An investigation into DIC using free vortex wake methods

Identifying the optimal dynamic induction control law to trigger effective wake mixing in a two turbine system

Jop Prinsen

Delft University of Technology

TU Delft

An investigation into DIC using free vortex wake methods

Identifying the optimal dynamic induction control law to trigger effective wake mixing in a two turbine system

by

Jop Prinsen

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday June 23, 2023 at 15:00.

Student number:	5195802		
Project duration:	September 6, 2021 – June 23, 2023		
Thesis committee:	Prof. dr. ir.	J.W. van Wingerden,	TU Delft, supervisor
	Ir.	D. van den Berg,	TU Delft
	dr.	W. Yu	TU Delft

Cover:	Yuriy Laymin (modified) - From Pexels
Style:	TU Delft Report Style, with modifications by Daan Zwaneveld

TU Delft

Preface

The subject for this thesis was proposed to me by my supervisors Jan-Willem van Wingerden and Daniel van den Berg. I would like to thank Jan-Willem for his enthusiasm and support during the project and especially my day to day supervisor Daniel for his invaluable guidance, suggestions and contributions to the process of writing my thesis.

In addition I would like to thank my friends, family and girlfriend Demi for supporting me throughout my bachelor and masters.

Jop Prinsen
Delft, June 2023

Abstract

Wind farm control is an area of research that looks to maximise the performance of individual but most importantly the collective performance of wind farms. The performance of turbines located in a wind farm is heavily dependent on the effects of turbine wakes, and improving the efficiency of wind farms by mitigating the wake effects is an ongoing and promising field of research. Simulating the wake effects of a turbine is generally done with computationally expensive physics based modelling techniques like CFD simulations. The computational complexity of these models limits the scope of optimisation that can be performed on wake mitigating controls strategies such as dynamic induction control. A substitute engineering model like the free vortex wake model could provide better insight into dynamic induction control by virtue of its computational efficiency allowing a wider optimisation parameter space. In this thesis an optimisation framework based on a 2D implementation of the free vortex wake model and genetic algorithm optimisation is used to investigate dynamic induction control for a simple two turbine wind farm beyond the research that has been done up to this point. While yet to be confirmed with higher fidelity simulations, initial results corroborate earlier findings for simple dynamic induction signals and additionally indicate that dynamic induction control could benefit from multiple harmonics. The optimisation framework achieves its goal in allowing a wider parameter space to be searched for optimisation, even on consumer grade desktop hardware, and shows potential as a tool for further investigation of dynamic induction control.

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
2D	Two Dimensional
3D	Three Dimensional
AWC	Active Wake Control
CFD	Computational Fluid Dynamics
DIC	Dynamic Induction Control
EU	European Union
FVWM	Free Vortex Wake Model
GA	Genetic Algorithm
ISA	International Standard Atmosphere
LCOE	Levelised Cost Of Energy
LES	Large Eddy Simulation
MPPT	Maximum Power Point Tracking
SIC	Static Induction Control

Symbols

Symbol	Definition	Unit
A	Area	$[m^2]$
a	Induction factor	$[-]$
C_A	Power correction coefficient A	$[-]$
C_B	Power correction coefficient B	$[-]$
C_P	Power coefficient	$[-]$
C_T	Thrust coefficient	$[-]$
c	Characteristic core size	$[m]$
D	Rotor Diameter	$[m]$
f	Frequency	$[Hz]$
k	Discrete time step	$[-]$
m	Mass	$[Kg]$
P	Power	$[W]$
p	Pressure	$[Pa]$
r	Distance between vortex points	$[m]$
St	Strouhal number	$[-]$
t	Time	$[s]$
T	Thrust force	$[N]$
U	Velocity	$[m/s]$
v	Induced velocity	$[m/s]$
V_∞	Free inflow velocity	$[m/s]$
x	X coordinate of vortex point	$[m]$
y	Y coordinate of vortex point	$[m]$
Γ	Vortex Strength	$[m^2/s]$
ω	Frequency	$[rad/s]$
ρ	Density	$[kg/m^3]$

1

Introduction

Global warming is one of the most important issues that threatens our world today [1], and tackling it in time will be a major challenge. A key part in solving this challenge is the transition from fossil fuels towards renewable energy sources to meet our ever increasing energy consumption. Even in recent years the consumption of fossil fuels has steadily increased, despite a global increase in renewable energy production [2][3]. Figure 1.1 below shows this trend in the composition of global electrical energy generation since the 2000s.

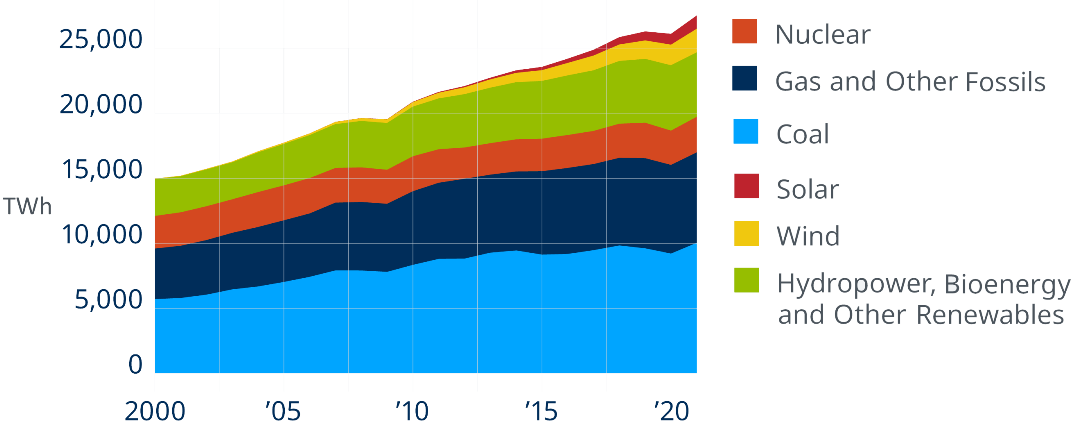


Figure 1.1: Global composition of generated electrical energy. Source [3]

An increase in renewable energy alone will not suffice to stem global warming. In order to make a meaningful impact renewable energy will need to start outpacing the growing energy demands to combat the increase in fossil fuel usage. In addition, with the Russian invasion of Ukraine in 2022 and the resulting energy crisis it has become clear that the energy independence of the EU from Russia is essential for the energy security going forward.

Wind energy is one of the key contributors to renewable energy production today according to the European Commission [4]. The European Commission has estimated that by 2030 up to 453 GW of wind energy is needed to comply with the 2015 Paris agreement. More recently in response to the Russia-Ukraine war the European Commission has declared that it aims to replace a large part of the fossil energy imports from Russia with renewable energy, accelerating previous set targets and aims to reach a target of 510 GW of wind power capacity by 2030 [5]. As of 2022 Europe has a combined total wind power capacity of 255 GW, with 19 GW of new wind power capacity installed in 2023 [6]. This falls significantly short of the estimated 30 GW a year needed over the period 2023-2027 to meet the 2030 targets. Figure 1.2 below shows the projected gap between the estimated current trend marked by the central scenario and the required trend to reach the 2030 targets

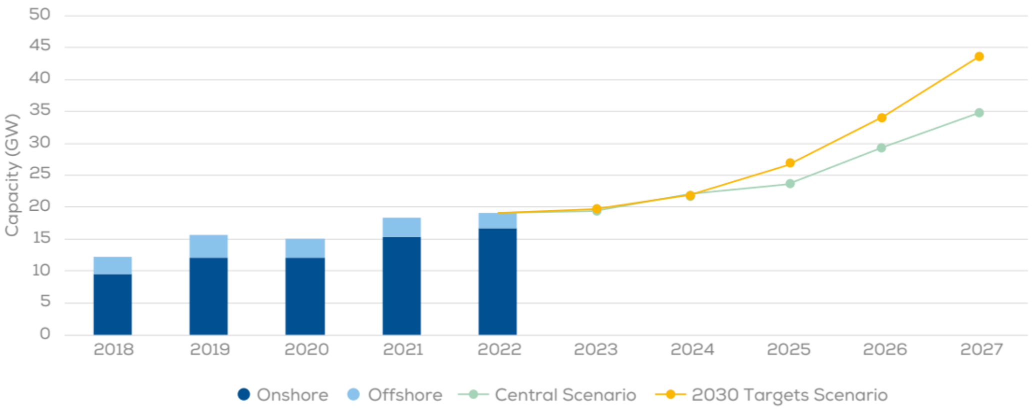


Figure 1.2: WindEurope scenario projections for newly installed wind power capacity in the EU. Source [6]

An important aspect of increasing the total wind power capacity in the EU is increasing the performance of wind turbines and wind farms as a whole. In both onshore and offshore sectors the average power rating of installed wind turbines have more than doubled over the last decade [6], with one of the main driving forces behind the increasing power rating being the increase in rotor diameter [7]. In addition the capacity factors of wind farms have been increasing due to the improvements in turbine technology, allowing modern turbines to operate at their maximum power rating for a larger portion of time.

The performance of turbines located in wind farm is heavily dependent on the effects of turbine wakes [8]. Long term average power losses in wind farms due to wake effects can exceed 10% of the total energy production [9]. Mitigating these wake effects is an ongoing field of research which shows great potential to improve wind farm efficiency. Methods include wind farm layout optimisation or a variety of turbine control methods which include wake steering, static induction control and dynamic induction control [8]. Dynamic induction control in particular shows great potential in numerical studies [8][10][11].

Dynamic induction control (DIC) involves dynamically derating turbines in such a way that downwind turbines are less affected by their wake in terms of power production. Derating is the process of operating a turbine outside of its maximum potential power extraction. DIC works by giving up individual performance to benefit the collective of the wind farm. Previous research into DIC using LES simulations [10] has identified a simple sinusoidal induction control signal with a characteristic optimal frequency. Further research using free vortex wake methods and adjoint optimisation has found a roughly periodic signal at around this same frequency [11].

This thesis aims to further explore DIC by building on this simple sinusoidal signal, expanding the signal to include multiple harmonics and optimising the parameters. To this end an implementation of a 2D free vortex wake model is used in conjunction with genetic algorithm optimisation. The effects of inter turbine spacing and a semi-free optimisation method will also be discussed.

Chapter 2 will cover the theoretical framework that serves as the basis for the thesis. This chapter also presents the implementation of the free vortex wake model, and provide a convergence study for model validation. Chapter 3 will discuss the genetic algorithm optimisation method that is used. Subsequently Chapter 4 will provide an overview and discussion of conducted simulations and their results. Finally a conclusion is drawn based on all findings.

2 Theoretical Framework

This chapter aims to provide a theoretical framework as a basis for the thesis by providing the necessary background knowledge and context on which the research builds. The chapter covers wind turbine theory, previous research on wake mitigating control, the simulation framework and its implementation as used in this study.

2.1. Basic wind turbine theory

This section introduces some basic concepts of wind turbine modelling and derives key quantities like the coefficient of thrust, the induction factor and a relation for the power output of wind turbines that are built upon in this thesis.

Most horizontal axis wind turbines have a three bladed rotor. As each blade is subjected to the incoming wind its aerofoil generates a lift force along the blade that exerts a torque on the rotor shaft. The combined torques of the blades drive the generator housed in the turbine nacelle. Modelling three turbine blades individually introduces a lot of complexity. It is useful to represent the rotor as an actuator disk eliminating the need to individually model the blades, instead focusing on their combined effects. The actuator disk is assumed to be a non-moving disk that is normal to the inflow direction and over which the forces on the flow are uniformly distributed. In addition the flow is assumed to be an incompressible, isentropic and inviscid fluid [7].

The actuator disk representation now allows for the application of conservation laws. The conservation of mass, momentum and energy, and by extension the law of Bernoulli, allow the derivation of key quantities that are useful in modelling wind turbines. The actuator disk is placed in an incoming flow velocity U in $[m/s]$ from which it extracts power P in $[W]$ while exerting a thrust force T in $[N]$ on the flow itself. Due to the thrust force exerted by the rotor upon the in flowing wind a high pressure region is created in front of the disk, while a low pressure field is created just behind it. The pressure jump over the actuator disk itself is equal to $\Delta p = \frac{T}{A}$. Figure 2.1 below shows an actuator disk in a flow as described.

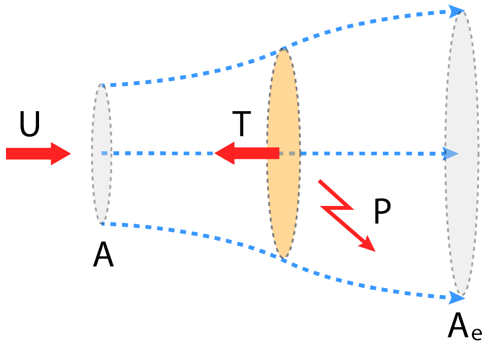


Figure 2.1: Actuator Disk placed in flow.

Figure 2.1 shows the expansion of the stream-tube. With the cross sectional area of the stream tube before and after the rotor denoted as A and A_e respectively this expansion can be expressed as: $A_e > A$. The velocity after the turbine is denoted as U_e. Thus the velocity reduction in turn can be expressed as ($U > U_e$), where U is the inflow velocity as stated earlier. Following the derivation in Zaayer 2018 [7], this effect can be explained by using the conservation of mass. The mass flow is conserved when passing through the rotor as:

$$\rho UA = \rho U_e A_e,$$

with air density ρ. This expression reduces to:

$$\frac{U}{U_e} = \frac{A_e}{A}. \tag{2.1}$$

By using the velocity deficit of the wake, $U > U_e$, in equation 2.1 it can be shown that the cross sectional area of the wake has to expand to compensate for the loss of velocity in order to satisfy mass conservation. Leading to the relation $A_e > A$.

By applying conservation of momentum the thrust force can be related to the loss in momentum between the inflow and outflow. The reduction in velocity is proportional to the thrust as:

$$T = m(U - U_e). \tag{2.2}$$

Using the principle of conservation of energy the extracted power can be expressed as:

$$P = \frac{1}{2}m(U^2 - U_e^2). \tag{2.3}$$

Since the power extraction is a result of the work done by the thrust force T, it can also be expressed as:

$$P = TU_r, \tag{2.4}$$

where U_r is the velocity at the rotor. Combining these expressions allows the velocity at the rotor to be written as:

$$U_r = \frac{1}{2}(U + U_e). \tag{2.5}$$

Using these results non-dimensional performance indicators can be deduced. The induction factor a represents the difference in velocity between the incoming flow and the speed at the rotor and is given by:

$$a = \frac{U - U_r}{U}. \tag{2.6}$$

Using (2.5), U_r and U_e can now be expressed as:

$$U_r = U(1-a), \tag{2.7}$$

$$U_e = U(1-2a). \tag{2.8}$$

This is useful as it allows the expressions for thrust force and power to be rewritten in terms of the non-dimensional induction factor a. Also observe that for $a \geq 0.5$, (2.8) becomes negative and thus momentum theory only holds for $a < 0.5$. The thrust and power can be expressed as:

$$T = \frac{1}{2}\rho U^2 A 4a(1-a), \tag{2.9}$$

$$P = \frac{1}{2}\rho U^3 A 4a(1-a)^2. \tag{2.10}$$

Equations 2.9 and 2.10 can be used to derive non-dimensional performance coefficients that only depend on the induction factor a. The thrust coefficient C_T and the power coefficient C_P are defined as:

$$C_T = 4a(1-a), \tag{2.11}$$

$$C_P = 4a(1-a)^2. \tag{2.12}$$

The optimal power coefficient can be found at $C_P = \frac{16}{27} \approx 0.59$ with $a = \frac{1}{3}$, with the corresponding thrust coefficient $C_T = \frac{8}{9} \approx 0.89$. This is the point at which the turbine reaches it maximum efficiency. This limit is called the Betz limit.[7] Maximising the thrust force instead of the power will result in no power generation as at that point ($a = 0.5$) the velocity at the rotor equals zero and thus the power, 2.4, also equals zero.

For purposes of simulation it is useful to express the thrust coefficient in terms of the local rotor velocity U_r. This allows the rotor velocity at the rotor in the simulation to be directly used for calculation of turbine power. The coefficient of thrust expressed in the rotor velocity is denoted as $C_T^{'}$ can be derived as follows:

$$C_T^{'} = \frac{C_T}{(1-a)^2} = \frac{4a}{1-a} \tag{2.13}$$

Now the maximum efficiency can again be obtained by maximising the power coefficient. The corresponding thrust coefficient is then $C_T^{'} = 2$. The turbine power can be expressed in terms of U_r and $C_T^{'}$ as:

$$P = TU_r = \frac{1}{2}\rho A U_r^3 C_T^{'}. \tag{2.14}$$

This formulation will prove useful for use with dynamic induction control signals expressed in the local coefficient of thrust. This motivates the use of the local coefficient of thrust definition for this thesis.

2.2. Wind farm control

Wind turbines are continuously controlled in order to ensure optimal operation. This can be to achieve different, sometimes conflicting, objectives such as power production optimisation or load mitigation. This section will provide a short overview of relevant contemporary wind farm control methods and introduces the concept of dynamic induction control and previous research into this topic.

Greedy control

The simplest and most common control strategy for a wind farm is called greedy control. In this configuration every turbine in the wind farm tries to extract as much energy from the incoming wind as possible. This is called maximum power point tracking (MPPT) [8], and is done by keeping the power coefficient as high as possible, in other words as close as possible to the Betz limit. As the objective for each turbine is to produce as much energy as possible, the wake effects on other turbines downwind are not taken into account. Therefor while each turbine is operating at or near its optimal power production given its respective incoming wind, the power production of the entire farm as a whole is likely not optimal due to the wake effects reducing the available energy for downwind turbines. Greedy control can be seen as a baseline performance to which wake management control strategies can be compared.

Wind farms are usually orientated such that wake interactions are minimised under the prevailing wind direction[12]. Therefor wake management with the aim of increasing power production is mainly beneficial when the incoming wind direction is off-design, which is when the wake interactions are most prevalent[8]. In addition if an implementation of wake mixing achieves faster wake recovery it could be used to make more compactly designed wind farms viable. More compact wind farms would lead to a lower levelized cost of energy (LCOE), by increasing either the density of turbines in- or decreasing the total required surface area of a plot of land or sea.

Wake steering

Wake steering is a method to improve the power production of a wind farm by steering the wake of a turbine in such a way that it has less of an effect on downstream turbines. The wake can be steered by yawing the rotor, sacrificing some of the effectiveness of the yawed turbine. Since the thrust force is now no longer parallel with the flow it induces a perpendicular component on it. This perpendicular force component will push the wake sideways when compared to the flow direction. Figure 2.2 illustrates this concept. Wake steering can improve the power production but it could also help with load mitigation for downstream turbines. While wake steering does increase the loads on the yawed turbine itself, it could also reduce the load on downwind turbines as turbines under a waked inflow experience greater fatigue loads due to the turbulence introduced by the wake [8][13]. However, since downwind turbines that are now no longer either fully or partially covered by a wake experience greater inflow velocities, the experienced loads could also increase, ultimately leading to a trade-off.

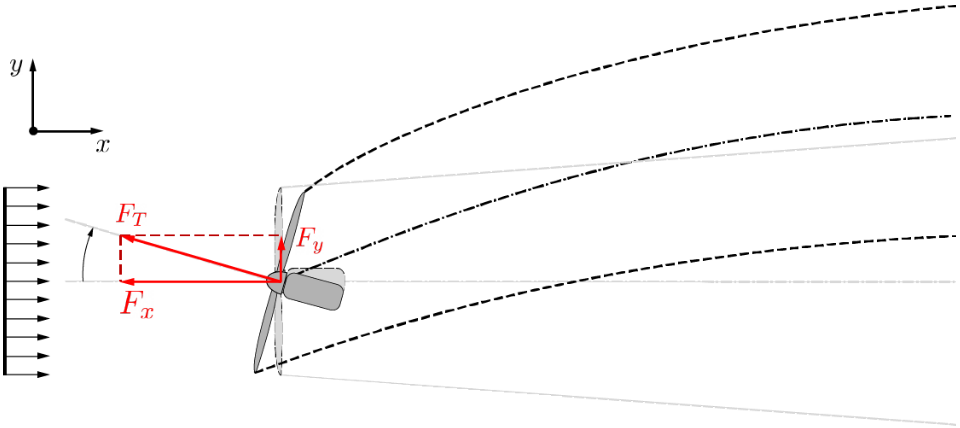


Figure 2.2: Wake steering illustrated.

Induction control

Induction control regulates the induction factor of a wind turbine by either altering the rotor blade pitch or using the generator to produce less power, thereby leaving more energy in the wake for the downstream turbine. The induction factor represents the decrease in speed of the wind flowing through the actuator disk that represents the turbine blades. Both over and under induction compared to the optimal induction factor can be used to derate the turbine. Over induction has been found to be more effective than under induction at improving wake recovery [14]. By incrementally derating each successive downstream turbine less allows for a more even energy distribution over the total wind farm and in theory yielding an overall higher energy output. The induction factor can be changed independent of time or with a very low frequency leading to almost static behaviour. This is called Static Induction Control (SIC).

However, research into the application of SIC has not yielded significant benefits. The theoretical larger benefits of SIC could have been a result of the wake models used. While SIC has been studied using higher fidelity CFD models, and even in wind tunnel and field tests, the majority of models used are low fidelity which could skew the results [8]. In addition Annoni et al [15] showed that while SIC does leave more kinetic energy in the wake of a turbine, the conserved kinetic energy in the wake is mainly concentrated at the edges of the wake. Due to wake expansion and meandering the additional kinetic energy concentrated at the wake edges is not entirely captured by a second downwind turbine and thus the downwind turbine is not able to benefit, leading to a net energy loss. Considering all of this, the benefits of SIC appear to be limited, and unproven.

Active wake control

Active wake control (AWC) is a term used to describe a dynamic wake control approach where the turbine is actively controlled in order to influence the wake as opposed to a static approach. Dynamic Induction Control (DIC) is an extension on the SIC method discussed earlier. In this case the turbines are derated in a dynamic, time-varying manner allowing the farm to adapt to changing inflow conditions and each turbine to adjust its configuration to the wake effects of upstream turbines and to the benefit of donwstream turbines. However, in addition the periodic induction changes of AWC can be used to make the wake mix and break-up faster. Leading to more energy being recovered into the wake before it reaches the next downwind turbine. AWC can be achieved by dynamically adjusting the yaw, pitch or torque of the turbine.

In [16] Munters and Meyers studied DIC with the aim of experimentally identifying the characteristics of an optimal DIC input signal using LES simulations on a wind farm. Gradient based optimisation of these LES simulations with the goal of maximising aggregate wind farm power production lead to optimal input signals that had a somewhat erratic and seemingly periodic nature. Munters and Meyers also made a few observations pertaining to the effects of DIC in larger wind farms. The first row of a wind farm with respect to the incoming wind direction has the most potential in improving wind farm efficiency, which intuitively makes sense as the first row produces the most power and its wake affects all downstream turbines in the wind farm. In addition self optimisation of turbines proved to be limited to non existent leading to a almost greedy control case for the last row of turbines. Since there are no significant wake effects propagating upstream, and there are no turbines in the wake of the last row again it is logical that these turbines would resort to greedy control to maximise their power output.

In order to mimic the found optimal input signals form the gradient based optimisation Munters and Meyers[16] used a sine in order to replicate the thrust coefficient variations that are the driving factor in triggering vortex shedding and therefor wake mixing. This sinusoidal input signal was parameterised in terms of its amplitude A and the non-dimensional Strouhal number St as:

$$C_T^{'} = 2 + Asin\left(2\pi St\frac{tU_\infty}{D}\right) \tag{2.15}$$

Where in order to characterise the frequency of excitation independent of turbine size and inflow conditions the dimensionless Strouhal number (St) is defined as:

$$St = \frac{fD}{u_\infty}. \tag{2.16}$$

Note that the base level or mean of the sinusoidal signal in Equation 2.15 was not parameterised and instead fixed at the Betz optimum thrust coefficient of 2. The remaining parameters were optimised using a parameters sweep to avoid the computationally expensive gradient based optimisation otherwise required [16]. While sufficient for the optimisation of the input signal as defined in Equation 2.15, this optimisation method does not lend itself well for optimisation of larger sets of parameters, especially with a computationally expensive model like used in an LES Simulation. The final identified parameters for turbines at spacing of 5 times the rotor diameter, or $5D$, resulting from the parameter sweep were chosen as $(St, A) = (0.25, 1.50)$ [16]. It should be noted that both the Strouhal number and amplitude were chosen at points interpolated from larger well performing ranges for these parameters.

More recently, and in parallel to this thesis, van den Broek et. al. [11] implemented both a 2D and a 3D free vortex wake model (FVWM) as surrogate models to simulate turbine wakes. Using the 2D model and adjoint optimisation for a two turbine setup at $5D$ spacing and optimising for maximum power production they found an almost sinusoidal signal with a Strouhal number of 0.2, slightly lower than Munters and Meyers but close to the earlier discussed range of well performing parameters in their parameter sweep. The signal found by van den Broek et. al. much more closely resembles a single sinusoidal signal than what Munters and Meyers originally observed in their gradient based optimisation using LES. Taking into account the much greater complexity of modelling involved with LES when compared to free vortex methods a simpler optimal signal is to be expected, as mechanisms like turbulence are not present. Van den Broek et. al. found that the mean value of the induction signal was slightly derated when compared to greedy control at $C_T^{'} = 1.75$, and an amplitude of $A = 0.8$ which is significantly lower than the amplitude determined Munters and Meyers at $A = 1.5$. The derating could be related to the derating that occurs more frequently in low fidelity models when looking at SIC.

Munters and Meyers [10] also investigated the effectiveness of multiple AWC control strategies using large eddy simulations on a 4x4 wind farm and found that significant improvements to the baseline greedy control could be achieved using both DIC and dynamic yawing of the turbine, with especially promising results for combined dynamic (over) induction and yaw control. Yawing at a frequency that corresponds to a Strouhal number in the range of $St = 0.1 - 0.3$ was found to trigger wake meandering [17]. For the tested configuration of turbines dynamic yaw control was found to be more effective than dynamic induction control. It has to be noted though that the effectiveness of dymanic yawing and wake steering in general is likely dependant on the wind farm size as for larger farms the turbines are more likely to be in the (steered) wake of another turbine. In such farms DIC might become the more effective mechanism for increasing power extraction. Van den Broek et. al. also investigated the combination of DIC and yaw control using a 3D implementation of the FVWM, again using adjoint optimisation but noted that the optimiser struggles with overcoming local optima. Other optimisation methods may be required to study the DIC and Yaw control problem in this case.

2.3. Wake simulation

In order to effectively investigate wake management and control techniques to improve the performance of a wind farm a sufficiently accurate wake model needs to be used that can capture the complex behaviour of turbine wakes, while also being efficient enough to allow fast simulation and optimisation. In the previous section both LES and FVWM are mentioned as being used for this goal. This section will discuss wake simulation models and the choice for a FVWM implementation in this thesis.

Engineering models

There are different approaches to model turbine wakes. In general, there are two types of models; low fidelity or surrogate/engineering models, and high fidelity models. The free vortex wake model can be categorised as a low fidelity model whereas LES is a high fidelity model. Engineering models are often simplified representations of a process or behaviour. These models can be very useful when used to describe the behaviour of the system they are intended to represent, especially when the physics of the real system are complex and require a lot of computational power to compute. Engineering models can be derived from experimental data or by using physical laws and applying the necessary assumptions in order to simplify the system to an appropriate level of complexity. Using an engineering model instead of a more complex higher fidelity model for the purpose of wind farm optimisation applications can allow conventional control techniques to be applied more easily by reducing the computational complexity enough for the model to be used in varying control methods in real time, whereas conventional fluid simulations cannot.

The Jensen model [18] was one of the first wake models to be developed and is based on the conservation of mass principle. The Jensen model is relatively simple and is sufficient to calculate the steady state wind velocity profile in the wake of a turbine. But it lacks the complex dynamic behaviours that are required to model the mechanisms and interactions in wakes. This is a result of its steady state nature and the use of assumptions and simplifications in its derivation. The Jensen model for example doesn't enforce conservation of both energy and momentum, and assumes the wake to retain a cone-like shape where in reality the cone tapers off over distance.

The FLORIDyn model by Gebraad and Van Wingerden [19] is an example of a dynamic engineering model that is simple enough in terms of computational complexity to be used in real time control applications and describes the wake interactions between the turbines in a wind farm. The FLORIDyn model is an extension on the earlier steady state FLORIS model, also by Gebraad, which itself is based on the Jensen model.

Free vortex wake model

In order to develop a controller that enhances wake mixing while also being efficient enough for real time simulation an engineering model is needed that can represent the dynamic behaviour in a wake to a sufficiently accurate level. For yaw controlled wake steering it has been experimentally found that the vortices that are released from the rotor tips drive the wake deformation in dynamic cases [20]. An important limitation in many engineering models is that they do not describe the aerodynamic mechanisms governing wake effects.

An alternative to resorting to LES simulations to model dynamic wake deformation is to use a free vortex wake model (FVWM) approach. This allows the impact of the tip vortices to be investigated separately as a key mechanism that drives wake deformation and mixing. Free vortex wake methods have been used in previous research to study the effects of yawing the rotor and originate in the modelling of helicopter rotors [21]. Troldborg et al. [22] compared the results of a 3D free vortex wake model of a turbine to CFD simulations and concluded that the free vortex wake model gave an accurate representation of the wake for a wide range of yaw angles except for higher angles. The relative simplicity of the model provides opportunity to develop real time control methods based around wake mixing. The advantages of the free vortex wake model have also been recognised by van den Broek et. al. in [11] and [23]. In their research they empahsised that the FVWM allows simulations and optimisations to be run on basic consumer grade hardware, in contrast to the hardware needed to perform complex LES simulations. This is a distinct advantage of using free vortex methods as they allow much more expansive search and optimisation of the control signal parameters. For the goal of this thesis the free vortex wake model is the most suitable model for expanding the search for the optimal DIC control signal for a two turbine system.

The free vortex wake model models the wake of a turbine as a set of discrete points representing vortices shed by the rotor tips and their interactions. For simplicity a 2 dimensional FVWM will be concidered. The rotor is represented by an actuator disk that is assumed to be under a uniform load. The rotor diameter D and the inflow velocity V_∞ are normalised to equal 1. The flow itself is assumed to be incompressible and inviscid and assumed to be always perpendicular to the actuator disk i.e. aligned with the x axis. At each time step a point is shed from each rotor tip as shown in Figure 2.3.

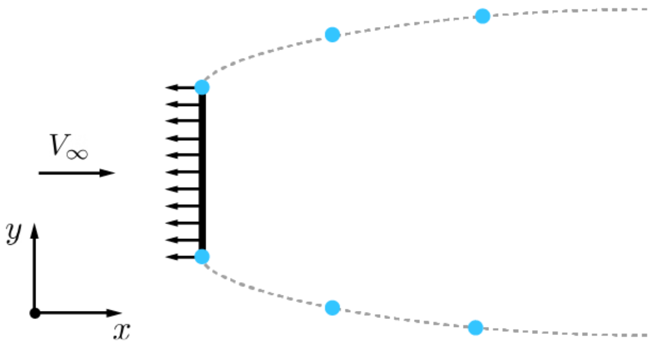


Figure 2.3: Vortex Points shed from the edges of the 2D actuator disk.

Each of the shed vortex points has three states consisting of x and y coordinates and the vortex strength Γ. The strength of the vortices at each point is derived from the vorticity equation in combination with momentum theory and can be expressed as in equation 2.17. It is assumed that the vortex strength is constant in time.

$$\Gamma = \frac{1}{2} V_\infty^2 C_T \Delta t \tag{2.17}$$

The points move in space under the influence of the local velocity at their location, which is composed of the incoming wind speed V_∞ and the contribution of all other vortices by means of induced velocities v as:

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} V_\infty \\ 0 \end{bmatrix} + \begin{bmatrix} \sum_{j\neq i}^{N} v_x(i,j) \\ \sum_{j\neq i}^{N} v_y(i,j) \end{bmatrix}. \tag{2.18}$$

The induced velocity components v_x and v_y can be calculated for each pair (i, j) as:

$$v_x = \frac{\Gamma_j}{2\pi} \frac{y_j - y_i}{r_{i,j}} \left(1 - e^{-\frac{r_{i,j}}{c^2}}\right), \tag{2.19}$$

$$v_y = \frac{\Gamma_j}{2\pi} \frac{x_j - x_i}{r_{i,j}} \left(1 - e^{-\frac{r_{i,j}}{c^2}}\right). \tag{2.20}$$

Where c is the characteristic core size with the distance between points i and j. The exponential term containing the core size is required because without it, as $r_{i,j}$ becomes smaller the induced velocity will increase boundlessly. The core size is taken constant for simplicity here, but there are also variants of the free vortex wake model where core growth is taken into account [21]. The distance between point i and j is given by $r_{i,j}$ as in equation 2.21 below:

$$r_{i,j} = \sqrt{(y_j - y_i)^2 + (x_j - x_i)^2}. \tag{2.21}$$

Figure 2.4 illustrates for the point at index 1, denoted by a green point, the relative distance vectors $r_{i,j}$ to all of the other vortex points. In this example only six points are used. One can imagine that as the amount of simulated points increases the amount interactions also increases dramatically. The orange circles represent the characteristic core size c. A balance between the simulation fidelity and simulation speed has to be found in order to achieve efficient simulations.

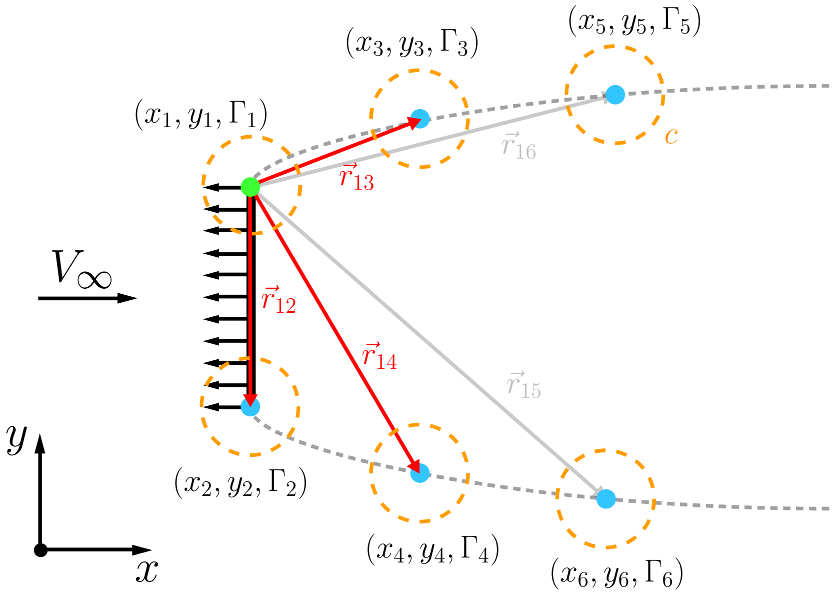


Figure 2.4: Vortex Points shed from the edges of the 2D actuator disk

Because each point moves as a result of its local velocity as denoted by equation 2.18, the simulation can calculate the new position of each point in discrete time iteratively. For example the vortex point at index 1 will move to index 3 and its new position can be calculated as:

$$\begin{bmatrix} x_3 \\ y_3 \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} V_x \\ V_y \end{bmatrix} \Delta t. \tag{2.22}$$

A state vector z can be chosen that contains all x, y and Γ states for each point of the simulation. Since at every time step k a new point is shed from the rotor, the amount of states would increase over time. In order to have a constant amount of states the model is initialised with all states present in a grid pattern, and zero vortex strength. By reassigning the states for every point at every time step, and re-initialising the first points a constant amount of points is retained. Essentially the points are being continuously recycled. The FVWM can now be described as an update equation of the form:

$$\underbrace{\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ x_4(k+1) \\ \vdots \\ x_n(k+1) \\ y_1(k+1) \\ y_2(k+1) \\ y_3(k+1) \\ y_4(k+1) \\ \vdots \\ y_n(k+1) \\ \Gamma_1(k+1) \\ \Gamma_2(k+1) \\ \Gamma_3(k+1) \\ \Gamma_4(k+1) \\ \vdots \\ \Gamma_n(k+1) \end{bmatrix}}_{z(k+1)} = \underbrace{\begin{bmatrix} P & \mathbf{0}_{n\times n} & \mathbf{0}_{n\times n} \\ \mathbf{0}_{n\times n} & P & \mathbf{0}_{n\times n} \\ \mathbf{0}_{n\times n} & \mathbf{0}_{n\times n} & P \end{bmatrix}}_{R} \underbrace{\begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \\ \vdots \\ x_n(k) \\ y_1(k) \\ y_2(k) \\ y_3(k) \\ y_4(k) \\ \vdots \\ y_n(k) \\ \Gamma_1(k) \\ \Gamma_2(k) \\ \Gamma_3(k) \\ \Gamma_4(k) \\ \vdots \\ \Gamma_n(k) \end{bmatrix}}_{z(k)} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ v_{x,1} \\ v_{x,2} \\ \vdots \\ v_{x,n} \\ 0 \\ 0 \\ v_{y,1} \\ v_{y,2} \\ \vdots \\ v_{y,n} \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{v(k)} \Delta t + \begin{bmatrix} 0 \\ 0 \\ V_\infty \\ V_\infty \\ \vdots \\ V_\infty \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \Delta t + \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ 0 \\ 0 \\ \vdots \\ 0 \\ y_1 \\ y_2 \\ 0 \\ 0 \\ \vdots \\ 0 \\ \Gamma_k \\ \Gamma_k \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{z_I(k)} . \tag{2.23}$$

with the number of points denoted as n, $m = n - 2$ and matrix P as:

$$P = \begin{bmatrix} \mathbf{0}_{2\times m} & \mathbf{0}_{2\times 2} \\ \mathbf{I}_{m\times m} & \mathbf{0}_{m\times 2} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \dots & 0 & 0 \\ & \ddots & \ddots & & & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 1 & 0 & \dots & 0 & 0 \end{bmatrix} \tag{2.24}$$

The vector $z_I(k)$ contains the x, y and Γ values of the points being initialised at the rotor tips. The coordinates x_1, y_1 and x_2, y_2 are constants and $\Gamma_k = \frac{1}{2} V_\infty^2 C_T^{'}(k) \Delta t$ is dependent on $C_T^{'}(k)$. The induced velocity vector $v(k)$ is nonlinear as the induced velocities depend non linearly on x_n, y_n as per equations 2.19, 2.20 and 2.21. The matrix P is stacked diagonally in a reassignment matrix R such that it is multiplied with each of the sets of states x_k, y_k and Γ_k. Multiplication by R of the states in combination with the addition of vector $z_I(k)$ handles the recycling and initialisation of the points with index 1. Additionally, it handles the reassignment of all states for the next iteration k, clipping the now redundant final states.

Implementation

A 2D free vortex wake model of a two turbine system was implemented in MATLAB in order to allow it to be used in numerical optimisation. To this end the model was formulated as having the input signal $C_T^{'}(k)$ as sole input and a performance metric as sole output. This output was chosen to be combined power output of a theoretical two turbine system, as maximisation of this value is desired. In addition the implementation has a set of environment parameters that are determined by the experimental setup and a set of model parameters pertaining to the implementation of the free vortex wake model itself.

The free vortex wake model requires the induced velocity vector $v(x)$ to be calculated at each time step. Since the distance between each set of points r_{ij} is needed for the calculation of the induced velocity of each point in the simulation the computational complexity is $O(n^2)$ if implemented with a nested for loop over all points $1, 2, ..., n$. Alternatively, a method was implemented that utilises matrix operations instead.

For the convenience of the calculation of the induced velocities the coordinates of points $1, 2, ..., n$ are stored in two matrices with odd numbered points in the top row, and even numbered points in the bottom row as:

$$C_x = \begin{bmatrix} x_1 & x_3 & ... & x_{n-1} \\ x_2 & x_4 & ... & x_n \end{bmatrix}$$

$$C_y = \begin{bmatrix} y_1 & y_3 & ... & y_{n-1} \\ y_2 & y_4 & ... & y_n \end{bmatrix}$$

In order to calculate the induced velocities the differences between the individual coordinates of all points need to be calculated. These differences can be subdivided into four parts: Top-Top, Top-Bottom, Bottom-Top and Bottom-Bottom differences.

The Top-Top differences in x, (TT_x), can be calculated from the top row of C_x:

$$TT_x = \begin{bmatrix} x_1 & x_1 & \dots & x_1 \\ x_3 & x_3 & \dots & x_3 \\ \vdots & \vdots & \ddots & \vdots \\ x_{n-1} & x_{n-1} & \dots & x_{n-1} \end{bmatrix} - \begin{bmatrix} x_1 & x_3 & \dots & x_{n-1} \\ x_1 & x_3 & \dots & x_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ x_1 & x_3 & \dots & x_{n-1} \end{bmatrix}$$

$$= \begin{bmatrix} x_1 - x_1 & x_1 - x_3 & \dots & x_1 - x_{n-1} \\ x_3 - x_1 & x_3 - x_3 & \dots & x_3 - x_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n-1} - x_1 & x_{n-1} - x_3 & \dots & x_{n-1} - x_{n-1} \end{bmatrix}$$

The Top-Bottom differences in x, (TB_x), can be calculated from the top and bottom rows of C_x as:

$$TB_x = \begin{bmatrix} x_2 & x_2 & \dots & x_2 \\ x_4 & x_4 & \dots & x_4 \\ \vdots & \vdots & \ddots & \vdots \\ x_N & x_N & \dots & x_N \end{bmatrix} - \begin{bmatrix} x_1 & x_3 & \dots & x_{n-1} \\ x_1 & x_3 & \dots & x_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ x_1 & x_3 & \dots & x_{n-1} \end{bmatrix}$$

$$= \begin{bmatrix} x_2 - x_1 & x_2 - x_3 & \dots & x_2 - x_{n-1} \\ x_4 - x_1 & x_4 - x_3 & \dots & x_4 - x_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ x_N - x_1 & x_N - x_3 & \dots & x_N - x_{n-1} \end{bmatrix}$$

The Bottom-Top and Bottom-Bottom differences can be calculated from C_x in the same fashion. By grouping the calculated differences in a matrix of the form:

$$D_x = \left[\begin{array}{c|c} TT_x & BT_x \\ \hline TB_x & BB_x \end{array}\right]$$

We obtain the following matrix:

$$D_x = \left[\begin{array}{cccc|cccc} x_1 - x_1 & x_1 - x_3 & \dots & x_1 - x_{n-1} & x_1 - x_2 & x_1 - x_4 & \dots & x_1 - x_n \\ x_3 - x_1 & x_3 - x_3 & \dots & x_3 - x_{n-1} & x_3 - x_2 & x_3 - x_4 & \dots & x_3 - x_n \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{n-1} - x_1 & x_{n-1} - x_3 & \dots & x_{n-1} - x_{n-1} & x_{n-1} - x_2 & x_{n-1} - x_4 & \dots & x_{n-1} - x_n \\ \hline x_2 - x_1 & x_2 - x_3 & \dots & x_2 - x_{n-1} & x_2 - x_2 & x_2 - x_4 & \dots & x_2 - x_n \\ x_4 - x_1 & x_4 - x_3 & \dots & x_4 - x_{n-1} & x_4 - x_2 & x_4 - x_4 & \dots & x_4 - x_n \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \dots & \vdots \\ x_n - x_1 & x_n - x_3 & \dots & x_N - x_{n-1} & x_n - x_2 & x_n - x_4 & \dots & x_n - x_n \end{array}\right]$$

The columns of D_x each contain the differences at point i to all points j with $i = j$ equal to 0. This will prove useful when calculating the sum of the induced velocities for each point i. The difference matrix for y, D_y, is analogous to D_x. And a matrix containing the distances between all points, R can be calculated from D_x and D_y as $D_x^2 + D_y^2$ where the squares are the element wise squares (`.^2` in MATLAB). The induced velocites for x and y can then be calculated by applying equation 2.19 and 2.20 to each element in D_x and D_y respectively in combination with R. If we then sum the elements per each column we obtain the sum of all induced velocities in x and y per point.

The model parameters require tuning as they have a large effect on the model output, to this end a convergence study has been performed and will be discussed later in this chapter.

Input signal	Environment parameters	Model parameters
Thrust coefficient: $C_T^{'}(k)$	Inflow velocity: V_∞	Time step: Δt
	Turbine diameter: D	Core size: c
	Turbine spacing: S	Vortex Points: P
	Air density: ρ	Grid spacing d_{grid}
		Simulation time: t_{sim}
		Performance delay: t_{delay}

Table 2.1: Free vortex wake model parameters.

Wind farm implementation

Now that the model itself has been discussed, the two turbine wind farm setup itself will be discussed. The two turbines are assumed to be of the same specification and are represented in the model in terms of a set of parameters as found in Table 2.1. The turbine dimensions are expressed in terms of rotor diameters in order to scale the problem. Because the goal is to study wake mixing, the relative positioning of the turbines is simplified to be inline of each other with respect to the incoming flow direction as illustrated in Figure 2.5. This setup represents the worst case scenario in terms of off design wind direction for such a two turbine system.

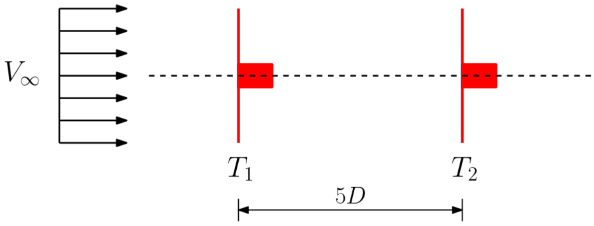


Figure 2.5: Two Turbine configuration.

In order to model the wake effects of the upstream turbine on the downstream turbine two distinct free vortex wake models were implemented. The models operate separately with an interface layer to model their interactions. This interface layer works by first calculating the intermediate average induced rotor velocity around the downstream turbine rotor and then using this as a velocity input for the free vortex wake model of the second turbine. In this way the effects of the velocity in the wake of the upstream turbine are propagated to the downstream turbine, while direct interaction between the sets of vortex points of the two models is avoided. Direct interaction between points of the two models would drastically increase the computational complexity, as for every point in both models the induced velocity from every other point will have to be calculated.

The velocity fields in the simulation can be extracted by calculating the induced velocities at every point in a 2D grid spanning the simulation environment. Figure 2.6 below shows an example of an isolated flow field for the wake of turbine 1 under a static $C_T^{'}$ generated as a result of the implemented free vortex wake model. As can be clearly seen in Figure 2.6, the simulated wake shows a significant reduction in axial velocity compared to undisturbed air. In addition the wake also shows known characteristics like a sharp boundary layer in the near wake, with a more turbulent boundary layer as the wake propagates downstream and wake mixing starts to occur. In the case of the FVWM however this turbulence is not the result of any modelled wake mechanism but the result of instability of the model itself.

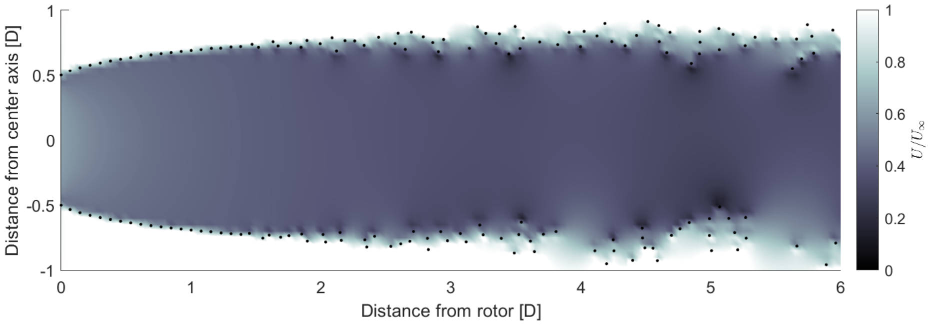


Figure 2.6: Turbine 1 flow field visualisation under a static coefficient of thrust signal.

When a sinusoidal signal is applied to the thrust coefficient the flow field now exhibits a curling up of the boundary layer and its subsequent breakup and thus wake mixing. The periodic nature of the curls causes high and low velocity regions propagating down the wake. In this case in terms of velocity it is not clear visually whether the average velocity in the wake has increased compared to a static thrust coefficient. The effectiveness of such an input signal will be investigated later in Chapter 4.

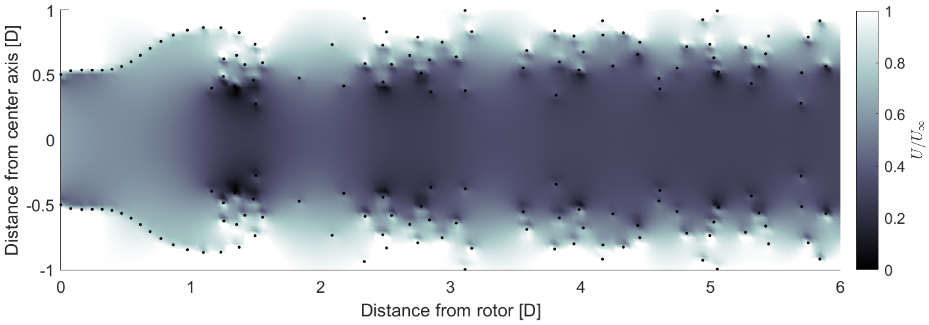


Figure 2.7: Turbine 1 flow field visualisation under a sinusoidal coefficient of thrust signal.

Power output

For use in optimisation, and more specifically the maximisation of the power output of the two turbine system, the power output needs to be extracted from the model. Since the velocity in the simulation can be determined at every time step and the local coefficient of thrust is an input to the model, an expression can be derived for the the total instantaneous power output. Equation 2.25 is the sum of the two individual turbines and is based on Equation 2.14.

$$P(k) = \sum_{i=1}^{2} \frac{1}{2}\rho\frac{\pi}{4}D^2 C^{'}_{T,i}(k) {V_{rot,i}}^3 \Delta t \tag{2.25}$$

At each time step the power $P(k)$ is only dependent on the thrust coefficients, $C^{'}_{T,i}(k)$, and the air velocity at the rotors $V_{rot,i}(k)$. All other terms are constant in time. As the second turbine does not have another turbine in its wake it does not need to take any wake mitigating measures. Therefor the thrust coefficient for turbine 2, $C^{'}_{T,2}(k)$, can be fixed at the Betz optimum, extracting as much energy from the incoming flow as possible i.e. operating under greedy control. Additionally the area of the actuator disk is now expressed in turbine diameter D.

The currently implemented model requires a certain amount of time before the first points that are shed from the edges of the actuator disk reach and pass the second turbine. Therefor, the velocity field around the second actuator disk can only accurately be calculated after the wake has developed to cover the simulation space. To account for this the simulation does not initially track performance until a time t_{delay} has passed. This time is chosen such that the simulation has enough time to develop the wake to reach the second turbine with some margin. Additionally the simulation time t_{Sim} represents the total simulation time over which the performance is tracked.

The expression for turbine power output, Equation 2.25, is based on the axial flow velocity at the rotors of the turbines $V_{rot,i}$. Thus the computation of the combined power output over time interval Δt requires the computation of the velocity at the rotors of both turbines. Since it is not computationally efficient to calculate the entire velocity field in the simulation environment when only the rotor velocities are required for the result, the rotor velocity is taken as the average velocity in a small area around the actuator disk that represents the rotor. By placing a grid in a small area around the actuator disks and averaging the calculated induced velocity over each point in this grid as a result of the vortex points released by the rotor, an approximation of the axial rotor velocities can be found. Figure 2.8 illustrates the grids around both rotors. Note that in the current implementation the rotor velocity of the second rotor is calculated from FVWM representing the second turbine, which is fed the velocity calculated at the second turbine from the first FVWM using the same grid.

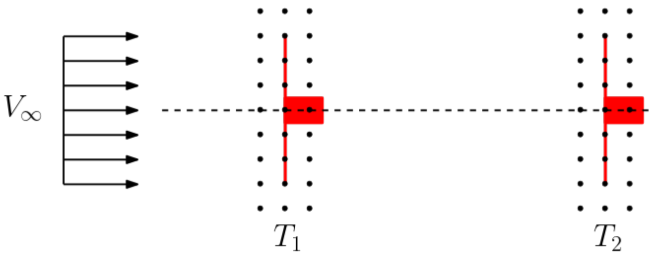


Figure 2.8: Rotor velocity grid layout in the simulation.

Power output correction

As mentioned in Section 2.1, the maximum theoretical power output is achieved at $C^{'}_{T} = 2$ and the power output follows Equation 2.13. From the theory in Section 2.1 we know that the maximum power production should be achieved at $C^{'}_{T} = 2$. In order to verify that the power output of the free vortex wake model follows this expected trend a set of simulations was conducted with static input signal over a range of $C^{'}_{T}$ between 1 and 3 for $D = U_\infty = 1$. As can be seen in Figure 2.9 the unmodified instantaneous power output not only severely overestimates the power produced, but also has its optimum around values of $C_T^{'}$ that are significantly larger than those of the theoretical model. In order to avoid the results of the future optimisations being skewed by this power output behaviour a correction function was applied, bringing the calculated power from actual rotor velocities in line with the theoretical model. The correction function is a simple linear function that is multiplied with the power output calculated by the FVWM from the rotor velocities to counteract the overestimation at larger $C_T^{'}$ values. The correction function can be parameterised as: $C_A C_T^{'} + C_B$, with $C_A = -0.0676$ and $C_B = 0.899$. Note that the optimum of the corrected power output function is now much closer to $C_T^{'} = 2$, at $C_T^{'} = 1.96$.

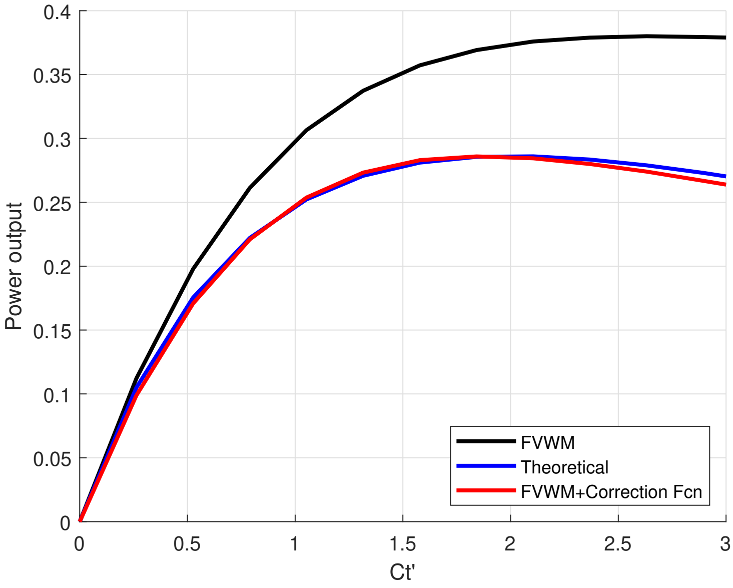


Figure 2.9: Power output plots for theoretical, corrected and original model.

The corrected system power output can now be denoted as:

$$P(k) = \sum_{i=1}^{2} \frac{1}{2} \rho \frac{\pi}{4} D^2 C_{T,i}^{'}(k) {V_{rot,i}}^3 \Delta t \left(C_A C_{T,i}^{'}(k) + C_B \right) \tag{2.26}$$

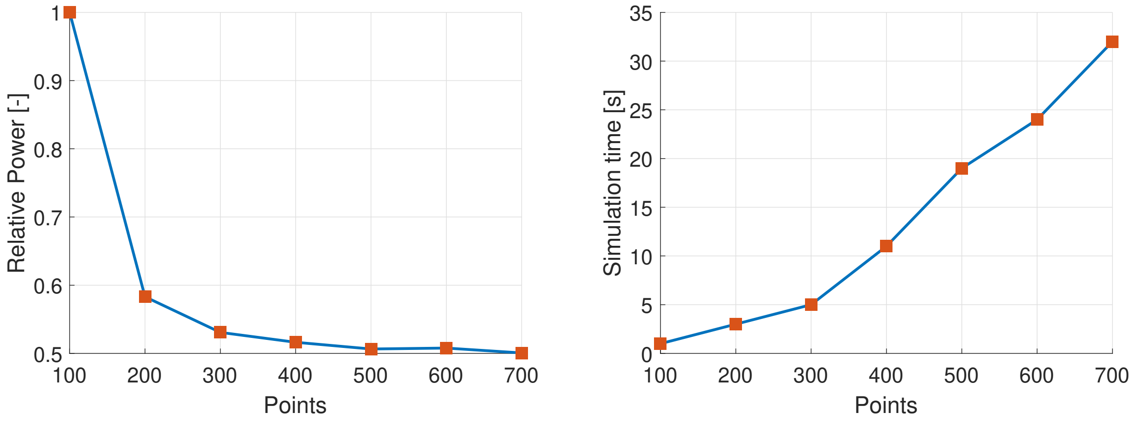
Convergence Study

The input and environment parameters in Table 2.1 are physical parameters that are set for a specific problem. The remaining model parameters do not directly relate to set physical properties of a given turbine set up and have to be chosen before an optimisation can be performed. The simulation time t_{sim} and performance delay t_{delay} can be chosen in such a way that the wakes in the simulation have time to develop before performance is tracked and such that the performance is tracked over a long enough time period to capture a realistic picture of the performance of the applied input signal. If the simulation time is taken too short bias will occur when the input signal will not contain enough full periods, shifting the mean $C_T^{'}$. The time step, core size, amount of vortex points and the grid spacing of the two grids around both turbines are parameters that need to be tuned in order to find a good balance between computational complexity and accuracy. To this end a convergence study is performed on these parameters to determine around what value the increase in precision no longer justifies the added performance cost.

The convergence study was performed on the implemented FVWM with a performance metric as output. This performance metric is the corrected aggregate power output of the two turbine system as defined by Equation 2.26. The power output is the value that needs to converge in order to have confidence in the precision of the FVWM when applying it for the identification of induction signals later in this thesis. The convergence of the power output was studied relative to the maximum power output result in the set of studied parameter values for each parameter. In this way the differences between the results can be viewed independently of the absolute values giving a more clear picture of the convergence.

The amount of vortex points and the time step can not be studied for convergence independently. Because at each time step a set of vortex points is released from the rotor tips and the number of vortex points is limited, decreasing the time step will lead to a situation where there are not enough points in the simulation to cover the wake to the downstream turbine. Therefor for each time step a separate study for the number of vortex points is needed.

For each Δt the convergence of the number of points was studied first. For a Δt of 0.1 Figure 2.10a below shows the model outputs as a function of the amount of points used in the simulation, and Figure 2.10b shows the associated required simulation time. Increasing the number of vortex points will increase the distance the wake will span behind the first rotor, but also drastically increase simulation time. The number of vortex points was chosen as $P = 300$ as any increases in the number of points beyond that point no longer justifies the added computation time.

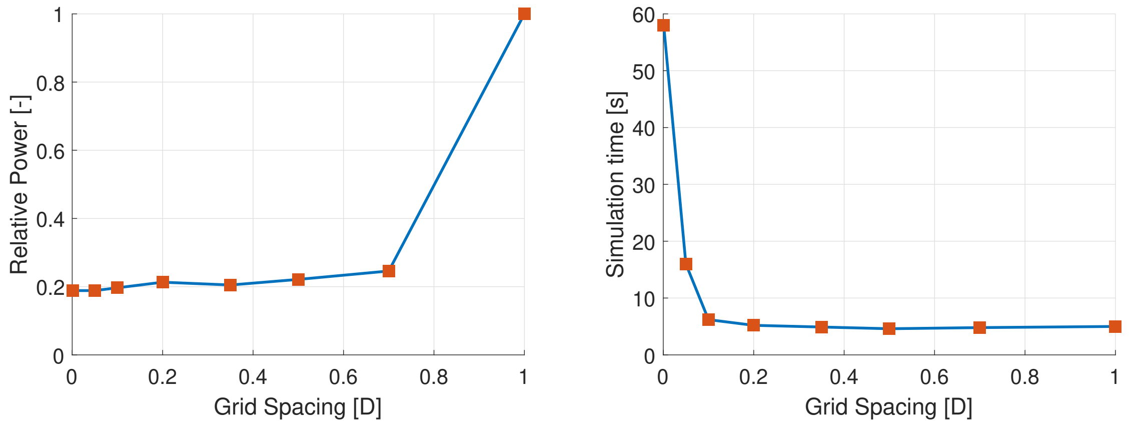


(a) Relative Power convergence.

(b) Simulation time.

Figure 2.10: Convergence study for number of vortex points for $\Delta t = 0.1$.

The grid around the actuator disks in the implemented model is defined by a set extents with variable resolution. This resolution determines the amount of points that make up the grid, and in this case was defined by the spacing between grid points. As with the vortex points, increasing the amount of points will lead to greater accuracy of the rotor velocities that are essential for calculating the power output, but at the cost of increasing computational time. Figure 2.11a shows the convergence of the model output for different grid spacing settings, and Figure 2.11b shows the associated simulation time. The grid spacing was chosen at $d_{grid} = 0.1$ as decreasing the spacing further does not provide any significant improvements to the convergence of the output, but does dramatically increase the simulation time.

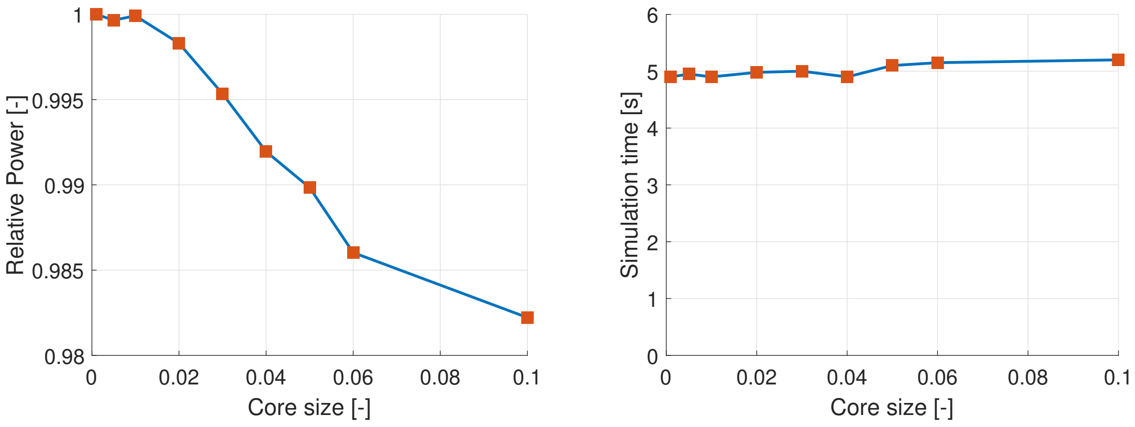


(a) Relative Power convergence.

(b) Simulation time.

Figure 2.11: Convergence study for grid spacing for $\Delta t = 0.1$.

The characteristic core size was also studied for convergence. Decreasing the core size will not lead to the model performing additional calculations every time step, but it does impact how the vortex points interact with each other through the induced velocities as governed by equations 2.19 and 2.20. Figure 2.12b confirms the core size did not have a significant impact on simulation time as expected. Figure 2.12a shows that the core size itself also has minimal effect on the model output, as the relative power remains within 2% of the largest tested value, but does seem to converge slightly below $c = 0.01$, motivating the choice for the core size at this value.



(a) Relative Power convergence.

(b) Simulation time.

Figure 2.12: Convergence study for core size for $\Delta t = 0.1$.

3 Genetic Algorithm optimisation

In this chapter the genetic algorithm optimisation method and why it is well suited for the optimisation problem of finding DIC control signals by simulating with the FVWM will be discussed.

3.1. Origins

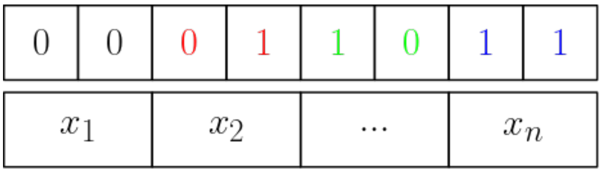
Genetic algorithms are part of a field in technology generally referred to as evolutionary computing. The algorithms involved with evolutionary computing in turn are referred to as evolutionary algorithms. Evolutionary computing is based on the Darwinian principles of evolution which centers around survival of the fittest as a way of solving problems [24]. The algorithms mirror how in biology the genetic diversity of a population in combination with mutations allows adaptation of the genome to environmental circumstances through evolution. The insight that these principles could be used in automated problem solving can be traced back to as early as the 1940s [24]. Unlike conventional convex optimisation techniques GA optimisation is not based on model characteristics like the gradient or hessian and is only dependent on the model output. This allows genetic algorithms to be used in problems where gradient based optimisation methods generally struggle, for example problems that have multiple local maxima or minima. The two main applications of genetic algorithms are in the optimisation of the performance of a certain system, and the fitting of quantitative models [25]. The first and most well known method is the version of GA that this chapter will focus on.

3.2. Main Principle

In GA optimisation the optimisation problem is approached as an evolutionary process, where model parameters fill the role of genes. A set of genes can be combined in the evolutionary equivalent, namely a chromosome. Each chromosome is a possible optimal parameter set that will be tested for fitness against a large set of other chromosomes that make up the population. The GA then simulates reproduction between the fittest chromosomes within the population in order to generate a new population that can be tested for fitness. This reproduction part is usually performed with a crossover step and a mutation step. In the crossover step chromosomes of the best performing parents (elite parents) are split and recombined into new chromosomes called children. The mutation step generates additional children by randomly mutating the genes of elite parents into new children. These newly generated children along with a set of the elite parents then make up the new population. By performing both crossover and mutation the new population contains a set of chromosomes that are genetically different but based upon their parents thereby allowing the next iteration to search a new part of the parameter space that is availible to the GA. Thus with every generation the GA aims to iteratively improve the fitness of the population. Because between every generation a set of elite parents are maintained each generation will at least contain an equal performing chromosome to the best of the last generation.

Parameter definition

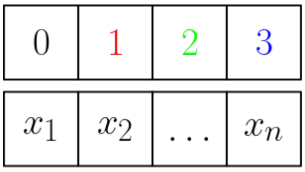
Historically the optimisation variables in genetic algorithms are defined in binary form, i.e. the optimisation parameters are encoded in binary and these methods are sometimes referred to as classic GAs [25][26]. In this case each optimisation parameter uses one or multiple genes to encode a numerical value. An example of a binary encoded chromosome that encodes three parameters with values $1, 2, 3$ can be seen in Figure 3.1. $x = 1$ for example is encoded in binary as 01. Note that in this case it takes six genes to encode three integer parameters with values $x_i < 3$. Optimising a larger set of parameters, or problems that require a higher level precision will require encoding of their numerical values in binary with a larger amount of genes thereby making the search space increasingly bigger causing inefficiencies [25]. In addition the conversion between real and binary representation for testing the fitness of the parameters at each step requires extra computational power [25].



0	0	0	1	1	0	1	1
x_1		x_2		$\cdots$		x_n	

Figure 3.1: Binary coded chromosome encoding $x = [0, 1, 2, 3]$.

Encoding the parameter values in real numbers allows the chromosomes to consist of a single gene for each parameter. This representation is simpler to implement and use since it avoids any conversion and potential issues with binary representations like bias [26].



0	1	2	3
x_1	x_2	$\ldots$	x_n

Figure 3.2: Real value coded chromosome encoding $x = [0, 1, 2, 3]$.

Initial population generation

First the initial population of chromosomes is generated. This population is usually simply generated randomly within the given set constraints [26], but depending on the specific optimisation problem could also involve more complicated generation methods [24]. The size of the population is an important parameter in the GA optimization problem as it determines the gene pool size. A larger population size will allow more distinct genes to be evaluated, but also increase the computational cost of evaluating a single generation for fitness. Introducing previously found solutions into the initial population is also possible and could aid the optimisation by adding well performing genes to the population early.

Selection

At every generation each chromosome is tested using a fitness function which outputs a fitness score that is used to compare the performance of the competing chromosomes. Next the score is scaled according to a specified method which is called the fitness scaling function, resulting in what are called expectation values. In practice the fitness function represents the performance of the system that is being optimised. With each chromosome scored they can be ranked from best to worst performing. In order to generate next generations children from the current population parents need to be selected for crossover. The choice of parents can follow differing methods but in general better performing chromosomes are more likely to become parents for the next generation. This is essential in driving the evolutionary process that is the basis for GA optimisation [26].

Crossover

In the crossover phase the genes of two selected parents are combined in a certain way as to produce genetically unique children. By performing crossover with top performing parents, the genes of the generated children will consist of a mix of successful genes from the previous generation. Crossover can be performed in a multitude of ways. The most basic crossover method is single point crossover. In single point crossover a random crossover point is selected where the chromosomes of the parents are both cut and the genes past the crossover point are swapped to create two unique children chromosomes. Figure 3.3 below illustrates this process for chromosomes with 8 genes.

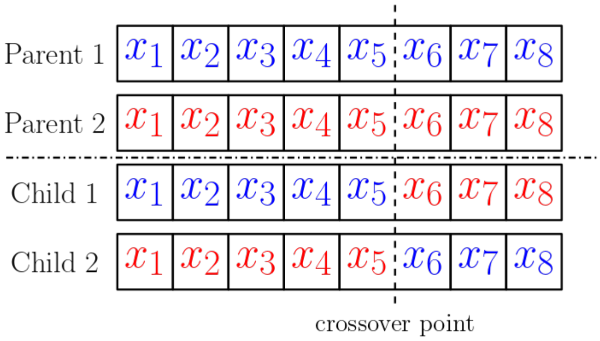


Figure 3.3: Example of crossover.

Crossover could also be performed using multiple crossover points or even on a per gene based swapping approach like uniform crossover. Again a real coded genetic algorithm avoids some of the issues that occur when performing crossover on binary coded chromosomes [26]. One of these issues is that multiple genes are needed for each parameter in binary encoded GAs leading to crossover occurring within the genetic material of a parameter value instead of crossing over whole parameters with real valued GAs. Crossover is however not limited to just swapping genes, it can also involve any method merging or blending genes by averaging for example. Crossover functions can be customised or even combined with other crossover functions in order to include extra selection steps or other adaptations in order to suit specific problems [24].

Mutation

The mutation phase is essential in introducing new genes into the population, thereby creating genetic diversity. This genetic diversity allows the GA to explore new genes and thus improves the ability to search for the optimal solution. In the mutation phase the genes of a parent are mutated on a per gene basis with a small probability. Like with the crossover step, the mutation methods can be chosen to suit the specific optimisation problem. Two types of mutation are uniform and dynamic mutation [26] as the name suggests in dynamic mutation the mutation process changes over the course of generations, while static mutation does not. This is useful as it could be beneficial to introduce a larger diversity in the initial generations in order to explore the parameter space but a more granular approach to refine the genes in the later stages of the optimisation problem.

Iteration and termination

The population for the next generation is built from a combination of the children generated though the crossover and mutation phases and a set of elite parents that remain unmodified. Therefor, the top performing genes will remain and the genetic algorithm will not lose performance over generations. The ratio between elite parents and new children again can be tuned to suit the specific optimisation problem.

Termination of the GA is generally initiated when one of multiple criteria are met. These criteria can again differ per implementation but in general these include [24]:

1. Maximum number of generations is exceeded
2. Maximum total elapsed CPU time is exceeded
3. The improvement between generations has fallen within the set threshold during a set number of generations

Once the algorithm is terminated the top performing chromosome of the current population is selected as the solution to the GA.

Considerations

The performance of genetic algorithms is dependant on a multitude of settings, parameters and functions that influence the algorithm. These options can be tuned to better suit the given optimisation problem in order to improve the efficiency of the optimisation. The population size and number of generations are two of the most influential parameters in any GA, and choosing them appropriately impacts the ability of the GA to find a global optimum significantly.

Genetic algorithms have some key benefits over gradient based optimisation that motivate the choice for this optimisation method to be used with the task of optimising DIC signals. The main benefit relevant to this work is the ability to search a vast parameter space in a relatively efficient way. As van den Broek et. al. [11] noted, gradient based optimisation methods can be very dependent on the initial guess in an optimisation space that has multiple local optima or is otherwise very flat, suggesting to use multi-start optimisation in future research. Due to the nature of GA optimisation, with a selection of elite chromosomes progressing to the next generation, the solver can simultaneously investigate multiple local optima by retaining the genes of not just the best performing chromosome. Combined with the initial population being distributed over the parameter space the GA should be less sensitive to local optima affecting the result, while also being more efficient than multi-start gradient based optimisation due to the parallel investigation of optima. Another benefit is that the implementation of genetic algorithms does not require gradient information.

4 Results

This chapter will discuss the results of the different optimisation cases that were performed. First a short recap of the general optimisation problem is given. Then sinusoidal input signals will be covered, followed by the introduction of a semi-free optimisation method. Finally a discussion section will provide an overview of the obtained results and put them into context.

4.1. Optimisation of the free vortex wake model

In this work a two turbine wind farm simulation setup has been implemented in a 2D free vortex wake model in MATLAB. The turbines are placed in line with each other with respect to the incoming wind at a spacing of $5D$. The model uses the local coefficient of trust for the first turbine, $C_T^{'}(k)$, at every time step as an input. The second turbine is operated under greedy control as it does not need to take the performance of downwind turbines into account. The input can be generated at every time step by a parameterised function like a sine to represent the DIC signal. The goal is to optimise the parameters of this DIC signal to maximise the power output of the FVWM. Therefor the cost function used for evaluation of the fitness of the chromosomes in the GA will be the aggregate power output of the two turbine system as described by Equation 2.25 in Chapter 2.3. By using multiple distinct parameterised functions as input signals this setup allows further exploration of DIC. In addition model parameters like turbine spacing can be varied in order to investigate their effects on the DIC optimum. The input signal is constrained in order to prevent unrealistic solutions and define the parameter search space. These constraints will be adapted to the specific optimisation problem and hand and will be covered in the relevant sections.

4.2. Sinusoidal input signals

As discussed in Section 2.2 previous research into DIC input signals by Munters and Meyers [10] [16] has identified that a sinusoidal input with a characteristic frequency of $St = 0.25$ is effective in triggering wake mixing and thereby improving the performance of wind farms under waked conditions. Further research by van den Broek et. al. [11] has corroborated this phenomenon and found similar results for the characteristic frequency at $St = 0.2$. The different approach in modelling with Munters and Meyers using LES simulations and van den Broek et. al. using a FVWM implementation suggests that the lower fidelity FVWM could be used in order to study DIC imput signals more efficiently than LES would allow due to the computational complexity involved. The aim of this thesis is to expand on this previous research by investigating DIC beyond a single sine utilising the FVWM in conjunction with GA optimisation to search the large parameter space.

4.2.1. Single Sine

First an attempt was made to recreate the findings of Munters and Meyers using the Free vortex wake model and GA optimisation. To this end the GA was configured to optimise three parameters that describe a sinusoidal signal:

$$\alpha + \beta sin(\omega t) \tag{4.1}$$

Where α represents the base level of the signal, β represents the amplitude and ω represents the frequency in rad/s. For this optimisation the frequency was expressed in terms of the Strouhal number in the fitness function as:

$$St = \frac{fD}{V_\infty} = \frac{1}{2\pi}\frac{D}{V_\infty}\omega \tag{4.2}$$

Solutions to the genetic algorithm are of the form:

$$x^* = \begin{bmatrix}\alpha & \beta & \frac{1}{2\pi}\frac{D}{V_\infty}\omega\end{bmatrix} = \begin{bmatrix}C'_{Tbase} & dC'_T & St\end{bmatrix} \tag{4.3}$$

The optimisation parameters are constrained by an upper and lower bound in order to limit the parameter search space to a limited but realistic range and thereby improve the efficiency of the genetic algorithm. The simulation is run for 900 time steps, where during the first 30 time steps the performance is not tracked in order to allow the flow to develop to the quasi steady state that results from the sinusoidal input i.e. $t_{sim} = 90$, $t_{delay} = 3$ and $\Delta t = 0.1$. The turbines are placed at a distance of 5 rotor diameters (or 5D) apart, and it is assumed that the second turbine is operating at the Betz optimum of $C'_T = 2$, i.e. under greedy control policy. For the optimisation , the genetic algorithm is configured to use 15 generations and a population size of 50. Choosing the population size relatively large compared to the amount of optimisation parameters allows the GA to explore a larger parameter space. By experimentation it was found that for this case 15 generations was enough to converge to a solution and further generations would not lead to any significant improvements.

Derating

The genetic algorithm produces the following solution:

$$x^* = \begin{bmatrix}C'_{Tbase} & dC'_T & St\end{bmatrix} = \begin{bmatrix}0.881 & 0.854 & 0.418\end{bmatrix} \tag{4.4}$$

It can be observed that the base level, and thus mean, C'_T is significantly below the Betz optimum of $C'_T = 2$. This derating of the average coefficient of thrust for the upstream turbine is notable, as this phenomenon has also been observed in static induction control optimisation using other theoretical models [8], and to a lesser extent in [11]. However it has been shown that operating the upstream turbines in such a state does not provide a benefit to downstream turbines in real world applications [8]. Like in Munters and Meyers [16] the base level C'_T can also be fixed at the Betz optimum. Their implementation is shown in Equation 2.15. Fixing the mean coefficient of thrust will eliminate the tendency of the optimisation of the free vortex wake model to derate the first turbine, and allow optimisation of the amplitude and frequency of the input signal.

Optimising the same problem with a C'_T constrained to the Betz optimum yields the following result:

$$x^* = \begin{bmatrix}C'_{Tbase} & dC'_T & St\end{bmatrix} = \begin{bmatrix}2.000 & 1.984 & 0.284\end{bmatrix} \tag{4.5}$$

Comparison to existing research

The GA optimised input signal differs from the original Pulse signal identified by Munters and Meyers [16]. While the base level C'_T is the same, both the amplitude and the frequency in the GA optimum are slightly higher. Compared to the signal found by Broek et. al. [11], the amplitude is higher, the C'_T is not derated and the amplitude is significantly higher. An overview of the three signals discussed so far can be found in Table 4.1.

The mean velocities over the simulation time in the isolated wake of turbine 1 are shown in Figure 4.1 below. The plot shows the mean velocities over the wake cross section between $[-0.5D, 0.5D]$, i.e. over the actuator disk of turbine 2, over time and thus represent the mean wake velocity a hypothetical second turbine might experience at that specific distance behind turbine 1. As can be seen the pulse, the van den Broek pulse and the GA optimum show a higher average wind velocity when compared to greedy control, with the pulse and GA optimum also exhibiting earlier wake velocity recovery starting around $2.5D$. The GA shows a higher average wake velocity near $5D$ than the pulse and van den

	Original Pulse	Broek et. al. Pulse	GA optimum
C_T'	2	1.75	2
dC_T'	1.5	0.87	1.98
St	0.25	0.2	0.28

Table 4.1: Sinusoidal input signal parameter comparison

Broek pulse do in these simulations. The van den Broek pulse also seems to have a later start to wake recovery, seemingly starting around $4.5D$.

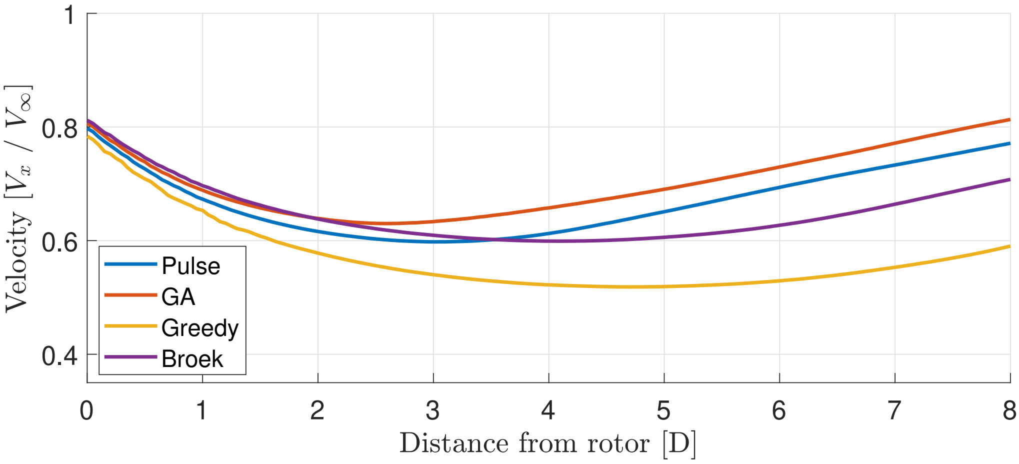


Figure 4.1: Mean velocities for the isolated wake of turbine 1.

Figure 4.2 shows the mean centerline wake velocities of the pulse, GA optimum, the van den Broek Pulse and greedy control strategies over time. As one would expect from a wake velocity profile the centerline velocities lie below the average wake velocities shown which in Figure 4.1. The aim of plotting the centerline velocity is to illustrate the wake recovery and rollover occurring as a result of the applied DIC. The centerline velocity is taken as the velocity on the x axis has the largest reduction in velocity as a result of the wake, and its recovery trails all other parts of the wake cross section, therefor recovery of centerline velocity is and indication for general wake recovery.

The differences between the DIC methods and greedy control are more apparent in these centerline plots than they are in the average wake velocity plots as both the pulse and the GA optimum cause a sharp drop in velocity at around $2.5D$, $3D$ respectively, while the van den Broek Pulse has a slight drop around $6.25D$. These drops in velocity can be associated with the wakes collapsing over themselves for the first time around these areas. Following their collapse, the wakes of the DIC methods recover velocity quickly which is a likely result of the expected wake mixing that occurs when the wake starts that characteristic curling effect that is in this case induced by DIC. Notably the van den Broek Pulse experiences the rollover later than the other two DIC signals, and also behind the second turbine placed at $5D$. This might be due to the significantly lower amplitude compared to the two other DIC signals, leading to less aggressive DIC and thus a smaller effect. Similarly to the other two DIC methods, the wake of the van den Broek Pulse recovers velocity quickly after the rollover compared to baseline greedy control recovery. In these simulations the GA optimum control method results in an earlier initial roll over effect than the pulse, with the centerline velocity also recovering significantly earlier even surpassing the centerline velocity in greedy control before reaching $5D$ where the 2nd turbine was placed in the optimisation. The pulse does not recover to above greedy control centerline velocity before this point, while the van den Broek Pulse remains higher than greedy almost entirely. It is interesting that while the centerline velocity of the pulse is lower than the greedy control centerline velocity at the second turbine, the Pulse still outperforms greedy control in the performed simulations.

However considering the mean wind velocity over the actuator disk of the Pulse at $5D$ is higher than that of greedy control this difference in performance does make sense.

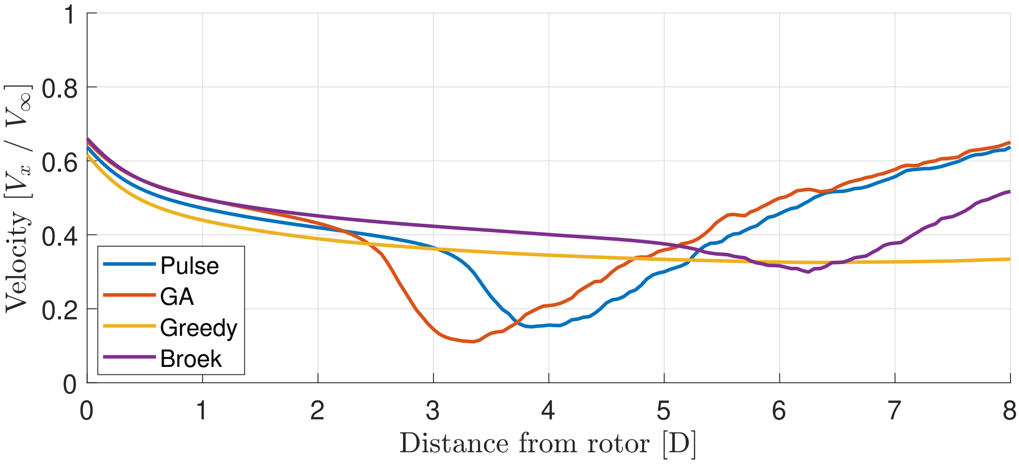


Figure 4.2: Centerline velocities for the isolated wake of turbine 1.

4.2.2. Additional Harmonics

The GA optimisation method allows easy expansion of the amount of optimisation parameters that would be too costly to perform using parameter searches like those performed by Munters and Meyers [16] and Frederik et al. [27] to study DIC. In order to give the optimisation more freedom to explore solutions that could replicate the erratic input signal that was found to trigger wake mixing the simple sine model proposed by Munters and Meyers in [16] is expanded to include additional harmonics. This is achieved by superposition of multiple parameterised sine waves and the addition of the required additional constraints.

Optimization with two sines

First an input signal with two superimposed sine waves will be investigated and compared against the original pulse and GA optimised pulse. The parameterised input signal requires two additional optimisation parameters and now takes the form:

$$C_T^{'}(t) = C_{Tbase}^{'} + dC_{T,1}^{'} sin\left(2\pi St_1 \frac{U_\infty}{D} t\right) + dC_{T,2}^{'} sin\left(2\pi St_2 \frac{U_\infty}{D} t\right) \tag{4.6}$$

where again the frequencies are expressed as St_1, St_2 respectively. In order to keep the input signal within realistic bounds, a constraint was added on the amplitudes of the first and second sines such that conjunction with the already present bound that $C_{Tbase}^{'} > 0$, $dC_{T,i}^{'} > 0$. the $C_T^{'}(k)$ values remain realistic.

$$C_{Tbase}^{'} - dC_{T,1}^{'} - dC_{T,2}^{'} > 0 \tag{4.7}$$

In addition in order to ensure uniqueness of each evaluated chromosome and thereby improve efficiency a constraint was added to the frequencies of both sines as:

$$St_1 > St_2 \tag{4.8}$$

By applying this constraint the situation where two chromosomes contain two of the same superimposed sine waves but encoded on different genes within the chromosome is prevented.

With the added optimisation parameters for the second sine the chromosomes now have the form:

$$x^* = \begin{bmatrix} C_{Tbase}^{'} & dC_T^{'} & St & dC_{T,2}^{'} & St_2 \end{bmatrix} \tag{4.9}$$

Optimising using the GA with increased population size of $P = 100$ and number of generations $G = 50$ to reflect the increase in number of optimisation parameters the following solution was found:

$$x^* = \begin{bmatrix} C'_{Tbase} & dC'_{T,1} & St & dC'_{T,2} & St_2 \end{bmatrix} = \begin{bmatrix} 2 & 1.71 & 0.24 & 0.24 & 0.26 \end{bmatrix} \tag{4.10}$$

$dC'_{T,1} = 1.71$ has decreased compared to the single sine optimum to reflect the addition of the second sine with its amplitude $dC'_{T,2} = 0.24$ and to respect the constraints. Summing the results in a $dC'_{T,1} + dC'_{T,2} = 1.95$ which is very close to the original dC'_{T}. Both frequencies St_1 and St_2 are very close to each other which would suggest the superposition of both sines to appear as a single sine signal. However looking at the plot of $C'_T(t)$ in Figure 4.3 it can be observed that the signal appears to have two very distinct frequencies superimposed, with the highest frequency being visually similar to the frequency found for the single sine case. Interestingly a Fourier transform only shows the two original frequencies. This effect of the two very close frequencies causing a signal that visually has two very distinct frequencies could be caused by interference [28].

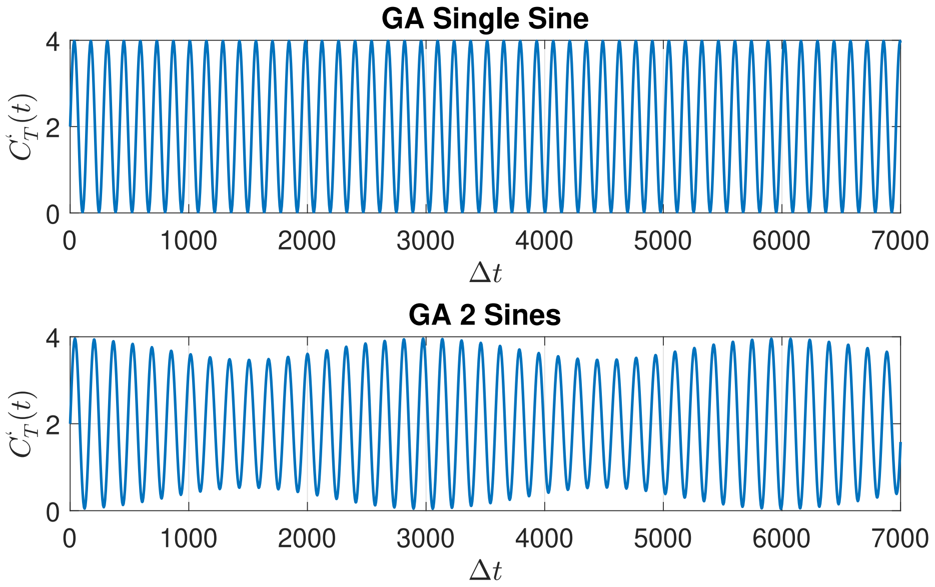


Figure 4.3: Comparison of $C'_T(t)$ plot over time for the single sine and 2 sine GA solutions.

In terms of performance the 2 sine GA optimum performs slightly better than the single sine variant by about 4% in the implemented FVWM, indicating that there might be more complex DIC signals than just the single sine that have potential.

Optimization with three and more sines

Given the fact that superimposing a second sine wave to the signal seems to provide a performance boost the optimisation was extended to investigate whether more than two sines would also yield benefits. To this end the optimisation parameters were once again expanded to accommodate an additional sine signal with an amplitude and frequency term, namely $dC'_{T,3}$ and St_3. With the added parameters the parameter space that needs to be explored expands significantly and thus the GA was adapted to use $P = 200$ population size with $G = 50$ generations. With a resulting amplitude $dC'_{T,3} = 0.005$ and a frequency of $St_3 = 0.08$ the third sine was insignificant compared to the first two. This leads to the conclusion that for the current implementation of the FVWM superimposing more than three sines into the DIC signal does not lead to any performance benefits.

4.2.3. Phase offset

In addition to superimposing two sines of different amplitudes and frequencies a phase offset parameter was introduced. The inclusion of a parameter for the phase offset between the two sine waves allows for further exploration of input signals that trigger wake mixing in the free vortex wake model. The extra parameters give the input signal more degrees of freedom to replicate the erratic nature of the type of signal that Munters and Meyers identified in [16].

The parameterised input signal now takes the form:

$$C_T^{'}(t) = C_{Tbase}^{'} + dC_{T,1}^{'} sin(2\pi St_1 \frac{U_\infty}{D} t) + dC_{T,2}^{'} sin(2\pi St_2 \frac{U_\infty}{D} t + \phi)$$

where ϕ represents the phase offset parameter. To reflect the added parameter in the optimisation the population size was increased to $P = 200$. In this case increasing the amount of generations did not yield any further benefits, with the GA converging to a solution before reaching $G = 50$.

With the phase offset added the GA reached the following solution:

$$x^* = \begin{bmatrix} C_{Tbase}^{'} & dC_{T,1}^{'} & St & dC_{T,2}^{'} & St_2 & \phi \end{bmatrix} = \begin{bmatrix} 2 & 1.71 & 0.24 & 0.24 & 0.26 & -0.64 \end{bmatrix} \tag{4.11}$$

The optimisation now finds the same parameter values as the optimisation without a phase offset with the addition of a $\phi = -0.64$ $[rad]$ phase offset. Figure 4.4 shows the resulting signal compared to the two sine GA optimum. Again the base frequency of the signal remains visually the same, which is expected given that the frequencies of the two sines have not changed in the optimisation including phase compared to the optimisation without. The phase offset does however change how the two sine components interact and causes the signal to visually have a slightly higher apparent second frequency superimposed over the base frequency. Visually being the key word as again a Fourier transform only shows the two original frequencies, with the apparent visual frequencies likely being a result of interference. The performance of the signal including a phase offset is almost exactly equal to that of the signal with just the two sines with a 0.25% performance in the benefit of the latter.

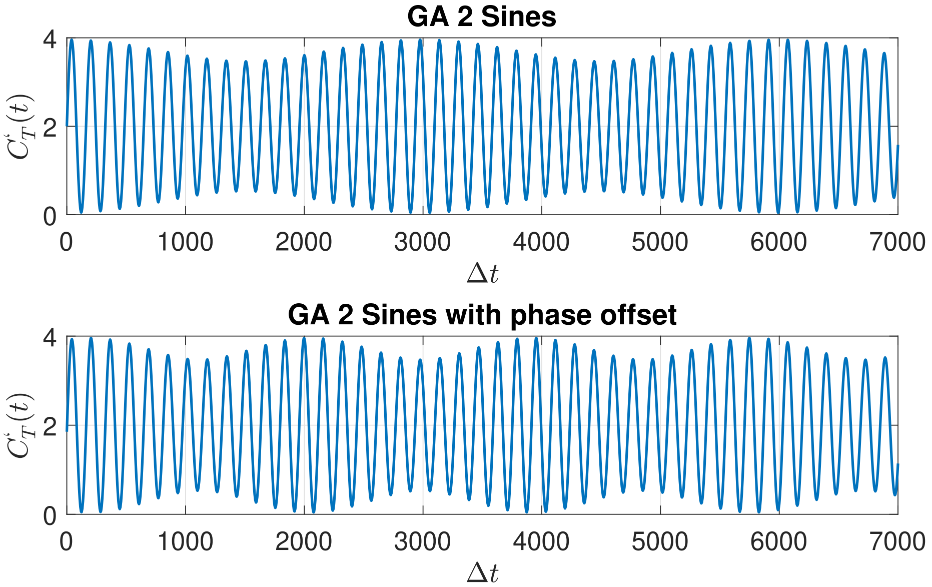


Figure 4.4: Comparison of $C_T^{'}(t)$ plot over time for the 2 sine and 2 sine with phase offset GA solutions.

4.2.4. Performance comparison

Gathering all results obtained in this chapter so far, Table 4.2 can be expanded to include the newly found signals for comparison and easy reference in Table 4.2.

Figure 4.5 shows a comparison of the performance of all discussed signals ordered from least to highest performing. In the implemented FVWM all of the signals outperform greedy control by a significant margin ranging from $11 - 21$%. The van den Broek pulse is the worst performer which is most likely caused by the amplitude being so low relative to the other signals, failing to cause the wake

	Original Pulse	Broek et. al. Pulse	GA 1 Sine	GA 2 Sines	GA 2 Sines + Phase offset
$C_T^{'}$	2	1.75	2	2	2
$dC_T^{'}$	1.5	0.87	1.98	1.71	1.71
St_1	0.25	0.2	0.28	0.24	0.24
$dC_{T,2}^{'}$	0	0	0	0.24	0.24
St_2	0	0	0	0.26	0.26
ϕ	0	0	0	0	-0.64

Table 4.2: Sinusoidal input signal parameter comparison.

rollover effect associated with faster wake recovery. The differences in found optima could be a result of differences in the implementation of the FVWM in both simulation frameworks. The signals including a second harmonic component appear to benefit from a significant jump in performance over the single sine DIC signals. The additional sine causing an an apparent added low frequency oscillation likely due to interference with the two harmonic components of similar frequency appears to accelerate wake recovery and improve performance.

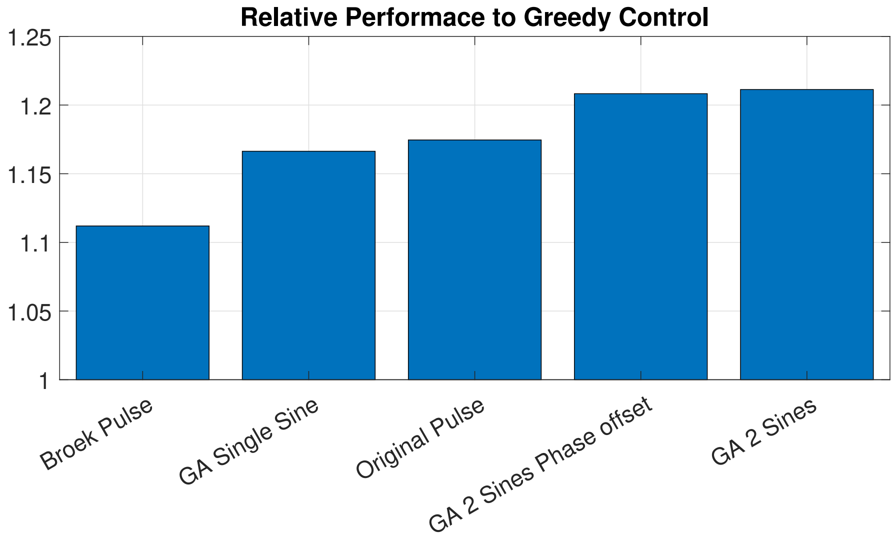


Figure 4.5: Performance comparison relative to greedy control for all discussed signals.

4.2.5. Turbine spacing

The spacing between wind turbines is an important parameter in wind farms and thus it would be interesting to study the effects of this spacing on the optimal input signal. Larger spacing between turbines allow for more natural wake recovery and thus less overall power losses than smaller spacing would. The benefits and characteristics of applying a sinusoidal input on the coefficient of thrust are therefor expected to change and provide a smaller benefit to the overall power production. Interestingly Munters and Meyers [16] found that in their simulations the parameters of their optimal input signal did not depend on stream wise turbine spacing, and even for a larger spacing significant gains could be achieved from applying DIC. Van den Broek et. al. only investigated turbines at a spacing of $5D$.

Performance of DIC at different spacings

Using the free vortex wake model, simulations were performed for greedy control, the original pulse from [16], the van den Broek pulse from [11] and the GA optimised pulse over a range of turbine spacing values. The relative performance compared to greedy control for each combination of input signal and turbine spacing are shown in Figure 4.6. From Figure 4.6 it is clear that the original pulse, the van den

Broek Pulse and the GA optimised pulse perform significantly better than the greedy control case in these simulations, with the van den Broek pulse not performing as well as the other two. In addition it can be observed that the system of two turbines performs better for larger turbine spacings. The GA optimised pulse is almost equivalent in performance to the original pulse, indicating that there is a range of well performing parameter combinations like the parameter sweep that Munters and Meyers performed in [16] already suggested. In their case Strouhal numbers between 0.2 and 0.3 seem to perform well in combination with amplitudes between 1 and 2. Their suggested optimum of $(St, A) = (0.25, 1.5)$ is around the geometric centre of this approximate range. The lesser performance of the van den Broek pulse compared to the other two might be explained by the lower amplitude not causing the rollover effect in this implementation of the FVWM leading to less wake mixing occurring. An interesting point of future research might be to investigate the penalisation of higher frequencies and amplitudes in the cost function of the GA in order to arrive at an efficient Pulse variant with respect to the stresses that are introduced on the components in the turbine by applying more aggressive DIC.

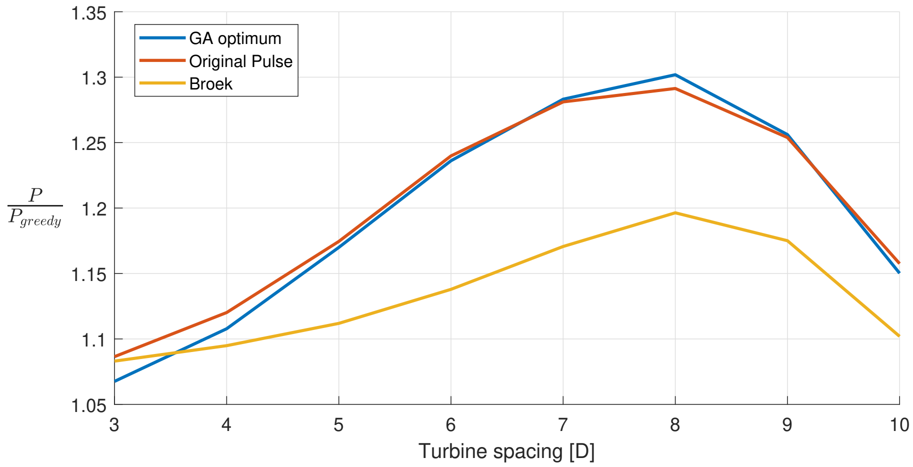
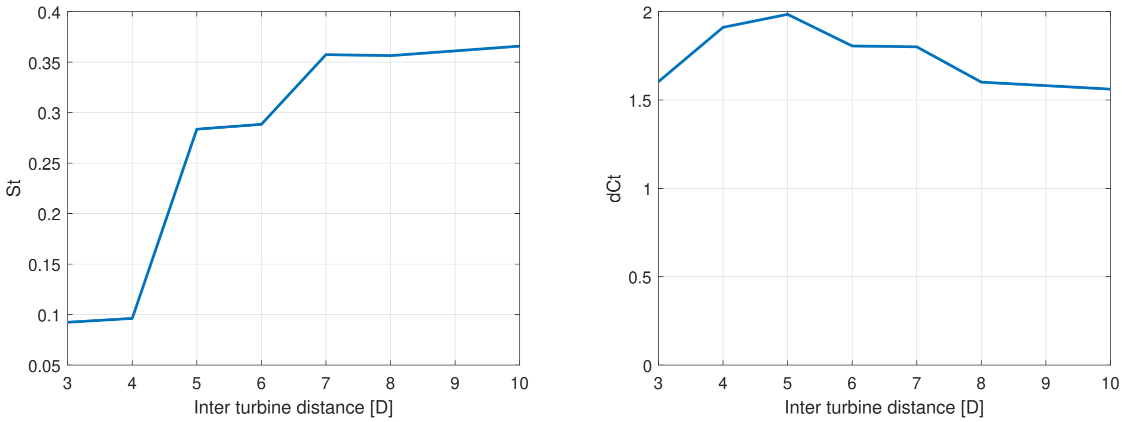


Figure 4.6: Performance at different turbine spacings for the original pulse, the van den Broek pulse and the GA optimised pulse relative to greedy control.

Optimization at different spacings

The GA optimised pulse was optimised at a turbine spacing of 5D. In order to investigate the claim that the original pulse is robust with respect to turbine spacing [16] additional optimisations were performed. For a range of turbine spacings the genetic algorithm was run under the same parameters as the initial 5D optimisation and the resulting optimum parameters in terms of Strouhal number and Amplitude are plotted in Figure 4.7a and Figure 4.7b respectively. The base level $C_T^{'}$ is kept at 2.

Interestingly in contrast to the findings of Munters and Meyers, in these simulations both the Strouhal number and amplitude do not seem to be robust with respect to turbine spacing. This is not a novel result as other studies into DIC have actually found a dependency between turbine spacing and Strouhal number like Frederik et al. but did not identify any specific relation between the two [27]. In addition, Frederik et al. performed their simulations with a fixed amplitude expressed in pitch angle of $4°$, while Figure 4.7 shows the results of a combined optimisation for frequency and amplitude. One could argue however that excluding $D = 3$ and $D = 4$ the results fall around the previously mentioned optimum range or plateau of $St = (0.2 - 0.3)$, $dCt = (1 - 2)$. The results at $D = 3$ and $D = 4$ might indicate that the turbine spacing in these cases is too short to allow DIC to trigger beneficial wake mixing. Further research into this topic is required before a conclusion can be drawn regarding the robustness of the Pulse with respect to turbine spacing.



(a) Strouhal number optima.

(b) Amplitude optima.

Figure 4.7: GA optimised pulse parameters for different turbine spacings.

4.3. Periodic full freedom

In addition to limiting the optimisation to sinusoidal signals, it is also interesting to test the capabilities of the combination of the free vortex wake model and GA optimisation in a more general setting. The results could provide more insight into how well the results obtained using the current implementation of the free vortex wake model correlate with findings based on other models like in the work of Munters and Meyers, and the possibilities provided by using GA optimisation with a simple engineering model.

Basic idea

The aim is to allow the GA to have significantly more freedom in finding a DIC input signal that would be beneficial to the two turbine setup. By broadening the search and expanding the parameters space that the GA has to optimise the required computational power can quickly become excessive. In order to limit this it is desirable to retain some periodicity in the input signal while giving the GA full freedom in between the start and end of this period.

This periodic full freedom signal can be framed in a GA optimisation problem where the genes encoded in the chromosomes represent the input signal at each time step i.e. $C_T'(t)$, with an equality constraint between the first and last gene in each chromosome to force periodicity. The amount of parameters in combination with Δt will determine the period of input signal, which should be sufficiently long to allow the optimisation to explore solutions while being short enough to allow a realistic computation time. The chromosomes in the optimisation will have the following form:

$$x^* = \begin{bmatrix} C_T'(1) & C_T'(2) & C_T'(3) & \dots & C_T'(n-1) & C_T'(1) \end{bmatrix} \tag{4.12}$$

with n being the period of the signal. For the optimisation problem to work there needs to be a set of constraints to keep the input signal within realistic bounds. Therefor constraints on the maximum and minimum $C_T'(t)$ are implemented. Additionally a constraint was added to keep the mean C_T' at 2 to prevent derating. The first simulations were run with these basic settings. But the method does allow easy addition of more detailed constraints to obtain a DIC input signal of desirable qualities.

As the amount of parameters in the optimisation has increased significantly compared to the earlier sinusoidal optimisation cases the population size in the GA was also increased to $P = 300$, with the number of generations initially being kept at $G = 50$. The simulation at $P = 300$ takes a considerable amount of time to run on the consumer grade hardware. Unfortunately the optimisation failed to converge to an optimum within the 50 generations. The resulting signal is shown in Figure 4.8.

The signal exhibits some of the same erratic characteristics that Munters and Meyers found in their initial LES simulations [16]. However without converging the signal is most likely not an optimum and thus no real conclusions can be drawn from this figure. Given the amount of optimisation parameters it is very likely that the population size needed is significantly larger than the $P = 300$ used. An attempt

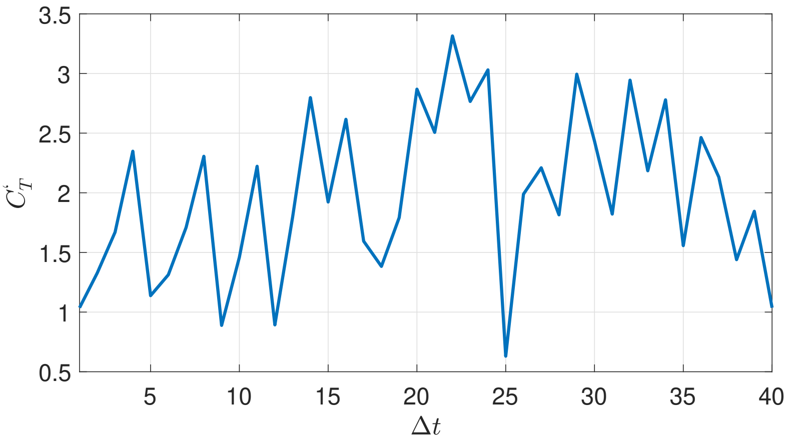


Figure 4.8: Periodic full freedom signal over time.

was made to increase the population size and the number of generations in order to facilitate more extensive exploration of the parameter space but this lead to simulations that were so slow it became unfeasible to run them on consumer hardware.

In further research it would be interesting to see if GA optimisation can be used with the FVWM in a setting like periodic full freedom optimisation on more powerful hardware as it would allow more extensive search into DIC beyond just sinusoidal signals.

4.4. Discussion

The results achieved with the current implementation of the FVWM and GA optimisation appear to be promising. The results for single sine optimisation yielding DIC signals that lie within the optimum plateau found by Munters and Meyers [16] add validity to the effectiveness of the FVWM in modelling wake mixing. The results from optimising a DIC signal with multiple harmonics appear to indicate that two superimposed sines provide better performance than single sine DIC signals. It is interesting that the effect of a lower and higher frequency harmonic was achieved by likely interference between to very close frequencies rather than two distinct frequencies. There might also be a well performing two sine DIC signal that does actually have two distinct frequencies but were slightly outperformed by the found optimum and thus did not surface during optimisation. Again further research is needed to confirm found results preferably in higher fidelity simulations. The results from gradient based optimisation by van den Broek et. al. [11] indicate that the optimum DIC signal for their FVWM implementation is very similar to a single sine, not observing any additional harmonics. However as van den Broek et. al. noted gradient based solvers are more susceptible to local optima, which could explain the difference.

The fact that the FVWM with genetic algorithm optimisation can run on consumer hardware with a mind range processor like the AMD Ryzen R5 3600 used for this thesis illustrates the efficiency of the model. However the results remain to be confirmed by higher fidelity simulation, and thus it would be overzealous to draw hard conclusions. The FVWM requires many simplifications, assumptions and constraints in order to work within the implemented framework. One example is static vortex strength that is used in the current implementation, or the constant core size. The advantage of the FVWM with GA optimisation is that it allows faster exploration of DIC signals and ideally the results should be taken as pointers to investigate specific signals in more detail by means of higher fidelity simulation of experimental testing. The developed framework also enables future research to include more specific constraints and cost function components to add competing objectives. In the current implementation only power production is taken into account for the fitness in the GA, however one could imagine for example that applying DIC also has mechanical downsides such as added stress on the turbine structure which could be taken into account by means of a term in the fitness function, for example penalising higher frequencies and amplitudes.

With more powerful hardware more free optimisation such as the periodic full freedom attempted in this thesis could be achieved. This would allow more insight into wheter sinusoidal signals are in fact optimal or there is another distinct signal shape that provides additional benefit. Again this is a point for future research.

5 Conclusion

The goal of this thesis was to investigate dynamic induction control beyond the currently found sinusoidal signals in the work of Munters and Meyers [16] using free vortex wake methods. To this end the free vortex wake model was implemented in MATLAB to allow a less computationally demanding optimisation that would in turn allow a much greater parameter space to be searched. This enabled the investigation into optimisation of DIC signals with multiple harmonics and the effects on the performance of a wind farm. The implemented FVWM was tuned by means of a convergence study to determine the model parameters before optimisation was attempted in order to have confidence in the precision of the model. For optimisation the genetic algorithm method was chosen as it is suitable to the problem at hand, namely the efficient investigation of large parameter spaces while being less affected by local optima than gradient based optimisation methods.

The characteristic frequency found to trigger wake mixing in the work of Munters and Meyers [16] and more recently corroborated by van den Broek et. al. [11] is around a non dimensional Strouhal number of $St = 0.25$. From the results of Munters and Meyers a plateau of well performing DIC signals appears to lie between $St = 0.2 - 0.3$ and $dC_T^{'} = 1 - 2$. Investigation of single sine DIC using the FVWM framework developed in this thesis resulted in an optimal frequency of $St = 0.28$, further adding strength to the hypothesis that the optimal DIC frequency is around this $St = 0.25$ number. In terms of amplitude again the found optimum of $dC_T^{'} = 1.98$ lies within the suggested plateau.

While first observation of the results might suggest otherwise, the investigation of the robustness of the pulse with respect to turbine spacing appears to confirm that while for different spacing the optimum DIC signal might vary slightly, the signals are robust within the well performing plateau. But further research is required on this topic to confirm this as the results are too inconclusive at this time.

Optimisation of multiple harmonics has yielded DIC signals concisting of two superimposed sines that offer an up to 10 percent point boost in performance over traditional single sine DIC when compared to greedy control in the current FVWM implementation. It was also found that the addition of more than two sines does not provide any additional benefit. While these results need to be confirmed in higher fidelity simulations, given that the results from the single sine optimisation were in line with previous high fidelity work the result seems promising.

The developed optimisation framework consisting of an implementation of the FVWM and GA optimisation could prove to be a useful tool that allows promising DIC signals to be found relatively efficiently before confirming them with higher fidelity simulations.

References

[1] Richard S. Eckaus. “Comparing the Effects of Greenhouse Gas Emissions on Global Warming”. In: *The Energy Journal* 13.1 (Jan. 1992). ISSN: 01956574. DOI: 10.5547/issn0195-6574-ej-vol13-no1-2. URL: www.iaee.org/energyjournal/article/1056.

[2] Vaclav Smil. “Energy Transitions: Global and National Perspectives (Expanded and updated edition)”. In: 30.4 (2017), p. 297. URL: http://vaclavsmil.com/2016/12/14/energy-transitions-global-and-national-perspectives-second-expanded-and-updated-edition/.

[3] *Renewable energy transition in 5 charts – DW – 11/08/2022*. URL: https://www.dw.com/en/renewable-energy-transition-graphs/a-63674250.

[4] European Commision. *Onshore and offshore wind | Energy*. Nov. 2021. URL: https://ec.europa.eu/energy/topics/renewable-energy/onshore-and-offshore-wind_en.

[5] *Joint European Action for more affordable, secure and sustainable energy*. Tech. rep. European Commision, 2022. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52022DC0108.

[6] WindEurope. *Wind energy in Europe; 2022 Statistics and the outlook for 2023-2027*. Tech. rep. 2023. URL: https://windeurope.org/intelligence-platform/product/wind-energy-in-europe-2022-statistics-and-the-outlook-for-2023-2027/#presentations.

[7] Michiel Zaayer and Axelle Vire. *Introduction to wind turbines: physics and technology*. 2018.

[8] Daniel R. Houck. “Review of wake management techniques for wind turbines”. In: *Wind Energy* May (2021), pp. 1–26. ISSN: 1095-4244. DOI: 10.1002/we.2668.

[9] M. Stoelinga. *A Multi-Project Validation Study of Vaisala’s Wake Loss Estimation Method*. Tech. rep. 2019. URL: https://www.vaisala.com/sites/default/files/documents/WEA-WNA-WL-ERG-EN-US-PUB-Validation%20study%20of%20wake%20loss%20estimation%20method-2019-09-%20B211843EN-A.pdf.

[10] Wim Munters and Johan Meyers. “Dynamic strategies for yaw and induction control of wind farms based on large-eddy simulation and optimization”. In: *Energies* 11.1 (Jan. 2018), p. 177. ISSN: 19961073. DOI: 10.3390/en11010177. URL: https://www.mdpi.com/1996-1073/11/1/177/htm%20https://www.mdpi.com/1996-1073/11/1/177.

[11] Maarten J. van den Broek et al. “Adjoint optimisation for wind farm flow control with a free-vortex wake model”. In: *Renewable Energy* 201 (Dec. 2022), pp. 752–765. ISSN: 0960-1481. DOI: 10.1016/J.RENENE.2022.10.120.

[12] Andrew Kusiak and Zhe Song. “Design of wind farm layout for maximum wind energy capture”. In: *Renewable Energy* 35.3 (Mar. 2010), pp. 685–694. ISSN: 0960-1481. DOI: 10.1016/J.RENENE.2009.08.019.

[13] Stoyan Kanev, Edwin Bot, and Jack Giles. “Wind Farm Loads under Wake Redirection Control”. In: *Energies 2020, Vol. 13, Page 4088* 13.16 (Aug. 2020), p. 4088. ISSN: 19961073. DOI: 10.3390/EN13164088. URL: https://www.mdpi.com/1996-1073/13/16/4088/htm%20https://www.mdpi.com/1996-1073/13/16/4088.

[14] Wim Munters and Johan Meyers. “Effect of wind turbine response time on optimal dynamic induction control of wind farms”. In: *Journal of Physics: Conference Series* 753.5 (Sept. 2016), p. 052007. ISSN: 17426596. DOI: 10.1088/1742-6596/753/5/052007. URL: https://iopscience.iop.org/article/10.1088/1742-6596/753/5/052007%20https://iopscience.iop.org/article/10.1088/1742-6596/753/5/052007/meta.

[15] Jennifer Annoni et al. “Analysis of axial-induction-based wind plant control using an engineering and a high-order wind plant model”. In: *Wind Energy* 19.6 (June 2016), pp. 1135–1150. ISSN: 1099-1824. DOI: 10.1002/WE.1891. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/we.1891%20https://onlinelibrary.wiley.com/doi/abs/10.1002/we.1891%20https://onlinelibrary.wiley.com/doi/10.1002/we.1891.

[16] Wim Munters and Johan Meyers. “Towards practical dynamic induction control of wind farms: Analysis of optimally controlled wind-farm boundary layers and sinusoidal induction control of first-row turbines”. In: *Wind Energy Science* 3.1 (2018), pp. 409–425. ISSN: 23667451. DOI: 10.5194/wes-3-409-2018.

[17] D. Medici et al. “Measurements behind model wind turbines: further evidence of wake meandering”. In: *WiEn* 11.2 (Mar. 2008), pp. 211–217. ISSN: 10954244. DOI: 10.1002/WE.247. URL: https://ui.adsabs.harvard.edu/abs/2008WiEn...11..211M/abstract.

[18] N O Jensen. “A note on wind generator interaction”. In: *Risø-M-2411 Risø National Laboratory Roskilde* (1983), pp. 1–16. ISSN: 01676105. URL: http://www.risoe.dk/rispubl/VEA/veapdf/ris-m-2411.pdf.

[19] P. M.O. Gebraad, P. A. Fleming, and J. W. Van Wingerden. “Wind turbine wake estimation and control using FLORIDyn, a control-oriented dynamic wind plant model”. In: *Proceedings of the American Control Conference*. Vol. 2015-July. Institute of Electrical and Electronics Engineers Inc., July 2015, pp. 1702–1708. ISBN: 9781479986842. DOI: 10.1109/ACC.2015.7170978.

[20] Michael F. Howland et al. “Wake structure in actuator disk models of wind turbines in yaw under uniform inflow conditions”. In: *Journal of Renewable and Sustainable Energy* 8.4 (July 2016), p. 043301. ISSN: 19417012. DOI: 10.1063/1.4955091. URL: https://aip-scitation-org.tudelft.idm.oclc.org/doi/abs/10.1063/1.4955091.

[21] Shreyas Ananthan and J. Gordon Leishman. “Role of filament strain in the free-vortex modeling of rotor wakes”. In: *Journal of the American Helicopter Society* 49.2 (Apr. 2004), pp. 176–191. ISSN: 00028711. DOI: 10.4050/JAHS.49.176. URL: https://www.researchgate.net/publication/245449526_Role_of_Filament_Strain_in_the_Free-Vortex_Modeling_of_Rotor_Wakes.

[22] Niels Troldborg, Mac Gaunaa, and Srinivas Guntur. “Modelling the influence of yaw using a simple vortex rotor model”. In: *European Wind Energy Conference and Exhibition 2012, EWEC 2012*. Vol. 2. 2012, pp. 1107–1111. ISBN: 9781627482912.

[23] Maarten J Van Den Broek et al. *Optimal Control for Wind Turbine Wake Mixing on Floating Platforms*. Tech. rep. URL: https://arxiv.org/abs/2210.17347v2.

[24] A E Eiben and J E Smith. *Natural Computing Series Introduction to Evolutionary Computing*. ISBN: 978-3-662-44873-1. DOI: 10.1007/978-3-662-44874-8.

[25] CHAPMAN & HALL/CRC. *The Practical Handbook of Genetic Algorithms*. 2001. ISBN: 1584882409. URL: https://doc.lagout.org/science/0_Computer%20Science/2_Algorithms/The%20Practical%20Handbook%20of%20Genetic%20Algorithms_%20Applications%20%282nd%20ed.%29%20%5BChambers%202000-12-07%5D.pdf.

[26] A.R. Parkinson, R. Balling, and J.D. Hedengren. “Real-Valued Genetic Algorithms”. In: *Optimisation Methods for Engineering Design*. URL: https://apmonitor.com/me575/uploads/Main/optimization_book.pdf.

[27] Joeri A. Frederik et al. “The helix approach: Using dynamic individual pitch control to enhance wake mixing in wind farms”. In: *Wind Energy* 23.8 (Aug. 2020), pp. 1739–1751. ISSN: 10991824. DOI: 10.1002/we.2513.

[28] Annette Grathoff. “On the explicit function of life within a physical universe”. In: *Philosophies* 6.3 (Sept. 2021). ISSN: 24099287. DOI: 10.3390/philosophies6030059.