K_a}\right); \qquad c_2 = \frac{\rho * b^2}{m^e} * \frac{a_L^2}{K_a} * \frac{C_c^2}{St^4} * \frac{b}{h};$$

$$c_1 = \frac{0.4^2}{2} * \left(1 - \frac{137.8}{4 * \pi * 6}\right) = -0.1 \qquad c_2 = \frac{1.25 * 15.4^2}{2.11E^5} * \frac{0.4^2}{6} * \frac{0.04^2}{0.8^4} * \frac{15.4}{144} = 0.00$$

(E.16)

For a circular cylinder and a square cross-section the constants C_c, $K_{a,max}$ and a_L are given in Table E.6.

$$C_c = 0.04 \qquad K_{a,\max} = 6 \qquad a_L = 0.4$$

The peak factor k_p should be determined.

$$k_p = \sqrt{2} * \left(1 + 1.2 \arctan\left(0.75 * \frac{Sc}{(4 * \pi * K_a)^4}\right)\right)$$

$$k_p = \sqrt{2} * \left(1 + 1.2 \arctan\left(0.75 * \frac{137.8}{(4 * \pi * 6)^4}\right)\right) = 1.415$$

The number of load cycles may be obtained from E.1.5.2.6 using a bandwidth factor of $\varepsilon_0 = 0.15$

The maximum acceleration is:

$$a_{max} = y_{F,max} * \left(2 * \pi * f_e\right)^2 = 0.746 * \left(2 * \pi * 0.13\right)^2 = 0.531 \text{ m/s}^2 .$$

Appendix 5 Verification Student building "Voorhof"

In this appendix, the dynamic model for bending and torsion is determined and verified in Simulink for the student building "Voorhof" before and after renovation.

Simulink is a sub program of Mathlab in which graphical programming can be done for modeling, simulating, and analyzing dynamic systems. Simulink enables you to pose a question about a system, model it, and see what happens.

The equation of motion for bending and torsion motion can be written as: ([12])

$$\mathrm{M}_{ben}\ddot{u} + C_{ben}\dot{u} + \mathrm{K}_{ben}u = F$$
$$\mathrm{M}_{tor}\ddot{\psi} + C_{tor}\dot{\psi} + \mathrm{K}_{tor}\psi = \mathrm{M}$$

The number of elements for this building model is 19, with at each node 2 degrees of freedom. The bending and torsion equations of motions are loaded in separate matrices in Simulink. The solution of the eigenmatrices required two separate matrices for bending and torsion. The M, C and K matrices for bending and torsion are (19 x 19) matrices and u, ψ and F, M are (19 x1) vectors. State space formulation had to be introduced for the bending and torsion equations of motion because Simulink cannot process a second order differential equation.

Verification of dynamic bending behaviour: eigenfrequency before renovation with calibration

Computation of the bending natural frequency

In Matlab the natural frequency ω_i can be determined. With $\varsigma_1 = 0.0108$ and $\omega_1 = 3.92$ rad/s the eigenfrequency of the damped system can be determined:

$$\omega_e = \omega_1\sqrt{1-\varsigma_1^{\,2}} = 3.92\sqrt{1-0.0108^2} = 3.92 \text{ rad/s}$$

Manual calculation of the bending natural frequency

A hand calculation will be performed to check that the eigenfrequency calculated by Simulink is correct. The first natural frequency is calculated using the Raleigh Quotient ([2] p.80])

$$\omega_n = C\sqrt{\left(EI / \rho A l^4\right)}$$

	$C = 3.52$	bending stiffness	$EI = 2.085E^{12}$ Nm^2
mean density	$\rho = 193$ kg/m^2	Area building	$A = 1047.50$ m^2
height building	$l = 51.3$ m		

$$\omega_n = C\sqrt{\left(EI / \rho A l^4\right)} = 3.52\sqrt{\left(\left(2.085E^{12}\right) / \left(193 * 1047.50 * 51.3^4\right)\right)} = 4.30 \text{ rad/s}$$

The first natural frequency from Mathlab corresponds to the first natural frequency of the Raleigh Quotient ([2] p.80]) is 1 - (3.92 / 4.30)* 100% = 8.7%

$$T_1 = \frac{2\pi}{\omega_1} = \frac{2\pi}{4.30} = 1.46 \text{ s}$$

Verification of the dynamic torsional behaviour: eigenfrequency before renovation with calibration

Computation of the natural torsion frequency

In Matlab the natural frequency ω_i can be determined. With $\varsigma_1 = 0.0108$ and $\omega_1 = 4.021$ rad/s the eigenfrequency of the damped system can be determined:

$$\omega_e = \omega_1\sqrt{1-\varsigma_1^{\,2}} = 4.021\sqrt{1-0.0108^2} = 4.021 \text{ rad/s}$$

Manual calculation of the torsion natural frequency

$$\omega_n = (\frac{(2*n-1)\pi}{2l})\sqrt{\left(\frac{GJ}{I_p}\right)}$$

mode $n = 1$ torsion stiffness $GJ = 1.863\ E^{12}\ Nm^2$

building height $l = 51.3$ m $I_p = \frac{2.827E^8\ \text{kgm}^2}{2.65m} = 106679245\text{kgm}$

$$\omega_n = (\frac{(2*n-1)\pi}{2l})\sqrt{\left(\frac{GJ}{\rho I_p}\right)} = (\frac{(2*1-1)\pi}{2*51.3})\sqrt{\left(\frac{1.863\ E^{12}}{106679245}\right)} = 4.05 \text{ rad/s}$$

With the outcome of ω_{n_tor} above on can see why the building characteristic values have to be calibrated. The torsion stiffness is 0.7% off. (1 - (4.021 / 4.05) * 100% = 0.7%) The torsion stiffness of the structure is most likely larger that the value determined in section 9.2.1.

Verification of dynamic bending behaviour: eigenfrequency after renovation with calibration

Computation of the bending natural frequency

In Matlab the natural frequency ω_i can be determined. With $\varsigma_1 = 0.0146$ and $\omega_1 = 5.35$ rad/s the eigenfrequency of the damped system can be determined:

$$\omega_e = \omega_1\sqrt{1-\varsigma_1^{\ 2}} = 5.35\sqrt{1-0.0146^2} = 5.35 \text{ rad/s}$$

Manual calculation of the bending natural frequency

A hand calculation will be performed to check that the eigenfrequency calculated by Simulink is correct. The first natural frequency can be calculated by using the Raleigh Quotient ([2] p.80])

$$\omega_n = C\sqrt{\left(EI / \rho Al^4\right)}$$

	$C = 3.52$	bending stiffness	$EI = 4.26E^{12}$ Nm^2
mean density	$\rho = 213$ kg/m^2	Area building	$A = 1047.50$ m^2
height building	$l = 51.3$ m		

$$\omega_n = C\sqrt{\left(EI / \rho Al^4\right)} = 3.52\sqrt{\left(\left(4.26E^{12}\right) / \left(213 * 1047.50 * 51.3^4\right)\right)} = 6.14 \text{ rad/s}$$

The first natural frequency from Mathlab corresponds to the first natural frequency of the Raleigh Quotient ([2] p.80]) is 1 - (5.35 / 6.14)* 100% = 12.9%

$$T_1 = \frac{2\pi}{\omega_1} = \frac{2\pi}{6.14} = 1.02 \text{ s}$$

Verification of the dynamic torsional behaviour: <u>eigenfrequency after renovation with calibration</u>

Computation of the torsion natural frequency

In Matlab the natural frequency ω_i can be determined. With $\varsigma_1 = 0.0146$ and $\omega_1 = 4.84$ rad/s the eigenfrequency of the damped system can be determined:

$$\omega_e = \omega_1\sqrt{1 - \varsigma_1^{\,2}} = 4.84\sqrt{1 - 0.0146^2} = 4.84 \text{ rad/s}$$

Manual calculation of the torsion natural frequency

$$\omega_n = \left(\frac{(2 * n - 1)\pi}{2l}\right)\sqrt{\left(\frac{GJ}{I_p}\right)}$$

mode $n = 1$ torsion stiffness $GJ = 3.187\ E^{12}\ Nm^2$

building height $l = 51.3$ m $I_p = \dfrac{3.367E^8\ kgm^2}{2.65m} = 127056604kgm$

$$\omega_n = (\frac{(2*n-1)\pi}{2l})\sqrt{\left(\frac{GJ}{I_p}\right)} = (\frac{(2*1-1)\pi}{2*51.3})\sqrt{\left(\frac{3.187\ E^{12}}{127056604}\right)} = 4.85 rad/s$$

With the outcome of ω_{n_tor} above on can see why the building characteristic values have to be calibrated. The torsion stiffness is 0.2% off. (1 - (4.84 / 4.85) * 100% = 0.2%) The torsion stiffness of the structure is most likely larger that the value determined in section 9.2.1.

Appendix 6 Matlab code Voorhof before renovation (time domain analysis)

The matlab code which is the same as the Juffertoren is not given.

Below the Matlab code is given for the Voorhof before renovation for return period of one year, the only difference with the Matlab code for return period of 12.5 years is that the wind parameters for Harris and Deaves (9.5) are taken and inputted in Matlab file: A_File_to_run_programs.m and $v_{10} = 15.79 m/s$ and $\sigma_v = 6.65 m/s$ (9.6) are inputted in Matlab file: Inputvalues.m

A_File_to_run_programs.m

```
% Program to run the different modules
%
%
% Thesis: Torsion motions of high-rise buildings due to wind loading.
% Author: H.A.O.Richardson (Anthony)
%
%---------------------------------------------------------------------------
%

tic
disp('-----------------------------------------------------')
disp('Thesis: Torsion motions of high-rise buildings')
disp('-----------------------------------------------------')

% Clear memory

clear
clear all
clc
close all

% Start timer

tic

%---------------------------------------------------------------------------
% Building input
%---------------------------------------------------------------------------


disp('-Reading problem data')

%inputting_variables_of_the_building_construction

save matrices
```

```
B_Variables_Wallplaces;

% Determining of the mass, bendingstiffness polarmoments and
% torsionalstiffness of the cross section.


C_Voorhof_before_renovation


% Determining maximum wind profile

u_star=3.836;         % Friction velocity
kappa=0.4;           % Terrain Roughness
d=3.5;               % average height of buildings (m)
z_0=0.7;             % Terrain Roughness according NEN 6702
step_z=1.325;        % Height of grid in which the building is divided


% DETERMING V_10

step_z=1.0;

[v_mean]=velocitymeanlogmodified(L,u_star,kappa,d,z_0,step_z);

Vhub=v_mean(1,2);


% meanwind velocity according to Harris and Deaves

step_z=1.325;

[v_mean]=velocitymeanlogmodified(L,u_star,kappa,d,z_0,step_z);

vh=max(v_mean);

Iz_top=(u_star/kappa)/vh;
Iz_10=(u_star/kappa)/Vhub;

save matrices vh Vhub v_mean Iz_top Iz_10 -append


%-------------------------------------------------------------------------
% Simulink input
%-------------------------------------------------------------------------

disp('-Input Simulink data')


[k,K_ben,M_ben,omega_eig_ben,a_ben,Cd_ben,phsi_ben,ktor,K_tor,M_tor,omega_e
ig_tor,a_tor,Cd_tor,Cd,E_ben]=D_Stiffness_matrix(v,EI,l,Cd,massfloor,mass1,
mass19,G,Jtot,Ip,Ip1,Ip19,xi1,xi2);


save matrices K_ben M_ben omega_eig_ben Cd_ben a_ben v k EI E_ben -append;
save matrices K_tor M_tor omega_eig_tor Cd_tor a_tor ktor -append;
```

```
%-------------------------------------------------------------------------
% Determining of space state formulations for simulink
%-------------------------------------------------------------------------


f=1350000;% force on node is

l_crossc=B;

[A_ben,B_ben,C_ben,D_ben,fe,fe_tor,A_tor,B_tor,C_tor,D_tor]=E_Space_state_f
ormulation(f,v,K_ben,M_ben,Cd_ben,omega_eig_ben,EI,L,l_crossc,K_tor,M_tor,C
d_tor,omega_eig_tor,l,G,Jtot);

save matrices A_ben B_ben C_ben D_ben v -append;

save matrices fe fe_tor A_tor B_tor C_tor D_tor -append;


%-------------------- -----------------------------------------------------
-
% Fluctuating wind (wind generator)
%-------------------------------------------------------------------------


%{
load matrices

uiimport UC1.mat %loading the fluctuating velocities %undo

%}

Inputvalues;

tic
[t,UC]=wind0(yr,zr,U,sigma_v,N,deltat,fmax);
toc


save ('UC')
save ('UC1','UC')  % original
%{
%}

%Are=2.5*1.325;  % area of one node which is loaded by wind

Are=(yr(2)-yr(1))*(zr(2)-zr(1));% area of one node which is loaded by wind


%-------------------- -----------------------------------------------------
-
% From wind velocities to forces and moments
%-------------------------------------------------------------------------
```

```
[F,M,t,FvoorWoudenberg,F_woud]=F_Forces_moments_19_DOF(v_mean,N,Ch,Are,rho,
B,deltat,L);

% Saving generated forces and Moments

save ('F','F','t') % Saving force of each time step
save ('M','M','t') % Saving moment of each time step


%------------------------------------------------------------------------
% Simulink Run
%------------------------------------------------------------------------


disp('-Run Simulink ')


%sim('G_Simulink_run_file');



%------------------------------------------------------------------------
% Getting the maximums of bending, torsion and bending and torsion
%------------------------------------------------------------------------

%H_Max_acceleration_values_and_graphs_output

 -----------------------------------------------------------------------
-

disp( '-Finished analysis. Time:' )
toc
```

B_Variables_Wall_Places.m

```
% Inputting varibles of the construction
%
% Thesis: Torsion motions of high-rise buildings due to wind loading.
% Author: H.A.O.Richardson (Anthony)
%
%--------------------------------------------------------------------------

% The outer dimensions of the total building

    L=51.3;                % [m] length of the building
    B=80.81;              % [m] width of the building
    H=14.2;              % [m] height of the building
    rhoconcrete= 2400;    % [kg/m3]specific density concrete
    rho=1.29;             % [kg/m3]specific density air
    Ch=1.2;               % [-] Thrust and Suction shape factor

    h=2.65;               % [m] height of story of the building
    l=h;                  % [m] lenght of one element

    hfl=0.15;             % [m] thickness of floor of the building

    hwall=h-hfl;          % [m] height of the walls of the building

    nbfloors=L/h;         % [-] amount of floors of the building

    Pvb=58.3;               % [kg/m2] weight of load on the floor (0,7kN/m2)

    qfacade=120;          % [kg/m2] weight of facade elements per m2
(1,2kN/m2)
                          % Limitation to the benchmark Skyscraper page 111)

    E=3e10;               % [N/m2] Youngsmodulus of the concrete C35/B65

    vu=0.2;               % [-] Poisson ratio

    g=9.81;               % [m/s2] Gravitational acceleration

 %-------------------------------------------------------------------------

    v= 19;                % number of elements

    dof= 2;               % degrees of freedom per node

 % units (meters and kN)

    xi1=0.0108;            % damping ratio of the first eigenmode
    xi2=0.0108;            % damping ratio of the first eigenmode
    tend=10;             % duration of the simulink simulation time
    timestep=5e-5;       % fixed timestep in simulink
    deltat=0.1;          % delta t

% nwallseg is the amount wall segments that the walls of the cross section
is
```

```
% divided into.

    nwallsegm=16;

% Declaring matrix for the wall segments.

    P=zeros(nwallsegm,4);


% The cross-section is divided into 16 wall segments for calculation.

%  The origin of the cross sectiion of the building is taken in the left
%  bottom corner.

%    Length x         Length y         Distance x midpoint to  Distance y
midpoint to
%                                           Startpoint O
Startpoint O

P(1,1)=15.440;  P(1,2)=0.500;     P(1,3)=15.440/2;
P(1,4)=26.340/2-0.5/2;
P(2,1)=6.920;   P(2,2)=0.600;     P(2,3)=6.920/2;
P(2,4)=0.5+4.7+0.6+14.74+0.6/2;
P(3,1)=1.000;   P(3,2)=0.300;     P(3,3)=6.620+0.3+1.1+1.0/2;
P(3,4)=0.5+4.7+0.6+14.74+0.3/2;
P(4,1)=0.300;   P(4,2)=3.720;     P(4,3)=6.620+0.3+1.1+1.0+0.3/2;
P(4,4)=0.5+4.7+0.6+14.74+0.3-2.72/2;
P(5,1)=6.120;   P(5,2)=0.300;     P(5,3)=15.44-6.120/2;
P(5,4)=0.5+4.7+0.6+14.74-2.1-0.3/2;
P(6,1)=3.100;   P(6,2)=0.300;   P(6,3)=15.44-3.02-0.3-3.1/2;
P(6,4)=0.5+4.7+0.6+14.74-2.1-0.3-1.02-0.3/2;
P(7,1)=3.100;   P(7,2)=0.200;   P(7,3)=15.44-3.02-0.3-3.1/2;
P(7,4)=0.5+4.7+0.6+14.74-2.1-0.3-1.02-0.3-2.3-0.2/2;
P(8,1)=0.300;   P(8,2)=14.740;  P(8,3)=6.620+0.3/2;
P(8,4)=0.5+4.7+0.6+14.74/2;
P(9,1)=0.300;   P(9,2)=7.900;   P(9,3)=15.44-3.02-0.3/2;
P(9,4)=0.5+4.7+0.6+14.74/2;
P(10,1)=3.100;  P(10,2)=0.200;  P(10,3)=15.44-3.02-0.3-3.1/2;
P(10,4)=0.5+4.7+0.6+2.1+0.3+1.02+0.3+2.3+0.2/2;


P(11,1)=3.100;  P(11,2)=0.300;  P(11,3)=15.44-3.02-0.3-3.1/2;
P(11,4)=0.5+4.7+0.6+2.1+0.3+1.02+0.3/2;
P(12,1)=6.120;  P(12,2)=0.300;  P(12,3)=15.44-6.120/2;
P(12,4)=0.5+4.7+0.6+2.1+0.3/2;
P(13,1)=0.300;  P(13,2)=3.720;  P(13,3)=6.620+0.3+1.1+1.0+0.3/2;
P(13,4)=0.5+5.0+2.72/2;
P(14,1)=6.920;  P(14,2)=0.600;  P(14,3)=6.920/2;
P(14,4)=0.5+4.7+0.6/2;
P(15,1)=1.000;  P(15,2)=0.300;  P(15,3)=6.620+0.3+1.1+1.0/2;
P(15,4)=0.5+5.0+0.3/2;

P(16,1)=15.440; P(16,2)=0.500;  P(16,3)=15.440/2;
P(16,4)=0.5/2;

%save matrices
save matrices L B H rhoconcrete rho Ch h l hwall g nbfloors hfl Pvb qfacade
E vu v xi1 xi2 tend timestep deltat nwallsegm  P -append
```

C_Voorhof_before_renovation.m

```
% Determining stiffness and torsion matrix for voorhof
%
% Thesis: Torsion motions of high-rise buildings due to wind loading
% Author: H.A.O. Richardson (Anthony)

%INPUT
   v=19;                  %   number of elements                       [-]
   l=2.65;                %   lenght of element                        [m]
   mvb=58.3;              %   variable floor mass                      [kg/m2]
   rhosteel=7900;         %   Mass denisity of steel                   [kg/m3]

   %-----------------------------------------------------------------------
   %Bending before renonvation
   %-----------------------------------------------------------------------


        %EI=2.53E12; %(1)        %   bending stiffness
        EI=2.085E12; %(2)        %   bending stiffness modified

        % maas buidling 10.943E6/19  page 129b[m]

        %{
        massfloor=(10.943E6-(10.943E6/(19*2)))/18;
        mass19=10.943E6/(19*2); % Weight of top floor is approximated half
of other floors
        %}

        %-----------------------------------------------------------------
        % estimating Mass of 19 th floor properly
        %-----------------------------------------------------------------

        % page 129 % M.Pols

            %mass steel            = 0.538E6
            %mass floor            = 7.672E6
            %mass partition walls  = 1.764E6
            %mass facade           = 0.969E6

            massfloor=580514;

            floormasstot= 10.943E6;

            floormass19= 7.672E6+0.5*(0.538E6+1.764E6+0.969E6);

            verh= floormass19/floormasstot; %0.85 without foundation

            %verh=floormass19/(floormasstot+1.451E6);%0.75 with foundation

            mass19=massfloor*verh;
           %{
            %}

           % Tekst f building is 0.641 to 0.750 HZ Page 139 [17]
```

```
        % Additional strucural stiffness (walls and steel frame) p.153

        % Strucural stiffness building (walls, steel frame, non structural
elements) p.153


   %-----------------------------------------------------------------------
   %Ixx structural steel (Pols)(Before Renovation)
   %-----------------------------------------------------------------------


        Ess=2.1E5*1000*1000;    %[N/m2]  2.1E5 N/mm2 modulus of elasticity

        Iss=EI/Ess;             %HEA 260

        %Issy=Iss/(3668/10455);

        %Issy=4*Iss;

        %Issy=16*Iss;

        % determining Issy

        Y1=(2*16130+2*2530)/(1000*1000); % mm2area of cross-section profile
Y1
        Y2=(2*11250+2*19160)/(1000*1000); %area of cross-section profile Y1
        Y3=(2*24020+2*2530)/(1000*1000); %area of cross-section profile Y1
        Y4=(2*24020+2*2530)/(1000*1000); %area of cross-section profile Y1



Issy=2*(Y3*4^2+Y3*8^2+Y3*12^2+Y3*16^2+Y3*20^2)+((Y3+Y4)*24^2)+((Y2+Y4)*28^2
)+((Y3+Y4)*32^2)+2*Y2*36^2+2*Y1*40^2;

   %-----------------------------------------------------------------------
   % Area for Woudenberg
   %-----------------------------------------------------------------------

    AreaWoud=2*(Y3+Y3+Y3+Y3+Y3)+((Y3+Y4))+((Y2+Y4))+((Y3+Y4))+2*Y2+2*Y1;

   %-----------------------------------------------------------------------



   %-----------------------------------------------------------------------
   % Torsion components Before renonvation
   %-----------------------------------------------------------------------

        % Shearmodulus of the structural steel .( Ridigidy modulus)


        %Ess=2.1E5*1000*1000;      %[N/m2]  2.1E5 N/mm2 modulus of
elasticity

        Gss=Ess/(2*(1+vu));       %[N/m2]
```

```
        %Gss=79.3E9;            %   Shear modulus struc steel  79.3
Gpa[N/mm2] website


        % Shearmodulus of the buidling.( Ridigidy modulus)


        G=Gss;          %   Shear modulus Steel (before renovation)
[N/m2]

        %G=3.6E11;


 %---------------------------------------------------------------------------
------------
    % Torsion constant structural steel
 %---------------------------------------------------------------------------
------------


        %torsion constant of structural steel

            % J van 1 profile is 54.2 cm4

             Jss=8*54.2/(10000*10000);  % [m4]

        %torsion constant of concrete walls

            %Jtot=Jss;         %   Torsion constant     (before)   [m4]

            %Jtot=12000000*Jss;

        % The torsional stiffness of one floor of the structure.

            %TorSTif=G*Jtot;

            %TorSTif=4.53E10;%(1)  % determined in Matrix CAE

            TorSTif=1.863E12; % modified

            Jtot=TorSTif/Gss;

  %--------------------------------------------------------------------------
-----------------------
  %Polar moment of inertia (Before renovation)
  %--------------------------------------------------------------------------

          %Strucural steel (before renovation)

            Ixtot=Iss;                     % HEA 260
            Iytot=Issy;

        % Polar inertia of the walls Ix and Iy of a floor
```

```
            Ipwall=Ixtot+Iytot;

        % Polar inertia of the floor Ix and Iy

            Ipfloor=((1/12)*H*B^3)+((1/12)*B*H^3);

        %----------------------
        %Before Renovation
        %----------------------

        % Polar inertia of the floor segments node 1 to node 18

            Ip=Ipwall*h*rhosteel+Ipfloor*(hfl*rhoconcrete+mvb);

        % Polar inertia of the top floor of the building

            Ip19=Ipwall*(h/2)*rhosteel+Ipfloor*(hfl*rhoconcrete+mvb);

        %----------------------------------------------------
        %Arbituarly chosen values (calibration)
        %----------------------------------------------------



   %-----------------------------------------------------------------------
      %Damping Ratios
   %-----------------------------------------------------------------------


        xi1=0.0108;           %   damping ratio of first eigenmode    [-]
        xi2=0.0108;           %   damping ratio of second eigenmode   [-]

        mass1=massfloor;
        Ip1=Ip;               %   Second moment of inertia  1 floor
[kgm2]

   %-----------------------------------------------------------------------
-
   %new added code 21-12
   %-----------------------------------------------------------------------


        % Total mass of the buidling

        % mass building of 18 floors walls + top floor

        mass=mass1+(v-2)*massfloor+mass19;

        % mass=v*massfloor+H*B*hfloor*rhoconcrete;

% Total weigth of the building

        Q_gebouw=mass*g;            % Gewicht building
```

```
% Spefic density of building                 [kg/m3]

        rhob=mass/(B*H*L);

% mass per meter building height             [kg/m1]

        rho11=mass/(L);

% mass per area              [kg/m2]

        rho_m2=mass/(B*H);
```

D_Stiffnes_matrix.m

```
function[k,K_ben,M_ben,omega_eig_ben,a_ben,Cd_ben,phsi_ben,ktor,K_tor,M_tor
,omega_eig_tor,a_tor,Cd_tor,Cd,E_ben]=D_Stiffness_matrix(v,EI,l,Cd,massfloo
r,mass1,mass19,G,Jtot,Ip,Ip1,Ip19,xi1,xi2)
%
% Determining stiffness and torsion matrix
%
%INPUT
%
%   v:              number of elements                   [-]
%   EI:             bending stiffness                    [Nm2]
%   l:              lenght of element                    [m]
%   massfloor:      mass of a floor                      [m]
%   mass1:          mass of bottom floor                 [m]
%   mass19:         mass of top floor                    [m]
%   G:              Shear modulus                        [N/mm2]
%   Jtot:           Torsion constant                     [mm4]
%   Ip:             Second moment of inertia             [mm4]
%   Ip1:            Second moment of inertia  1 floor    [mm4]
%   Ip19:           Second moment of inertia 19 floor    [mm4]
%   xi1:            damping ratio of first eigenmode     [-]
%   xi2:            damping ratio of second eigenmode    [-]
%
%OUTPUT
%
%   k:              Bending stiffness field elements
%   K_ben:          Bending stiffness matrix
%   M_ben:          Mass stiffness matrix
%   omega_eig_ben:  Eigen frequency bending matrix MDF
%   a_ben:          Damping martix first and second term
%   Cd_ben:         Damping matrix bending MDF
%   phsi_ben:       Damping ratios matrix bending
%   ktor:           Torsional stiffness field elements
%   K_tor:          Torsion stiffness matrix
%   M_tor           Mass torsion matrix
%   omega_eig_tor:  Eigen frequency torsion matrix MDF
%   a_tor:          Damping martix first and second term
%   cd_tor:         Torsion damping matrix MDF
%
%   Cd:             Damping matrix contoles system bending MDF
%   omega_eig:      Eigen frequency contolled bending matrix MDF
%
% Thesis: Torsion motions of high-rise buildings due to wind loading.
% Author: H.A.O.Richardson (Anthony)


%------------------------------------------------------------------------
% Assembling the stiffness matrix
%------------------------------------------------------------------------
% Bending Field elements

    k=EI/l^3*[1     -2     1;
            -2     4    -2;
            1     -2     1];% bending element stifnessmatrix

    K_ben=zeros(v+4,v+4); % Total systemmatrix bending
    for o=0:1:(v-1)%v-1
```

```
        for n=1:1:3
            for m=1:1:3
                K_ben(o+n,o+m)=K_ben(o+n,o+m)+k(n,m);
            end
        end
    end

 %Rearanging bending matrix for simulink

    K_ben(:,1)=[];
    K_ben(1,:)=[];

    K_ben(:,1)=[];
    K_ben(1,:)=[];


% Top element
% Delting the unwanted and coloum k-matrix for half element.

    K_ben(:,21)=[];         % K_ben(:,((Size(K_ben,2))+2)=[]
    K_ben(21,:)=[];

    K_ben(:,20)=[];
    K_ben(20,:)=[];


%---------------------------------------------------------------------------
% Assembling the mass matrix
%---------------------------------------------------------------------------

    M_ben=zeros(v,v);

    for n=1:1:v;
        M_ben(n,n)=massfloor;
    end

    % Making the first and last term of mass matrix correct

    M_ben(n,n)=mass19;


%---------------------------------------------------------------------------
% Determing the eigen frequency (MDF)
%---------------------------------------------------------------------------



    % E is the modal matrix; omegakw is modal K*E = M*E*OMEGAKW
    [E_ben,omegakw_ben] = eig(K_ben,M_ben);
    for n=1:1:v
        omega_eig_ben(n)=sqrt(omegakw_ben(n,n));
    end


%---------------------------------------------------------------------------
% Assembling of the damping matrix
%---------------------------------------------------------------------------
```

```
    a_ben=2*(inv([1/(omega_eig_ben(1)) (omega_eig_ben(1));
1/omega_eig_ben(2) omega_eig_ben(2)])*[xi1;xi2]);
    Cd_ben=a_ben(1,1)*M_ben+a_ben(2,1)*K_ben;


%---------------------------------------------------------------------------
% plotting damped motion
%---------------------------------------------------------------------------

% damping ratio of the eigenmodes of the structure

    for n=1:1:v;

phsi_ben(n)=a_ben(1,1)/(2*omega_eig_ben(n))+a_ben(2,1)/2*omega_eig_ben(n);
    end

    %{
    figure
    plot(phsi_ben)
    xlabel('eigenmode')
    ylabel('value')
    title('Plot Damping Ratio Bending')
    %}

    save matrices K_ben M_ben omega_eig_ben Cd_ben a_ben v -append;


%-------------------------------------------------------------------------
% Torsional movement
%-------------------------------------------------------------------------




%-------------------------------------------------------------------------
% Assembling the stiffness matrices
%-------------------------------------------------------------------------

    % Torsional Field elements


    ktor=(G*Jtot)/l*[1  -1
                    -1   1];    % 1 floor height is 2.65 meters


    K_tor=zeros(v+2,v+2); % Total systemmatrix bending

    for o=0:1:(v-1)
      for n=1:1:2
        for m=1:1:2
            K_tor(o+n,o+m)=K_tor(o+n,o+m)+ktor(n,m);
        end
      end
    end
```

```
% Bottom element
% Deleting the unwanted row and coloum k-matrix for half element.

    K_tor(:,1)=[];
    K_tor(1,:)=[];

% Top element
% Deleting the unwanted and coloum k-matrix for half element.

    K_tor(:,20)=[];
    K_tor(20,:)=[];


%------------------------------------------------------------------------------
% Assembling the polar mass matrix
%------------------------------------------------------------------------------

    M_tor=zeros(v,v);

    for n=1:1:v;
        M_tor(n,n)=Ip;
    end

% Making the first and last term of the torsional matrix correct.

    %M_tor(1,1)=Ip1;
    M_tor(n,n)=Ip19;


%------------------------------------------------------------------------------
% Determing the eigen frequency (MDF)
%------------------------------------------------------------------------------



    % E is the modal matrix; omegakw is modal K*E = M*E*OMEGAKW

    [E_tor,omegakw_tor] = eig(K_tor,M_tor);
    for n=1:1:v
        omega_eig_tor(n)=sqrt(omegakw_tor(n,n));
    end


%------------------------------------------------------------------------------
% Assembling of the damping matrix
%------------------------------------------------------------------------------

    a_tor=2*(inv([1/(omega_eig_tor(1)) (omega_eig_tor(1));
1/omega_eig_tor(2) omega_eig_tor(2)])*[xi1;xi2]);
    Cd_tor=a_tor(1,1)*M_tor+a_tor(2,1)*K_tor;


%------------------------------------------------------------------------------
% plotting damping ratios
%------------------------------------------------------------------------------

% damping ratio of the eigenmodes of the structure
```

```
    for n=1:1:v;

phsi_tor(n)=a_tor(1,1)/(2*omega_eig_tor(n))+a_tor(2,1)/2*omega_eig_tor(n);
    end
%{
    figure
    plot(phsi_tor)
    xlabel('eigenmode')
    ylabel('value')
    title('Plot Damping Ratio Torsion')
%}


save matrices K_tor M_tor omega_eig_tor Cd_tor a_tor  -append;
```

E_Space_state_formulation.m

```
function[A_ben,B_ben,C_ben,D_ben,fe,fe_tor,A_tor,B_tor,C_tor,D_tor]=E_Space
_state_formulation(f,v,K_ben,M_ben,Cd_ben,omega_eig_ben,EI,L,l_crossc,K_tor
,M_tor,Cd_tor,omega_eig_tor,l,G,Jtot)
%
% Determining Space state formulation bending and torsion,Plotting
% displacement of building bending and torsion,Space stateformulaton
% bending and torsion,Validation of dynamic behaviour bending and torsion.
%
%INPUT
%
%   f:              Total force of horizontal row of nodes   [N]
%   v:              Number of elements                       [-]
%   K_ben:          Bending stiffness matrix
%   M_ben:          Mass stiffness matrix
%   Cd_ben:         Damping matrix bending MDF
%   omega_eig_ben:  Eigen frequency bending matrix MDF       [rad]
%   EI:             bending stiffness                        [Nm2]
%   L:              Height building                          [m]
%   l_crossc:       Width building                           [m]
%   K_tor:          Torsion stiffness matrix
%   M_tor:          Mass torsion matrix
%   Cd_tor:         Damping matrix bending MDF
%   omega_eig_tor:  Eigen frequency torsion matrix MDF       [rad]
%   l:              Height building                          [m]
%   G:              Shear modulus                            [N/m2]
%   Jtot:           Torsion constant                         [mm4]
%
%OUTPUT
%
%   A_ben           State space bending input matrix for simulink
%   B_ben           State space bending input matrix for simulink
%   C_ben           State space bending input matrix for simulink
%   D_ben           State space bending input matrix for simulink
%   fe              Bending Frequency                        [Hz]
%   fe_tor          Torsion Frequency                        [Hz]
%   A_tor           State space torsion input matrix for simulink
%   B_tor           State space torsion input matrix for simulink
%   C_tor           State space torsion input matrix for simulink
%   D_tor           State space torsion input matrix for simulink
%
%
% Thesis: Torsion motions of high-rise buildings due to wind loading.
% Author: H.A.O.Richardson (Anthony)



%--------------------------------------------------------------------------
% Space state formulation
% Changing the matrices into formulation readable for simulink
%--------------------------------------------------------------------------



%--------------------------------------------------------------------------
% Determining static diplacement
%--------------------------------------------------------------------------

    F= zeros(v,1);
```

```
        for  lr=1:1:v
             F(lr,1)=f;
        end

% Bending Displacement


    u_ben=(-inv(K_ben))*F;

%-----------------------------------------
% Coordinates in z direction
%-----------------------------------------



    % The z cordinates of the nodes in vertical direction

        Z= zeros(v,1);

        zzz=2.65;

        for  zrn=1:1:v

            Z(zrn,1)=(zzz+(zrn-1)*2.65);
        end

    %---------------------------------------------------------------------
----
    %Plotting  displacment of the buiding.
    %---------------------------------------------------------------------
----


        figure
        plot(abs(u_ben),Z)

        axis([0 0.21 0 55])
        set(gca,'YTick',0:10:55)
        set(gca,'XTick',0:0.05:0.20)

        xlabel('u(m)ben')
        ylabel('z(m)')
        title('Plot Static deflection')


%-------------------------------------------------------------------------
%Space State Formulation.
%-------------------------------------------------------------------------

% Placing space state formlation Matrix A

    A2_ben=eye(v,v);
    A3_ben=-inv(M_ben)*K_ben;
    A4_ben=-inv(M_ben)*Cd_ben;

% Placing space state formlation Matrix B
```

```
    B1_ben=inv(M_ben);
    C1_ben=eye(v,v);

    A_ben(1:v,v+1:1:2*v)=A2_ben;
    A_ben(v+1:1:2*v,1:v)=A3_ben;
    A_ben(v+1:1:2*v,v+1:1:2*v)=A4_ben;
    B_ben(v+1:1:2*v,1:v)=B1_ben;

    C_ben=eye(2*v,2*v);
    D_ben=zeros(2*v,v);


%---------------------------------------------------------------------------
% Validation of dynamic behavior
%---------------------------------------------------------------------------

% Zie matrix omega_eigen_ben(1,1) demping first natural frequency

zeta1=0.01;

omega_e_ben=omega_eig_ben(1,1)*(1-zeta1^2)^0.5;

% Equivelent stiffness

    keq=(8*EI)/(1.0*L^4);

    verp=500000*(1.0*L^4)/(8*EI);  %500 kn/m1 (story 2.65)=1350 kn/m1
    diff=verp+u_ben(19,1);
    diffper=diff/verp*100;
    diffper2=(1-(verp/u_ben(19,1)));

%-----------------------------------------------------------------------

% zeta1=0.01; not used because in the memory already

    Ftop=-1*(keq*u_ben(19,1));

% Equivelent Mass

    meq=keq/(omega_eig_ben(1,1)^2);
    zetameq=0.05;
    we=omega_e_ben;
    fe=we/(2*pi);

%------------------------------------------------------------------
% Plot Dynamic behavior
%------------------------------------------------------------------
    %{
for n=1:1:20000
    t=n/100;

    %x_48(n)=(Ftop/keq)*(cos(we*t));
    x_48(n)=1.05*(Ftop/keq)*(1-(exp(-
zetameq*(((keq/meq)*t)^0.5)))*(cos(we*t)));
    %x_47(n)=(Ftop/keq)*((exp(-zetameq*(((keq/meq)*t)^0.5)))*(cos(we*t)));
```

```
    tijd(n)=n/100;
    end

    figure
    plot(tijd,x_48,'b')

    hold on
    %plot(tijd,x_47,'r')
    hold off

    axis([0 80 0 0.4])
    xlabel('t(s)')
    ylabel('u(m)')
    %}

%--------------------------------------------------------------------------
% Torsional movement
%--------------------------------------------------------------------------



%--------------------------------------------------------------------------
% Determining static diplacement Torsion
%--------------------------------------------------------------------------


% over on each side 0.17
    over=0.17;
% horizondat division length
    Yrd=2.6/2;
    Arm=l_crossc/2-over-Yrd;
    f=186300; %  Force on each node in N
    m=f*Arm*10; % *10 voor groter moment 21797100 Nm



% To validate the static torsional displacement we will put a positive
moment and then a negative moment on each consecative node.

      M= zeros(v,1);

    for  lr=1:1:v
         M(lr,1)=-m;
    end

u_tor=-inv(K_tor)*M;

% The z cordinates of the nodes in vertical direction

Zy_tor= zeros(v,1);

    %zzz= 1.5; undo for normal run

for  zrn=1:1:v

     Zy_tor(zrn,1)=(zzz+(zrn-1)*3);
end

%--------------------------------------------------------------------------
```

```
%Plotting rotational displacment of the building.
%-----------------------------------------------------------------------------

%{
 figure
 plot (u_tor,Z)

 %axis([0 0.21 0 145])
 %set(gca,'YTick',0:20:144)
 %set(gca,'XTick',0:0.05:0.21)

 xlabel('u_ tor(m)')
ylabel('z(m)')
title('Plot Static Rotational Displacement')
%}


%-----------------------------------------------------------------------------
%Space State Formulation.
%-----------------------------------------------------------------------------

% Placing space state formlation Matrix A

    A2_tor=eye(v,v);
    A3_tor=-inv(M_tor)*K_tor;
    A4_tor=-inv(M_tor)*Cd_tor;

% Placing space state formlation Matrix B

    B1_tor=inv(M_tor);
    C1_tor=eye(v,v);

    A_tor(1:v,v+1:1:2*v)=A2_tor;
    A_tor(v+1:1:2*v,1:v)=A3_tor;
    A_tor(v+1:1:2*v,v+1:1:2*v)=A4_tor;
    B_tor(v+1:1:2*v,1:v)=B1_tor;

    C_tor=eye(2*v,2*v);
    D_tor=zeros(2*v,v);


%-----------------------------------------------------------------------------
% Validation of dynamic behavior
%-----------------------------------------------------------------------------

% Zie matix omega_eigen_tor(1,1) demping first natural frequency

    omega_e_tor=omega_eig_tor(1,1)*(1-zeta1^2)^0.5;

    fe_tor=omega_eig_tor(1,1)/(2*pi); % added 04/01

% Equivelent stiffness

    keq_tor=(l*Arm)/G*Jtot;

% zeta1=0.01; not used because in the memory already
```

```
    Mtop=-1*(keq_tor*u_tor(18,1));

% Equivelent Polar moment

    Ip_eq=keq_tor/(omega_eig_tor(1,1)^2);
    zetameq=0.05;

%{
    for n=1:1:20000
        t=n/100;

        %x_48(n)=(Mtop/keq_tor)*(1-cos(we*t));
        x_48_tor(n)=1.05*(Mtop/keq_tor)*(1-(exp(-
zetameq*(((keq_tor/Ip_eq)*t)^0.5)))*(cos(omega_e_tor*t)));
        x_47_tor(n)=(Mtop/keq_tor)*(1-(exp(-
zetameq*(((keq_tor/Ip_eq)*t)^0.5)))*(cos(omega_e_tor*t)));

        tijd(n)=n/100;
    end

    figure
    plot (tijd,x_48_tor,'b')

    hold on
    plot(tijd,x_47_tor,'r')
    hold off

    %axis([0 80 -0.0001 0.0001])

    %}
```

F_Forces_moments_19_DOF.m

```
function[F,M,t,FvoorWoudenberg,F_woud]=F_Forces_moments_19_DOF(v_mean,N,Ch,
Are,rho,B,deltat,L)
% Syntax: function[F,M]=Force_mod(v_mean)
%
% Summation of mean wind speed and flucuating wind speed determining the
% Nodal forces and Nodal moments for the grided area.
%
%INPUT
%   vmean:        Mean wind speead at height
[kg/m]
%   UC1:          Fluctuating velocity from the random generator
[m/s]
%
%OUTPUT
%   F             Force martix                                          [N]
%   M             Moment matrix
[Nm]
%
%
% Thesis: Torsion motions of high-rise buildings due to wind loading.
% Author: H.A.O.Richardson (Anthony)
%---------------------------------------------------------------------------

load matrices

load UC1                  %Importing the fluctuating velocity from the random
generator.
Uf(:,:,5:1:36)=UC;        %places matrix UC in matrix Uf from position 8 to
37

Uf(:,:,1)=Uf(:,:,8);      %places the fluctuating values of Uf(:,:,8) in
Uf(:,:,2)and Uf(:,:,1)
Uf(:,:,2)=Uf(:,:,8);
Uf(:,:,3)=Uf(:,:,8);      %places the fluctuating values of Uf(:,:,8) in
Uf(:,:,4)and Uf(:,:,3)
Uf(:,:,4)=Uf(:,:,8);



%N=8192; %take away
% Mean velocity according to NEN 6702
%v_mean(1,1,3:1:93)=u_star/kappa*log(z-d/z_0);    % mean wind speed at
reference height
%for n=1:

v_mean2(1,1,5:1:36)=v_mean;

%end
for x=1:1:4

v_mean2(1,1,x)=v_mean2(1,1,3);  %velocity out of vmean(1,1,3) is given to
vmean(1,1,1-7)

end

v_mean=v_mean2;
```

```
% Mean velocity is determined for each time step
% There is 10 timesteps in a second this is the 1 to 10 in vmean
% v_mean(:,1,:)or N
% v_mean(:,1,:) nodes in the width direction which are 10  for this
% buidling


v_mean(:,1,:)=v_mean(1,1,:);
v_mean(:,2,:)=v_mean(1,1,:);
v_mean(:,3,:)=v_mean(1,1,:);
v_mean(:,4,:)=v_mean(1,1,:);
v_mean(:,5,:)=v_mean(1,1,:);
v_mean(:,6,:)=v_mean(1,1,:);
v_mean(:,7,:)=v_mean(1,1,:);
v_mean(:,8,:)=v_mean(1,1,:);
v_mean(:,9,:)=v_mean(1,1,:);
v_mean(:,10,:)=v_mean(1,1,:);

v_mean(:,11,:)=v_mean(1,1,:);
v_mean(:,12,:)=v_mean(1,1,:);
v_mean(:,13,:)=v_mean(1,1,:);
v_mean(:,14,:)=v_mean(1,1,:);
v_mean(:,15,:)=v_mean(1,1,:);
v_mean(:,16,:)=v_mean(1,1,:);
v_mean(:,17,:)=v_mean(1,1,:);
v_mean(:,18,:)=v_mean(1,1,:);
v_mean(:,19,:)=v_mean(1,1,:);
v_mean(:,20,:)=v_mean(1,1,:);

v_mean(:,21,:)=v_mean(1,1,:);
v_mean(:,22,:)=v_mean(1,1,:);
v_mean(:,23,:)=v_mean(1,1,:);
v_mean(:,24,:)=v_mean(1,1,:);
v_mean(:,25,:)=v_mean(1,1,:);
v_mean(:,26,:)=v_mean(1,1,:);
v_mean(:,27,:)=v_mean(1,1,:);
v_mean(:,28,:)=v_mean(1,1,:);
v_mean(:,29,:)=v_mean(1,1,:);
v_mean(:,30,:)=v_mean(1,1,:);

v_mean(:,31,:)=v_mean(1,1,:);
v_mean(:,32,:)=v_mean(1,1,:);


v_mean=repmat(v_mean,[N 1 1]);

U=Uf+v_mean;        %Sommation of the fluctuation velocity and the mean
velocity in matrix U
U(:,:,32)=0;        % U(:,:,94)=0;This term in needed for the active
damping system ?? on even number at top

F=1/2*Are*Ch*rho*(U).^2; % turrning velocities on area into forces


%---------------------------------------------------------------------------
% Moments code
```

```
%--------------------------------------------------------------------------


M=zeros(size(F));   % making Moment matrix (M)the same size as F matrix


for n=1:1:36         % Reading 32 half heights of 1,5 meters into 14

           % make 32 a variable
    for p=1:1:32          % horizontal distance is divided in 32 pieces Yr

    if p<16
        M(:,p,n)=F(:,p,n)*(-1)*((B/2-(p-0.5)*2.5)-0.405);       % B/2-over-
Yrd;
    else
        M(:,p,n)=F(:,p,n)*(1)*((B/2+((p-16.5)*2.5)-0.405));

    end
    end

end



M=sum(M,2);              % Summation of the moments of each row in the
matrix. 6 horizontal places
M=squeeze(M);            % remove the singleton variable.


m(1:1:N,1:1:4)=2*M(1:1:N,1:1:4);     % bottom 2 nodes

m(1:1:N,1)=0;                        %half of bottoms node            %
kelder

for n=5:1:19                         % Summating the system 90 half heights
of 1,5 meters into 45
    m(:,n)=M(:,2*n-3)+M(:,2*n-2);    % into 45 spaces of 3 meters
    m(:,n)=M(:,2*n-3)+M(:,2*n-2);    % into 45 spaces of 3 meters
end
M=m;

%--------------------------------------------------------------------------
%Ending of moment code
%--------------------------------------------------------------------------


F=sum(F,2);              % Summation of the forces of each row in the
matrix. 6 horizontal places
F=squeeze(F);            % remove the singleton variable.

f(1:1:N,1:1:4)=2*F(1:1:N,1:1:4);     % bottom 2 nodes

f(1:1:N,1:1)=0;          %Firstnode

for n=5:1:19
    f(:,n)=F(:,2*n-3)+F(:,2*n-2);
```

```
end
F=f;


t=[deltat:deltat:(N*deltat)]';


FvoorWoudenberg=sum(F,2);
F_woud=(FvoorWoudenberg(1))/L;
```

Inputvalues.m

```
% Input values for simulation of a turbulent wind field
%
% INPUT:
%    yr, zr: specification of coordinates on the facade of the structure
%    v_10: mean wind velocity at 10 m above the surface of the earth (m/s)
%    sigma: standard deviation of the fluctuating part of the wind speed
(m/s)
%    N: number of time points (including zero); N must be a power of 2
%    deltat: time step (s)
%    fmax: maximum frequentie spectrum (Hz)
% OUTPUT:
%    UC: constrained turbulent wind velocities (m/s)
%
% Thesis: Torsion motions of high-rise buildings due to wind loading
% Author: H.A.O. Richardson (Anthony)



yr=1.655:2.5:80.81;
zr=(10.225:1.325:51.3);
v_10=21.45; %(Urban 2)

U=v_10;

sigma=2.44;
sigma_v=sigma;

N=8192;
deltat=.1;
fmax=5;
timestep=0.1;

save matrices yr zr U sigma_v N deltat timestep fmax -append
```

specest_test_run.m

```
% Checking  the Spectral density function
%
% Thesis: Torsion motions of high-rise buildings due to wind loading
% Author : H.A.o.Richardson (Anthony)

clear
close all

% Loading the generated wind velocities

load UC1

% Selecting a node and generating a time vector

u=UC(:,1,1);
t=(1:1:8192)*0.1;

% Plotting figure time versus the generated wind velocities

figure
plot(t,u);

% Calling the program specest.m (different ranges to choose from)

[Pxx,freq] = specest(u,1024,10);
[Pxx,freq] = specest(u,512,10);
[Pxx,freq] = specest(u,256,10);


% Plotting output specest

figure
plot(freq,Pxx);

figure
semilogy(freq,Pxx);

% Making a frequency time line and inputting the inputted values for
windspeed at hub height and standard deviation of wind speed in the
autopower spectrum

f=(0.001:0.001:5);
S=autopow(f,21.45,2.44);

% Plotting frequency versus autopower spectrum

figure
semilogy(f,S);

% Plotting frequency versus autopower spectrum and generated wind
velocities

figure
semilogy(f,S,freq(2:end),Pxx(2:end));
```

Appendix 7 Matlab code Voorhof after renovation (time domain analysis)

The matlab code which is the same as the Juffertoren and Voorhof before renovation is not given.

Below the Matlab code is given for the Voorhof after renovation for return period of one year, the only difference with the Matlab code for return period of 12.5 years is that the wind parameters for Harris and Deaves (9.5) are taken and inputted in Matlab file: A_File_to_run_programs.m and $v_{10} = 15.79 m/s$ and $\sigma_v = 6.65 m/s$ (9.6) are inputted in Matlab file: Inputvalues.m

A_File_to_run_programs.m

```
% Program to run the different modules
%
% Thesis: Torsion motions of high-rise buildings due to wind loading.
% Author: H.A.O.Richardson (Anthony)
%
%-------------------------------------------------------------------------
%


tic
disp('-----------------------------------------------------')
disp('Thesis: Torsion motions of high-rise buildings')
disp('-----------------------------------------------------')

% Clear memory

clear
clear all
clc
close all

% Start timer

tic


%-------------------------------------------------------------------------
% Building input
%-------------------------------------------------------------------------


disp('-Reading problem data')

%inputting_variables_of_the_buidling_construction

save matrices
```

```
B_Variabels_Wallplaces;

% Determining of the mass, bendingstiffness polarmoments and
% torsionalstiffness of the cross section.

C_Voorhof_after_renovation


% Determining maximum wind profile

u_star=3.836;        % Friction velocity
kappa=0.4;           % Terrain Roughness
d=3.5;               % average height of buildings (m)
z_0=0.7;             % Terrain Roughness according NEN 6702
step_z=1.325;        % Half of floor length


% DETERMING V_10

step_z=1.0;

[v_mean]=velocitymeanlogmodified(L,u_star,kappa,d,z_0,step_z);

Vhub=v_mean(1,2);


% meanwind velocity according to Harris and Deaves

step_z=1.325;

[v_mean]=velocitymeanlogmodified(L,u_star,kappa,d,z_0,step_z); %CHANGE


vh=max(v_mean);

Iz_top=(u_star/kappa)/vh;
Iz_10=(u_star/kappa)/Vhub;

save matrices vh Vhub v_mean Iz_top Iz_10 -append


%-------------------------------------------------------------------------
% Simulink input
%-------------------------------------------------------------------------

disp('-Input Simulink data')



[k,K_ben,M_ben,omega_eig_ben,a_ben,Cd_ben,phsi_ben,ktor,K_tor,M_tor,omega_e
ig_tor,a_tor,Cd_tor,Cd,E_ben]=D_Stiffness_matrix(v,EI,l,Cd,massfloor,mass1,
mass19,G,Jtot,Ip,Ip1,Ip19,xi1,xi2);


save matrices K_ben M_ben omega_eig_ben Cd_ben a_ben v k EI E_ben -append;
```

```
save matrices K_tor M_tor omega_eig_tor Cd_tor a_tor ktor -append;


%------------------------------------------------------------------------
% Determining of space state formulations for simulink
%------------------------------------------------------------------------

f=1350000;% force on node is

l_crossc=B;

[A_ben,B_ben,C_ben,D_ben,fe,fe_tor,A_tor,B_tor,C_tor,D_tor]=E_Space_state_f
ormulation(f,v,K_ben,M_ben,Cd_ben,omega_eig_ben,EI,L,l_crossc,K_tor,M_tor,C
d_tor,omega_eig_tor,l,G,Jtot);

save matrices A_ben B_ben C_ben D_ben v -append;

save matrices fe fe_tor A_tor B_tor C_tor D_tor -append;


%-------------------- -----------------------------------------------------
-
% Fluctuatiing wind (wind generator)
%------------------------------------------------------------------------

%{
load matrices

uiimport UC1.mat %loading the fluctuating velocities %undo

%}

Inputvalues;


tic
[t,UC]=wind0(yr,zr,U,sigma_v,N,deltat,fmax);
toc

save ('UC')
save ('UC1','UC')  % orignal modify
%{
%}

%Are=2.5*1.325;  % area of one node which is loaded by wind

Are=(yr(2)-yr(1))*(zr(2)-zr(1));% area of one node which is loaded by wind

%-------------------- -----------------------------------------------------
% From wind velocities to forces and moments
%------------------------------------------------------------------------

[F,M,t,FvoorWoudenberg,F_woud]=F_Force_moments_19_DOF(v_mean,N,Ch,Are,rho,B
,deltat,L);
```

```
save ('F','F','t') % Saving moment of each time step
save ('M','M','t') % Saving moment of each time step


%---------------------------------------------------------------------------
% Simulink Run
%---------------------------------------------------------------------------

disp('-Run Simulink ')


%sim('G_Simulink_run_file');


%---------------------------------------------------------------------------
% Getting the maximums of bending, torsion and bending and torsion
%---------------------------------------------------------------------------

%H_Max_acceleration_values_and_graphs_output


disp( '-Finished analysis. Time:' )
toc
```

B_Variables_Wall_Places.m

```
% Inputting varibles of the construction
%
% Thesis: Torsion motions of high-rise buildings due to wind loading.
% Author: H.A.O.Richardson (Anthony)
%
%--------------------------------------------------------------------------


% The outer dimensions of the total building

    L=51.3;                % [m] length of the building
    B=80.81;              % [m] width of the building
    H=14.2;              % [m] height of the building
    rhoconcrete= 2400;    % [kg/m3]specific density concrete
    rho=1.29;             % [kg/m3]specific density air
    Ch=1.2;               % [-] Thrust and Suction shape factor

    h=2.65;               % [m] height of story of the building
    l=h;                  % [m] lenght of one element

    hfl=0.15;             % [m] thickness of floor of the building

    hwall=h-hfl;          % [m] height of the walls of the building

    nbfloors=L/h;         % [-] amount of floors of the building

    %Pvb=70;               % [kg/m2] weight of load on the floor (0,7kN/m2)
    mvb=58.3; % 22/02      % [kg/m2] weight of load on the floor (0,7kN/m2)

    vu=0.2;                % [-] Poisson ratio

    %----------------------------------------------------------------------
    rhosteel=7900;         %   Mass denisity of steel             [kg/m3]
    %----------------------------------------------------------------------

        %qfacade=120;          % [kg/m2] weight of facade elements per m2
(1,2kN/m2)
                          % Limitation to the benchmark Skyscraper page 111)

        %E=3e10;               % [N/m2] Youngsmodulus of the concrete C35/B65

     %---------------------------------------------------------------------
-

    g=9.81;               % [m/s2] Gravitational acceleration

 %-------------------------------------------------------------------------

    v= 19;                % number of elements

    dof= 2;               % degrees of freedom per node

 % Damping ratios
```

```
    xi1=0.0146;           % damping ratio of the first eigenmode
    xi2=0.0146;           % damping ratio of the first eigenmode
    tend=10;            % duration of the simulink simulation time
    timestep=5e-5;      % fixed timestep in simulink
    deltat=0.1;         % delta t


% nwallseg is the amount wall segments that the walls of the cross section
is
% divided into.

    nwallsegm=16;

% Declaring matrix for the wall segments.

    P=zeros(nwallsegm,4);

% The cross-section is divided into 16 wall segments for calculation.

%  The origin of the cross sectiion of the building is taken in the left
%  bottom corner.

%    Length x         Length y         Distance x midpoint to  Distance y
midpoint to
%                                           Startpoint O
Startpoint O

P(1,1)=15.440;  P(1,2)=0.500;     P(1,3)=15.440/2;
P(1,4)=26.340/2-0.5/2;
P(2,1)=6.920;   P(2,2)=0.600;     P(2,3)=6.920/2;
P(2,4)=0.5+4.7+0.6+14.74+0.6/2;
P(3,1)=1.000;   P(3,2)=0.300;     P(3,3)=6.620+0.3+1.1+1.0/2;
P(3,4)=0.5+4.7+0.6+14.74+0.3/2;
P(4,1)=0.300;   P(4,2)=3.720;     P(4,3)=6.620+0.3+1.1+1.0+0.3/2;
P(4,4)=0.5+4.7+0.6+14.74+0.3-2.72/2;
P(5,1)=6.120;   P(5,2)=0.300;     P(5,3)=15.44-6.120/2;
P(5,4)=0.5+4.7+0.6+14.74-2.1-0.3/2;
P(6,1)=3.100;   P(6,2)=0.300;   P(6,3)=15.44-3.02-0.3-3.1/2;
P(6,4)=0.5+4.7+0.6+14.74-2.1-0.3-1.02-0.3/2;
P(7,1)=3.100;   P(7,2)=0.200;   P(7,3)=15.44-3.02-0.3-3.1/2;
P(7,4)=0.5+4.7+0.6+14.74-2.1-0.3-1.02-0.3-2.3-0.2/2;
P(8,1)=0.300;   P(8,2)=14.740;  P(8,3)=6.620+0.3/2;
P(8,4)=0.5+4.7+0.6+14.74/2;
P(9,1)=0.300;   P(9,2)=7.900;   P(9,3)=15.44-3.02-0.3/2;
P(9,4)=0.5+4.7+0.6+14.74/2;
P(10,1)=3.100;  P(10,2)=0.200;  P(10,3)=15.44-3.02-0.3-3.1/2;
P(10,4)=0.5+4.7+0.6+2.1+0.3+1.02+0.3+2.3+0.2/2;

P(11,1)=3.100;  P(11,2)=0.300;  P(11,3)=15.44-3.02-0.3-3.1/2;
P(11,4)=0.5+4.7+0.6+2.1+0.3+1.02+0.3/2;
P(12,1)=6.120;  P(12,2)=0.300;  P(12,3)=15.44-6.120/2;
P(12,4)=0.5+4.7+0.6+2.1+0.3/2;
P(13,1)=0.300;  P(13,2)=3.720;  P(13,3)=6.620+0.3+1.1+1.0+0.3/2;
P(13,4)=0.5+5.0+2.72/2;
P(14,1)=6.920;  P(14,2)=0.600;  P(14,3)=6.920/2;
P(14,4)=0.5+4.7+0.6/2;
P(15,1)=1.000;  P(15,2)=0.300;  P(15,3)=6.620+0.3+1.1+1.0/2;
P(15,4)=0.5+5.0+0.3/2;
```

```
P(16,1)=15.440; P(16,2)=0.500;  P(16,3)=15.440/2;
P(16,4)=0.5/2;

save matrices L B H rhoconcrete rho Ch h l hwall rhosteel g nbfloors hfl
mvb v xi1 xi2 tend timestep deltat nwallsegm  P -append
```

C_Voorhof_after_renovation.m

```
% Determining stiffness and torsion matrix for voorhof
%
% Thesis: Torsion motions of high-rise buildings due to wind loading
% Author: H.A.O. Richardson (Anthony)
%
%INPUT

   %v=19;                %   number of elements               [-]
   %l=2.65;              %   lenght of element                [m]
   %mvb=58.3;            %   variable floor mass              [kg/m2]
   %rhosteel=7900;       %   Mass denisity of steel           [kg/m3]

   %------------------------------------------------------------------------
   %Bending before renovation
   %------------------------------------------------------------------------

        %EIb=2.53E12; %(1)    %   bending stiffness
        EIb=2.085E12; %(2)    %   bending stiffness

        %{
        EI=2.53E12;           %   bending stiffness
        massfloor=607944;     %   mass of a floor   maas buidling
10.943E6/19  page 129b
        mass19=303972;        %   mass of top floor
        %}

        % Tekst f building is 0.641 to 0.750 HZ Page 139
        % Additional strucural stiffness (walls and steel frame) p.153
        % Strucural stiffness building (walls, steel frame, non structural
elements) p.153


   %------------------------------------------------------------------------
   %Bending After renovation
   %------------------------------------------------------------------------



        %EI=5.17E12; %(1)   %   bending stiffness  p.155
        EI=4.26E12;  %(2)   %   bending stiffness  modified

        %massfloor=636421;%(1) %   mass of a floor   maas buidling
12.092E6/19  page 130b[m]

        %------------------------
        % Original values mass
        %-------------------------

        %{
        massfloor=(12.092E6-(12.092E6/(19*2)))/18;
        mass19=12.092E6/(19*2); % Weight of top floor is approximated half
of other floors
        %}
```

```
%-----------------------------------------------------------------------
% estimating Mass of 19 th floor properly
%-----------------------------------------------------------------------

% Appendix F page 6 % M.Pols

    %mass steel                = 0.650E6
    %mass floor                = 7.672E6
    %mass partition walls      = 1.469E6
    %mass facade               = 1.134E6
    %stabilty walls            = 1.165E6



    massfloor=642499;%(2)  %   mass of a floor maas buidling
modified
    floormasstot= 12.090E6;
    floormass19= 7.672E6+0.5*(0.650E6+1.469E6+1.134E6+1.165E6);
    verh= floormass19/floormasstot; %0.8173 without foundation

    %verh=floormass48/(floormasstot+1.451E6);%0.75 with foundation

    mass19=massfloor*verh;


%-----------------------------------------------------------------------
% E and Ixx and Izz structural walls   p.153 (Pols)(After renovation)
%-----------------------------------------------------------------------


EIw=2.39E12;              %   [N/m2] Concrete B22.5 pol
Ew=27875*1000*1000;  %   [N/m2] is 27875[N/mm2] Concrete B22.5
Ixxw=EIw/Ew;
bw=0.25;
hw=10.1;

% Calculated Second Moment of Inertia Concrete walls (Richardson)

Ixxwm=4*(1/12)*bw*hw^3;
Iyywm=4*(1/12)*hw*bw^3+2*hw*bw*8^2+2*hw*bw*40^2;

%-----------------------------------------------------------------------
%Ixx structural steel (Pols)(Before Renovation)
%-----------------------------------------------------------------------


Ess=2.1E5*1000*1000;    %[N/m2]  2.1E5 N/mm2 modulus of elasticity
Iss=EIb/Ess;             %HEA 260

% determining Issy

Y1=(2*16130+2*2530)/(1000*1000); % mm2area of cross-section profile
Y1
Y2=(2*11250+2*19160)/(1000*1000); %area of cross-section profile Y1
Y3=(2*24020+2*2530)/(1000*1000); %area of cross-section profile Y1
Y4=(2*24020+2*2530)/(1000*1000); %area of cross-section profile Y1
```

```
Issy=2*(Y3*4^2+Y3*8^2+Y3*12^2+Y3*16^2+Y3*20^2)+((Y3+Y4)*24^2)+((Y2+Y4)*28^2
)+((Y3+Y4)*32^2)+2*Y2*36^2+2*Y1*40^2;

   %-------------------------------------------------------------------------
   % Torsion componets Before renonvation
   %-------------------------------------------------------------------------

        % Shearmodulus of the structural steel .( Ridigidy modulus)

        %Ess=2.1E5*1000*1000;      %[N/m2]  2.1E5 N/mm2 modulus of
elasticity
        Gss=Ess/(2*(1+vu));       %[N/m2]
        %Gss=79.3E9;            %   Shear modulus struc steel  79.3
Gpa[N/mm2] website

        % Shearmodulus of the concrete.( Ridigidy modulus)

        % Ew=2.1E5*1000*1000;      %[N/m2]  2.1E5 N/mm2 modulus of
elasticity
        Gw=Ew/(2*(1+vu));       %[N/m2]


        % Shearmodulus of the buidling.( Ridigidy modulus)


        %G=Gss;          %   Shear modulus Steel (before renovation)
[N/m2]
        G=Gss+Gw;      %   Shear modulus Concrete (after renovation) [N/m2]

 %---------------------------------------------------------------------------
------------
 % Torsion constant walls + structural steel
 %---------------------------------------------------------------------------
------------

           %torsion constant of concrete walls

           Jw=((1/3)*bw^3*hw)*4;  % [m4]

       %torsion constant of structural steel

            % J van 1 profile is 54.2 cm4
            Jss=8*54.2/(10000*10000);  % [m4]
            Jss=12000000*Jss;
        %torsion constant of concrete walls

            %Jtot=Jss;         %   Torsion constant     (before)   [m4]
            %Jtot=Jw+Jss;      %   Torsion constant     (after)    [m4]
            %Jtot=8*Jw;        %    Arbituary chosen number


        % The torsional stiffness of one floor of the structure.

            %TorSTif=G*Jtot;
```

```
            %TorSTif=1.3046E12; %(1) % determined in Matrix
            TorSTif=3.187E12;   %(2) % modified
            Jtot=TorSTif/G;

  %-------------------------------------------------------------------------
-----------------------
  %Polar moment of inertia (Before renovation)
  %-------------------------------------------------------------------------

          %Strucural steel (before renovation)

            %Ixtot=Iss;                       % HEA 260
            %Iytot=Issy;

        %Strucural steel with concrete walls (after renovation)

            Ixtot=Iss+Ixxwm;                % HEA 260
            Iytot=Issy+Iyywm;

        % Polar inertia of the walls Ix and Iy of a floor

            Ipwall=Ixtot+Iytot;

        % Polar inertia of the floor Ix and Iy

            Ipfloor=((1/12)*H*B^3)+((1/12)*B*H^3);

        %-----------------------
        %Before Renovation
        %-----------------------

        % Polar inertia of the floor segments node 1 to node 18

            %Ip=Ipwall*h*rhosteel+Ipfloor*(hfl*rhoconcrete+mvb);

        % Polar inertia of the top floor of the building

            %Ip19=Ipwall*(h/2)*rhosteel+Ipfloor*(hfl*rhoconcrete+mvb);

        %-----------------------
        %After Renovation
        %-----------------------

        % Polar inertia of the floor segments node 2 to node 19

            %Ip=Iss+Ixxwm+Issy+Iyywm;
            %Ip=Ipwall*h*rhoconcrete+Ipfloor*(hfl*rhoconcrete+mvb);

Ip=(Iss+Issy)*h*rhosteel+(Ixxwm+Iyywm)*h*rhoconcrete+Ipfloor*(hfl*rhoconcre
te+mvb);

        % Polar inertia of the top floor of the building


Ip19=(Iss+Issy)*(h/2)*rhosteel+(Ixxwm+Iyywm)*(h/2)*rhoconcrete+Ipfloor*(hfl
*rhoconcrete+mvb);
```

```
        %----------------------------
        %Arbituarly chosen values
        %----------------------------


   %-----------------------------------------------------------------------
   % Damping Ratios
   %-----------------------------------------------------------------------


        %xi1=0.0146; %Appendix I  %   damping ratio of first eigenmode
[-]
        %xi2=0.0146; %M.Pols      %   damping ratio of second eigenmode
[-]

        mass1=massfloor;
        Ip1=Ip;

  %------------------------------------------------------------------------
  % new added code
  %-----------------------------------------------------------------------


        % Total mass of the buidling

        % mass building of 48 floors walls + top floor

        mass=mass1+(v-2)*massfloor+mass19;     % edit 10-07-2021


% Total weigth of the building

        Q_gebouw=mass*g;           % weight building

% Spefic density of building             [kg/m3]

        rhob=mass/(B*H*L);

% mass per meter building height         [kg/m1]

        rho11=mass/(L);

% mass per area             [kg/m2]

        rho_m2=mass/(B*H);


save matrices EIb EI massfloor mass19 EIw Ixxw Ixxwm Iyywm Ess Ess Issy Gss
Gw G Jw TorSTif Jtot Ixtot Iytot Ipwall Ipfloor Ip Ip19 mass  -appen
```

Appendix 8 The Voorhof before renovation with calibration output of the 100 simulations for return period of one year

	a_ben	a_tor	a_ben_tor		sigma_a_ben	sigma_a_tor	sigma_a_ben_tor
1	0.06968581	0.2323952	0.2425436		0.02073865	0.06999912	0.073308746
2	0.09459908	0.3324859	0.3363541		0.027142111	0.07454332	0.075802652
3	0.07981099	0.319223	0.3241027		0.027187926	0.08776767	0.086216955
4	0.08852405	0.2386805	0.2732928		0.02529268	0.06354616	0.063661003
5	0.10780719	0.2161534	0.2432105		0.028723298	0.0691151	0.068706984
6	0.10393684	0.2722841	0.3006352		0.033922993	0.08044134	0.084569951
7	0.13983604	0.3107522	0.3275735		0.030809479	0.07559226	0.07568096
8	0.11266355	0.2910735	0.3173023		0.028387038	0.07251213	0.072839736
9	0.10118157	0.2694308	0.2850848		0.028625121	0.06897747	0.071688119
10	0.094874	0.2707378	0.3083641		0.027963864	0.07755586	0.07657867
11	0.09103684	0.4306071	0.4846919		0.026045572	0.07736404	0.076885403
12	0.1192959	0.3678819	0.3972573		0.028212741	0.08383017	0.082239681
13	0.10037784	0.2339367	0.2736085		0.027185709	0.06804569	0.067141071
14	0.09650082	0.2271779	0.2656233		0.029410598	0.07123267	0.069525583
15	0.10526886	0.2346658	0.2880861		0.031548737	0.07795531	0.074358185
16	0.08657789	0.2325341	0.2744868		0.02724236	0.07279285	0.073425833
17	0.11537893	0.2451352	0.3466249		0.033849215	0.07160541	0.070007816
18	0.11905488	0.2312176	0.2878769		0.027666468	0.07070848	0.072233234
19	0.09318995	0.2418974	0.2804626		0.02809755	0.06859543	0.063949594
20	0.10733049	0.2521839	0.2895513		0.031121525	0.067547	0.070439647
21	0.09586113	0.2451115	0.2811913		0.03141114	0.07305372	0.075522359
22	0.09422124	0.2265356	0.2635227		0.027370288	0.07430079	0.076066951
23	0.10157598	0.2246911	0.2598122		0.02789948	0.07375591	0.073268751
24	0.11371913	0.2483587	0.286389		0.029893843	0.06998979	0.067275778
25	0.08609646	0.3200284	0.336961		0.025601263	0.07542853	0.075973793
26	0.11820078	0.2420689	0.2877038		0.027224473	0.07450878	0.073830983
27	0.08709277	0.2638212	0.2796252		0.027543432	0.08106	0.080698627
28	0.14168451	0.231262	0.2448968		0.03149429	0.07406183	0.077670899
29	0.08330143	0.2337006	0.2787854		0.025219417	0.07723349	0.075716244
30	0.08021751	0.2170814	0.2520742		0.023898409	0.07047064	0.070266123
31	0.08588611	0.3210228	0.3652437		0.02579652	0.0858468	0.084254363
32	0.09998934	0.272798	0.2725435		0.029668637	0.07291383	0.07705215
33	0.10289949	0.2616814	0.3183038		0.027851542	0.0856182	0.084307902
34	0.10610138	0.2852645	0.3140343		0.033193161	0.07980131	0.08163677
35	0.08840539	0.2084205	0.254627		0.025457889	0.0742258	0.07430919
36	0.09243891	0.2222352	0.2810523		0.027631785	0.06747276	0.069142719
37	0.12466199	0.3424759	0.3807099		0.038632567	0.09088218	0.094380345
38	0.09872472	0.269861	0.3366244		0.029828716	0.07001902	0.069683742
39	0.08354948	0.2167586	0.2676394		0.025113691	0.06962668	0.065923367
40	0.10392215	0.2797081	0.3320671		0.028126512	0.07911515	0.076944732
41	0.08686265	0.2508106	0.3235119		0.026257703	0.07027384	0.069922742
42	0.07195468	0.2427826	0.2880514		0.023674742	0.06952888	0.067695942
43	0.09215262	0.2260216	0.2656651		0.029505594	0.07200773	0.072720948
44	0.09873573	0.251212	0.3167027		0.030004219	0.07201672	0.072308785
45	0.08631015	0.2804093	0.3114786		0.027549207	0.0689346	0.064698772
46	0.08635972	0.2672592	0.2867134		0.025917354	0.09075052	0.090831298
47	0.1095098	0.2121968	0.2644319		0.032696609	0.06738026	0.065425947
48	0.11374896	0.2173716	0.2559811		0.028621996	0.07053527	0.074426434
49	0.09264699	0.3025396	0.308538		0.025843507	0.07944526	0.082390185
50	0.09703506	0.2700021	0.2906911		0.028619461	0.07868107	0.07888268

51	0.10873508	0.2072597	0.2614061		0.031599256	0.0659975	0.064526159
52	0.0923521	0.2440859	0.2755421		0.024305516	0.0794656	0.078577627
53	0.09445844	0.231962	0.2820508		0.026853739	0.07149167	0.073339116
54	0.09421834	0.3422295	0.3850462		0.026924778	0.0826733	0.083165378
55	0.09778426	0.2955212	0.3097031		0.028865149	0.07258231	0.072972648
56	0.08390384	0.2362627	0.2716683		0.026761413	0.07123976	0.068628016
57	0.09782747	0.2788395	0.2992525		0.029860589	0.0812375	0.082455696
58	0.13366137	0.2831212	0.3426366		0.030759259	0.0756661	0.076699284
59	0.10567685	0.2423624	0.2704474		0.026326732	0.06760412	0.072412651
60	0.08798872	0.2118911	0.2537081		0.027425236	0.06315061	0.061990113
61	0.07904974	0.2814664	0.3283452		0.026636855	0.0748814	0.071497644
62	0.10573049	0.2927015	0.3119549		0.032826907	0.08291921	0.083417278
63	0.08741304	0.3024827	0.3106739		0.025400672	0.07986983	0.07884547
64	0.11248963	0.2621553	0.3210056		0.030682309	0.06659212	0.070077987
65	0.07881942	0.270469	0.3484155		0.026831758	0.07242281	0.07224251
66	0.10318829	0.2460462	0.2806025		0.031540559	0.07417331	0.077465654
67	0.07481322	0.2455863	0.3141527		0.025999075	0.07047544	0.068163067
68	0.08150063	0.2563806	0.2766879		0.024163349	0.08013136	0.081246231
69	0.08712915	0.2746013	0.2777074		0.025078503	0.08088448	0.078169162
70	0.10507293	0.2682962	0.3208513		0.032095268	0.0856253	0.083670384
71	0.09451755	0.2337716	0.2895274		0.028013751	0.07162449	0.070207607
72	0.12155937	0.3392853	0.3275946		0.033036569	0.08093443	0.079103053
73	0.09736786	0.2721337	0.3362344		0.033092115	0.07420541	0.071937256
74	0.09847456	0.2781615	0.3131084		0.024613638	0.07212286	0.069200064
75	0.09139178	0.2338469	0.2836615		0.028068663	0.0727667	0.07052742
76	0.08482044	0.2453991	0.2841524		0.025374028	0.07457693	0.071919917
77	0.09906002	0.2634985	0.3318196		0.029588047	0.08350898	0.083478467
78	0.10582695	0.2212191	0.2712574		0.031142613	0.06920075	0.065285556
79	0.08307613	0.224204	0.2594861		0.026015836	0.07088928	0.071236395
80	0.11542022	0.2865435	0.3209713		0.029305174	0.07708191	0.078923558
81	0.10305177	0.2571853	0.3481012		0.034307146	0.08126231	0.080446386
82	0.10427048	0.2248203	0.265095		0.029796679	0.06936201	0.073322147
83	0.08883197	0.2867356	0.2971252		0.02786614	0.07567606	0.077415261
84	0.09806447	0.2217529	0.2433477		0.032649971	0.06511947	0.065380014
85	0.15201754	0.2793504	0.3045406		0.034961936	0.07841229	0.082474226
86	0.09095487	0.2607863	0.2364978		0.028472871	0.07164584	0.072057077
87	0.09392804	0.2808446	0.2736179		0.026417386	0.06772404	0.070060998
88	0.09687944	0.2302677	0.2767709		0.02613545	0.07984792	0.079293729
89	0.09061728	0.2302001	0.2509917		0.027921541	0.06643955	0.067576221
90	0.11660052	0.2591182	0.3125904		0.029627042	0.07474458	0.075528149
91	0.10656865	0.214868	0.2556724		0.03004593	0.07198833	0.071358262
92	0.08485424	0.2959943	0.3702379		0.028127996	0.08208029	0.082476402
93	0.08046477	0.2342171	0.2630544		0.024627958	0.07340187	0.07084539
94	0.08716646	0.2976506	0.3044983		0.028733526	0.07463346	0.07320379
95	0.09455275	0.2098555	0.249255		0.031035693	0.06627594	0.068296085
96	0.10655824	0.213618	0.2301995		0.030327627	0.06583291	0.070223143
97	0.10979627	0.3018169	0.3230461		0.027651079	0.07856811	0.079641814
98	0.09345996	0.3027426	0.285258		0.028791126	0.07877702	0.079269441
99	0.08193727	0.2365753	0.2532		0.02532765	0.07676286	0.077607367
100	0.10534629	0.2523827	0.3239992		0.030169527	0.07875074	0.079003235

Appendix 9 The Voorhof after renovation with calibration output of the 100 simulations for return period of one year

	a_ben	a_tor	a_ben_tor		sigma_a_ben	sigma_a_tor	sigma_a_ben_tor
1	0.05976313	0.1513281	0.1579253		0.015979828	0.04487348	0.047365481
2	0.05652056	0.1448139	0.1418168		0.01610822	0.04536202	0.047454826
3	0.05136874	0.1652138	0.1685007		0.014352097	0.04573289	0.048301589
4	0.05146955	0.1208907	0.1291855		0.013923399	0.03936891	0.042159703
5	0.05020175	0.1591799	0.1773519		0.016325876	0.0461672	0.048715556
6	0.05170829	0.14767	0.1474878		0.014938506	0.04305784	0.045641543
7	0.04974838	0.1435849	0.1569219		0.015119611	0.04073288	0.043116046
8	0.05842028	0.156474	0.1678372		0.015727833	0.03909816	0.042111825
9	0.05780921	0.1419809	0.1701099		0.015617824	0.04313251	0.045433173
10	0.04743413	0.1644738	0.1720629		0.013692645	0.04665831	0.048474041
11	0.05774946	0.138435	0.1403325		0.014230101	0.0416503	0.043700851
12	0.05336552	0.1733502	0.2009804		0.015702391	0.04031342	0.043635609
13	0.05925306	0.1589006	0.1792989		0.015705833	0.04037993	0.04282512
14	0.04658816	0.1376759	0.1549639		0.01498649	0.04167663	0.043995902
15	0.04380667	0.1491126	0.152079		0.013359213	0.04570731	0.046507203
16	0.04723745	0.1649446	0.1841781		0.01338408	0.04614615	0.047394367
17	0.05651648	0.1612043	0.1828555		0.016104617	0.0432206	0.045646527
18	0.04410889	0.17573	0.1989676		0.013750655	0.04641823	0.047596894
19	0.05076669	0.1588492	0.1603332		0.013619825	0.04590617	0.046592558
20	0.04810722	0.160807	0.1766668		0.0149198	0.05052763	0.052851411
21	0.05127209	0.1340369	0.1463355		0.015639961	0.04003163	0.042276237
22	0.04629481	0.1328333	0.1500292		0.01403408	0.04339174	0.045044851
23	0.05784195	0.1592646	0.1698472		0.017027428	0.04565665	0.048258154
24	0.04573132	0.1469331	0.1695548		0.01318519	0.04306062	0.044281785
25	0.05553655	0.138333	0.1668156		0.017001229	0.04349419	0.046333235
26	0.0578549	0.1442753	0.1627096		0.017725141	0.04086919	0.043875677
27	0.05812749	0.1511259	0.1594743		0.014505834	0.04553826	0.046965834
28	0.04610464	0.1659812	0.182418		0.013865843	0.04346201	0.044839009
29	0.05561656	0.131993	0.1493497		0.016405747	0.04238426	0.044446187
30	0.05660978	0.1455675	0.1501849		0.015629025	0.03992041	0.042436553
31	0.05083463	0.1492727	0.1674445		0.015763018	0.04945467	0.051493028
32	0.04793023	0.1485064	0.1479009		0.015261762	0.04362482	0.045707947
33	0.04747013	0.18731	0.1961606		0.014707412	0.04592213	0.047568223
34	0.05265228	0.1451135	0.1612663		0.015881331	0.04849516	0.050926866
35	0.05571721	0.1455886	0.1619306		0.015532608	0.04446265	0.046316213
36	0.0525307	0.1556255	0.1802466		0.014495112	0.04518252	0.047230382
37	0.05119899	0.1522708	0.1566731		0.014178271	0.04607297	0.047517554
38	0.0543459	0.1444395	0.1704945		0.015179308	0.04195046	0.044226519
39	0.0454023	0.1781848	0.1779274		0.015842054	0.04836891	0.050822861
40	0.0448714	0.1365988	0.1514352		0.014365612	0.0395114	0.04116435
41	0.05746137	0.1515396	0.1683724		0.015413266	0.04046936	0.043642018
42	0.05715052	0.1546987	0.165354		0.017789997	0.04105911	0.044748767
43	0.05311228	0.1633059	0.181997		0.015226414	0.04493326	0.047262839
44	0.04797795	0.1636256	0.1963769		0.013733967	0.04613675	0.046921203
45	0.04898144	0.1421144	0.1564908		0.014898677	0.04037391	0.042710616
46	0.05908468	0.1903077	0.2005996		0.016285859	0.04803129	0.050152288
47	0.04450188	0.1710141	0.2080492		0.013018289	0.04109678	0.042436913
48	0.05402815	0.1381079	0.1608818		0.014039757	0.04237056	0.043995177
49	0.04829275	0.1626424	0.1892593		0.015113721	0.04546485	0.047416154
50	0.04892869	0.1374643	0.1525066		0.014341509	0.03893	0.040389363

51	0.04626648	0.1622078	0.1744252		0.015012276	0.04603069	0.047585064
52	0.05568904	0.1390987	0.1614398		0.015989496	0.04064272	0.04242614
53	0.04995129	0.1523983	0.1665253		0.015901319	0.0432281	0.046185205
54	0.05351534	0.1371881	0.1578774		0.016401646	0.04321826	0.046084335
55	0.05152323	0.1670306	0.1783497		0.016059036	0.04684048	0.049401808
56	0.06208938	0.1614985	0.164503		0.01740832	0.0437006	0.046513219
57	0.05832961	0.1500825	0.1554171		0.015309016	0.04079667	0.043141716
58	0.06076961	0.1429302	0.1578035		0.012998786	0.04414658	0.045445878
59	0.04762482	0.1769906	0.193227		0.015216546	0.04446804	0.046671188
60	0.05898548	0.1915251	0.2201322		0.015982533	0.04840857	0.050907472
61	0.05517837	0.1777094	0.1996463		0.014512543	0.045855	0.047203861
62	0.04826472	0.1850609	0.1842753		0.014835978	0.04342645	0.045622228
63	0.05517632	0.1277348	0.1492548		0.014442591	0.03992322	0.041933685
64	0.05796889	0.1717115	0.188073		0.0157031	0.04725519	0.049597022
65	0.05167892	0.162092	0.1749119		0.014801526	0.04567126	0.048291949
66	0.04571168	0.1695468	0.1791248		0.013555639	0.04227943	0.043738053
67	0.06502127	0.1456628	0.1642747		0.01673845	0.04292546	0.045292988
68	0.05296632	0.1595158	0.1714528		0.015891463	0.04641633	0.049125485
69	0.04820408	0.15865	0.1800312		0.014259737	0.04137286	0.042881078
70	0.05331303	0.1503899	0.1463386		0.01530971	0.04078819	0.043505915
71	0.05332907	0.1970937	0.217179		0.015382382	0.04500305	0.046755203
72	0.04603305	0.1563576	0.1922291		0.014849345	0.04337498	0.045625652
73	0.04721129	0.1613374	0.1683834		0.014162639	0.04468793	0.046734205
74	0.05041464	0.1436732	0.1679368		0.013591215	0.04502103	0.046330115
75	0.0574447	0.1634421	0.157613		0.014969693	0.04228537	0.044653705
76	0.0619812	0.1375816	0.1451502		0.016099118	0.03701516	0.040088012
77	0.04446912	0.1715323	0.158808		0.013739941	0.04558275	0.046679773
78	0.04273304	0.1297246	0.139635		0.013075981	0.04170955	0.043564785
79	0.05718429	0.1599348	0.1587893		0.015813166	0.04158941	0.043693702
80	0.05005585	0.1505995	0.1572004		0.014294989	0.04125553	0.043529218
81	0.04498865	0.1522213	0.1722845		0.012774149	0.04358175	0.044517048
82	0.0507932	0.1736167	0.1825878		0.014471183	0.04151254	0.042958293
83	0.05362913	0.1412566	0.1697407		0.015566066	0.04377198	0.046279072
84	0.05597482	0.1806593	0.2046209		0.01420719	0.04647772	0.04862704
85	0.05189925	0.1634807	0.1759104		0.015060425	0.03941479	0.041724603
86	0.06062021	0.1380548	0.1651255		0.016632908	0.03716163	0.040392267
87	0.04471026	0.1685392	0.1795652		0.01337834	0.04258251	0.044333356
88	0.05934646	0.1402346	0.1655277		0.016587615	0.04066268	0.043215022
89	0.04403393	0.1581042	0.160948		0.013490785	0.04449465	0.0460693
90	0.0589074	0.1496075	0.1670196		0.016291575	0.0473275	0.049962752
91	0.0439191	0.1557181	0.1800662		0.013009044	0.04421942	0.045381321
92	0.05129651	0.1834788	0.1854588		0.014702558	0.04392213	0.045820645
93	0.04434232	0.1554517	0.1617167		0.013761609	0.04252297	0.043981332
94	0.05282164	0.1337995	0.1491153		0.014494773	0.03777531	0.039675116
95	0.05290772	0.1512687	0.1554517		0.015246723	0.0448809	0.048016921
96	0.04933598	0.1542397	0.1667761		0.01463108	0.04480414	0.046378334
97	0.04915997	0.183186	0.2017828		0.014411971	0.04495599	0.04649084
98	0.05005978	0.1515308	0.161145		0.013481495	0.0451663	0.047418932
99	0.05702237	0.1645609	0.1768556		0.014929579	0.04382237	0.046138172
100	0.05569363	0.1373605	0.1654077		0.015555812	0.04267954	0.044781568

Appendix 10 The Voorhof before renovation with calibration output of the 50 simulations with wind speed 18.4 m/s.

	a_ben	a_tor	a_ben_tor		sigma_a_ber	sigma_a_tor	sigma_a_ben_tor
1	0.075951	0.275219	0.296667		0.02205787	0.077935512	0.08097891
2	0.116354	0.38499	0.496571		0.02934827	0.085387039	0.085768877
3	0.125639	0.308054	0.345731		0.02789903	0.075791855	0.075612101
4	0.111138	0.254214	0.302103		0.03083876	0.078047774	0.078449199
5	0.13691	0.31785	0.362222		0.03684749	0.089956313	0.094213544
6	0.11959	0.363527	0.382644		0.0318823	0.086364083	0.086363845
7	0.095157	0.284833	0.313466		0.0297067	0.079322538	0.079175701
8	0.109519	0.349301	0.371306		0.03083661	0.077325458	0.078821247
9	0.101143	0.260907	0.30009		0.02836518	0.086321166	0.084618535
10	0.091989	0.521309	0.587117		0.02773489	0.090298353	0.088893984
11	0.122772	0.456135	0.498462		0.03034366	0.094735008	0.093719553
12	0.119281	0.298151	0.351204		0.02897868	0.075369711	0.07446363
13	0.102315	0.27578	0.333013		0.03171778	0.081856675	0.080619329
14	0.131628	0.271551	0.322184		0.03306813	0.08702367	0.083158091
15	0.090834	0.28419	0.336033		0.02833336	0.080443273	0.082259205
16	0.12494	0.306281	0.407121		0.03752326	0.082171174	0.080811723
17	0.093891	0.284868	0.322766		0.02860962	0.080723526	0.081674337
18	0.107705	0.364787	0.419803		0.03000754	0.076276314	0.072462337
19	0.100615	0.273087	0.29411		0.03146495	0.075416824	0.077547837
20	0.101387	0.251612	0.304095		0.03339892	0.080225878	0.08336529
21	0.104952	0.265737	0.295904		0.0307203	0.086433092	0.088999204
22	0.088687	0.302337	0.318548		0.02904207	0.084514423	0.083163006
23	0.118102	0.236254	0.267206		0.03128528	0.079031666	0.076150198
24	0.099476	0.405285	0.424333		0.0266435	0.088783036	0.088487475
25	0.09527	0.267601	0.305551		0.02834139	0.082427408	0.082102513
26	0.099682	0.339449	0.377458		0.0293891	0.092854391	0.09160806
27	0.160978	0.267433	0.28012		0.03293008	0.081693593	0.084900834
28	0.091412	0.291127	0.355401		0.02688705	0.086791734	0.085830209
29	0.095799	0.26765	0.29821		0.02569651	0.078952674	0.078939933
30	0.09758	0.339032	0.387828		0.02763985	0.09570591	0.094503769
31	0.128575	0.285043	0.284602		0.03064045	0.080547468	0.084087427
32	0.116318	0.346442	0.393124		0.03037045	0.097044296	0.096385459
33	0.110485	0.329353	0.346056		0.03680632	0.09139412	0.092720692
34	0.086404	0.260392	0.285721		0.02655553	0.08497409	0.084473465
35	0.106755	0.258123	0.27755		0.02921664	0.074674957	0.07656011
36	0.14836	0.386234	0.434988		0.04179915	0.09926104	0.103167414
37	0.103544	0.33343	0.324843		0.03241928	0.079908988	0.080150382
38	0.088436	0.26879	0.32643		0.0279111	0.078296605	0.074591237
39	0.116763	0.300454	0.354985		0.03112015	0.088910909	0.087460097
40	0.09355	0.263213	0.323435		0.02674557	0.075950028	0.076950632
41	0.074032	0.288135	0.319766		0.02452914	0.079271229	0.077376487
42	0.100418	0.27619	0.320656		0.03067193	0.080754693	0.0821582
43	0.116427	0.308517	0.380815		0.03225737	0.080898442	0.082079481
44	0.097737	0.387174	0.421509		0.02968658	0.079645262	0.074606365
45	0.098265	0.326089	0.375677		0.02800039	0.104749963	0.103797592
46	0.118548	0.245881	0.298171		0.03441489	0.076457148	0.074591045
47	0.104248	0.326436	0.341187		0.03196974	0.087657022	0.090981399
48	0.093448	0.29009	0.353642		0.02573315	0.090830689	0.091803848
49	0.087994	0.253067	0.315519		0.02640019	0.078095689	0.077460038
50	0.113726	0.343226	0.403138		0.03137786	0.085235526	0.085226307
Average	0.1067	0.3089	0.3508		0.0301	0.0839	0.0839

Appendix 11 The Voorhof after renovation with calibration output of the 50 simulations with wind speed 9.2 m/s.

	max_a_ben	max_a_tor	max_a_ben_tor		sigma_v_a_ben	sigma_v_a_tor	sigma_v_a_ben_tor
1	0.00721456	0.0222206	0.027162306		0.00207776	0.00568695	0.006042697
2	0.00772237	0.0236591	0.022862488		0.00211326	0.006621345	0.006949389
3	0.00736908	0.0307778	0.028869059		0.001798732	0.005586459	0.005752898
4	0.00867452	0.0231883	0.024087504		0.002217215	0.005696801	0.006064925
5	0.00660858	0.0258441	0.028212485		0.001823923	0.006443393	0.006704333
6	0.00662554	0.0233309	0.025951907		0.001922658	0.006287209	0.006455571
7	0.0085662	0.0309846	0.030720525		0.001908064	0.005970272	0.006222889
8	0.00968254	0.0256271	0.025865787		0.001953385	0.006149118	0.006451267
9	0.00799978	0.0280257	0.025891687		0.001789693	0.00630833	0.006547832
10	0.00871002	0.0210739	0.025239678		0.001870554	0.005533957	0.005802478
11	0.00757419	0.022029	0.020946658		0.002061742	0.005964676	0.006387874
12	0.00787641	0.024902	0.027980679		0.002049076	0.006293811	0.006586584
13	0.00653486	0.0200402	0.020911764		0.002005823	0.005747416	0.006092743
14	0.00920319	0.0259208	0.031218609		0.001858194	0.006416712	0.006565894
15	0.00690278	0.0189183	0.021773334		0.00175652	0.005943998	0.006144334
16	0.00782178	0.0237568	0.023876732		0.002116888	0.005620063	0.006016481
17	0.00646338	0.0247695	0.029287303		0.001898632	0.006409668	0.006536027
18	0.007182	0.0313307	0.032814154		0.001805355	0.006150608	0.006307286
19	0.00704502	0.0207398	0.026178543		0.001956456	0.00601466	0.006343596
20	0.00780784	0.0256325	0.022985937		0.002091921	0.005956086	0.006233525
21	0.00619033	0.0232419	0.024689783		0.001805199	0.00557438	0.005811322
22	0.01037292	0.0279572	0.032049781		0.002314844	0.006240383	0.006584279
23	0.00625954	0.0280836	0.025265932		0.00176731	0.006277103	0.006455471
24	0.00828672	0.0269073	0.028242942		0.002079865	0.006644806	0.006889912
25	0.00755832	0.023153	0.025282217		0.00226188	0.006319953	0.006650617
26	0.00865531	0.0226225	0.024805918		0.00189692	0.005763315	0.005919474
27	0.00672448	0.0216023	0.02414147		0.0017408	0.005970644	0.00607484
28	0.00779923	0.01964	0.022387351		0.002080794	0.005297639	0.005572366
29	0.00786504	0.0265386	0.028761761		0.001955834	0.005907859	0.006170222
30	0.00769052	0.0204848	0.021948174		0.002135833	0.005765999	0.006044473
31	0.00624388	0.0203235	0.022754743		0.001905744	0.005620407	0.00591163
32	0.0083853	0.0227692	0.027462224		0.002062959	0.00566111	0.005865816
33	0.00760509	0.0309411	0.031441901		0.002115818	0.007033567	0.007274473
34	0.00682642	0.0250015	0.026678327		0.001960763	0.006305736	0.006473427
35	0.00659029	0.0261365	0.025831684		0.001863525	0.00581853	0.006070116
36	0.00661796	0.0232756	0.027428488		0.001808215	0.006368448	0.006594826
37	0.00716542	0.0234552	0.025307669		0.001845597	0.006086952	0.006261165
38	0.0074945	0.0234143	0.023974672		0.002068432	0.005783729	0.006042626
39	0.0060533	0.0206309	0.024728352		0.001893227	0.005794858	0.00601112
40	0.00837527	0.0225067	0.023451362		0.002036368	0.006137302	0.006509209
41	0.00895996	0.0240274	0.025237471		0.002209085	0.006095478	0.006436268
42	0.00703792	0.0213733	0.023993265		0.001945985	0.005357071	0.005702202
43	0.00604619	0.0198596	0.021103946		0.001715548	0.005680292	0.005846591
44	0.00578906	0.0231722	0.022178032		0.00191262	0.005712881	0.005972499
45	0.00772706	0.0236537	0.025251181		0.002247997	0.005934119	0.00625711
46	0.007007	0.022369	0.023901151		0.00172952	0.006194247	0.00635461
47	0.00837941	0.0236048	0.02997162		0.001994836	0.006364139	0.00656652
48	0.00862992	0.0308411	0.032800994		0.002241268	0.00689173	0.007178617
49	0.01022218	0.021016	0.022849959		0.002262717	0.00554361	0.005913361
50	0.0082164	0.0252751	0.024620442		0.002200198	0.006308139	0.006675295
Average	0.0076	0.0241	0.0258		0.0020	0.0060	0.0063

Appendix 12 Formulas to determine the maximum acceptable acceleration for Student building "Voorhof" after renovation

In this appendix, the calculated acceleration out of matlab for the Student building "Voorhof" (bending, torsion and bending and torsion) are compared to design formulas.

Formula	Natural frequency		Along wind			Across wind		
	Hz	rad/s	Max bending acceleration m/s^2	Max torsional acceleration m/s^2	Max total acceleration m/s^2	Max bending acceleration m/s^2	Max torsional acceleration m/s^2	Max total acceleration m/s^2
NEN	0.996	6.258	0.061		0.061			
Eurocode	0.897	5.634	0.852		0.852	NVT		NVT
NBCC	0.996	6.259				0.013		0.013
Woudenberg (emp)	0.897	5.634	1.628	0.267	1.895			
Woudenberg	0.698	4.388	0.211		0.211			
Schueller	0.801	5.034	1.300		1.300			
Dicke/Nijsse	0.130	0.816	0.049		0.049			

Table 38: Resulting annual maxima after renovation

Matlab	0.851	5.346	0.052	0.155	0.170			

Table 39: The average out of a 100 simulations occurring acceleration in Simulink for the Voorhof after renovation for return period of one year

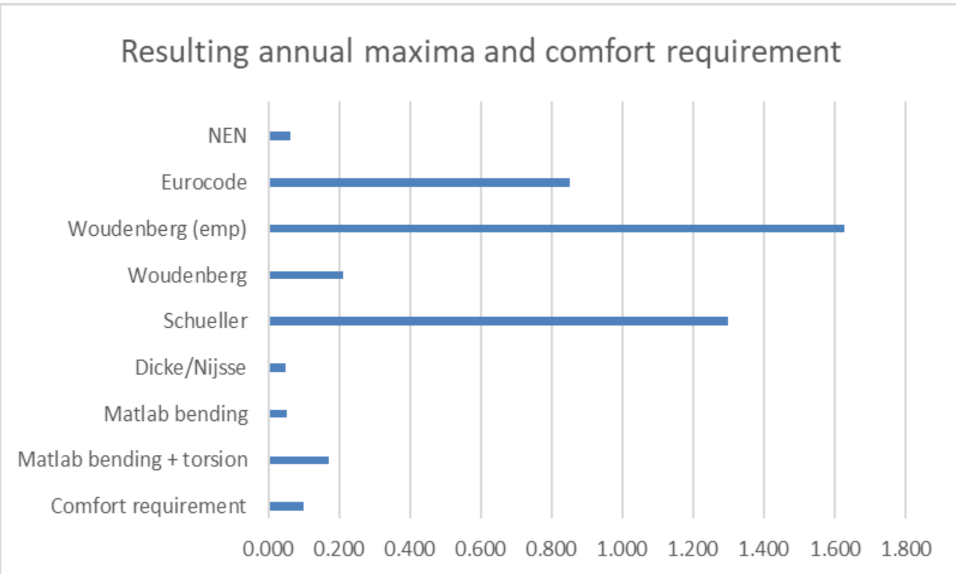


Figure 14.6: Resulting annual maxima and comfort requirement after renovation

NEN

According to NEN 6702 ([3]) vibrations are annoying when the acceleration exceeds a value depending on the frequency (Figure 14.2).

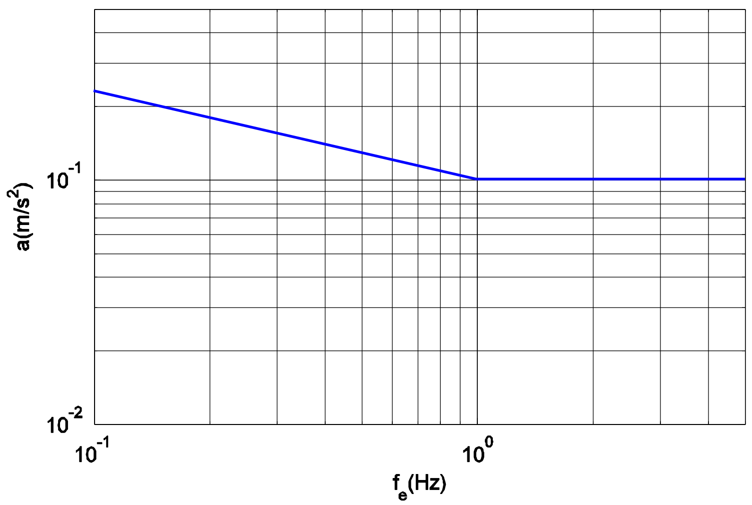


Figure 14.7: Limitation demand for the acceleration according to NEN 6702 figure 21

The NEN 6702 ([3]) which gives the limitation demand for the peak value of the acceleration follows from:

$$a_{max} = 1.6 \frac{\phi_2 \tilde{p}_{w;1} C_t b_m}{\rho_1} < a$$

Where:

a limitation demand of the acceleration for once per year a = 0.10 m/s^2 ([3])

ρ_1 mass of the building per meter height ρ_1 = 2.36*10^5 kg/m

b_m width of the building perpendicular to the wind direction b_m = 80.81 m

C_t summation of the shape factor C_t = 0.8 + 0.4 = 1.2 ([3])

$\tilde{p}_{w;1}$ value for the varying part of the wind pressure $\tilde{p}_{w;1} = 100\ln(h/0.2) = 555$ N/m^2

ϕ_2 value depending on the dimensions, the eigenfrequency and the damping

$$\phi_2 = \sqrt{\frac{0.0344 * f_e^{-2/3}}{\varsigma(1+0.12 f_e h)(1+0.2 f_e b_m)}} = \sqrt{\frac{0.0344*0.996^{-2/3}}{0.01*(1+0.12*0.996*51.3)(1+0.2*0.996*80.81)}} = 0.168$$

Where:

f_e eigenfrequency $f_e = T^{-1} = 0.85\ \text{Hz}$ (From Mathlab)

ς damping ratio section $\varsigma = 0.01$ (Section 9.2.5)

f_e eigenfrequency $f_e = \sqrt{\frac{a}{\delta}} = \sqrt{\frac{0.384}{4.0}} = 0.996\ \text{Hz}$ (NEN 6702 [3])

a acceleration $a = 0.384\,\text{m/s}^2$

δ displacement $\delta = \frac{ql^4}{8EI} = \frac{2.31E^6 * 51.3^4}{8*2.1E^{11}*12.076} = 4.0\ \text{m}$

In which:

l = The building length:	51.3	m	
Q = The building weight:	107.3	MN	(Pols)
q = dead weight of the structure:	2.31	MN/m	(Pols)
g = The gravitational acceleration:	9.81	m/s^2	
E = Young's modulus:	2.1E10	N/m^2	
I = The bending stiffness:	12.076	m^4	

The deflection is determined by using the equation which is given ([14] figure 6.19). In this equation, variable q is the total deadweight of the structure. The total deadweight of the structure is the summation of the deadweight of the buildings floors, load-bearing elements and façade.

Floors:

The representative deadweight of the floor is $q_{g;rep} = 0.15*24 = 3.6kN/m^2$.The design value is of $q_{g;d} = 3.6*1.2 = 4.32kN/m^2$. The total area of the building is $26.3*15.4 \approx 405m^2$. The total deadweight of a storey of the building is: $1147*4.32 = 4955kN$. The storey height is 2.65 metres so it follow's that the deadweight perimeter height due to dead weight is $4955 : 2.65 \approx 1891kN$.

Vertical load-bearing elements:

The area of the concrete walls is $4*10.10*0.25=10.1\ \text{m}^2/\text{m}$ The deadweight of the concrete walls is $q_{g;rep}=10.1*24=242kN/m^2$.

Steel Structure:

The mass of the steel structure $m=6.5\ \text{E}^5\ \text{kg}$. The weight per meter is $q_s=\dfrac{6.5\ \text{E}^5*9.81}{51.3}\approx 125\ \text{kN/m}$.

Partition walls

The mass of the steel structure $m=1.134\ \text{E}^6\ \text{kg}$. The weight per meter is $q_{pw}=\dfrac{1.469\ \text{E}^6*9.81}{51.3}\approx 281\ \text{kN/m}$.

Facade:

The mass of the façade $m=1.134\ \text{E}^6\ \text{kg}$. The weight per meter is $q_{pw}=\dfrac{1.134\ \text{E}^6*9.81}{51.3}\approx 217\ \text{kN/m}$.

Total:

The total deadweight per meter building height is: $1891.2+242+125+281+217=2726.2\ \text{kN/m}$.

This gives:

$$\phi_2=\sqrt{\frac{0.0344*0.996^{-2/3}}{0.01*(1+0.12*0.996*51.3)(1+0.2*0.996*80.81)}}=0.168$$

Filling in the formula for limitation demand for peak acceleration gives:

$$1.6\frac{0.168*555*1.2*80.81}{2.36*10^5}=0.061\ \text{m/s}^2<0.10\ \text{m/s}^2$$

Matlab

With the use of a Simulink model the maximum acceleration due to bending and torsion is calculated for a time span of 10 minutes. For the calculation in this paragraph we take the values form one of the 100 simulations, with mean velocity $V_{10} = 21.45$ m/s and return period of one year. (9.8)

The acceleration of the top of the building (Bending and Torsion motion added together)

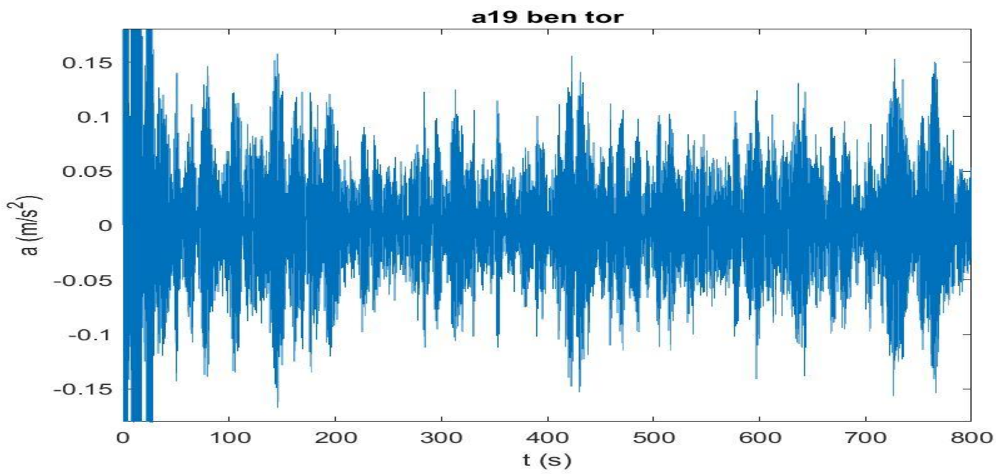


Figure 14.8: Acceleration of the top of the building (bending and torsion motion added together)

The frequency belonging to this acceleration equals the lowest eigenfrequency of the structure $f_e = T_e^{-1} = 0.851$ Hz (section 9.2.4). The peak acceleration of the top of the building for bending and torsion motion added together equals: $a_{19;\max} = 0.170 m/s^2$, which is higher than the maximal acceptable annual acceleration of $0.1 m/s^2$ according to NEN. In which the number 19 is the node number at the top of the building.

We may assume that a_{19} is a normally distributed signal. The standard deviation of a_{19} can be determined from:

$$\sigma_{a;19} = \left(\frac{1}{n} \sum_{i=1}^{n} \left(a_{19;i} - \mu_{a;19} \right)^2 \right)^{\frac{1}{2}}$$

Where:

n the number of discrete time points for which a_{19} have been calculated

$a_{19;i}$ the acceleration of the top of the building at time point i

$\mu_{a;19}$ the mean value of a_{19}

With $\sigma_{a;19}$ = 0.047 m/s^2 and $\mu_{a;19}$ = 0 m/s^2 calculated with Matlab, the expected peak value of the acceleration for a given time range follows from:

$$a_{19;peak} = \mu_{a;19} + \sigma_{a;19}\sqrt{2\ln(N)} \quad \text{([1] eq. 3.38)}$$

Note: The formula above assumes that all peaks are independent of each other which is not true in reality, so this formula will give a value larger than the real peak acceleration.

Where N is the number of draws which follows from the number of local peaks in the total time range:

$$N = T_s f_e$$

Where:

T_s time range of the signal; $T_s = 600$ s

f_e natural frequency which comes out of the mathlab model; $f_e = 0.851Hz$

Filling in peak value of acceleration gives:

$$a_{19;peak;600} = 0.0.047\sqrt{2\ln(600 * 0.851)} = 0.167 \text{ m/s}^2$$

The difference between the expected peak value and the actual value is: $(1-(0.158/0.167))*100 = 5.4\%$, which is reasonably accurate. The expected peak value for a storm of 6 hours (21600 s) can be calculated from:

$$a_{19;peak;21600} = 0.047\sqrt{2\ln(21600 * 0.851)} = 0.208 \text{ m/s}^2$$

The expected peak value for a storm of 1 hour (3600 s) can be calculated from:

$$a_{19;peak;3600} = 0.047\sqrt{2\ln(3600 * 0.851)} = 0.188 \text{ m/s}^2$$

The expected peak value for a hour long storm derived from the Simulink simulation values can be compared with the maximum acceleration according to NEN 6702,

$$\frac{a_{\max;NEN}}{a_{19;peak;3600}} = \frac{0.061}{0.188} = 0.324 \cdot$$

The maximum acceleration due to **bending and torsion motion added together** is exceeded by a factor 3.1. The acceleration of the building is unacceptable.

In this section the maximum acceleration will be calculated for bending acceleration only because this part of the model can be compared to known literature.

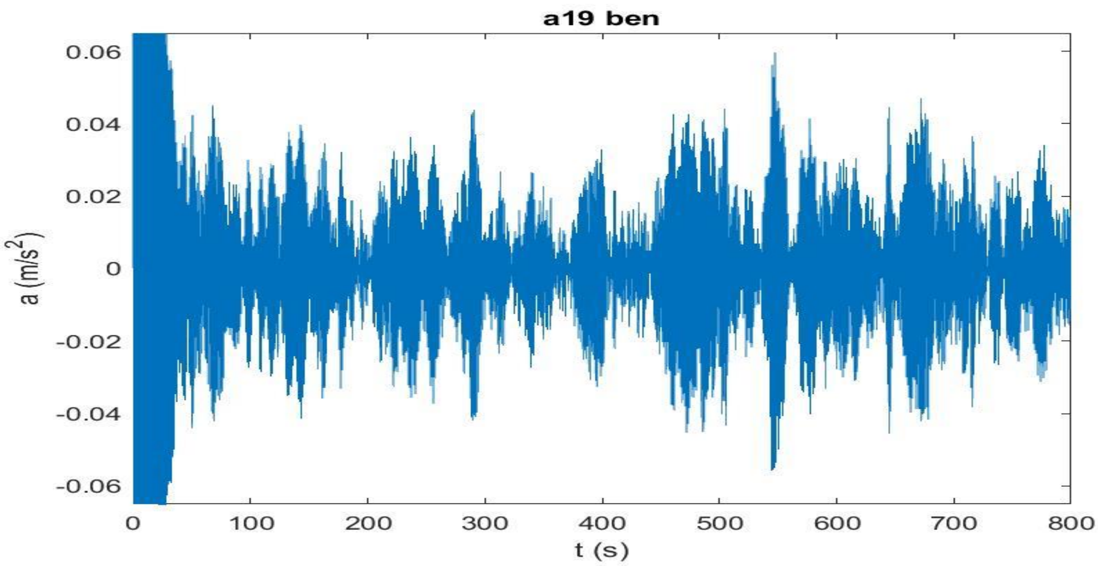


Figure 14.9: Bending acceleration of the top of the building

The frequency belonging to this acceleration equals the lowest eigenfrequency of the structure $f_e = T_e^{-1} = 0.851$ Hz (section 9.2.4). The peak acceleration of the top of the building equals: $a_{19;\max} = 0.060 m/s^2$, which is lower than the maximal acceptable acceleration. We may assume that a_{19} is a normally distributed signal. View p.328 for the formula of the standard deviation of a_{19}.

With $\sigma_{a;19} = 0.016$ m/s^2 and $\mu_{a;19} = 0$ m/s^2.

The expected peak value for a hour long storm derived from the Simulink simulation values can be compared with the maximum acceleration according to NEN 6702,

$$\frac{a_{\max;NEN}}{a_{19;peak;3600}} = \frac{0.061}{0.064} = 0.953 \cdot$$

The maximum acceleration due to **bending** is exceeded by about 5%. The acceleration of the building is unacceptable.

Eurocode

The wind force acting on a structure or structural component is: ([10] p.25])

$$F_w = c_s c_d * c_f * q_p(z_e) * A_{ref}$$

$$F_w = 0.77 * 0.67 * q_p(z_e) * A_{ref}$$

With:

$c_s c_d$:	Structural factor
c_f :	Force coefficient
$q_p(z_e)$:	peak velocity pressure at reference height z_e
A_{ref} :	reference area of the structure

Force coefficient. [10] p.67

$$c_f = 0.67$$

Structural factor ([10] p.28])

$$c_s c_d = \frac{1 + 2 * k_p * I_v(z_e) * \sqrt{B^2 + R^2}}{1 + 7 * I_v(z_e)}$$

$$c_s c_d = \frac{\left(1 + 2 * 3.69 * 0.29 * \sqrt{0.35 + 0.033}\right)}{(1 + 7 * 0.29)} = 0.76$$

With:

z_e :	reference height or height of structure.
k_p :	peak factor.
I_v :	turbulence intensity
B^2 :	Background factor.
R^2 :	resonance response factor.

Background factor ([10] p.110]) **Eurocode procedure 2**

$$B^2 = \frac{1}{1+\frac{3}{2}\sqrt{\left(\frac{b}{L(z_e)}\right)^2+\left(\frac{h}{L(z_e)}\right)^2+\left(\frac{b}{L(z_e)}*\frac{h}{L(z_e)}\right)^2}}$$

$$B^2 = \frac{1}{1+\frac{3}{2}\sqrt{\left(\frac{80.81}{85.6}\right)^2+\left(\frac{51.3}{85.6}\right)^2+\left(\frac{80.81}{85.6}*\frac{51.3}{85.6}\right)^2}} = 0.35$$

Wind Turbulence ([10] p.104])

$$L(z_e) = L_t\left(\frac{z_e}{z_t}\right)^{\alpha} = 300\left(\frac{30.78}{200}\right)^{0.67} = 85.6$$

$$\alpha = 0.67 + 0.05*\ln(z_0) = 0.67 + 0.05*\ln(1) = 0.67$$

With:

b, h: width and height of structure

$L(z_e)$: Turbulence length scale. It is on the safe side to use $B^2 = 1$

$z_e = 0.6*h \geq z_{\min} = 0.6*51.3 = 30.78$:([10] p.29])

$$k_p = \sqrt{2*\ln(v*T)} + \frac{0.6}{\sqrt{2*\ln(v*T)}}$$

$$k_p = \sqrt{2*\ln(0.80*600)} + \frac{0.6}{\sqrt{2*\ln(0.80*600)}} = 3.69$$

Resonance response factor ([10] p.110])

$$R^2 = \frac{\pi^2}{2*\delta}*S_L(z_e, n_{1,x})*K_s(n_{1,x})$$

$$R^2 = \frac{\pi^2}{2*\delta}*S_L\left(z_e,n_{1,x}\right)*K_s\left(n_{1,x}\right)$$

$$R^2 = \frac{\pi^2}{2*0.15}*0.068*0.0143 = 0.033$$

With:

δ	The total logarithmic decrement of damping	([10] pp. 145-147])
S_L	wind power spectral density function given	B. 1 (2)
$n_{1,x}$	natural frequency of the structure	([10] p.144])
K_s	size reduction function	([10] p.111])

The total logarithmic decrement of damping

$$\delta = \delta_s + \delta_a + \delta_d = 0.07 + 0.08 + 0 = 0.15$$

The total logarithmic decrement of structural damping

$$\delta_s = 0.07$$

The total logarithmic decrement of aerodynamic damping for fundamental mode

$$\delta_a = 0.08$$

The total logarithmic decrement of damping due to special devices

$$\delta_d = 0$$

wind power spectral density function given ([10] p.104])

$$S_L(z_e, n_{1,x}) = \frac{n * S_v(z_e, n)}{\sigma_v^2} = \frac{6.8 * f_L(z_e, n)}{(1 + 10.2 * f_L(z_e, n))^{5/3}} = \frac{6.8 * 2.754}{(1 + 10.2 * 2.754)^{5/3}} = 0.068$$

$$f_L(z_e, n) = \frac{n * L(z_e)}{v_m(z_e)} = f_L(85.6, 0.80) = \frac{0.80 * 85.6}{21.68} = 2.754$$

$$L(z) = L_t\left(\frac{z}{z_t}\right)^{\alpha} = 300\left(\frac{30.78}{200}\right)^{0.67} = 85.6$$

$$\alpha = 0.67 + 0.05 * \ln(z_0) = 0.67 + 0.05 * \ln(1) = 0.67$$

natural frequency of the structure [10] p.144]

$$\delta = \frac{ql^4}{8EI} = \frac{2.31E^6 * 51.3^4}{8 * 5.29E^{10} * 97.85} = 0.39 \text{ m}$$

$$n = n_{1,x} = \frac{1}{2\pi}\sqrt{\left(\frac{g}{x_1}\right)} = \frac{1}{2\pi}\sqrt{\left(\frac{9.81}{0.39}\right)} = 0.80\text{Hz}$$

size reduction function ([10] p.111])

$$K_s(n) = \frac{1}{1 + \sqrt{(G_y * \phi_y)^2 + (G_z * \phi_z)^2 + \left(\frac{2}{\pi} * G_y * \phi_y * G_z * \phi_z\right)^2}}$$

$$K_s(n) = \frac{1}{1 + \sqrt{(0.5 * 34.35)^2 + (0.278 * 21.80)^2 + \left(\frac{2}{\pi} * 0.5 * 34.35 * 0.278 * 21.80\right)^2}} = 0.0144$$

$$\phi_y = \frac{c_y * b * n}{v_m(z_e)} = \frac{11.5 * 80.81 * 0.80}{21.68} = 34.35$$

$$\phi_z = \frac{c_y * h * n}{v_m(z_e)} = \frac{11.5 * 51.3 * 0.80}{21.68} = 21.80$$

Decay constants

$c_y = c_z = 11.5$

The peak velocity pressure is calculated using $q_p(z) = 1 + 7 * I_v(z) * \frac{1}{2} * \rho * v_m^2(z)$

With:

$I_v(z)$:	turbulence intensity
ρ:	air density
$v_m^2(z)$:	mean wind velocity

Turbulence intensity ([10] p.21])

$$I_v(z) = \frac{\sigma_v}{v_m(z)} = \frac{k_l}{c_0(z) * \ln(z/z_0)} = \frac{1}{1 * \ln(51.3/1)} = 0.25$$

$$I_v(z_e) = \frac{\sigma_v}{v_m(z_e)} = \frac{k_l}{c_0(z_e) * \ln(z_e/z_0)} = \frac{1}{1 * \ln(30.78/1)} = 0.29$$

Terrain roughness ([10] p.20])

$$k_r(z) = 0.19 * \ln\left(\frac{z_0}{z_{0,II}}\right)^{0.07}$$ (Terrain category IV)

$$k_r(z) = 0.19 * \ln\left(\frac{1}{0.05}\right)^{0.07} = 0.23 \text{ [-]}$$

$$c_r(z) = k_r(z) * \ln\left(\frac{z}{z_0}\right)$$

$$c_r(z) = 0.23 * \ln\left(\frac{51.3}{1}\right) = 0.92 \text{ [-]}$$ (Terrain category IV)

$$c_r(z_e) = 0.23 * \ln\left(\frac{30.78}{1}\right) = 0.80 \text{ [-]}$$

Mean wind velocity ([10] p.21])

$$v_m(z) = c_r(z) * c_0(z) * v_b = 0.92 * 1 * 27 = 24.9 \text{ m/s}^2$$
$$v_m(z_e) = c_r(z_e) * c_0(z_e) * v_b = 0.80 * 1 * 27 = 21.68 \text{ m/s}^2$$

With:

$c_r(z)$	roughness factor
$c_0(z)$	orography factor
v_b	basic wind velocity

Basic wind velocity ([10] p.19])

$$\text{v}_\text{b}(\text{z}) = \text{c}_\text{dir} * \text{c}_\text{season} * \text{v}_\text{b,0} = 1 * 1 * 27 = 27 \text{ m/s}^2$$

With:

c_{dir}	directional factor
c_{season}	season factor
$v_{b,0}$	fundamental value of basic wind velocity

For the Netherlands, the euro code states that the fundamental value of basic wind velocity is equal to 27 m/s.

Acceleration for serviceability assessments ([10] p.111])

$$\sigma_{\mathrm{a,x}}\left(\mathrm{y,z}\right)=\mathrm{c_f}*\rho*\mathrm{I_v}\left(z_e\right)*\mathrm{v_m^2}\left(z_e\right)*R*\frac{K_y*K_z*\Phi\left(\mathrm{y,z}\right)}{\mu_{ref}*\Phi_{\max}}$$

$$\sigma_{\mathrm{a,x}}\left(\mathrm{y,z}\right)=0.67*1.29*0.29*\left(21.68\right)^2*0.182*\frac{1*5/3*1}{205*1}=0.231\ \mathrm{m/s^2}$$

The standard deviation $\sigma_{\mathrm{a,x}}$ of the characteristic along-wind acceleration of the structural point with coordinates (*y*,*z*) is approximately given by Expression ([10])

where:

$\mathrm{c_f}$	force coefficient
ρ	air density
$\mathrm{I_v}\left(z_e\right)$	turbulence intensity at height z_e above ground
$\mathrm{v_m^2}\left(z_e\right)$	characteristic mean wind velocity at height z_e
R	square root of the resonant response
K_y, K_z	constants given in C.2 (6)
μ_{ref}	the reference mass per unit area
$\Phi\left(\mathrm{y,z}\right)$	the mode shape
$\Phi_{\max}$	mode shape value at the point with maximum amplitude

The characteristic peak accelerations are obtained by multiplying the standard deviation by the peak factor in B. 2 (3) using the natural frequency as upcrossing frequency, i.e. $v = n_{1,x}$.

$$a_{\max}=k_p\sigma_{\mathrm{a,x}}\left(\mathrm{y,z}\right)=3.69*0.231=0.852\ \mathrm{m/s^2}$$

Woudenberg empirical formula's

The formula to calculate the frequency in this article is ([19] p.32]):

$$f_{e\_Woudenburg} = \frac{46}{h}$$

In which:

height building: 51.3 m

Substituting the variables into the frequency formula gives:

$$f_{e\_Woudenburg} = \frac{46}{51.3} \approx 0.897 \text{ Hz}$$

The cyclic frequency $\omega_{e\_Woudenberg} = 2 * \pi * f_e \approx 5.63$ rad/s .

The maximum acceptable deflection of the top of the building is $u_{top} = H/500$. When the height is substituted we get a deflection of $51.3/500 = 0.1026m$. The amplitude A of our building is the half of the maximum acceptable deflection of the top of the building, this is $0.1026/2 = 0.0513m$.

The formula for the natural vibration can be written as $u(t) = A\sin(\omega t + \varphi)$. The acceleration is the second derivative of the natural vibration formula, $a(t) = \ddot{u}(t) = A\omega^2 \sin(\omega t + \varphi)$, of which the maximum bending acceleration is $a = A * \omega^2$. Substituting the variables into the formula gives $a = 0.0513 * 5.63^2 = 1.628 \text{ m/s}^2$.

This maximum occurring acceleration is **far above** the maximum acceptable acceleration of $a = 0.1 m/s^2$.

Determining the torsional acceleration with the empirical formula in Woudenberg article ([19] p.32]):

$$f_{e\_Woudenburg} = \frac{72}{h_{building}}$$

In which:

h = height building: 51.3 m

Substituting the variables into the frequency formula gives:

$$f_{e\_Woudenburg} = \frac{72}{51.3} = 1.404 \text{ Hz}$$

The angular cyclic frequency is $\omega_{e\_Woudenberg} = 2 * \pi * f_e \approx 8.819$ rad/s.

The maximum acceptable deflection of one floor of the building is $u_{top} = H/500$. When the height is substituted we get a deflection of $51.3/500 = 0.103$ m. The maximum acceptable bending deflection per story is $0.103/19 = 5.4E^{-3}$ m. To determine the torsion on a floor we have to look at the interstory drift of a building. The maximum acceptable displacement due to interstory drift is $u_{story} = h_{story}/300$. Substitution gives $u_{story} = 2.65/300 = 8.83E^{-3}$ m. The displacement left for torsion motion is $8.83E^{-3} - 5.40E^{-3} = 3.43E^{-3}$m.

We assume in our model that the bending and torsional motions are uncoupled. The maximum amplitude due to torsional motion is $0.00343m$. The maximum amplitude for torsion, should not be desregarded for the maximum deflection of a building.

$$u_{floor} = \frac{h_{story}}{300} - \frac{h}{500 * n_{floors}} = \frac{2.65}{300} - \frac{144}{500 * 48} = 8.83E^{-3} - 5.40E^{-3} = 3.43E^{-3}m$$

The formula for the natural vibration can be written as $u(t) = A\sin(\omega t + \varphi)$. The acceleration is the second derivative of the natural vibration formula $a(t) = \ddot{u}(t) = A\omega^2 \sin(\omega t + \varphi)$. Substituting the variables into the formula gives $a = 0.00343 * 8.819^2 = 0.267$ m/s^2, which is the maximum torsional acceleration.

This maximum occurring acceleration due to bending and torsion motion is $a = 1.628 + 0.267 = 1.895 \text{ m/s}^2$ far above the maximum acceptable acceleration of $a = 0.1 \text{ m/s}^2$.

Woudenberg formulas

The formulas to calculate the frequency in this article “Windbelasting en het hoogbouwontwerp” ([19] p.28-36]):

Formulas

Bending displacement at top due to wind load

$$\partial_{EI} = \left(\frac{q_w * h^4}{8EI}\right) = \left(\frac{33553 * 51.3^4}{8 * 5.17\text{E}^{12}}\right) = 5.62\text{E}^{-3}\ \text{m}$$

Shear displacement at top due to wind load

$$\partial_{GA} = \left(\frac{q_w * h^2}{2GA}\right) = \left(\frac{33553 * 51.3^2}{2 * 2.09\text{E}^{11}}\right) = 2.11\text{E}^{-4}\ \text{m}$$

Displacement at the top due to foundation rotation

$$C_f = 20 * EI / L = 2.02E^{12}\ \text{Nm} \qquad ([5])$$

$$\partial_{C_f} = \left(\frac{h^3}{2C_f}\right) = \left(\frac{51.3^3}{2 * 2.02E^{12}}\right) = 1.12\text{E}^{-3}\ \text{m}$$

Total displacement at top of the building

$$\partial_{Top} = \partial_{EI} + \partial_{GA} + \partial_{C_f} = 7.0\text{E}^{-3}\ \text{m}$$

Critical buckling force

$$Q_k = \left(\frac{q_w * h^2}{\partial_{Top}}\right) = \left(\frac{33553 * 51.3^2}{7.0\text{E}^{-3}}\right) = 12703689795\ \text{N}$$

Second order effect

$$\text{second order} = \frac{n}{n-1} = \frac{107.111}{107.111-1} = 1.01$$

$$n = \frac{Q_k}{Q_{optr}} = \frac{1}{q_m} \frac{1}{\left(\frac{h^3}{8EI} + \frac{h}{2GA} + \frac{h^2}{2C_f}\right)} = \frac{1}{2.31E^6} \frac{1}{\left(\frac{51.3^3}{8*5.17E^{12}} + \frac{51.3}{2*2.09E^{11}} + \frac{51.3^2}{2*2.02E^{12}}\right)} = 107.111$$

$$m_{buidling} = 1.21\ E^7\ \text{kg (Mathlab)} \quad q_m = \frac{m_{buidling} * g}{h} = \frac{1.21E^7 * 9.81}{51.3} = 2.31E^6\ \text{N/m}$$

Total displacement due to wind load and foundation rotation (including second order effect)

$$\partial_{Top\,sec} = \frac{n}{n-1}\partial_{Top} = 1.01 * 0.007 = 0.00707\ \text{m}$$

Amplitude

$$A = 0.00707 / 2 = 3.54E^{-3}\ \text{m}$$

Natural frequency

$$\omega_n = C\sqrt{\left(EI / \rho Al^4\right)} = 3.52\sqrt{\left(\left(2.54E^{12}\right) / \left(205.39 * 1147.50 * 51.3^4\right)\right)} = 4.39\ \text{rad/s}$$

$$\omega_{e\_Woudenberg} = \omega_n \approx 4.39\ \text{rad/s}$$

The formula for the natural vibration can be written as $u(t) = A\sin(\omega t + \varphi)$. The acceleration is the second derivative of the natural vibration formula, $a(t) = \ddot{u}(t) = A\omega^2 \sin(\omega t + \varphi)$. Of which the maximum acceleration is, $a = A*\omega^2$. Substituting the variables into the formula gives $a = 0.0035*4.39^2 = 0.068\ \text{m/s}^2$.

This maximum occurring acceleration is **below** the maximum acceptable acceleration of $a = 0.1\ \text{m/s}^2$.

Schueller

The eigenfrequency of the structure must be known, to determine the acceleration what occurs ([18] p. 176]).

The formula to calculate the vibration time is:

$$T = 2 * \pi * \sqrt{((Q * H \wedge 3)/(8 * EI * g))}$$

In which:

H =	The building Height:	51.3	m
Q =	The building weight:	119	MN (Pols)
g =	The gravitational acceleration:	9.81	m/s^2
E =	Young's modulus:	5.29E10	N/m^2
I =	The bending stiffness:	97.82	m^4

Substituting the variables into the formula gives:

$$T = 2 * \pi * \sqrt{((Q * H \wedge 3)/(8 * EI * g))} = 1.248 \text{ s}$$

The eigenfrequency $f = 1/T = 1/1.248 \approx 0.801 \text{ Hz}$. The cyclic frequency is $\omega = 2 * \pi * f = 5.034 \text{ rad}$.

The maximum acceptable deflection of the top of the building is $u_{top} = H/500$. When the height is substituted we get a deflection of $51.3/500 = 0.103m$. The amplitude A of our building is the half of the maximum acceptable deflection of the top of the building, this is $0.103/2 = 0.051m$.

The formula for the natural vibration can be written as $u(t) = A\sin(\omega t + \varphi)$. The acceleration is the second derivative of the natural vibration formula, $a(t) = \ddot{u}(t) = A\omega^2 \sin(\omega t + \varphi)$, of which the maximum acceleration is $a = A * \omega^2$. Substituting the variables into the formula gives $a = 0.051 * 5.034^2 = 1.300 \text{ m/s}$.

This maximum occurring acceleration is **above** the maximum acceptable acceleration of $a = 0.1 \text{ m/s}$.

Dicke/Nijsse

The formula to calculate the frequency in this article is ([20] p.32]):

The values of c by different types of fixations

c=1	spring
c=0.83	Cantilever
c=0.67	**on piles**
c=0.64	on rock ground

The variables of the building loaded by wind

L	Length of building	[m]
m	Mass of the building	[kg]
I	Second Moment of Area	[m^4]

The gravitational effect of the upside-down pendulum

K_e = Effective stiffness of the system	[N/m]
K_b = Spring stiffness	[N/m]
K_p = Negative stiffness of the upside-down pendulum	[N/m]

Mathematic Pendulum

$T=2*\pi*\sqrt{\frac{l}{g}}$	(Independent of Mass)	[S]
$T=\frac{2*\pi}{\omega}$	(From vibration time to cyclic frequency)	[S]

summation of $\omega^2=\frac{g}{l}$

$\omega^2 = \sqrt{\frac{k}{m}}$ (Single degree of freedom system) $\omega^2 = \frac{K_p}{m}$

Combination of $\omega^2 = \frac{K_p}{m}$ and $\omega^2 = \frac{g}{l}$ gives us $K_p = \frac{mg}{l}$

with $G = m * g$

Upside-down pendulum

l_p pendulum length

c values of cantilevers.

l building length

$l_p = c * l$

Calculations

Weight building

$G = m * g = 1.21\ E^7 * 9.81 = 118602900$ N

Pendulum length

$l_p = 0.67 * 51.3 = 34.37$ m

Negative stiffness of the upside-down pendulum

$$K_p = \frac{m*g}{l_p} = \frac{1.21E^7*9.81}{37.37} = 34506677 \text{ N/m}$$

Spring stiffness

$$K_b = \frac{3EI}{l^3} = \frac{3*5.17E^{12}}{51.3^3} = 114920410 \text{ N/m}$$

Effective stiffness of the system

$K_e = K_b - K_p = 80413733$ N/m

$$\omega_p = \sqrt{\frac{K_p}{m}} = \sqrt{\frac{34506677}{1.209E^7}} = 0.534 \text{ rad/s}$$

$$\omega_b = \sqrt{\frac{K_b}{m}} = \sqrt{\frac{114920410}{1.209E^7}} = 0.975 \text{ rad/s}$$

$$\omega_e = \sqrt{\frac{K_e}{m}} = \sqrt{\frac{80413733}{1.209E^7}} = 0.816 \text{ rad/s}$$

Frequency of Professor Dicke

$$f = \frac{\omega_e}{2\pi} = \frac{0.816}{2\pi} = 0.130 \text{ Hz}$$

Vibration time of Professor Dicke

$$T = \frac{1}{f_e} = \frac{1}{0.130} = 7.704 \text{ sec}$$

Critical buckling factor

$$n = \frac{\omega_b^2}{\omega_p^2} = \frac{0.975^2}{0.534^2} = 3.330$$

Second order effect

$$n_{sec} = \frac{n}{n-1} = \frac{3.330}{2.330} = 1.43$$

Amplitude

$$A = \frac{h}{500} / 2 = 0.103 / 2 = 0.0515 \text{ m}$$

Maximum acceleration

The maximum acceleration is $a = A * \omega^2 * n_{sec}$. Substituting the variables into the formula gives $a = 0.0515 * 0.816^2 * 1.43 = 0.049 \text{ m/s}^2$

This maximum occurring acceleration is **below** the maximum acceptable acceleration of $a = 0.1 \text{ m/s}^2$.

NBCC Acrosswind acceleration

The acrosswind acceleration will be larger than the alongwind acceleration if $(bd)^{0.5} < 0.33$ ([17] p.61]). This is the case in most buildings with a rectangular cross section, so the acrosswind acceleration will be dominant.

The National Building Code of Canada (NBCC) will be used to determine the across wind acceleration of the building ([17] p.61]).

$$a_y = \frac{f_e g_p a_r \sqrt{bd}}{\rho_b g \sqrt{\beta_w}}$$

With:

f_e	frequency of the building	Hz
g_p	peak factor	[-]
b_m	width of the building	m
ρ	average density of the building	kg/m3
g	gravitational acceleration	m/s2
β_w	lift damping ratio	[-]
b	width of the building	m
d	depth of the building	m

a_r is calculated by the following equation:

$$a_r = 0.0785\left(\frac{v_h}{f_e\sqrt{b_m * b_m}}\right)^{3.3}$$

With:

f_e	frequency of the building	Hz
b_m	width of the building	m
v_h	mean wind speed at top of the building	m/s^2

$$a_y = \frac{f_e g_p a_r \sqrt{bd}}{\rho_b g \sqrt{\beta_w}} = \frac{0.996*3.74*0.0288*\sqrt{80.81*14.2}}{205.38*9.81*\sqrt{0.020}} = 0.013 \text{ m/s}^2$$

a_r is calculated by the following equation:

$$a_r = 0.0785\left(\frac{v_h}{f_e\sqrt{b_m * b_m}}\right)^{3.3} = 0.0785\left(\frac{36.8}{0.996\sqrt{80.81*14.2}}\right)^{3.3} = 0.0288 \text{ [-]}$$

Eurocode Acrosswind acceleration

E.1.2 Criteria for vortex shedding ([10] p.116)

(3) The effect of vortex shedding should be investigated when the ratio of the largest to the smallest crosswind dimension of the structure, both taken in the plane perpendicular to the wind, exceed 6.

(4) The effect of vortex shedding need not be investigated if

$$v_{crit,i} > 1.25 * v_m \qquad \text{(E.1)}$$

$$106.34 > 1.25 * 24.91$$

where:

$v_{crit,i}$ is the critical wind velocity for mode I, as defined in E.1.3.1

v_m is the characteristic 10 minutes mean wind velocity specified in 4.3.1 (1) at the cross section where vortex shedding occurs. (Figure E.3).

Vortex shedding need not be investigated

E.1.3 Basic parameters for vortex shedding

E.1.3.1 Critical wind velocity $v_{crit,i}$

The critical wind velocity for bending vibration mode *i* is defined as the wind velocity at which the frequency of vortex shedding equals a natural frequency of the structure or a structural element and is given in Expression (E.2).

$$v_{crit,i} = \frac{b * n_{i,y}}{St} \qquad \text{(E.2)}$$

$$v_{crit,i} = \frac{14.2 * 0.80}{0.107} = 106.34 \text{ m/s}^2$$

where:

b is the reference width of the cross-section at which resonant vortex shedding occurs and where the modal deflection is maximum for the structure or structural part considered; for circular cylinders the reference width is the outer diameter.

$n_{i,y}$ is the natural frequency of the considered flexural mode *i* of cross-wind vibration; approximations for $n_{1,y}$ are given in F.2

St Strouhal number as defined in E.1.3.2.

.

E.1.3.2 Strouhal number *St*

The Strouhal number St for different cross-sections may be taken from Table E.1

$$d/b = 80.81/14.2 = 5.69 \text{ [-]}$$

$$St = 0.107 \text{ [-]}$$

E.1.3.3 Scruton number *Sc* ([10] p.119)

The susceptibility of vibrations depends on the structural damping and the ratio of structural mass to fluid mass.

This is expressed by the Scruton number *Sc*, which is given in Expression (E.4).

$$Sc = \frac{2 * \delta_s * m_{i,e}}{\rho * b^2} \quad \text{(E.4)}$$

$$Sc = \frac{2 * 0.08 * 2.36E^5}{1.29 * 14.2^2} = 145.0$$

where:

δ_s is the structural damping expressed by the logarithmic decrement.

ρ is the air density under vortex shedding conditions.

$m_{i,e}$ is the equivalent mass m_e per unit length for mode *i* as defined in F.4 (1)

b is the reference width of the cross-section at which resonant vortex shedding occurs

E.1.5 Calculation of the cross wind amplitude

E.1.5.1 General

(1) Two different approaches for calculating the vortex excited cross-wind amplitudes are given in E.1.5.2 and E.1.5.3.

(2) The approach given in E.1.5.2 can be used for various kind of structures and mode shapes. It includes turbulence and roughness effects and it may be used for normal climatic conditions.

(3) The approach given in E.1.5.3 may be used to calculate the response for vibrations in the first mode of cantilevered structures with a regular distribution of cross wind dimensions along the main axis of the structure.

Typically structures covered are chimneys or masts. It cannot be applied for grouped or in-line arrangements and for coupled cylinders. This approach allows for the consideration of different turbulence intensities, which may differ due to meteorological conditions. For regions where it is likely that it may become very cold and stratified flow conditions may occur (e.g. in coastal areas in Northern Europe), approach E.1.5.3 may be used.

E.1.5.2 Approach 1 for the calculation of the cross wind amplitudes

E.1.5.2.1 Calculation of displacements ([10] p.121)

The largest displacement $y_{F,max}$ can be calculated using Expression (E.7).

$$\frac{y_{F,\max}}{b} = \frac{1}{St^2} * \frac{1}{Sc} * K * K_w * c_{lat} \qquad \text{(E.7)}$$

$$y_{F,\max} = \frac{1}{0.107^2} * \frac{1}{145.0} * 0.13 * 0.60 * 1.1 * 14.2 = 0.7341 \text{ m}$$

where:

St is the Strouhal number given in Table E.1

Sc is the Scruton number given in E.1.3.3

K_w is the effective correlation length factor given in E.1.5.2.4

K is the mode shape factor given in E.1.5.2.5

c_{lat} is the lateral force coefficient given in Table E.2

E.1.5.2.2 Lateral force coefficient c_{lat}

The basic value, $c_{lat,0}$, of the lateral force coefficient is given in Table E.2.

$$c_{lat,0} = 1.1$$

E.1.5.2.3 Correlation length L ([10] p.124)

The correlation length L_j, should be positioned in the range of antinodes. Examples are given in Figure E.3. For guyed masts and continuous multispan bridges special advice is necessary.

$$K_w = 3\frac{L_j / b}{\lambda} * \left[1 - \frac{L_j / b}{\lambda} + \frac{1}{2} * \left(\frac{L_j / b}{\lambda}\right)^2\right]$$

$$K_w = 0.6 \; [-]$$

E.1.5.2.5 Mode shape factor ([10] p.125)

The mode shape factor *K* is given in Expression (E.9).

$$K = \frac{\sum_{j=1}^{m} \left| \int_{l_j} \Phi_{i,y}(s) \right| ds}{4 * \pi * \sum_{j=1}^{m} \int_{l_j} \Phi_{i,y}^2(s) ds}$$

$$K = 0.13 \qquad \text{(E.9)}$$

where:

m	is defined in E.1.5.2.4 (1)
$\Phi_{i,y}(s)$	is the cross-wind mode shape i (see F.3)
l_j	is the length of the structure between two nodes (see Figure E.3)

The maximum acceleration is:

$$a_{\max} = y_{F,\max} * \left(2*\pi* f_e\right)^2 = 0.7341*\left(2*\pi*0.8\right)^2 = 18.55 \text{ m/s}^2 .$$

Appendix 13 Matlab code NDOF Juffertoren (frequency domain analysis)

Below the Matlab code is given for the Juffertoren for return period of one year, the only difference with the Matlab code for return period of 12.5 years is that $v_{10} = 22.62 m/s$ and $\sigma_v = 6.23 m/s$ (6.3) are inputted in Matlab file: Spectra_acceleration.m

For this code to run take generated file matrices.mat from the time domain analysis file folder.

Spectra_acceleration.m

```
clear
close all

%
% Calculating the spectra of acceleration
%
%   Input: K_ben M_ben Cd_ben
%
%   Output:  Saa
%
% Author: H.A.O.Richardson (Anthony)
% Thesis: Wind induced torsions in high-rise buildings due to wind loading.
%
%-------------------------------------------------------------------------

tic


% Loading Mass-,  Stiffness-, Damping- Velocity-, eigen matrix bending
matrix

%load matrices M_ben K_ben Cd_ben v_mean omega_eig_ben E_ben V_Hub yr zr
%edit 09-07
load matrices M_ben K_ben Cd_ben v_mean omega_eig_ben E_ben  %edit 28-04-
2019

zr=3:3:144; %48 ndes of freedom system
yr(:)=0; % there is no coherence in y direction

%---------------------------------------------------------------

v=48;                % Bending degrees of freedom
L=1200;              % characteristic length Davenport
sigma_v=2.52;        % standard deviation of the wind speed variation with
height
v_10=21.45;          % mean wind speed at 10 m height

U=v_10;
```

```
V_Hub=U;

rho=1.25;              % Air density
A=3*26.34;             % Area of the building surfice one node height *
Building width
Ch=1.2;                % Summation of suction and drag coefficient

omega_max=max(omega_eig_ben)+200; % Maximum value of the natural frequency
matrix plus 200

delta=.1;


%-------------------------------------------------------------------------
% Putting velocity in 48 node in correspondence with the nodes of structure
%-------------------------------------------------------------------------

for c=1:1:v-2
        v_mean_48(1,2+c)=v_mean((2*c-1));
end
        v_mean_48(1,1)=v_mean_48(1,3);
        v_mean_48(1,2)=v_mean_48(1,3);



%------------------------------------------------------------
% Fixing the correct Eigenvectors
%------------------------------------------------------------

Transversion_E_ben


%------------------------------------------------------------
% Generalised mass, damping-, spring constant added
%------------------------------------------------------------

for n=1:1:v;

    m_k(n)=transpose(E_ben(:,n))*M_ben*E_ben(:,n);

end

xx=transpose(E_ben(:,1));


   for n=1:1:v;

    c_k(n)=transpose(E_ben(:,n))*Cd_ben*E_ben(:,n);

   end


    for n=1:1:v;

    k_k(n)=transpose(E_ben(:,n))*K_ben*E_ben(:,n);
```

```
    end


%-------------------------------------------------------------------------
% Determing the velocity spectra
%-------------------------------------------------------------------------


S_vv=zeros(v,v,2);
S_FF=zeros(v,v,2);
S_vv_1=zeros(v,v,2);

for n=1:1:(25*10);

    omega(n)=n/10;% plotting omega for every interval

    for p=1:1:v


    %-----------------------------------------------------------------------
--
    % Depending on height (only autospectra)
    %-----------------------------------------------------------------------
--
    %

    % making 48 *48 matrix to be able to do the coherence

S_vv(p,p,n)=((omega(n).*L./(2.*pi.*v_10)).^2)./((1+(omega(n).*L./(2.*pi.*v_
10)).^2).^(4/3)).*(((2/3).*sigma_v.^2)./omega(n));

    end


    %-----------------------------------------------------------------------
--
    % Depending on height (autospectra and cross spectra)
    %-----------------------------------------------------------------------
--





    %-----------------------------------------------------------------------
--
    % Coherence of velocity spectra
    %-----------------------------------------------------------------------
--


    Coh=cohorence_2_run(n,omega,yr,zr,U);
    Cohm=Coh;

    %-----------------------------------------------------------------------
---
```

```
    for aa=1:1:v
        for bb=1:1:v
                if (aa==bb)


S_vv(aa,bb,n)=(Cohm(aa,bb).*(S_vv(aa,aa,n).*S_vv(bb,bb,n)).^.5);%edit 20-
04-2019

                else


S_vv(aa,bb,n)=(Cohm(aa,bb).*(S_vv(aa,aa,n).*S_vv(bb,bb,n)).^.5);
                end
        end
    end

    S_vvco(:,:,n)=S_vv(:,:,n);


    %----------------------------------------------------------------------
    % Force spectra of the nodes
    %----------------------------------------------------------------------


    % Determining the force spectra for ever node on diagonal
    for d=1:1:v


            S_FF(d,d,n)=(rho.*A.*Ch.*v_mean(d)).^2.*S_vv(d,d,n); % CT4145
Page 26
            % no aerodynamic admittance in the function

            % At ever node there is a different velocity,for each velocity
a
            % force spectra must be determined.
    end

    %----------------------------------------------------------------------
--
    % Aerodynamic admittance
    %----------------------------------------------------------------------

    ADM=admittance_2_run(n,omega,yr,zr,U);

    %----------------------------------------------------------------------
    % Fitting the aerodynamic admittance into the force spectrum
    %----------------------------------------------------------------------



    for aaa=1:1:v
        for bbb=1:1:v
                if (aaa==bbb)
```

```
S_FF(aaa,bbb,n)=S_vv(aaa,bbb,n).*((rho.*A.*Ch).^2).*v_mean(aaa).*v_mean(bbb
).*(ADM(aaa,bbb)^2);

                else

S_FF(aaa,bbb,n)=S_vv(aaa,bbb,n).*((rho.*A.*Ch).^2).*v_mean(aaa).*v_mean(bbb
).*(ADM(aaa,bbb)^2);
                end
        end
    end

S_FFco(:,:,n)=S_FF(:,:,n);


end


%% Quasi-static part of the response spectra


    ii=48;
    j=48;

    ps=v;qs=v;ks=v;ls=v;

    %starting calculation

    S_UU_quasi=zeros(v,v);


  for n=1:1:(25*10)

        omega(n)=n/10;% plotting omega for every interval
        omega_quasi(n)=0;% plotting omega for every interval

        S_UU_ls_quasi=0;
        S_UU_quasi(ii,j)=0;

        for ps=1:1:v

            for qs=1:1:v

                for ks=1:1:v

                    for ls=1:1:v


S_UU_ls_quasi=S_UU_ls_quasi+((((E_ben(ii,ks))*(E_ben(ps,ks)))/((k_k(ks)-
omega_quasi(n)^2*m_k(ks)+i*omega_quasi(n)*c_k(ks)))))*((((E_ben(j,ls))*(E_b
en(qs,ls)))/((k_k(ls)-omega_quasi(n)^2*m_k(ls)-
i*omega_quasi(n)*c_k(ls)))))*(S_FF(ps,qs,n));


                    end
```

```
                end

            end

        end

        S_UU_psf_quasi(n)=S_UU_ls_quasi;

 end

figure
h=semilogy(omega,abs(S_UU_psf_quasi));

xlabel('\omega(rad/s)','FontSize',8)
ylabel('S_u_u_q_u_a_s_i(m2s/rad)','FontSize',8)


%% Dynamic part of the response spectra


%------------------------------------------------------------
%  Spectrum of displacement 4.32 CT4145  page 8
%------------------------------------------------------------



    ii=48;
    j=48;

    ps=v;qs=v;ks=v;ls=v;

    %starting calculation

    S_UU_dyn=zeros(v,v);


 for n=1:1:(25*10)

        omega(n)=n/10;% ploting omega for every interval

        S_UU_ls_dyn=0;
        S_UU_dyn(ii,j)=0;

        for ps=1:1:v

            for qs=1:1:v

                for ks=1:1:v

                    for ls=1:1:v


S_UU_ls_dyn=S_UU_ls_dyn+((((E_ben(ii,ks))*(E_ben(ps,ks)))/((k_k(ks)-
```

```
omega(n)^2*m_k(ks)+i*omega(n)*c_k(ks)))))*((((E_ben(j,ls))*(E_ben(qs,ls)))/
((k_k(ls)-omega(n)^2*m_k(ls)-i*omega(n)*c_k(ls)))))*(S_FF(ps,qs,n));


                    end

                end

            end

        end


        S_UU_psf_dyn(n)=S_UU_ls_dyn;

 end


 figure
h=semilogy(omega,abs(S_UU_psf_dyn));

xlabel('\omega(rad/s)','FontSize',8)
ylabel('S_u_u_d_y_n(m2s/rad)','FontSize',8)


 %% Adding the quasi-static and dynamic responce spectra


for n=1:1:(25*10)

 S_UU_psf(n)=S_UU_psf_quasi(n)+S_UU_psf_dyn(n);

end

figure
h=semilogy(omega,abs(S_UU_psf));

xlabel('\omega(rad/s)','FontSize',8)
ylabel('S_u_u_d_y_n(m2s/rad)','FontSize',8)


 %% Calculation of the acceleration spectra


  for n=1:1:(25*10)

        omega(n)=n/10;% ploting omega for every interval

        S_aa_psf(n)=S_UU_psf(n)*omega(n)^2;

 end

 figure
h=semilogy(omega,abs(S_aa_psf));
```

```
xlabel('\omega(rad/s)','FontSize',8)
ylabel('S_a_a(m^2/s^4)','FontSize',8)


%% Plotting and saving the figures

toc


I_spectrums_plot

save Output_spectral.mat


%----------------------------------------------------
%Getting the standard deviation of the acceleration
%----------------------------------------------------

maximum_value
```

admittance_2_run.m

```
function ADM=admittance_2_run(n,omega,yr,zr,U)
% syntax: function ADM=admittance_2_run(f,Yrtot,Zrtot,Vhub,zhub,option)
% Aerodynamic Admittance
%
% Input:
%
%   n:      number counted which divides the omega spectra              [-]
%   omega:  cyclic frequency
[Rad/s]
%   Yr:     lateral node places                                         [m]
%   Zr:     longditudinal node places                                   [m]
%   U:      the 10 minute average wind speed at hub height
[m/s}
%
%
% Output:
%   ADM: aerodynamic admittance (-)
%
% Thesis: Torsion motions of high-rise buildings due to wind loading
% Author: H.A.O. Richardson (Anthony)


% number of points in rotor plane
Ny=length(yr);
Nz=length(zr);
Np=Ny*Nz;

% y and z coordinates of all rotor points in one column vector
Yr=reshape(yr'*ones(1,Nz),Np,1);
Zr=reshape(ones(Ny,1)*zr,Np,1);

Yrtot=zeros(Np,Np);
for i=1:Np
   for j=i+1:Np
      % distances between points
      Yrtot(i,j)=Yr(i)-Yr(j);
      Yrtot(j,i)=Yrtot(i,j);
   end
end


Zrtot=zeros(Np,Np);
for i=1:Np
   for j=i+1:Np
      % distances between points
      Zrtot(i,j)=(Zr(i)-Zr(j));
      Zrtot(j,i)=Zrtot(i,j);
   end
end


f=omega(n)/(2*pi());

Cz=7;                       % Londitudinal  Coherence
Cy=10;                       % Lateral      Coherence
```

```
x=2*f.*((sqrt((Cz.^2).*((Zrtot).^2)+(Cy^2).*((Yrtot).^2)))/U);
ADM=1./(1+((x).^(4/3)));
```

coherence_2_run.m

```
function Coh=coherence_2_run(n,omega,yr,zr,U)
% syntax: function Coh=coher(f,Yrtot,Zrtot,Vhub,zhub,option)
% Coherency function of longitudinal wind velocity fluctuations
%
% Input:
%
%   n:      number counted which divides the omega spectra               [-]
%   omega:  cyclic frequency
[Rad/s]
%   Yr:     lateral node places                                          [m]
%   Zr:     longditudinal node places                                    [m]
%   U:      the 10 minute average wind speed at hub height
[m/s}
%
%
% Output:
%   Coh: coherency (-)
%
% Thesis: Torsion motions of high-rise buildings due to wind loading
% Author: H.A.O. Richardson (Anthony)


% number of points in rotor plane
Ny=length(yr);
Nz=length(zr);
Np=Ny*Nz;

% y and z coordinates of all rotor points in one column vector
Yr=reshape(yr'*ones(1,Nz),Np,1);
Zr=reshape(ones(Ny,1)*zr,Np,1);




Yrtot=zeros(Np,Np);
for i=1:Np
   for j=i+1:Np
      % distances between points
      Yrtot(i,j)=Yr(i)-Yr(j);
      Yrtot(j,i)=Yrtot(i,j);
   end
end




Zrtot=zeros(Np,Np);
for i=1:Np
   for j=i+1:Np
      % distances between points
      Zrtot(i,j)=(Zr(i)-Zr(j));
      Zrtot(j,i)=Zrtot(i,j);
   end
end


f=omega(n)/(2*pi());
```

```
Cz=7;                      % Londitudinal  Coherence
Cy=10;                      % Lateral      Coherence


    x=f.*((sqrt((Cz.^2).*((Zrtot).^2)+(Cy^2).*((Yrtot).^2)))/U);
    Coh=exp(-1.*x);
```

Transversion_E_ben.m

```
%Making the Modal eigen vectors (start with 1 in first column)
%
% Thesis: Torsion motions of high-rise buildings due to wind loading
% Author: H.A.O. Richardson (Anthony)


% making the bottom value of the eigen vector 1 and recalculating the other
% values.

E_ben_modified=zeros(v);


% putting 1 in the top row of the matrix

for n=1:1:v;

    E_ben_modified(1,n)=1;

end


% fulling in the calculated values in the first column vector

for n=2:1:v;
    for p=1:1:v;

        E_ben_modified(n,p)=2;

    end
end


% N Dof


for p=1:1:v;
    for n=2:1:v;

        if n==2;
            k_x=(E_ben_modified(n-1,p)/E_ben(n-1,p));
            E_ben_modified(n,p)=(E_ben_modified(n-1,p)/E_ben(n-
1,p))*E_ben(n,p);

        else
            E_ben_modified(n,p)=k_x*E_ben(n,p);

        end
    end

end

E_ben=E_ben_modified;
```

Maximum_value.m

```
%
% Thesis: Torsion motions of high-rise buildings due to wind loading
% Author: H.A.O. Richardson (Anthony)
%
%
%
%-------------------------------------------------------------------------
% Displacement Spectra
%-------------------------------------------------------------------------
%
%
mu_u=0;          % Mean of the displacement spectra                 [-]

sigma_u_square=trapz(omega,S_UU_psf);    % Variance of the displacement
spectra    [-] % 48 dof
sigma_u=sqrt(sigma_u_square); % Standard deviation of the displacement
spectra    [-] % 48 dof

T=800;           % Period of a signal
[seconds]

omega_eig_ben_1=omega_eig_ben(1);    % angular frequency [rad/s] %
omega=2*pi*f

f_0=omega_eig_ben_1/(2*pi); % radial frequency [Hz] or [1/s]   % f =
omega/(2*pi)

u_star=mu_u+sigma_u*(2*log(T*f_0))^.5; % N Dof



%-------------------------------------------------------------------------
% Acceleration Spectra
%-------------------------------------------------------------------------

mu_a=0;          % Mean of the acceleration spectra                 [-]

sigma_a_square=trapz(omega,S_aa_psf);      % Variance of the acceleration
spectra    [-] % 48 dof
sigma_a=sqrt(sigma_a_square);    % Standard deviation of the acceleration
spectra    [-] % 48 dof

T_a=800;           % Period of a signal
[seconds]

omega_eig_ben_1=omega_eig_ben(1);    % angular frequency [rad/s] %
omega=2*pi*f

f_0=omega_eig_ben_1/(2*pi); % radial frequency [Hz] or [1/s]   % f =
omega/(2*pi)

a_star=mu_a+sigma_a*(2*log(T_a*f_0))^.5; % N Dof
```

I_spectums_plot.m

```
%
% Thesis: Torsion motions of high-rise buildings due to wind loading
% Author: H.A.O. Richardson (Anthony)
%
%


figure
h=semilogy(omega,abs(S_UU_psf));

xlabel('\omega(rad/s)','FontSize',8)
ylabel('S_u_u(m2s/rad)','FontSize',8)

saveas(gcf, 'S_UU_psf.jpeg', 'jpeg');




S_vv_N=(zeros(1,size(S_vv,3)));

for xx=1:1:size(S_vv,3);
    S_vv_N(xx)=S_vvco(v,v,xx);
end


figure
h=loglog(omega,S_vv_N);

axis([1e-2 1e1 1e-1 1e2])
set(h,'LineWidth',.1)
set(gca,'FontSize',8)
xlabel('\omega(rad/s)','FontSize',8)
ylabel('S_v_v(m^2/s^2)','FontSize',8)
set(gcf, 'PaperPositionMode', 'manual');
set(gcf, 'PaperUnits', 'centimeters');
set(gcf, 'PaperPosition', [4 4 15 6]);

saveas(gcf, 'S_vv_N.jpeg', 'jpeg');



S_FF_N=(zeros(1,size(S_FF,3)));

for xx=1:1:size(S_vv,3);
    S_FF_N(xx)=S_FFco(v,v,xx);
end


figure
h=loglog(omega,S_FF_N);
set(h,'LineWidth',.1)
set(gca,'FontSize',8)
xlabel('\omega(rad/s)','FontSize',8)
ylabel('S_F_F(N^2)','FontSize',8)
set(gcf, 'PaperPositionMode', 'manual');
set(gcf, 'PaperUnits', 'centimeters');
set(gcf, 'PaperPosition', [4 4 15 6]);
```

```
saveas(gcf, 'S_FF_N.jpeg', 'jpeg');

figure
h=semilogy(omega,abs(S_aa_psf));
set(h,'LineWidth',.1)
set(gca,'FontSize',8)
xlabel('\omega(rad/s)','FontSize',8)
ylabel('S_a_a(m^2/s^4)','FontSize',8)
set(gcf, 'PaperPositionMode', 'manual');
set(gcf, 'PaperUnits', 'centimeters');
set(gcf, 'PaperPosition', [4 4 15 6]);

saveas(gcf, 'S_aa_psf', 'jpeg');


% saving values in matrices

save matrices omega S_UU_psf S_vv S_vv_N S_vvco S_FF_N S_FF S_FFco S_aa_psf
-append
```

Appendix 14 Matlab code SDOF Juffertoren (frequency domain analysis)

Below the Matlab code is given for the Juffertoren for return period of one year, the only difference with the Matlab code for return period of 12.5 years is that $v_{10} = 22.62 m/s$, $\sigma_v = 6.23 m/s$ and $v_{144} = v\_mean(1) = 39.60 m/s$ (6.3) are inputted in Matlab file: S_FF_we.m

For this code to run take generated file matrices.mat from the time domain analysis file folder.

S_dof_approximation.m

```
% 3.3 Responce to a arbitrary load [1]
%
% Thesis: Torsion motions of high-rise buildings due to wind loading
% Author: H.A.O. Richardson (Anthony)


close all;
clear;


% Determining S_FF(we)


ve=10000;     % length of omega divided by 100
deltaom=0.01;  % splitting 1 radian into 100 intervals

S_FF_we


From_48_DOF_to_1_Dof

% Formulas

w_e=(k/m)^.5;

zeta= (c)/(2*sqrt(k*m));


%original S_uu

omega=20;

S_uu_original=(S_0/(((k^2)*((1-(omega/w_e)^2)^2+(2*zeta*(omega/w_e))^2))));
```

```
% Plotting S_uu omega variable

deltaom=0.01;

for n=1:1:(ve)

    omega(n)=n*deltaom;


    S_uu(n)=(S_FF_N(n)./k.^2)+(S_0./(k^2*(((1-
(omega(n)./w_e).^2).^2)+(2*zeta*(omega(n)./w_e)).^2)));

    % checked with S_uu original formula is correct
    %S_aa=S_uu.*omega.^2;

    S_aa(n)=S_uu(n).*omega(n).^2;
    % the first value of the displacement spectrum has to be according to
    % figure 3.3. page 3 S_0/k^2 is corrrect 3*10-5
end


    S_uu(1);

    S_0/k^2;


    % Plotting Displacement Spectra

    figure

    plot (omega,S_uu);
    grid
    title('S_u_u')
    axis ([0 10 0 .10])

    figure
    plot (omega,S_aa);
    grid
    title('S_a_a')
    axis ([0 10 0 .15])

% Determining the Variance of the Displacement Spectra


Sigma_u_square_white_noise=(pi*w_e*S_0)/(4*zeta*k^2);

Sigma_F_square = trapz(omega,S_FF_N);
Sigma_u_square=((Sigma_F_square)/k^2)+(pi*w_e*S_0)/(4*zeta*k^2)

Sigma_u=Sigma_u_square^.5;


% Integrating the variance of Displacement Spectra

    Q = trapz(omega,S_uu)
```

```
    Q_root=sqrt(Q);

% Integrating the variance of Acceleration Spectra

    Sigma_a_square = trapz(omega,S_aa);
    Sigma_a=sqrt(Sigma_a_square);
```

S_FF_we.m

```
% Determining the load of the force spectra for the first natural
% frequency (S_FF(we))
%
% Thesis: Torsion motions of high-rise buildings due to wind loading
% Author: H.A.O. Richardson (Anthony)

load matrices omega_eig_ben % Loading of the eigenfrequencies

% First we have to determine the velocity spectra
%------------------------------------------------------------------------
% Determing the velocity spectra
%------------------------------------------------------------------------


v=1;                    % degrees of freedom
L=1200;                 % characteristic length Davenport
v_10=21.45;             % mean wind speed at 10 m height
sigma_v=2.52;           % standard deviation of the wind speed variation with
height


S_vv=zeros(v,v,ve);
S_FF=zeros(v,v,ve);
S_vv_1=zeros(v,v,ve);


for n=1:1:(ve)

    omega(n)=n*deltaom;


    for p=1:1:v

S_vv(p,p,n)=((omega(n).*L./(2.*pi.*v_10)).^2)./((1+(omega(n).*L./(2.*pi.*v_
10)).^2).^(4/3)).*(((2/3).*sigma_v.^2)./omega(n));
    end

end


% plotting velocity spectra

for xx=1:1:size(S_vv,3);
    S_vv_N(xx)=S_vv(v,v,xx);
end

figure
h=loglog(omega,S_vv_N);
title('S_v_v loglog')

figure
plot(omega,S_vv_N);
```

```
title('S_v_v')
axis([0 10 0 80])


% Next we have to determine the force spectra ( top of the buiding to be
% conservitive)


%-----------------------------------------------------------------------
% Determing the force spectra
%-----------------------------------------------------------------------

rho=1.25;            % Air density
A=3*26.34;           % Area of the building surfice one node height *
Ch=1.2;              % Summation of suction and drag coefficient
v_mean(1)=37.62;     %velociy at node 48 .... the velocity will always be
smaller than this so it is conservative.


for n=1:1:(ve)
    omega(n)=n*deltaom;

    % Determining the force spectrum for ever node on diagonal
    for d=1:1:v

            S_FF(d,d,n)=(rho.*A.*Ch.*v_mean(d)).^2.*S_vv(d,d,n); % CT4145
Page 26
            % no aerodynamic admittance in the function

            % At ever node there is a different velocity,for each velocity
a
            % force spectra must be determined.
    end
end


% plotting force spectra

for xx=1:1:size(S_FF,3);
    S_FF_N(xx)=S_FF(v,v,xx);
end

figure
h=loglog(omega,S_FF_N);
title('S_F_F loglog')

figure
plot(omega,S_FF_N);
title('S_F_F')


% Next we determine the S_FF(we)

omega_e=round(omega_eig_ben(1)*(1/deltaom));

S_0=S_FF_N(omega_e);
```

From_48_DOF_to_1_Dof.m

```
% Transfering from 48 DOF to 1 Dof for M,C,K-Matrix
%
% Thesis: Torsion motions of high-rise buildings due to wind loading
% Author: H.A.O. Richardson (Anthony)


% Loading the M, C, K matrix of the 48 DOF system and eigenfrequencies

load matrices K_ben Cd_ben M_ben omega_eig_ben v EI L


% Determining the total mass of the building

M_total=0;

for s=1:1:v

    M_total(s)=M_ben(s,s);

end

% M_s_dof=0.24*rho*A*l; % Equivilent mass page 79 CT 4140
M_s_dof=0.24*675.54*405.02*144; % Equivilent mass page 79 + 80 CT 4140


omega_eig_ben(1); % Natural eigenfrequency

% K_s_dof_mod=3*3*1000*1000*1000*10*667/(144^3) % K=(3*EI)/l^3
K_s_dof_mod=(3*EI)/(L^3); % spring stiffness for one dof Page .. CT 2022

w_e=sqrt(K_s_dof_mod/M_s_dof);

k=K_s_dof_mod;
m=M_s_dof;


% to determine c we look at Reader CT 2022 page 58

c_kr=2*sqrt(k*m);

zeta=0.01;

c=zeta*c_kr;
```

Appendix 15 Matlab code NDOF Voorhof after renovation (frequency domain analysis)

Below the Matlab code is given for the Voorhof after renovation for return period of one year, the only difference with the Matlab code for return period of 12.5 years is that $v_{10} = 15.79 m/s$ and $\sigma_v = 6.65 m/s$ (9.6) are inputted in Matlab file: Spectra_acceleration.m. For the Voorhof before renovation the a different file: matrices.mat has to be loaded.

For this code to run take generated file matrices.mat from the time domain analysis file folder. Frequency domain files what are the same as Juffertoren (Appendix 13) are not given.

Spectra_acceleration.m

```
clear
close all

%
% Calculating the spectra of acceleration
%
%   Input: K_ben M_ben Cd_ben
%
%   Output:  Saa
%
% Author: H.A.O.Richardson (Anthony)
% Thesis: Wind induced torsions in high-rise buildings due to wind loading.
%
%-------------------------------------------------------------------------

tic

% Loading Mass-,  Stiffness-, Damping- Velocity-, eigen matrix bending
matrix

%load matrices M_ben K_ben Cd_ben v_mean omega_eig_ben E_ben V_Hub yr zr
load matrices M_ben K_ben Cd_ben v_mean omega_eig_ben E_ben

zr=2.65:2.65:50.35; %19 ndes of freedom system

yr(:)=0; % there is no coherence in y direction

%---------------------------------------------------------------


v=19;                   % Bending degrees of freedom
L=1200;                 % characteristic length Davenport
sigma_v=2.44;           % standard deviation of the wind speed variation with
height
```

```
v_10=21.45;           % mean wind speed at 10 m height (Urban 2)


U=v_10;
V_Hub=U;

rho=1.25;            % Air density
A=2.65*80.81;           % Area of the building surfice one node height *
%Building width

Ch=1.2;              % Summation of suction and drag coefficient

omega_max=max(omega_eig_ben)+200; % Maximum value of the natural frequency
matrix plus 200


delta=.1;

ve=93*10; % Length of omega interval devided bij amount of steps.


%-------------------------------------------------------------------------
% Putting velocity in 19 node in correspondence with the nodes of structure
%-------------------------------------------------------------------------

for c=1:1:v-3 % 3 from the bottom missing
        v_mean_19(1,3+c)=v_mean((2*c-1));
end
        v_mean_19(1,1)=v_mean_19(1,4);
        v_mean_19(1,2)=v_mean_19(1,4);
        v_mean_19(1,3)=v_mean_19(1,4);




%-----------------------------------------------------------
% Fixing the correct Eigenvectors
%-----------------------------------------------------------

Transversion_E_ben


for n=1:1:v;

    m_k(n)=transpose(E_ben(:,n))*M_ben*E_ben(:,n);

end

xx=transpose(E_ben(:,1));


   for n=1:1:v;

    c_k(n)=transpose(E_ben(:,n))*Cd_ben*E_ben(:,n);

   end
```

```
    for n=1:1:v;

    k_k(n)=transpose(E_ben(:,n))*K_ben*E_ben(:,n);

    end


%------------------------------------------------------------------------
% Determing the velocity spectra
%------------------------------------------------------------------------


S_vv=zeros(v,v,2);
S_FF=zeros(v,v,2);
S_vv_1=zeros(v,v,2);


for n=1:1:(ve);

    omega(n)=n/10;% ploting omega for every interval

    for p=1:1:v

    %-----------------------------------------------------------------------
--
    % Depending on height (only autospectra)
    %-----------------------------------------------------------------------
--



    % making 48 *48 matrix to be able to do the coherence

S_vv(p,p,n)=((omega(n).*L./(2.*pi.*v_10)).^2)./((1+(omega(n).*L./(2.*pi.*v_
10)).^2).^(4/3)).*(((2/3).*sigma_v.^2)./omega(n));


    end


    %-----------------------------------------------------------------------
    % Depending on height (autospectra and cross spectra)
    %-----------------------------------------------------------------------





    %-----------------------------------------------------------------------
    % Coherence of velocity spectra
    %-----------------------------------------------------------------------


    Coh=cohorence_2_run(n,omega,yr,zr,U);
    Cohm=Coh;
```

```
    %-----------------------------------------------------------------------
---

    for aa=1:1:v
        for bb=1:1:v
                if (aa==bb)



S_vv(aa,bb,n)=(Cohm(aa,bb).*(S_vv(aa,aa,n).*S_vv(bb,bb,n)).^.5);

                else


S_vv(aa,bb,n)=(Cohm(aa,bb).*(S_vv(aa,aa,n).*S_vv(bb,bb,n)).^.5);
                end
        end
    end


    S_vvco(:,:,n)=S_vv(:,:,n);




    %-----------------------------------------------------------------------
    % Force spectra of the nodes
    %-----------------------------------------------------------------------


    % Determining the force spectra for ever node on diagonal
    for d=1:1:v


            S_FF(d,d,n)=(rho.*A.*Ch.*v_mean(d)).^2.*S_vv(d,d,n); % CT4145
Page 26 edit 20-04-2019
            % no aerodynamic admittance in the function

            % At ever node there is a different velocity,for each velocity
a
            % force spectra must be determined.
    end

    %-----------------------------------------------------------------------
--
    % Aerodynamic admittance
    %-----------------------------------------------------------------------

    ADM=admittance_2_run(n,omega,yr,zr,U);

    %-----------------------------------------------------------------------
---
    % Fitting the aerodynamic admittance into the force spectrum
    %-----------------------------------------------------------------------
```

```
    for aaa=1:1:v
        for bbb=1:1:v
                if  (aaa==bbb)

S_FF(aaa,bbb,n)=S_vv(aaa,bbb,n).*((rho.*A.*Ch).^2).*v_mean(aaa).*v_mean(bbb
).*(ADM(aaa,bbb)^2);

                else

S_FF(aaa,bbb,n)=S_vv(aaa,bbb,n).*((rho.*A.*Ch).^2).*v_mean(aaa).*v_mean(bbb
).*(ADM(aaa,bbb)^2);
                end
        end
    end

S_FFco(:,:,n)=S_FF(:,:,n);


end


%% Quasi-static part of the response spectra


    ii=19;
    j=19;

    ps=v;qs=v;ks=v;ls=v;

    %starting calculation

    S_UU_quasi=zeros(v,v);


 for n=1:1:(ve)

        omega(n)=n/10;% plotting omega for every interval
        omega_quasi(n)=0;% plotting omega for every interval

        S_UU_ls_quasi=0;
        S_UU_quasi(ii,j)=0;

        for ps=1:1:v

            for qs=1:1:v

                for ks=1:1:v

                    for ls=1:1:v


S_UU_ls_quasi=S_UU_ls_quasi+((((E_ben(ii,ks))*(E_ben(ps,ks)))/((k_k(ks)-
omega_quasi(n)^2*m_k(ks)+i*omega_quasi(n)*c_k(ks)))))*((((E_ben(j,ls))*(E_b
en(qs,ls)))/((k_k(ls)-omega_quasi(n)^2*m_k(ls)-
i*omega_quasi(n)*c_k(ls)))))*(S_FF(ps,qs,n));
```

```
                    end

                end

            end

        end


        S_UU_psf_quasi(n)=S_UU_ls_quasi;

 end

figure
h=semilogy(omega,abs(S_UU_psf_quasi));

xlabel('\omega(rad/s)','FontSize',8)
ylabel('S_u_u_q_u_a_s_i(m2s/rad)','FontSize',8)



%% Dynamic part of the response spectra


%------------------------------------------------------------
%  Spectrum of displacement 4.32 CT4145  page 8
%------------------------------------------------------------

    ii=19;
    j=19;

    ps=v;qs=v;ks=v;ls=v;

    %starting calculation

    S_UU_dyn=zeros(v,v);


 for n=1:1:(ve)

        omega(n)=n/10;% plotting omega for every interval

        S_UU_ls_dyn=0;
        S_UU_dyn(ii,j)=0;

        for ps=1:1:v

            for qs=1:1:v

                for ks=1:1:v

                    for ls=1:1:v
```

```
S_UU_ls_dyn=S_UU_ls_dyn+((((E_ben(ii,ks))*(E_ben(ps,ks)))/((k_k(ks)-
omega(n)^2*m_k(ks)+i*omega(n)*c_k(ks)))))*((((E_ben(j,ls))*(E_ben(qs,ls)))/
((k_k(ls)-omega(n)^2*m_k(ls)-i*omega(n)*c_k(ls)))))*(S_FF(ps,qs,n));


                    end

                end

            end

        end


        S_UU_psf_dyn(n)=S_UU_ls_dyn;

 end


 figure
h=semilogy(omega,abs(S_UU_psf_dyn));

xlabel('\omega(rad/s)','FontSize',8)
ylabel('S_u_u_d_y_n(m2s/rad)','FontSize',8)




 %% Adding the quasi-static and dynamic responce spectra



for n=1:1:(ve)

 S_UU_psf(n)=S_UU_psf_quasi(n)+S_UU_psf_dyn(n);


end

figure
h=semilogy(omega,abs(S_UU_psf));

xlabel('\omega(rad/s)','FontSize',8)
ylabel('S_u_u_d_y_n(m2s/rad)','FontSize',8)





 %% Calculation of the acceleration spectra


  for n=1:1:(ve)
```

```
        omega(n)=n/10;% plotting omega for every interval

        S_aa_psf(n)=S_UU_psf(n)*omega(n)^2;


  end

 figure
h=semilogy(omega,abs(S_aa_psf));

xlabel('\omega(rad/s)','FontSize',8)
ylabel('S_a_a(m^2/s^4)','FontSize',8)



%% Plotting and saving figures

toc

I_spectrums_plot

save Output_spectral.mat


%---------------------------------------------------
%Getting the standard deviation of the acceleration
%---------------------------------------------------

maximum_value
```

Appendix 16 Matlab code SDOF Voorhof after renovation (frequency domain analysis)

Below the Matlab code is given for the Voorhof after renovation for return period of one year, the only difference with the Matlab code for return period of 12.5 years is that $v_{10} = 15.79 m/s$, $\sigma_v = 6.65 m/s$ and $v_{51.3} = v\_mean(1) = 30.17 m/s$ (9.6) are inputted in Matlab file: S_FF_we.m. For the Voorhof before renovation the a different file: matrices.mat has to be loaded.

For this code to run take generated file matrices.mat from the time domain analysis file folder.

S_dof_approximation.m

```
% 3.3 Responce to a arbitrary load [1]

close all;
clear;

tic

% Determining S_FF(we)


ve=9300;     % length of omega divided by 100
deltaom=0.1;  % splitting 1 radian into 100 intervals

S_FF_we


% values from 19 by 19 matrix

From_19_DOF_to_1_Dof


% Formulas

w_e=(k/m)^.5;

zeta= (c)/(2*sqrt(k*m));

omega=20;

S_uu_original=(S_0/(((k^2)*((1-(omega/w_e)^2)^2+(2*zeta*(omega/w_e))^2))));


% Plotting S_uu omega variable
```

```
deltaom=0.01;

for n=1:1:(ve)

    omega(n)=n*deltaom;

    S_uu(n)=(S_FF_N(n)./k.^2)+(S_0./(k^2*(((1-
(omega(n)./w_e).^2).^2)+(2*zeta*(omega(n)./w_e)).^2)));


    S_aa(n)=S_uu(n).*omega(n).^2;

end


    S_uu(1);

    S_0/k^2;


    % Plotting Displacement Spectra

    figure

    plot (omega,S_uu);
    grid
    title('S_u_u')
    %axis ([0 10 0 .10])

    figure
    plot (omega,S_aa);
    grid
    title('S_a_a')
    %axis ([0 10 0 .15])


% Determining the Variance of the Displacement Spectra


Sigma_u_square_white_noise=(pi*w_e*S_0)/(4*zeta*k^2);

Sigma_F_square = trapz(omega,S_FF_N);
Sigma_u_square=((Sigma_F_square)/k^2)+(pi*w_e*S_0)/(4*zeta*k^2)

Sigma_u=Sigma_u_square^.5;


% Integrating the variance of Displacement Spectra

    Q = trapz(omega,S_uu)
    Q_root=sqrt(Q);

% Integrating the variance of Acceleration Spectra

    Sigma_a_square = trapz(omega,S_aa);
```

```
Sigma_a=sqrt(Sigma_a_square);
Sigma_a=sqrt(Sigma_a_square)


Toc
```

S_FF_we.m

```
%
% Thesis: Torsion motions in high-rise buildings due to wind loading
% Author: H.A.O. Richardson (Anthony)
%
%
% Determining the S_FF(we)

load matrices omega_eig_ben

% First we have to determine the velocity spectra

%-------------------------------------------------------------------------
% Determing the velocity spectra
%-------------------------------------------------------------------------


v=1;        %degrees of freedom
L=1200;     % characteristic length Davenport

v_10=21.45;        % mean wind speed at 10 m height (Urban 2)

sigma_v=2.44;      % standard deviation of the wind speed variation with
height


S_vv=zeros(v,v,ve);
S_FF=zeros(v,v,ve);
S_vv_1=zeros(v,v,ve);


for n=1:1:(ve)

    omega(n)=n*deltaom;

    for p=1:1:v

S_vv(p,p,n)=((omega(n).*L./(2.*pi.*v_10)).^2)./((1+(omega(n).*L./(2.*pi.*v_
10)).^2).^(4/3)).*(((2/3).*sigma_v.^2)./omega(n));
    end

end


% plotting velocity spectra

for xx=1:1:size(S_vv,3);
    S_vv_N(xx)=S_vv(v,v,xx);
end

figure
h=loglog(omega,S_vv_N);
title('S_v_v loglog')

figure
```

```
plot(omega,S_vv_N);
title('S_v_v')
axis([0 10 0 20])


% Next we have to determine the foce spectra (top of the buiding to be
% conservative)


%-------------------------------------------------------------------------
% Determing the force spectra
%-------------------------------------------------------------------------

rho=1.25;           % Air density
A=2.65*80.81;          % Area of the building surfice one node height *
%Building width

Ch=1.2;             % Summation of suction and drag coefficient

v_mean(1)=40.83;  %velociy at node 19 .... the velocity will always be
smaller than this so it is conservative.

for n=1:1:(ve)

    omega(n)=n*deltaom;

    % Determining the force spectra for ever node on diagonal
    for d=1:1:v

            S_FF(d,d,n)=(rho.*A.*Ch.*v_mean(d)).^2.*S_vv(d,d,n); % CT4145
Page 26
            % no aerodynamic admittance in the function

            % At ever node there is a different velocity,for each velocity
a
            % force spectra must be determined.
    end


end


% plotting force spectra

for xx=1:1:size(S_FF,3);
    S_FF_N(xx)=S_FF(v,v,xx);
end

figure
h=loglog(omega,S_FF_N);
title('S_F_F loglog')

figure
plot(omega,S_FF_N);
title('S_F_F')
%axis([0 10 0 80])
```

```
% Next we determine the S_FF(we)

omega_e=round(omega_eig_ben(1)*(1/deltaom));

S_0=S_FF_N(omega_e);
```

From_19_DOF_to_1_Dof.m

```
%
% Thesis: Torsion motions of high-rise buildings dueto wind loading
% Author: H.A.O. Richardson (Anthony)
%
%
% Transfering from 19 DOF to 1 Dof for M,C,K-Matrix


load matrices K_ben Cd_ben M_ben omega_eig_ben v EI L


% Determining the total mass of the building

M_total=0;

for s=1:1:v

    M_total(s)=M_ben(s,s);

end

M_s_dof=0.24*193*1047.50*51.3; % Equivilent mass page 79 + 80 CT 4140

% w_e_s_dof=sqrt(K_s_dof/M_s_dof);
omega_eig_ben(1);

K_s_dof_mod=(3*EI)/(L^3); % spring stiffness for one dof Page .. CT 2022

w_e=sqrt(K_s_dof_mod/M_s_dof);


k=K_s_dof_mod;
m=M_s_dof;


% to determine c we look at Reader CT 2022 page 58

c_kr=2*sqrt(k*m);

zeta=0.01;

c=zeta*c_kr;
```

Appendix 17 The Juffertoren output of the 100 simulations for return period of 12.5 years

	a_ben	a_tor	a_ben_tor		sigma_a_ben	sigma_a_tor	sigma_a_ben_tor
1	0.11155162	0.7240442	0.7529165		0.045232227	0.30425964	0.306685147
2	0.11154716	0.7535971	0.7523307		0.036706371	0.32850244	0.330266968
3	0.12207007	1.0275112	1.0439641		0.035139772	0.35889044	0.360490992
4	0.08123533	0.8335365	0.8895186		0.030133125	0.31225813	0.313517404
5	0.10649015	0.8463609	0.8026797		0.036741006	0.33094116	0.332579065
6	0.09709362	1.1307521	1.1032707		0.033369465	0.42835881	0.429987219
7	0.12203673	0.71412	0.7518118		0.038566534	0.26733601	0.270005669
8	0.07322891	0.7473262	0.7408825		0.025514988	0.27868555	0.278803869
9	0.14101176	1.2264156	1.1935362		0.036223971	0.4865884	0.487816669
10	0.08880326	0.6916713	0.7431123		0.033026409	0.21673336	0.219136729
11	0.12144215	0.9111311	0.9261201		0.038445007	0.40035031	0.401784007
12	0.0960403	0.9390306	0.96418		0.036537158	0.39780731	0.39933767
13	0.10590692	0.803761	0.8523703		0.03612345	0.3051248	0.307018163
14	0.10207226	0.7637198	0.7130528		0.033309708	0.28144942	0.283300145
15	0.09571262	1.0166797	1.0428508		0.032120528	0.40932801	0.410159476
16	0.10572411	0.6316116	0.6110616		0.035895362	0.2183409	0.220870349
17	0.10433771	1.0107509	1.0323002		0.036379406	0.32683324	0.328829347
18	0.09921264	0.9616628	0.9690588		0.033896618	0.40792548	0.409846019
19	0.11916774	1.2770978	1.3111538		0.037978374	0.38711023	0.38906324
20	0.13981833	0.9523267	0.967581		0.041539122	0.32315908	0.326658162
21	0.10212301	0.9262693	0.904917		0.031826813	0.34617496	0.347219038
22	0.09547567	0.7888494	0.7997443		0.032019423	0.28332144	0.284908351
23	0.11039158	0.5462329	0.5744725		0.042216799	0.21676839	0.221412682
24	0.13171787	0.7204073	0.7562632		0.041255215	0.21313584	0.217197241
25	0.10416771	1.6338502	1.6621087		0.039591149	0.58406417	0.585491302
26	0.12751228	0.9212298	0.9398589		0.039866864	0.40535691	0.407250369
27	0.09089154	0.5789663	0.6358961		0.031368715	0.22586404	0.228153975
28	0.08943601	1.2092585	1.1856382		0.031592693	0.45465738	0.4563995
29	0.13415781	1.376461	1.3726039		0.043328572	0.53524776	0.537351279
30	0.15054447	0.763181	0.7574961		0.041640781	0.26767704	0.270566379
31	0.13065983	0.79529	0.8208232		0.043233487	0.24437001	0.248277814
32	0.10094152	1.1494006	1.1607516		0.038896919	0.4238933	0.426084569
33	0.11291509	1.1492907	1.1569513		0.039993302	0.35994058	0.362214366
34	0.10648614	0.8206881	0.9268075		0.039011234	0.32223967	0.324345161
35	0.1227019	0.9654759	1.0449254		0.043479751	0.46199467	0.464531858
36	0.14317864	0.5946757	0.6223826		0.044515359	0.2204906	0.223769746
37	0.11538301	0.9727054	0.9461839		0.0406628	0.31518568	0.31680453
38	0.09644733	0.8677593	0.8143642		0.032433367	0.35651754	0.357668883
39	0.11370954	1.1524265	1.1689825		0.038258104	0.43551727	0.437273967
40	0.13974251	0.9469874	1.0257016		0.042549235	0.35605827	0.35906679
41	0.09797074	0.8191425	0.8745917		0.03294019	0.28403185	0.285821253
42	0.1145659	1.1371351	1.1722362		0.041707543	0.43238507	0.43434747
43	0.09642182	1.1995268	1.2137461		0.032154439	0.42101929	0.422354015
44	0.09262831	1.1624503	1.1095597		0.039713925	0.49429076	0.496064021
45	0.13460785	0.7684799	0.7604683		0.034111714	0.33278106	0.334155848
46	0.12248393	0.6111439	0.6368132		0.041109877	0.23812139	0.241670343
47	0.10740418	0.9199761	0.9146816		0.033453774	0.33998494	0.3411787
48	0.13134568	0.8374438	0.9246274		0.052261648	0.34454469	0.347869468
49	0.13513289	0.59933	0.5950603		0.038116629	0.20889168	0.213272253
50	0.08824578	0.8813119	0.8448471		0.031673141	0.38918575	0.390740443

51	0.09580143	0.8062081	0.8502503		0.038614348	0.27892686	0.281557931
52	0.07727698	1.2251641	1.2224122		0.027850488	0.46978914	0.470980064
53	0.11070123	1.0705018	1.1252116		0.043469381	0.45527329	0.456855312
54	0.11139118	0.7598042	0.7493865		0.03936605	0.38024833	0.382228226
55	0.09778569	0.81286	0.8386133		0.039958567	0.32637361	0.328661598
56	0.11458405	1.0809081	1.1318065		0.035376996	0.39657207	0.398451538
57	0.08261843	1.1702562	1.2096019		0.028760886	0.42131014	0.423015958
58	0.10373064	0.8772859	0.915561		0.037120748	0.26238075	0.264945704
59	0.13442928	0.9741576	1.0559203		0.041223902	0.38007053	0.381682196
60	0.13587528	0.7637833	0.7971412		0.039480519	0.29918353	0.301638626
61	0.10391039	0.7539139	0.7455163		0.038870177	0.32775544	0.330630914
62	0.13007813	0.8153754	0.8749848		0.044597976	0.33055357	0.333278841
63	0.12974387	1.5096973	1.5373902		0.035922622	0.55872755	0.560112709
64	0.08203779	0.9905184	0.9793398		0.030314132	0.36916452	0.370392922
65	0.09004452	1.1164703	1.1062243		0.030296912	0.35788098	0.35974387
66	0.09784234	1.0621523	1.0616422		0.033184098	0.38923999	0.391572271
67	0.11732801	1.1037101	1.0859689		0.043312559	0.46364103	0.465322342
68	0.10910267	0.6374185	0.6671115		0.037847288	0.2603165	0.263159439
69	0.12922448	0.8526041	0.8555102		0.045605869	0.29531877	0.297782079
70	0.10528255	0.7669827	0.8360539		0.032567044	0.31986756	0.322431036
71	0.11388114	1.1834371	1.2062057		0.032511507	0.44999605	0.451220532
72	0.09687676	0.6493315	0.7047312		0.033195809	0.27904404	0.281033794
73	0.11079533	0.9611422	0.9471013		0.037700275	0.36770925	0.369347995
74	0.11231311	0.9771492	1.0149984		0.041596316	0.41691538	0.419845326
75	0.12252081	0.8985467	0.9522642		0.040383648	0.37397927	0.376524539
76	0.09258126	1.1001019	1.1468076		0.02904632	0.38310239	0.384120826
77	0.12040577	1.20839	1.1740847		0.04806841	0.40583965	0.408735257
78	0.11993758	0.7813485	0.8196963		0.043245827	0.33125693	0.333705581
79	0.09132328	0.896939	0.8931612		0.030867305	0.32026334	0.321418751
80	0.12659561	0.9830988	1.0123525		0.043846424	0.43493308	0.437530457
81	0.13309465	0.9576078	1.0031816		0.047064159	0.29377657	0.297997385
82	0.11151291	0.5194993	0.536611		0.037799344	0.20076435	0.204361095
83	0.12568093	1.1553116	1.1198721		0.041052616	0.39546627	0.397190272
84	0.08540602	0.6915491	0.704115		0.033096045	0.25112652	0.253025573
85	0.08665458	1.0749553	1.0869305		0.02926637	0.41186051	0.412731835
86	0.12985464	0.8319342	0.8922802		0.041478877	0.27140727	0.274799743
87	0.1075108	0.742153	0.7084196		0.036035927	0.28034993	0.281892299
88	0.10199785	0.8730948	0.8577882		0.035818031	0.30330039	0.30523129
89	0.08609803	0.8628105	0.8601541		0.02713402	0.39743195	0.398807262
90	0.09728932	1.0090079	0.9877342		0.033325184	0.38348676	0.385046838
91	0.10023876	0.6917141	0.7398942		0.038193207	0.26712968	0.269175222
92	0.09909306	0.7866573	0.8036583		0.036017551	0.27079935	0.272771794
93	0.10937251	1.3202472	1.3643269		0.028758794	0.53360348	0.534489899
94	0.10229493	0.7771343	0.7706437		0.034776131	0.27534725	0.277754076
95	0.08740679	0.9927196	1.0131793		0.030698062	0.43257101	0.434160455
96	0.11044981	0.8396332	0.8216313		0.040653199	0.29233076	0.295234148
97	0.16412163	1.3239315	1.3273079		0.057281223	0.44499718	0.449037697
98	0.11201777	1.1198973	1.1272282		0.038716252	0.42393748	0.426235976
99	0.1223254	0.8216679	0.8598863		0.042505837	0.34846899	0.350903788
100	0.10820045	1.1216217	1.1246774		0.034769163	0.45157417	0.453210194

Average	0.1103	0.9308	0.9471		0.0374	0.3532	0.3553
	a_ben	a_tor	a_ben_tor		sigma_a_ben	sigma_a_tor	sigma_a_ben_tor
Max	0.1641	1.6339	1.6621		0.0573	0.5841	0.5855
Min	0.0732	0.5195	0.5366		0.0255	0.2008	0.2044
Max - Average	0.0538	0.7030	0.7150		0.0199	0.2309	0.2302
Average - Min	0.0371	0.4113	0.4105		0.0119	0.1524	0.1510
Percentage Max	48.79%	75.53%	75.49%		53.30%	65.37%	64.78%
Percentage Min	33.61%	44.19%	43.34%		31.71%	43.16%	42.48%

Appendix 18 The Voorhof before renovation with calibration output of the 100 simulations for return period of 12.5 years

	a_ben	a_tor	a_ben_tor		sigma_a_ben	sigma_a_tor	sigma_a_ben_tor
1	0.09649697	0.351435	0.3800669		0.028370068	0.09999785	0.103887581
2	0.14540427	0.4867301	0.6247119		0.038457497	0.10910995	0.109957001
3	0.12751248	0.5422838	0.5813574		0.03658601	0.12765911	0.125761152
4	0.15959592	0.3872076	0.4561543		0.036262761	0.09481892	0.094899466
5	0.14762725	0.3156306	0.3743804		0.041030072	0.09967165	0.100326703
6	0.1734521	0.4032311	0.4552789		0.048381454	0.11555663	0.121079228
7	0.15019483	0.4919103	0.5154553		0.041502701	0.11281542	0.112855361
8	0.12782259	0.3702924	0.4095992		0.038965912	0.10203626	0.102241538
9	0.13747845	0.4273922	0.4551484		0.040499428	0.09964971	0.102432878
10	0.12811846	0.3256745	0.3746686		0.036660426	0.11006152	0.107720172
11	0.12296435	0.659849	0.745356		0.036314136	0.11366039	0.112475043
12	0.1657138	0.5776488	0.6328011		0.040094384	0.12237036	0.121442329
13	0.15596778	0.3810602	0.4450258		0.037751248	0.09642428	0.095338617
14	0.12694036	0.3595883	0.4201109		0.041173853	0.10430388	0.103166627
15	0.17051178	0.3478351	0.4238272		0.043485582	0.11136169	0.106647472
16	0.11793732	0.3599206	0.4317439		0.037181927	0.10420938	0.106706257
17	0.15734492	0.3883633	0.5271338		0.049078363	0.10355698	0.101609115
18	0.13718916	0.3736449	0.4299218		0.038454404	0.10415848	0.105589189
19	0.14855405	0.4449869	0.5142121		0.039754527	0.09774031	0.092588603
20	0.1313605	0.3454546	0.3691304		0.041650078	0.09595602	0.098506235
21	0.13482855	0.3316444	0.3790342		0.043939146	0.10238576	0.106826957
22	0.13555948	0.3467348	0.4002336		0.039543674	0.1078547	0.110771679
23	0.11975998	0.3669065	0.3909308		0.03809152	0.10797074	0.106457106
24	0.15397082	0.298511	0.3833735		0.041685637	0.10239164	0.099013964
25	0.13343347	0.5351907	0.5604411		0.035328181	0.11457776	0.114089488
26	0.12794951	0.3537474	0.3910802		0.037619558	0.10753263	0.107212992
27	0.12729389	0.4224129	0.4684538		0.037867778	0.11859303	0.117487372
28	0.21121793	0.3511537	0.3652245		0.043646719	0.10703209	0.112017565
29	0.11532394	0.3580504	0.4337115		0.035552466	0.11261712	0.11158713
30	0.13112328	0.3454862	0.3906931		0.034329787	0.10350013	0.103557073
31	0.12266379	0.4431435	0.5034237		0.037049285	0.12380507	0.123014505
32	0.16672247	0.3517951	0.3578881		0.038924768	0.10146131	0.105833876
33	0.14775704	0.4401334	0.5040143		0.039897662	0.12142172	0.120579023
34	0.14756268	0.4296455	0.4616533		0.047742217	0.11788131	0.119317905
35	0.11159247	0.3239829	0.3594107		0.034596833	0.10903069	0.108812833
36	0.13783082	0.3511309	0.3688706		0.038622327	0.09658001	0.099178147
37	0.19519764	0.4895597	0.5504264		0.054803989	0.12469727	0.130231681
38	0.13784192	0.425173	0.4105403		0.042167971	0.10107685	0.10132235
39	0.11273059	0.3584888	0.432216		0.03724614	0.10240713	0.098366789
40	0.15064468	0.358091	0.4262803		0.04083239	0.11343239	0.112506468
41	0.11626953	0.3432656	0.4046392		0.035675635	0.09676606	0.09797102
42	0.1023177	0.3632445	0.4060627		0.032166809	0.10136995	0.098770275
43	0.13126381	0.346113	0.4070056		0.040047321	0.10191882	0.103737272
44	0.15456299	0.391567	0.4866338		0.042025769	0.10638957	0.107710902
45	0.1270725	0.4923767	0.5481358		0.039546052	0.10137904	0.095466907
46	0.13455244	0.3923775	0.4723063		0.037278563	0.13050725	0.129991701
47	0.15489461	0.3352314	0.4051269		0.045442732	0.09711204	0.09465116
48	0.11359417	0.3500217	0.4517313		0.039379507	0.10226301	0.105808147
49	0.13734674	0.4563548	0.4786169		0.038108706	0.1176309	0.121047408
50	0.14526906	0.4189269	0.4318263		0.040997761	0.11613052	0.11564979

51	0.1753547	0.2981933	0.392323		0.045566959	0.09942692	0.098188818
52	0.1354474	0.4191949	0.4734244		0.03370788	0.11350158	0.11160774
53	0.13884932	0.3680704	0.4911139		0.037412608	0.10218982	0.105018897
54	0.1283997	0.5534368	0.6244032		0.037418562	0.122866	0.122143746
55	0.13460068	0.4699993	0.5122507		0.03936288	0.10500092	0.105345061
56	0.1315033	0.4163951	0.4882474		0.039299678	0.10342747	0.101262536
57	0.13025327	0.4088049	0.4221163		0.042014836	0.1125111	0.116602499
58	0.13378487	0.4030406	0.4844331		0.039841884	0.1096787	0.108247257
59	0.13303794	0.3514702	0.3474696		0.037329139	0.09576697	0.102426657
60	0.13320796	0.3152209	0.395979		0.038298292	0.08945441	0.089945073
61	0.12014364	0.4406796	0.4875159		0.038024848	0.10904493	0.10604064
62	0.14859897	0.492519	0.5121671		0.046049981	0.1229759	0.123538375
63	0.13662566	0.4116096	0.4503083		0.035866665	0.11823777	0.116527472
64	0.17059657	0.4242329	0.510831		0.043168976	0.09765262	0.101764535
65	0.12300821	0.3734298	0.4414065		0.037416529	0.10288398	0.104031712
66	0.1468865	0.3813493	0.4503098		0.044613026	0.1014569	0.106581189
67	0.11964474	0.4139302	0.5215624		0.035037831	0.10192618	0.098414403
68	0.12728639	0.3893063	0.466125		0.034689838	0.11700555	0.118431364
69	0.11868043	0.4375379	0.4409886		0.035126296	0.11079614	0.10779417
70	0.16667924	0.4010538	0.4487732		0.045593702	0.1213264	0.118992552
71	0.14603586	0.3595576	0.4206958		0.03894886	0.10119474	0.098844495
72	0.15243331	0.4983419	0.5093322		0.044445257	0.11578715	0.113617826
73	0.14106339	0.4322909	0.5235611		0.046240201	0.1078804	0.103600887
74	0.15769345	0.3513879	0.4109179		0.035393499	0.10383696	0.099774167
75	0.1433859	0.3531954	0.443351		0.040707311	0.10573898	0.104238757
76	0.11704625	0.2986968	0.3632758		0.034768477	0.10534315	0.102966637
77	0.13362746	0.4062027	0.5083474		0.041835083	0.11999208	0.119880443
78	0.17342969	0.3473905	0.403453		0.044572452	0.10587119	0.100205391
79	0.11175981	0.3303821	0.3850093		0.034570399	0.10060112	0.100132861
80	0.17398775	0.4207488	0.4558118		0.042115642	0.10857504	0.11175142
81	0.15303541	0.3466968	0.4414335		0.048174618	0.11538078	0.115396219
82	0.15809405	0.3327233	0.4429751		0.044100881	0.10120486	0.105913118
83	0.135832	0.5352715	0.5191027		0.039210504	0.11279891	0.115187421
84	0.14967419	0.3466654	0.3696011		0.044346532	0.09274285	0.092730582
85	0.165844	0.4007122	0.4552371		0.047472458	0.10698722	0.111880416
86	0.12565201	0.2990651	0.3316148		0.040056389	0.09829931	0.099885341
87	0.13942383	0.5422419	0.5384726		0.036434504	0.10699189	0.107793013
88	0.14350422	0.3783525	0.4666652		0.036644465	0.11326301	0.112068567
89	0.12933969	0.4053049	0.434423		0.039074247	0.09267483	0.094960389
90	0.14576178	0.4047482	0.475232		0.041129122	0.10831271	0.108300794
91	0.16302922	0.3242777	0.4045413		0.043969651	0.10450883	0.103314743
92	0.12497326	0.4570684	0.5768062		0.03767892	0.12360524	0.123746292
93	0.11453794	0.3976571	0.436525		0.036207006	0.10820471	0.105210443
94	0.13519667	0.4139943	0.4340521		0.040337579	0.10616805	0.103541871
95	0.14605296	0.3155768	0.3664308		0.045310549	0.09804047	0.10174165
96	0.16812909	0.3281347	0.401734		0.045189926	0.0987467	0.104706395
97	0.11409273	0.42809	0.4499317		0.037275979	0.11492277	0.115795832
98	0.13114562	0.4345981	0.4367704		0.041020062	0.11108084	0.11235459
99	0.12295722	0.3408237	0.3927261		0.036259876	0.10776118	0.110096076
100	0.13892786	0.3919973	0.4663421		0.040668847	0.11194839	0.112259785

Average	0.1397	0.3970	0.4522		0.0399	0.1076	0.1078
	a_ben	a_tor	a_ben_tor		sigma_a_ben	sigma_a_tor	sigma_a_ben_tor
Max	0.2112	0.6598	0.7454		0.0548	0.1305	0.1302
Min	0.1023	0.2982	0.3316		0.0322	0.0895	0.0899
Max - Average	0.0715	0.2628	0.2931		0.0149	0.0229	0.0224
Average - Min	0.0374	0.0988	0.1206		0.0078	0.0181	0.0178
Percentage Max	51.14%	66.20%	64.82%		37.29%	21.31%	20.82%
Percentage Min	26.78%	24.89%	26.67%		19.42%	16.85%	16.56%

Appendix 19 The Voorhof after renovation with calibration output of the 100 simulations for return period once in 12.5 years

	a_ben	a_tor	a_ben_tor		sigma_a_ben	sigma_a_tor	sigma_a_ben_tor
1	0.08748323	0.2281594	0.2207814		0.022928099	0.06420448	0.067617603
2	0.08343268	0.1806239	0.2160944		0.022926468	0.06398548	0.066941249
3	0.07812935	0.23733	0.2296463		0.020923643	0.06578367	0.069742655
4	0.07498889	0.2082795	0.2364079		0.01934875	0.05676201	0.060530239
5	0.07812951	0.2635823	0.2830256		0.022750942	0.06886751	0.072293505
6	0.07009173	0.2328343	0.2283077		0.020467099	0.06186376	0.065497232
7	0.07205909	0.2287433	0.2528382		0.021318947	0.05897368	0.06231949
8	0.08463715	0.2367531	0.2519956		0.022310156	0.05800968	0.062075614
9	0.08740452	0.2391109	0.2391643		0.022123444	0.06284046	0.065928187
10	0.07328503	0.249453	0.2597503		0.019490812	0.06689762	0.069748819
11	0.09925264	0.2056909	0.2574586		0.02083733	0.06037943	0.063125497
12	0.0785463	0.2540401	0.2889154		0.022096088	0.05965718	0.064200866
13	0.09129559	0.2106645	0.2349373		0.022398842	0.05744393	0.060976972
14	0.06717727	0.2223697	0.243268		0.020748218	0.05825883	0.061755398
15	0.06212077	0.2290535	0.2456473		0.018834389	0.06491926	0.066095285
16	0.07332581	0.2405706	0.2745698		0.019071917	0.06574238	0.067687208
17	0.0855748	0.2085541	0.2290752		0.023093088	0.06177257	0.06565839
18	0.06422299	0.2738348	0.2974679		0.019830362	0.06777616	0.069246083
19	0.07188914	0.2500358	0.256723		0.019895679	0.06560618	0.066763143
20	0.06754884	0.2445067	0.258834		0.02155497	0.07272396	0.076133125
21	0.08147208	0.1910316	0.2035349		0.02281629	0.05761598	0.061126553
22	0.05896339	0.217577	0.2213009		0.020128232	0.06232499	0.065021138
23	0.09382381	0.2435451	0.2603529		0.024284692	0.06578894	0.069321463
24	0.07729272	0.2364881	0.2540303		0.019120373	0.06287577	0.064534327
25	0.08583633	0.2332802	0.2606821		0.0238697	0.06195219	0.065855529
26	0.08656234	0.2078057	0.2337801		0.024875053	0.05777183	0.061881537
27	0.09378299	0.2244188	0.250212		0.020528836	0.06407587	0.066015739
28	0.064008	0.2345957	0.2401558		0.019458442	0.06322424	0.064998768
29	0.08619728	0.1912027	0.2366897		0.02314442	0.06009877	0.0630542
30	0.08626555	0.2088685	0.2278196		0.022144238	0.05803004	0.061505937
31	0.07026398	0.2475704	0.269981		0.022259181	0.07084897	0.073685491
32	0.07064708	0.2215741	0.2403371		0.021817385	0.06173292	0.064791465
33	0.06990862	0.2613637	0.2781777		0.021024082	0.06783644	0.070104837
34	0.09099968	0.2258312	0.2384206		0.023132652	0.06895055	0.072623645
35	0.08395568	0.2285796	0.2558073		0.021598141	0.06255302	0.064941946
36	0.07775708	0.2484795	0.255256		0.020327872	0.06423521	0.06686882
37	0.07746178	0.2186691	0.2239203		0.020152614	0.06589457	0.067934369
38	0.07645566	0.2163365	0.2426436		0.020542099	0.05874916	0.06170581
39	0.08872555	0.2448551	0.2621398		0.022386809	0.07036159	0.073567101
40	0.06716545	0.2077217	0.2262275		0.020313638	0.05689091	0.059145265
41	0.08484253	0.1883053	0.2244157		0.02129132	0.05701925	0.061199213
42	0.08344099	0.2577632	0.2803028		0.023968075	0.06089766	0.065298602
43	0.07258027	0.2364256	0.25393		0.020996285	0.0644357	0.067454177
44	0.07346205	0.2856569	0.321719		0.019402897	0.06681873	0.06770359
45	0.06218325	0.2387973	0.2650501		0.021226447	0.06017099	0.063355529
46	0.10277139	0.3382014	0.3595778		0.024585392	0.07003386	0.073518688
47	0.07118809	0.2459559	0.2999107		0.018801957	0.05932458	0.061107563
48	0.08831073	0.2162611	0.2609306		0.019920528	0.06319581	0.065578853
49	0.06899945	0.2280141	0.2311113		0.021073101	0.06471865	0.067189369
50	0.07843753	0.236444	0.2534777		0.020513876	0.05665839	0.058824617

51	0.06955724	0.2551307	0.280121		0.021479398	0.06564061	0.067911055
52	0.08904057	0.2127689	0.2456701		0.022547118	0.05907937	0.061614912
53	0.07630033	0.2417611	0.2552657		0.022606589	0.06253608	0.066817893
54	0.07320222	0.2108218	0.2324764		0.022884091	0.06027073	0.06414595
55	0.0720731	0.2466039	0.2787788		0.023279655	0.0694646	0.072955599
56	0.10851564	0.2287132	0.2641741		0.024876678	0.06363638	0.067455292
57	0.09045926	0.2544316	0.2598717		0.021776155	0.05841701	0.061856505
58	0.08662741	0.2226464	0.2392706		0.019138709	0.06469715	0.066723068
59	0.07439614	0.2419094	0.2679006		0.021652615	0.06283449	0.066118768
60	0.09877084	0.2731522	0.3290516		0.022939162	0.07099487	0.074285614
61	0.07899144	0.2878056	0.3176168		0.020845676	0.06565255	0.067689447
62	0.06722461	0.2942286	0.2940876		0.021347553	0.06372721	0.067008846
63	0.07959919	0.2059838	0.2301513		0.021235409	0.05800744	0.061195151
64	0.09451489	0.2452086	0.2739212		0.023255588	0.06955204	0.073021515
65	0.0774023	0.2834301	0.3075374		0.020334194	0.06648289	0.069783049
66	0.06902913	0.2741004	0.2919384		0.019596141	0.06189101	0.064019548
67	0.0969677	0.2214668	0.2410853		0.023969064	0.06132233	0.064781374
68	0.07932535	0.2750316	0.2810199		0.023327298	0.06646147	0.070496318
69	0.07051959	0.2273734	0.2566797		0.020230285	0.05943879	0.061838866
70	0.08476181	0.2216636	0.2424201		0.022003345	0.05934493	0.063184433
71	0.07816418	0.3301339	0.3673266		0.02247661	0.06568833	0.068340517
72	0.07060447	0.2344967	0.2871761		0.021533292	0.06452004	0.067777138
73	0.06574023	0.260168	0.2811709		0.019823082	0.06446949	0.067285841
74	0.07517628	0.1977595	0.1969721		0.019493305	0.0626924	0.064667442
75	0.08787502	0.2239016	0.2099263		0.02068572	0.06009626	0.062986291
76	0.08088247	0.2394268	0.2496978		0.022703676	0.05381763	0.058173199
77	0.06378968	0.2554589	0.2350529		0.01964011	0.06442219	0.06611276
78	0.06544252	0.2137902	0.2172378		0.018201307	0.05953718	0.061716891
79	0.08681823	0.2566892	0.2679315		0.023090703	0.0603019	0.063709403
80	0.08017447	0.196986	0.2224708		0.019983934	0.06042992	0.063514748
81	0.06470438	0.264498	0.2685368		0.017896612	0.06241703	0.063474118
82	0.0731774	0.2678179	0.2794206		0.020597272	0.06076912	0.062841956
83	0.08895352	0.2135875	0.2397689		0.022984074	0.06153115	0.06575581
84	0.07777969	0.2444185	0.2519309		0.019419553	0.06673464	0.069325071
85	0.0905185	0.2487885	0.2622185		0.021895165	0.05786821	0.06112227
86	0.0926652	0.2368339	0.2661405		0.023326984	0.05408971	0.058474734
87	0.07249708	0.2374821	0.2526839		0.019778799	0.05973683	0.062639012
88	0.08548303	0.1978707	0.2312587		0.023874038	0.05758903	0.061543631
89	0.07011713	0.2457232	0.2529308		0.019736725	0.06579724	0.068058933
90	0.08352781	0.2557928	0.2848257		0.023409037	0.06644363	0.070212351
91	0.06061306	0.2434062	0.2406677		0.018452599	0.06458204	0.066269948
92	0.08084918	0.3004816	0.303575		0.020884712	0.06198633	0.064812225
93	0.05750803	0.2116206	0.2458869		0.01916567	0.06037551	0.062439342
94	0.07411018	0.1958417	0.2351502		0.020343714	0.05522453	0.057747261
95	0.08150271	0.214581	0.2274061		0.022311368	0.06272967	0.067254672
96	0.07102569	0.2477276	0.2767815		0.021459349	0.06765524	0.069871088
97	0.07622672	0.2399854	0.2665907		0.020507945	0.0642177	0.066494677
98	0.08915402	0.2768522	0.2649512		0.020149657	0.0654388	0.068932751
99	0.08169533	0.2349085	0.2744434		0.020897706	0.06269294	0.065632239
100	0.08629204	0.2036609	0.2687212		0.022050703	0.06119987	0.063910134

Average	0.0787	0.2374	0.2575		0.0213	0.0627	0.0657
	a_ben	a_tor	a_ben_tor		sigma_a_ben	sigma_a_tor	sigma_a_ben_tor
Max	0.1085	0.3382	0.3673		0.0249	0.0727	0.0761
Min	0.0575	0.1806	0.1970		0.0179	0.0538	0.0577
Max - Average	0.0298	0.1008	0.1098		0.0035	0.0100	0.0105
Average - Min	0.0212	0.0568	0.0606		0.0035	0.0089	0.0079
Percentage Max	37.80%	42.46%	42.63%		16.53%	15.95%	15.91%
Percentage Min	26.97%	23.92%	23.52%		16.17%	14.19%	12.08%

Appendix 20 Discrepancies in thesis Hans Breen

Formula Breen [5] equation 3.18 p.12 is not correct.

$$u(t) = \frac{F}{k}\left(1 - e^{-\varsigma\sqrt{\frac{k}{m_{eq}}}t} * \sin(\omega_e t)\right)$$

The correct formula **Fout! Verwijzingsbron niet gevonden.** p.64 under equation 4.68.

$$u(t) = \frac{F_0}{k}\left(1 - \frac{\omega_0}{\omega_e} e^{-\delta t} \cos\left(\omega_e t - \arctan\frac{\delta}{\omega_e}\right)\right)$$

With

$$\delta = \zeta\omega_n \qquad \omega_e = \omega_n\sqrt{1-\zeta^2}$$

Figure 4.2 Breen [5] p.16 is not correct.

For urban area II

$$v\_10 \text{ is not } 10\frac{m}{s} \text{ but } \bar{v}(10) = \frac{2.82}{0.9}\ln\left(\frac{10-3.5}{0.7}\right) = 15.71\frac{m}{s}$$

For vacant area II: because of open side with sea.

$$v\_10 \text{ is not } 10\frac{m}{s} \text{ but } \bar{v}(10) = \frac{2.30}{0.4}\ln\left(\frac{10-0}{0.2}\right) = 22.49\frac{m}{s}$$

Formula Breen [5] equation 4.9 p.17 is not correct.

The correct formula is:

$$S_{vv}(\omega) = \frac{1}{2\pi}\frac{2}{3}\frac{\left(\frac{\omega L}{2\pi\bar{v}(10)}\right)^2}{\left(1+\left(\frac{\omega L}{2\pi\bar{v}(10)}\right)^2\right)^{4/3}}\frac{2\pi\sigma_v^2}{\omega} = \frac{\left(\frac{\omega L}{2\pi\bar{v}(10)}\right)^2}{\left(1+\left(\frac{\omega L}{2\pi\bar{v}(10)}\right)^2\right)^{4/3}}\frac{2\sigma_v^2}{3\omega}$$

Formula Breen [5] equation 4.11 p.18 is not correct.

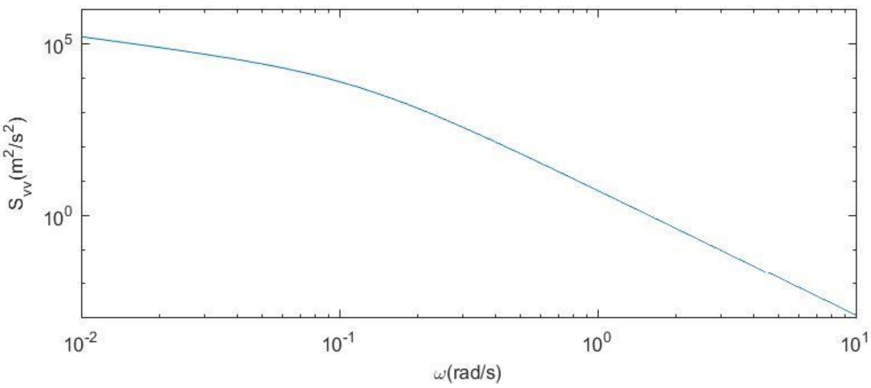
The correct formula is:

$$\sigma_v = \frac{ku_*}{\kappa} = \frac{1.0*2.30}{0.4} = 5.75\frac{m}{s}$$

Vacant area II not urban area II half of the building is facing the Mass river.

Figure 4.3 Breen [5] p.18 is not correct.

The correct figure :



Reference Breen [5] p.55 is not correct.

the reference to Appendix 1 is not correct … it has to be Appendix 8

Appendix Breen [5] file wind.m p.75 is not correct.

v_10 is not $10\frac{m}{s}$ $\quad$ $f\max$ is not 10 $\quad f\max = \frac{1}{2*\Delta t} = \frac{1}{2*0.1} = 5Hz$

Appendix Breen [5] file autospectrum….m p.99 is not correct.

v_10 is not $10\frac{m}{s}$ $\quad$ v_gem is not 35

$$v\_gem = \frac{22.62+39.60}{2} = 31.11\frac{m}{s}$$

Appendix 21 Modified Matlab code Hans Breen

Note: Below are the files that I have modified to determine the natural frequency with Breen's model. The coherence was kept constant in his model. From Breeen's files one could conclude that the return periode of 12.5 years was checked (sigma_v =6.345m/s [5] Appendix 5). Take the wind generator files (wind0.m, autopow.m and coher3.m) from Appendix 2.

A_File_to_run_programs.m

```
%
%Thesis: Torsion motions of high-rise buildings due to wind loading.
%Author: H.A.O.Richardson (Anthony)
%


%
%---------------------------------------------------------------------------
%
% Clear memory
clear all
clc
clear
close all

% Start timer
tic

disp('-----------------------------------------------------')
disp('Thesis: Torsion motions in high-rise buildings')
disp('-----------------------------------------------------')


disp('-Reading problem data')

% Problem data (change this to the name of the mesh, leaving off .m)

B_cross_section_prop;

%---------------------------------------------------------------------------
----

% Start solve
disp('-Starting solution procedure')

% Determining of the mass, bendingstiffness of the cross section.

C_data;

D_stiffness_matrix;

E_State_space_formulation;
```

```
%Inputvalues;

%wind0;

%Force;

%sim('Simulink_run_file);


%---------------------------------------------------------------------------
----


disp( '-Finished analysis. Time:' )
toc
```

B_cross_section_prop.m

```
load matrices %viuw values of all matrices

% The cross-section is divided into 8 walls,

% Location of the neutral-axis, the centerline is taken as the reference
line
% Area of the walls
Area(1)=2*0.5*15.44;
Area(2)=2*0.3*1;
Area(3)=2*0.3*3.72;
Area(4)=2*0.3*6.12;
Area(5)=2*(0.3+0.2)*3.1;
Area(6)=0.3*7.9;
Area(7)=0.3*14.74;
Area(8)=2*6.92*0.6;
% Perpendicular distance between center of mass and reference line
s(1)=0;
s(2)=0.8;
s(3)=1.3+0.15;
s(4)=6.12/2+1.6;
s(5)=3.1/2+1.3;
s(6)=0.3/2+3.1+1.3;
s(7)=-(0.15+0.8);
s(8)=-(6.92/2+0.8);
% Distance between the reference line and the neutral axis
na=dot(Area,s)/(sum(Area));

% The second moment of inertia
Ieig(1)=2*1/12*.5*15.44^3;
Ieig(2)=2*1/12*.3*1^3;
Ieig(3)=2*1/12*3.72*.3^3;
Ieig(4)=2*1/12*.3*6.12^3;
Ieig(5)=2*1/12*.5*3.1^3;
Ieig(6)=1/12*7.9*.3^3;
Ieig(7)=1*1/12*14.74*.3^3;
Ieig(8)=2*1/12*.6*6.92^3;
% Rule of Steiner
Isteiner=sum(s.^2.*Area);
% Total second moment of inertia
I=sum(Ieig)+Isteiner;
E=3e10;
EI=E*I;

% The mass of one storey existing of
%walls, floor and loading on the floor
mass=sum(Area)*2.75*2500+15.44*26.34*.25*2500+15.44*26.34*250;

save matrices EI mass na -append
```

C_data.m

```
%clear
clc
L=144; % height of building
v= 48; % dof
% units (meters and kN)
mass= 6.3182e+005;  % mass kg (maple file)+2,5kN/m^2
l=3;              % lenght of one element
xi1=0.01; % damping ratio of the first eigenmode
xi2=0.01; % damping ratio of the second eigenmode
tend=10; % duration of the simulink simulation time
%timestep=5e-5; %fixed timestep in simulink
timestep=0.1; %fixed timestep in simulink
%save matrices v l xi1 xi2 tend timestep -append;

%save matrices L v mass l EI xi1 xi2 q tend timestep -append;
save ('matrices','L','v','l','mass','xi1','xi2','timestep','tend');
% save ('matrices','v','l','xi1','xi2');
```

D_stiffness_matrix.m

```
clc

load matrices

%{
input
v    = degrees of freedom
EI   = bending stiffness
L    = height of the structure
l    = lenght of element
zeta1 = damping ratio of first eigenmode
zeta2 = damping ratio of second eigenmode
%}
%{
v    = 48;

L    = 144;
l    = 3;
mass=631815.9;
%}

%EI   = 2.003e+013;
zeta1 = 0.01;
zeta2 = 0.01;

Cr=20*EI/L;

% stiffness matrix
% Field elements
k=EI/l^3*[1     -2     1;
          -2     4    -2;
          1     -2     1];% bending element stifnessmatrix

K=zeros(v+4,v+4); % Total systemmatrix bending
for o=0:1:(v+1)
    for n=1:1:3
        for m=1:1:3
            K(o+n,o+m)=K(o+n,o+m)+k(n,m);
        end
    end
end
% Edge element (rotation of the foundation)
K(3,3)=K(3,3)+1/l^2*(1/(l/(2*EI)+1/Cr));
%K(2,1)=K(2,1)-1/l^2*(1/(l/(2*EI)+1/Cr));
%K(1,2)=K(1,2)-1/l^2*(1/(l/(2*EI)+1/Cr));
%K(1,1)=K(1,1)+1/l^2*(1/(l/(2*EI)+1/Cr));

% Bottom element
% Delting the unwanted row and coloum k-matrix for half element.

K(:,1)=[];           % restrained displacement node 0 = 0
K(1,:)=[];           % restrained twist = 0


K(:,1)=[];           % restrained displacement node 0 = 0
```

```
K(1,:)=[];           % restrained twist = 0

% Top element
% Delting the unwanted and coloum k-matrix for half element.

K(:,50)=[];           % restrained displacement node 0 = 0
K(50,:)=[];           % restrained twist = 0


K(:,49)=[];           % restrained displacement node 0 = 0
K(49,:)=[];           % restrained twist = 0
% end stiffness matrix

Cr=20*EI/L;


% mass matrix
M=zeros(v,v);
for n=1:1:v;
    M(n,n)=mass;
end
% end mass matrix

% E is the modal matrix; omegakw is modal K*E = M*E*OMEGAKW
[E,omegakw] = eig(K,M);
for n=1:1:v
    omega_eig(n)=sqrt(omegakw(n,n));
end
% end eigen frequency


% damping matrix
a=2*(([1/(omega_eig(1)) (omega_eig(1)); 1/omega_eig(2) omega_eig(2)])^-
1*[zeta1;zeta2]); %04-10
Cd=a(1,1)*M+a(2,1)*K;

% end damping matrix

 save matrices K M omega_eig Cd -append;


% damping ratio of the eigenmodes of the structure
for n=1:1:48;
    phsi(n)=a(1,1)/(2*omega_eig(n))+a(2,1)/2*omega_eig(n);
end
plot(phsi)

% This has been done already in a matrix

a0=2*omega_eig(1)*omega_eig(2)*(zeta1*omega_eig(2)-
zeta2*omega_eig(1))/((omega_eig(2))^2-(omega_eig(1))^2);
a1=2*(zeta2*omega_eig(2)-zeta1*omega_eig(1))/((omega_eig(2))^2-
(omega_eig(1))^2);

 save matrices a a0 a1 -append;
```

E_Space_state_formulation.m

```
clc % clears comand window
clf % clears figure
%clear
%clc
load matrices

%{
input
v   = degrees of freedom
EI  = bending stiffness
L   = height of the structure
l   = lenght of element
xi1 = damping ratio of first eigenmode
xi2 = damping ratio of second eigenmode
%}
Cr=20*EI/144;

% stiffness matrix
% Field elements
k=EI/l^3*[1    -2    1;
          -2    4   -2;
          1    -2    1];% bending element stifnessmatrix
K=zeros(v+1,v+1); % Total systemmatrix bending
for o=0:1:46
    for n=1:1:3
        for m=1:1:3
            K(o+n,o+m)=K(o+n,o+m)+k(n,m);
        end
    end
end
% Edge element (rotation of the foundation)
K(2,2)=K(2,2)+1/l^2*(1/(l/(2*EI)+1/Cr));
K(2,1)=K(2,1)-1/l^2*(1/(l/(2*EI)+1/Cr));
K(1,2)=K(1,2)-1/l^2*(1/(l/(2*EI)+1/Cr));
K(1,1)=K(1,1)+1/l^2*(1/(l/(2*EI)+1/Cr));
K(:,1)=[];          % restrained displacement node 0 = 0
K(1,:)=[];          % restrained twist = 0
% end stiffness matrix

% mass matrix
M=zeros(v,v);
for n=1:1:v;
    M(n,n)=mass;
end
% end mass matrix
[E,omegakw] = eig(K,M);
for n=1:1:v
    omega_eig(n)=sqrt(omegakw(n,n));
end
% end eigen frequency

% damping matrix
a=2*(([1/(omega_eig(1)) (omega_eig(1)); 1/omega_eig(2) omega_eig(2)])^-
1*[xi1;xi2]);
aa=2*(([1/(omega_eig(1)) (omega_eig(1)); 1/omega_eig(2) omega_eig(2)])^-
1*[xi1;xi2]); % edit 27-08-2018
```

```
Cd=a(1,1)*M+a(2,1)*K;

% end damping matrix

save matrices K M omega_eig Cd -append;


% damping ratio of the eigenmodes of the structure

for n=1:1:48;
    phsi(n)=a(1,1)/(2*omega_eig(n))+a(2,1)/2*omega_eig(n);
end

figure
plot(phsi)

% This has been doen already in a matrix

a0=2*omega_eig(1)*omega_eig(2)*(xi1*omega_eig(2)-
xi2*omega_eig(1))/((omega_eig(2))^2-(omega_eig(1))^2);
a1=2*(xi2*omega_eig(2)-xi1*omega_eig(1))/((omega_eig(2))^2-
(omega_eig(1))^2);
load matrices

% Placing space state formlation Matrix A

A2=eye(v,v);
A3=-inv(M)*K;
A4=-inv(M)*Cd;

% Placing space state formlation Matrix B

B1=inv(M);
C1=eye(v,v);

A(1:v,v+1:1:2*v)=A2;
A(v+1:1:2*v,1:v)=A3;
A(v+1:1:2*v,v+1:1:2*v)=A4;
B(v+1:1:2*v,1:v)=B1;
%C(1:1:v,1:1:v)=C1; corrected 17-11
C=eye(2*v,2*v);
D=zeros(2*v,v);

[E,p]=eig(A);
for n=1:1:2*48
a(n,1)=p(n,n);
end

for n=1:1:48
    phi(:,n)=E(1:1:48,2*n-1); %04-10
end

save matrices A B C D -append;

% Plot off  10-11-2008
```

```
R=real(E);
%
%for n=1:1:10000
%     t=n/100;
%     x_48(n)=phi(48,48)*exp(p(96,96)*t)+R(48,96)*exp(p(95,95)*t);
    %x_48(n)=phi(48,48)*exp(p(65,65)*t)+E(48,96)*exp(p(64,64)*t);
%     tijd(n)=n/100;
%end
%plot (tijd,x_48)



% Determining static diplacement

f=186300; %  put the values in Nm

F= zeros(v,1);

for  lr=1:1:v
     F(lr,1)=f;
end

u=-inv(K)*F;

% THE Z POSITION OF THE NODES

Z= zeros(v,1);

    zzz= 1.5;

for  zrn=1:1:v

     Z(zrn,1)=(zzz+(zrn-1)*3);
end


%Plotting  displacment of the buiding.

 figure
 plot(abs(u),Z)
 axis([0 0.21 0 145])
 set(gca,'YTick',0:20:144)
 set(gca,'XTick',0:0.05:0.21)

 xlabel('u(m)')
ylabel('z(m)')
title('Plot Static deflection')


% Validation of dynamic behavior

% Zie matix omega_eigen(1,1) demping first antral frequency

we=omega_eig(1,1)*(1-zeta1^2)^0.5;

wee=(1-zeta1^2)^0.5;
```

```
kk=(8*EI)/(1.2*L^4);

% zeta1=0.01; not used becasue in teh memeory already

Ftop=-1*(kk*u(48,1));

meq=kk/(omega_eig(1,1)^2);


zetameq=0.05;

alpha=0.01;
beta=-1;

for n=1:1:20000
    t=n/100;

    %x_48(n)=(Ftop/kk);
    %x_48(n)=(Ftop/kk)*sin(we*t);
    %x_48(n)=(Ftop/kk)*(1-cos(we*t));
    %x_48(n)=alpha*exp(beta*t/100);
    %x_48(n)=(Ftop/kk)*(exp(-0.95*1.48)*t)*(1-cos(we*t));
    %x_48(n)=(Ftop/kk)*(exp(-zetameq*(((kk/meq)*t)^0.5)))*cos(we*t);
    %x_48(n)=(Ftop/kk)*(1-cos(we*t));
    x_48(n)=1.05*(Ftop/kk)*(1-(exp(-
zetameq*(((kk/meq)*t)^0.5)))*(cos(we*t)));
    x_47(n)=(Ftop/kk)*(1-(exp(-zetameq*(((kk/meq)*t)^0.5)))*(cos(we*t)));
    %x_47(n)=(Ftop/kk)*(1+exp(-zetameq*(((kk/meq)*t)^0.5)))*sin(we*t);
    %x_48(n)=(Ftop/kk)*sin(we*t);
    %x_48(n)=(Ftop/kk)*((1-exp(-zeta1*(((kk/meq)*t)^0.5)))*sin(we*t));
    %x_48(n)=phi(48,48)*exp(p(96,96)*t)+R(48,96)*exp(p(95,95)*t);

    tijd(n)=n/100;
end

figure
plot (tijd,x_48,'b')

hold on
plot(tijd,x_47,'r')
hold off

axis([0 80 0 0.45])
```

Force.m

```
load matrices
rho=1.25;            % [kg/m^3]
Area=3*26.34/20;         % [m^2]
Ch=1.2;              % thrust coefficient
u_star=2.82;
kappa=0.4;
d=3.5;
z_0=2;
z=(9:1.5:144);
load UC1
Uf(:,:,3:1:93)=UC;
Uf(:,:,1)=Uf(:,:,3);
Uf(:,:,2)=Uf(:,:,3);
% average part
v_mean(1,1,3:1:93)=u_star/kappa*log(z-d/z_0);            % mean wind speed
at reference height
v_mean(1,1,1:1:2)=v_mean(1,1,3);
v_mean(:,1,:)=v_mean(1,1,:);
v_mean(:,2,:)=v_mean(1,1,:);
v_mean(:,3,:)=v_mean(1,1,:);
v_mean(:,4,:)=v_mean(1,1,:);
v_mean(:,5,:)=v_mean(1,1,:);
v_mean(:,6,:)=v_mean(1,1,:);
v_mean(:,7,:)=v_mean(1,1,:);
v_mean(:,8,:)=v_mean(1,1,:);
v_mean(:,9,:)=v_mean(1,1,:);
v_mean(:,10,:)=v_mean(1,1,:);
v_mean=repmat(v_mean,[N 1 1]);

U=Uf+v_mean;
U(:,:,94)=0;

F=1/2*Area*Ch*rho*(U).^2;
F=sum(F,2);
F=squeeze(F);

f(1:1:N,1:1:2)=2*F(1:1:N,1:1:2); % bottom 2 nodes
for n=3:1:48
    f(:,n)=F(:,2*n-3)+F(:,2*n-2);
end
F=f;
% F(:,49)=0;                      % for active demping place
t=[deltat:deltat:(N*deltat)]';
save ('F','F','t')
```

wind0.m

```
function [UC]=wind0(yr,zr,v_10,sigma,N,deltat,fmax);
% simulation of a turbulent wind field
%
% INPUT:
%    yr, zr: specification of coordinates on the facade of the structure
%    v_10: mean wind velocity at 10 m above the surface of the earth (m/s)
%    sigma: standard deviation of the fluctuating part of the wind speed
(m/s)
%    N: number of time points (including zero); N must be a power of 2
%    deltat: time step (s)
%    fmax: maximum frequentie spectrum (Hz)
% OUTPUT:
%    UC: constrained turbulent wind velocities (m/s)

%yr=1.3:2.6:24.7;
%zr=9:1.5:144;
%v_10=10;
%sigma=6.345;
%N=10;
%deltat=.1;
%fmax=10;

load matrices

% number of points in rotor plane
Ny=length(yr);
Nz=length(zr);
Np=Ny*Nz;

% y and z coordinates of all rotor points in one column vector
Yr=reshape(yr'*ones(1,Nz),Np,1);
Zr=reshape(ones(Ny,1)*zr,Np,1);

r=zeros(Np,Np);
for i=1:Np
   for j=i+1:Np
      % distances between points
      r(i,j)=sqrt((Yr(i)-Yr(j))^2+(Zr(i)-Zr(j))^2);
      r(j,i)=r(i,j);
   end
end
% time vector
t=[0:N-1]'*deltat;
% period
T=N*deltat;
% frequency step
deltaf=1/T;
% discretized frequencies
k=[1:N/2-1]';
f=k.*deltaf;
% autopower spectral density (one-sided)
Sa=autopow(f,v_10,sigma);
% spectrum is cut-off above fmax by application of window
Index=find(f>fmax);
if ~isempty(Index)
  Nw=Index(1);
```

```
  w=zeros(N/2-1,1);
  W=window('hann',2*Nw+1);w(1:Nw+1)=W(Nw+1:2*Nw+1);
  Sa=w.*Sa;
end
% renormalize Sa to variance
Sa=sigma^2/(sum(Sa)/T)*Sa;

% Fouriercoefficients points in rotor plane
ak=zeros(Np,N/2-1);
bk=zeros(Np,N/2-1);
for k=1:N/2-1
   Coh=coher(f(k),r,v_10);
   % Choleski decomposition
   L=sqrt(Sa(k)/T)*chol(Coh)';
   % vector of unit variance normal random numbers
   ran=randn(Np,1);
   ak(:,k)=L*ran;
   ran=randn(Np,1);
   bk(:,k)=L*ran;
end

% complex notation
i=sqrt(-1);
UC=zeros(N,Np);
for j=1:Np
   C=ak(j,:)'-i*bk(j,:)';
   C=1/2*[0;C;0;rot90(C')];
   % inverse FFT
   uc=N*ifft(C);
   if any(abs(imag(uc)) >= 1e-7*abs(uc) & abs(imag(uc)) >= 1e-12)
     max(abs(uc))
     max(imag(uc))
     error('imag too large uc')
   end
   UC(:,j)=real(uc);
end
% reshape UC: separate indices for y and z
UC=reshape(UC,N,Ny,Nz);


save ('UC')
save ('UC1','UC')

%save matrices;
%save matrices UC1 -append
```

AUTOPOW.m

```
function S=autopow(f,v_10,sigma)
% syntax: function S=autopow(f,v_10,sigma)
% Autopower spectral density function of turbulence
% Input:
%   f: frequency (Hz)
%   v_10: the mean wind speed at 10 m above (m/s)
%   sigma: standard deviation (m/s)
% Output:
%   S: autopower spectral density (m^2/s)

sigma_v=6.345;       % standard deviation of the wind speed variation with
height
L=1200;              % characteristic length Davenport
v_10=10;             % mean wind speed at 10 m height


S=2/3*(f.*L/v_10).^2 ./ ((1+(f.*L/(v_10)).^2).^(4/3)).*sigma_v^2./(f);
```

COHER.m

```
function Coh=coher(f,r,v_10)
% syntax: function Coh=coher(f,r,v_10)
% Coherency function
% of longitudinal wind velocity fluctuations
% Input:
%   f: frequency (Hz)
%   r: mutual distance coordinates
%   V_10: the 10 minute average wind speed at hub height (m/s)
% Output:
%   Coh: coherency (-)

C=10;
x=f.*C.*r./v_10
Coh=exp(-1.*x);
```