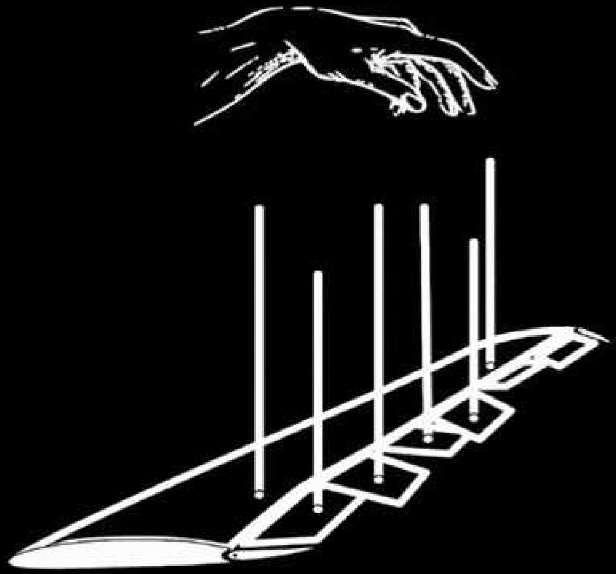
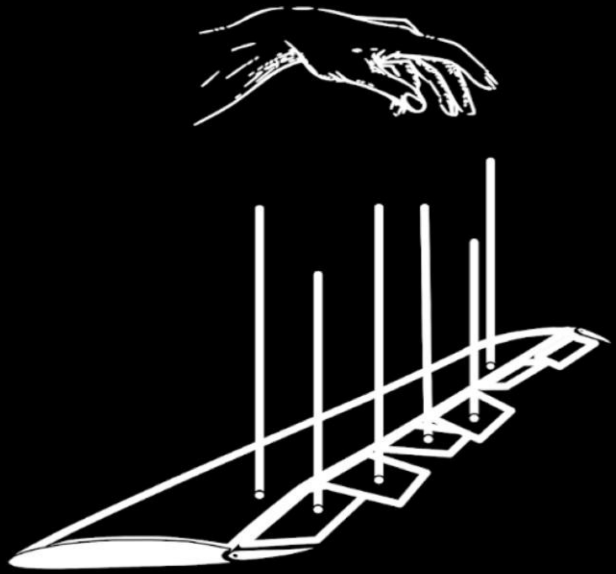
Yerevan, GIF FAST 2019 16 October 2019



TUDelft

Yerevan, GIF FAST 2019 16 October 2019

Adaptive state estimation and Real-Time tracking for aircraft control with ML and AI



Tigran Mkhoyan

TUDelft

Elegance and efficiency of birds

TU Delft

Elegance and efficiency of birds

TU Delft

Aeroelastic Structures

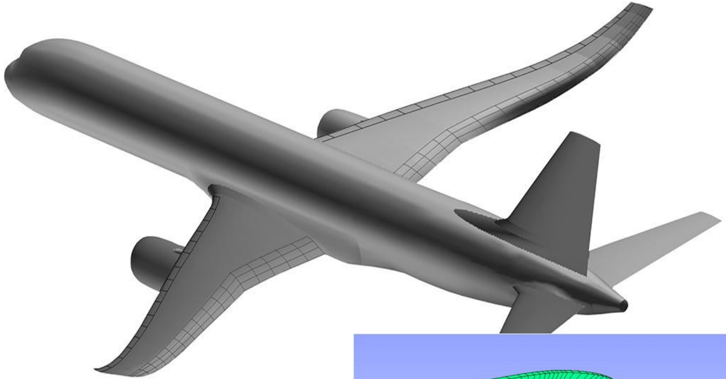
Introduction

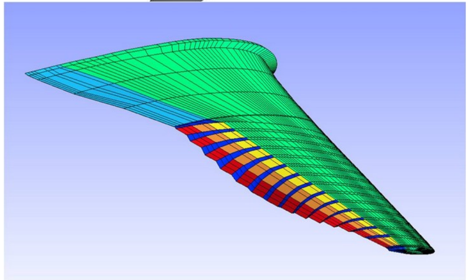
Trend towards flexible configurations:

Adaptive Compliant Trailing Edge (ACTE)

Variable camber continuous trailing edge flap flap (VCCTF)

(source: NASA/FlexSys)

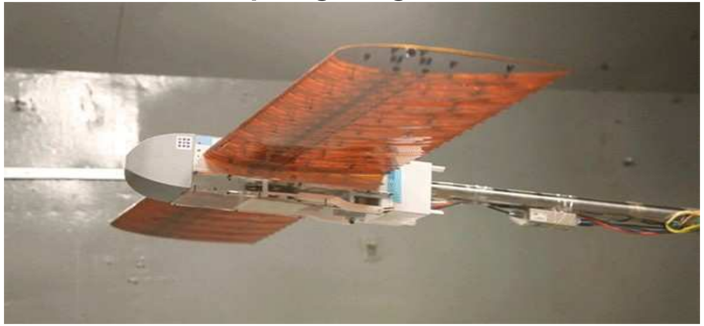




(source: NASA Ames)

Applications: slender flexible (morphing) aircraft

Cellular morphing wing



(source: NASA/MIT)

HALE solar power aircraft

(source: NASA)

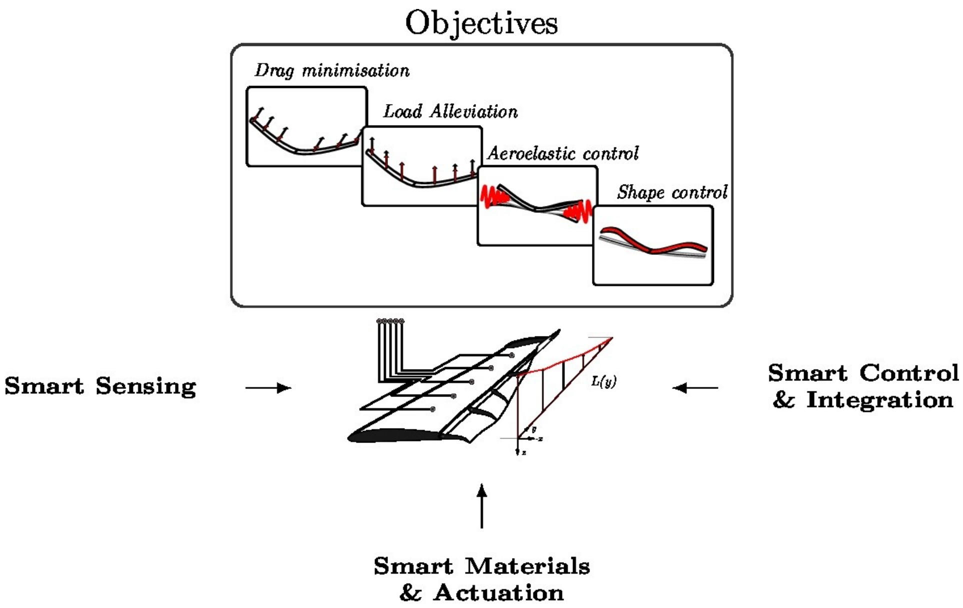
Facebook drone aquila

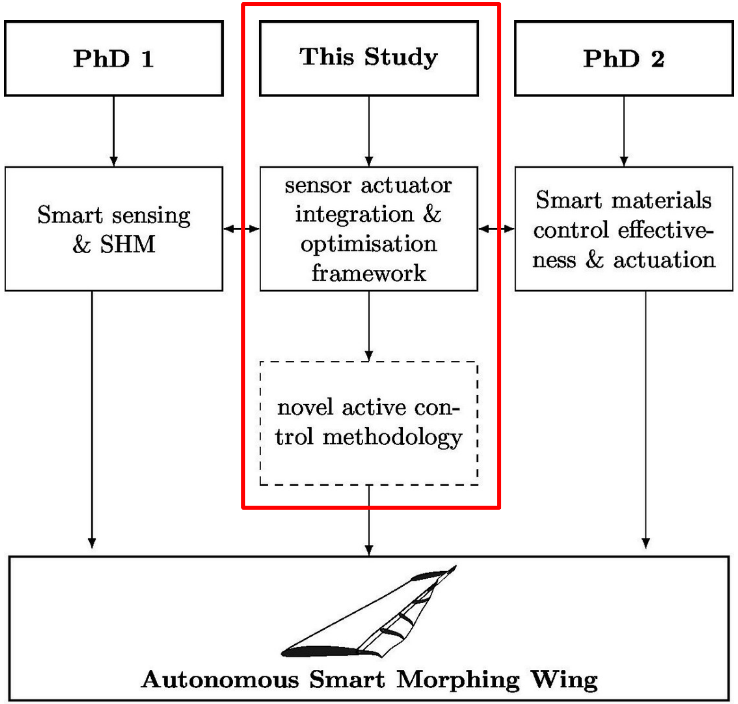
(source: Facebook)

TU Delft

Goal: the Smart Morphing Wing

integration of <u>novel control laws, sensing methods, and actuation mechanism</u>
for real-time, in-flight, multi-objective optimisation of actively morphing wing

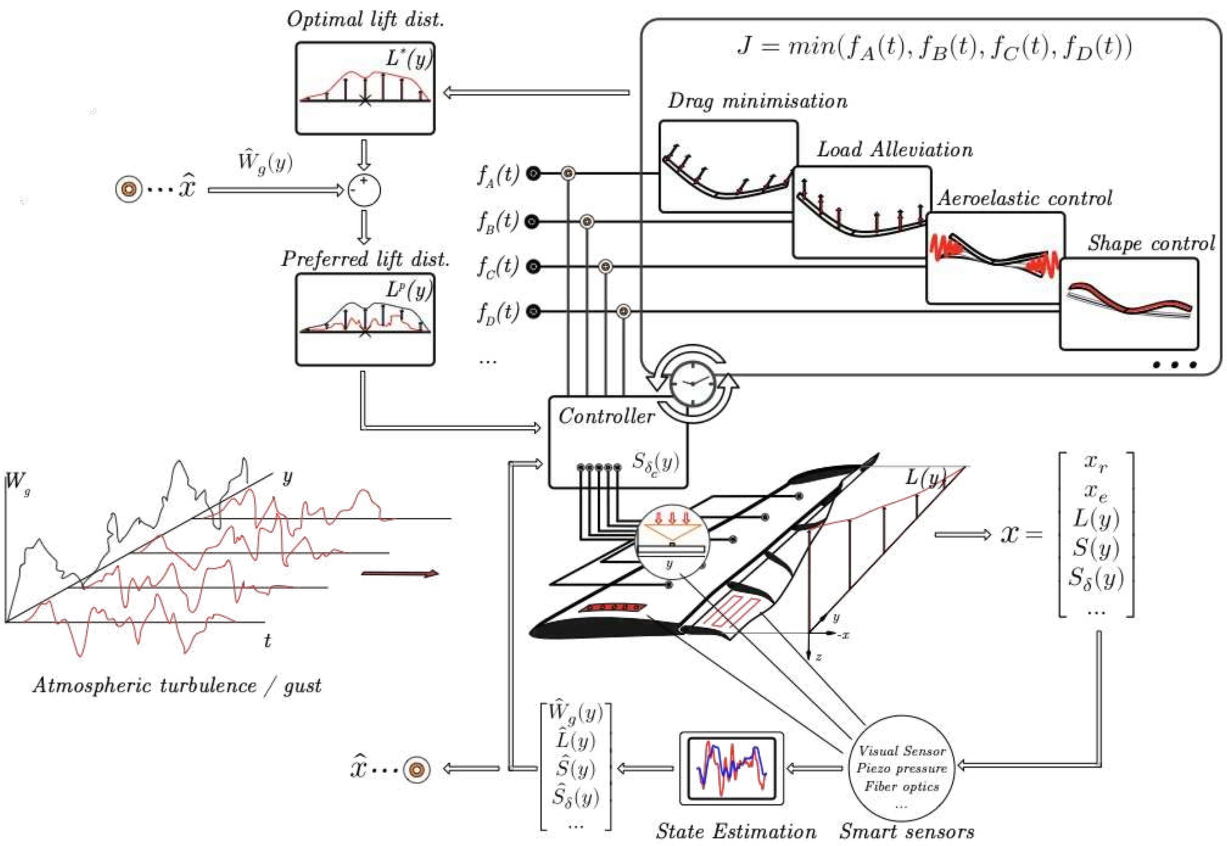




TU Delft

Smart-X

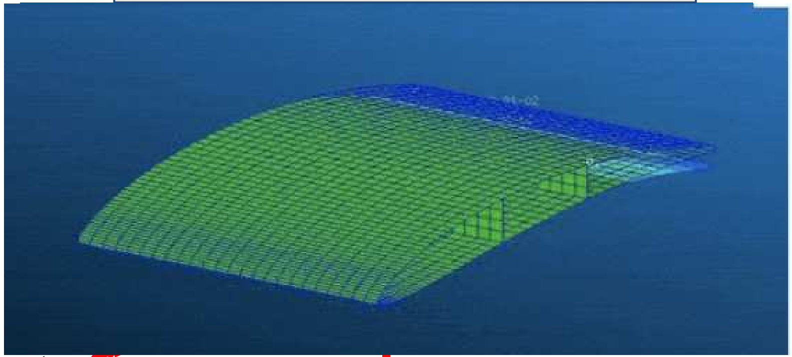
Real-time multi-objective performance optimisation



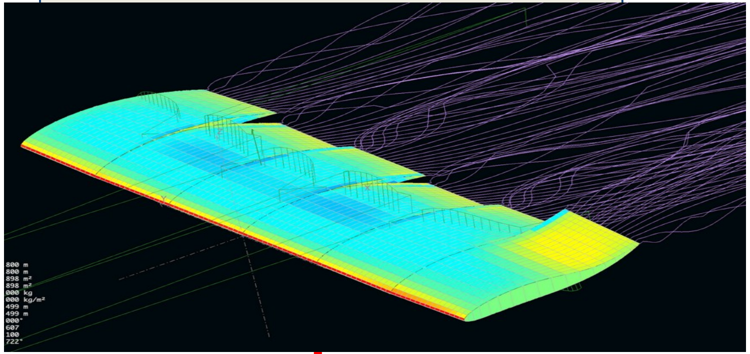
Smart-X design approach

Patented Translation Induced Camber (TRIC) concept

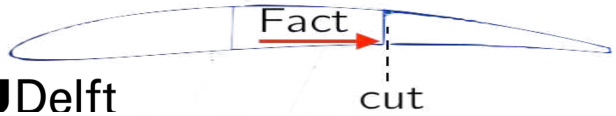
FEM Model

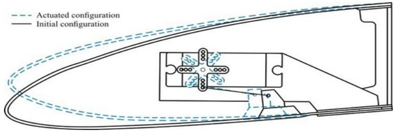


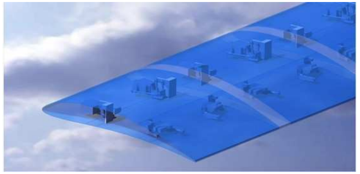
CFD Model



$F_{actuator}$







TU Delft

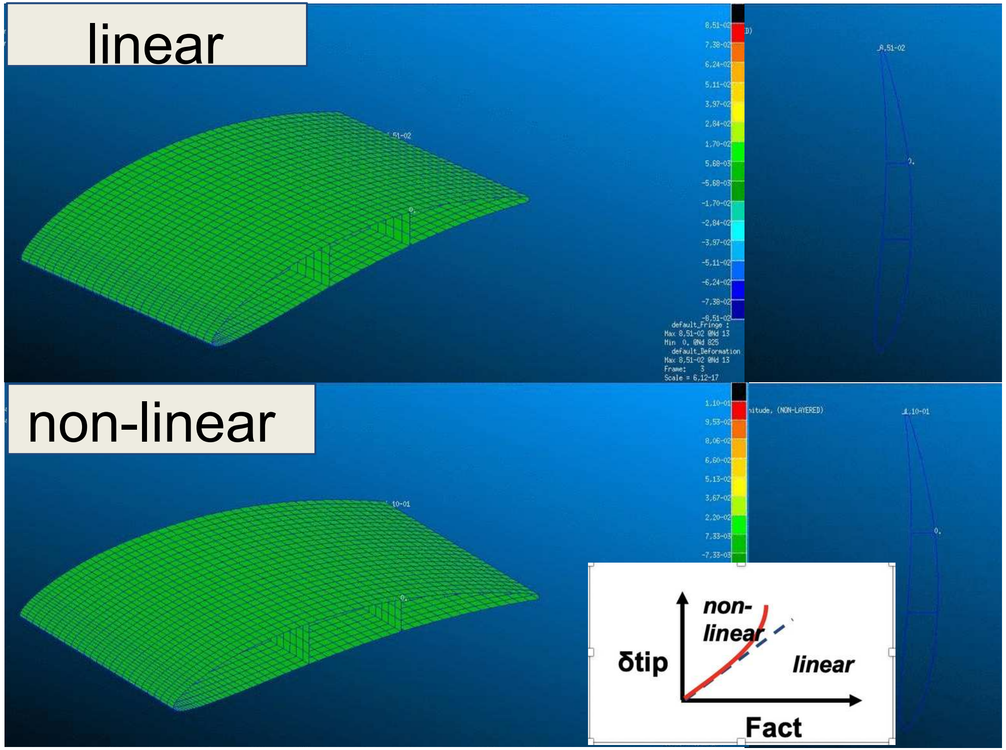
Challenges Morphing Structures

Challenges control design morphing: non-linearity

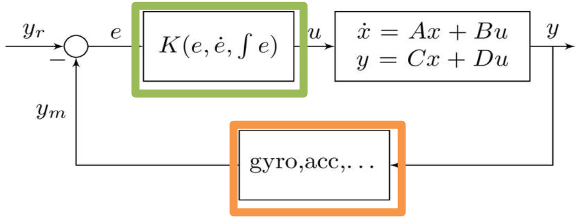
actuator force non-linear control effectiveness mapping:

sensor

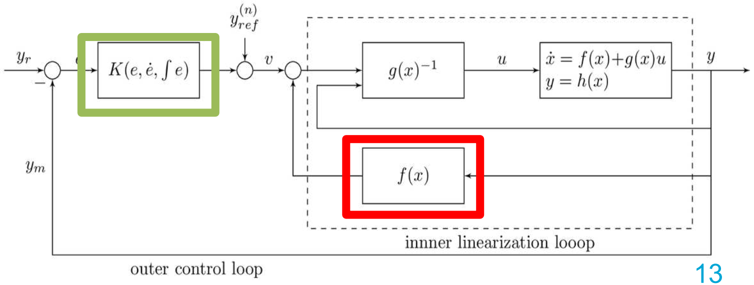
PID controller



Typical: Linear + PID



non-linear dynamics (act + model)



Gaps in Literature

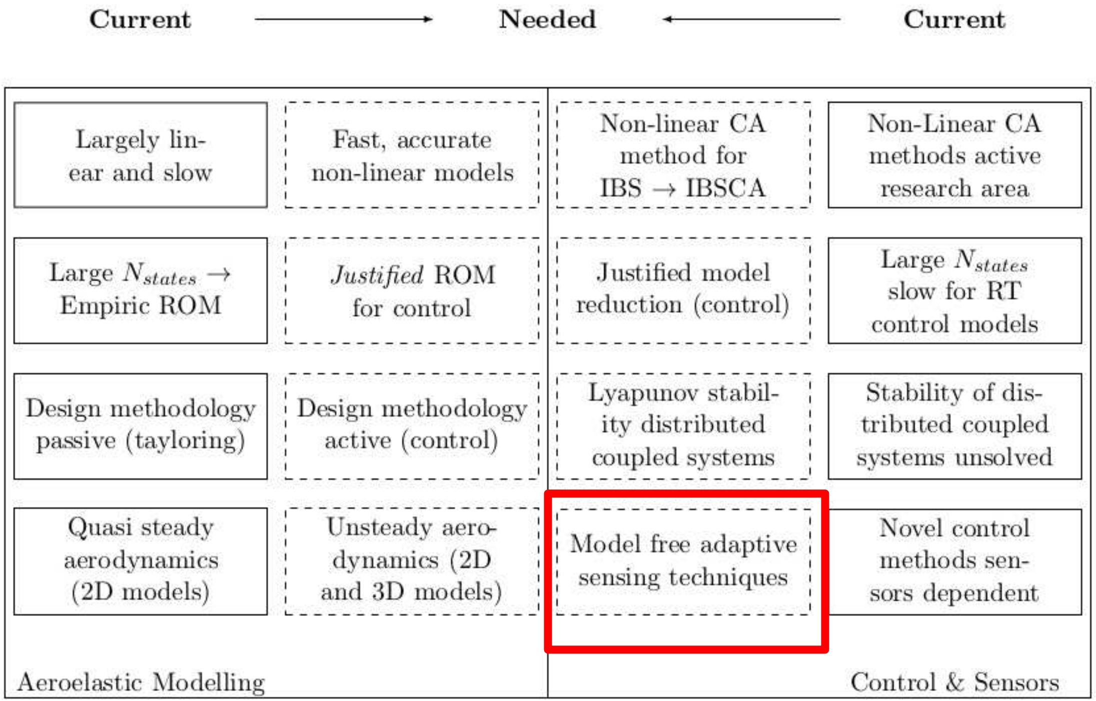
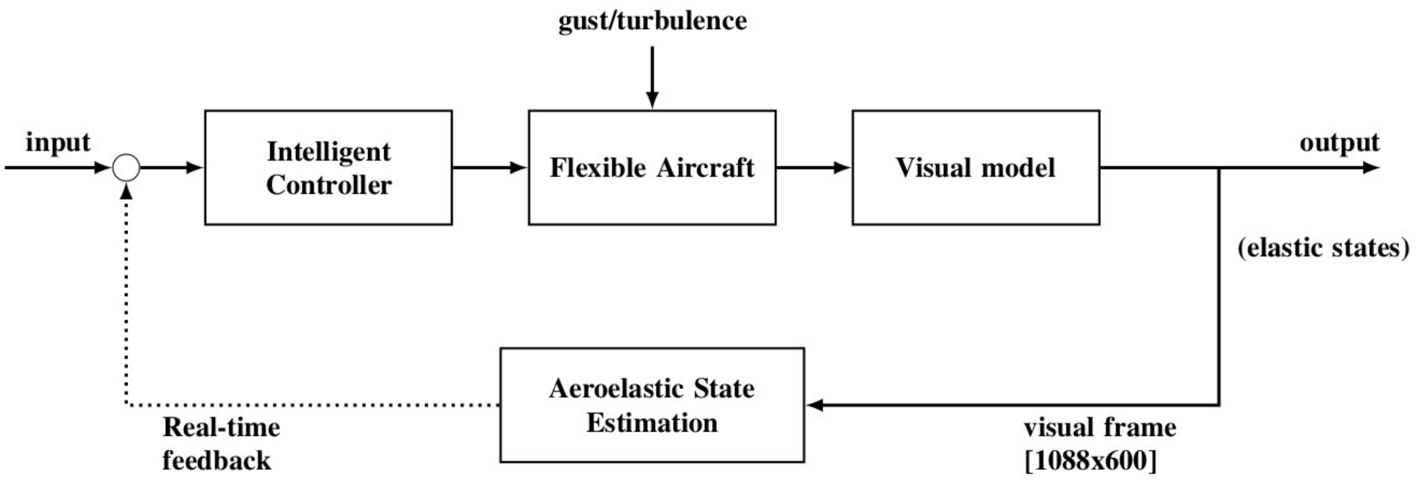


Figure 2: Gaps in Literature of Smart morphing wings

A Simplified Control diagram Visual Tracking



Question: *Can we provide aeroelastic feedback with alternative sensors for Real-time control?*

Purpose: *Investigate how to eliminate dependency on both model (f(x)) as control effectiveness (g(x))*

TU Delft

ML / Traditional CV

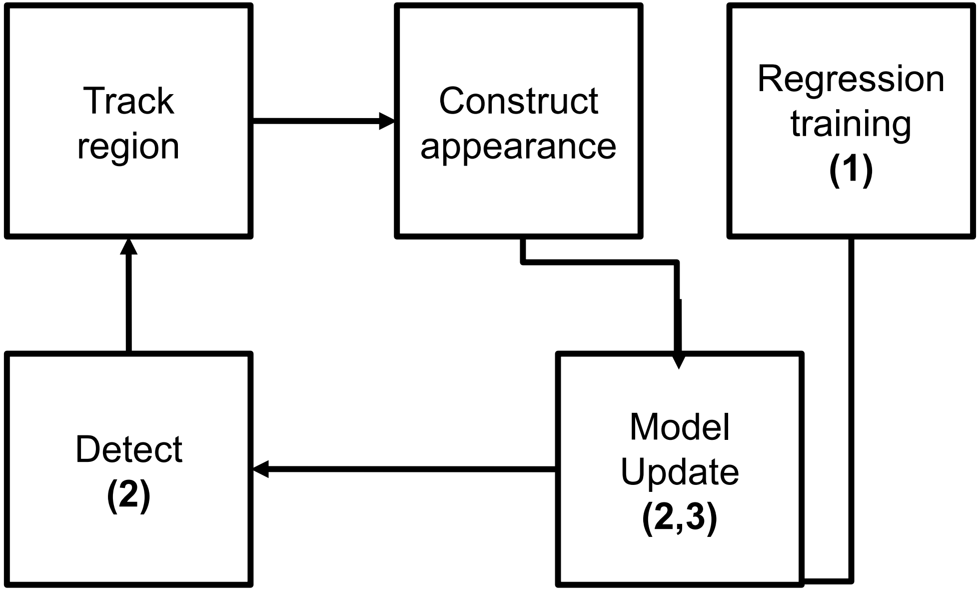
TU Delft

Visual tracking KCF-Kalman couple

- Novel method for monitoring wing displacements and loads real-time with simple camera feed (e.g. mounted in the fuselage)
- Combines speed of KCF (Kernelized Correlation filter) with robustness and prediction of the KF (Kalman Filter)

KCF: purely visual filter

Correlation filter with linear ridge regression based on circulant matrix properties and kernel functions



1: Training Kernel function:

$$\hat{\alpha} = \frac{\hat{z}}{\hat{k}^{aa} + \lambda}$$

2: Regression function:

$$\hat{f}(b) = \hat{k}^{ab} \odot \hat{\alpha}$$

3,4: Regression weight update lin. interpolation

$$\hat{\alpha} = ((1 - interpf) \times \hat{\alpha}) + (interpf \times \hat{\alpha}_{new})$$

$$a = ((1 - interpf) \times a) + (interpf \times a_{new})$$

[1] J. F. Henriques et. al. (2015) “High-Speed Tracking with Kernelized Correlation Filters.” IEEE Transactions on Pattern Analysis and Machine Intelligence 37(3): 583-596.

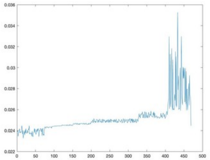
Kalman Filter: Adding Dynamics to visual motion:

$$x(t+h) = x(t) + f'(x(t))h$$

Kalman Filter (KF): Predicting linear motion

$$x_k = x_{k-1} + \dot{x}_{k-1}h$$
$$y_k = y_{k-1} + \dot{x}_{k-1}h$$
$$\dot{y}_k = \dot{y}_{k-1} + \ddot{y}_{k-1}h$$

Extended KF (EKF): Non-linear motion, non-uniform timestep



The general differential equation is given as:

$$\ddot{y}(t) = -\frac{c}{m}\dot{y}(t) - \frac{k}{m}y(t)$$

In state space form we have:

$$\frac{d}{dt}\begin{bmatrix} y_k \\ \dot{y}_k \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k/m & -c/m \end{bmatrix}\begin{bmatrix} y_k \\ \dot{y}_k \end{bmatrix}$$

Augmented (AEKF): Non-linear motion, time-varying, learn unknown dynamics

$$\ddot{y}(t) = -\frac{c(t)}{m(t)}\dot{y}(t) - \frac{k(t)}{m(t)}y(t)$$

$$\bar{x}_k = \begin{bmatrix} y_k \\ \dot{y}_k \\ K_k \\ c_k \\ m_k \end{bmatrix} = \begin{bmatrix} y_{k-1} + \dot{x}_{k-1}h \\ -K_{k-1}/m_{k-1} \cdot y_{k-1} - (1 - c_{k-1}/m_{k-1}h) \cdot \dot{y}_{k-1} \\ K_{k-1} + 0 \cdot h \\ c_{k-1} + 0 \cdot h \\ m_{k-1} + 0 \cdot h \end{bmatrix} \quad (13)$$

→

$$J(\bar{x}_k) = \begin{matrix} 1 & h & 0 & 0 & 0 \\ -K_{k-1} \cdot m_{k-1}^{-1} \cdot h & 1 - c_{k-1} \cdot m_{k-1}^{-1} \cdot h & m_{k-1}^{-1} \cdot y_{k-1} \cdot h & -m_{k-1}^{-1} \cdot \dot{y}_{k-1} \cdot h & m_{k-1}^{-2} \cdot c_{k-1} \cdot \dot{y}_{k-1} \cdot h - m_{k-1}^{-2} \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{matrix}$$

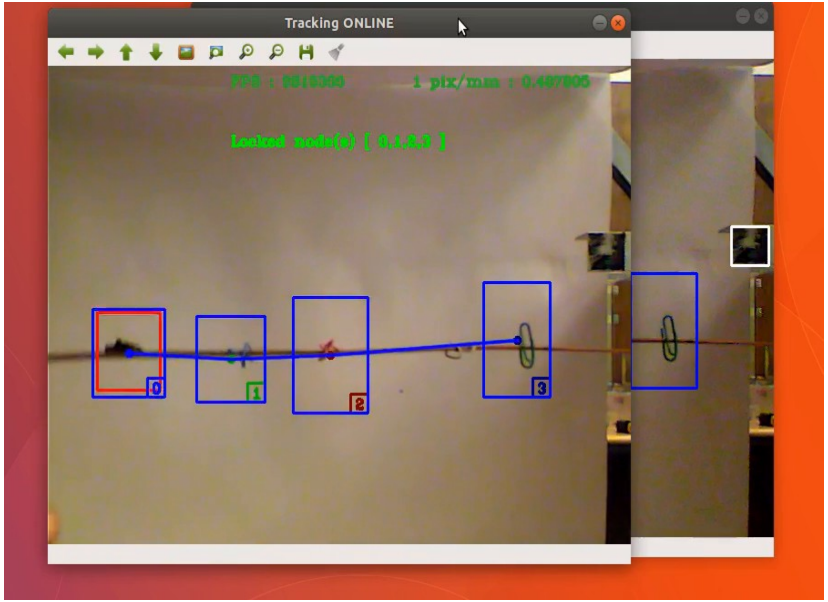
TU Delft

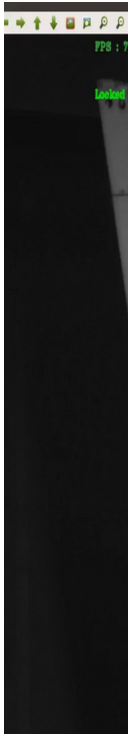
Smart State Estimation
What can you do with it?

TUDelft

Visual tracking KCF-Kalman couple

Examples: Clamped beam (right) and real wing (left)

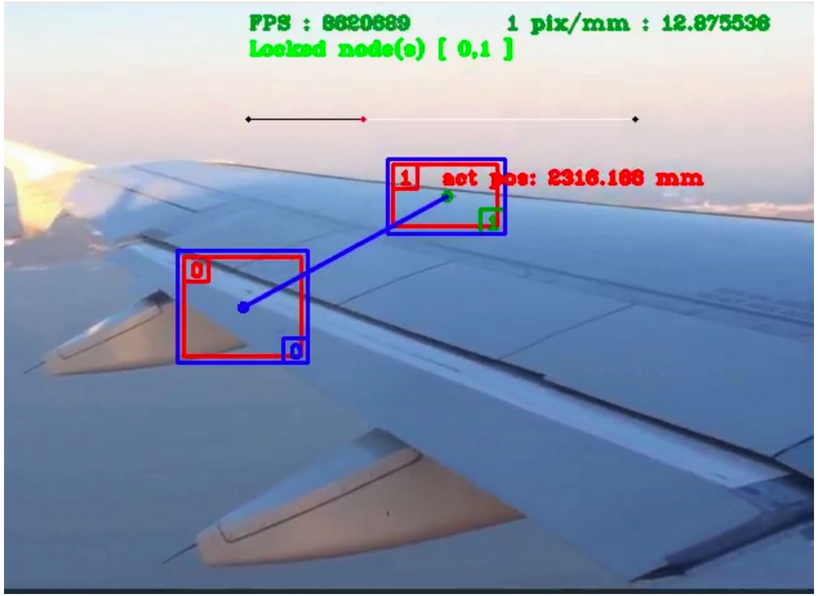




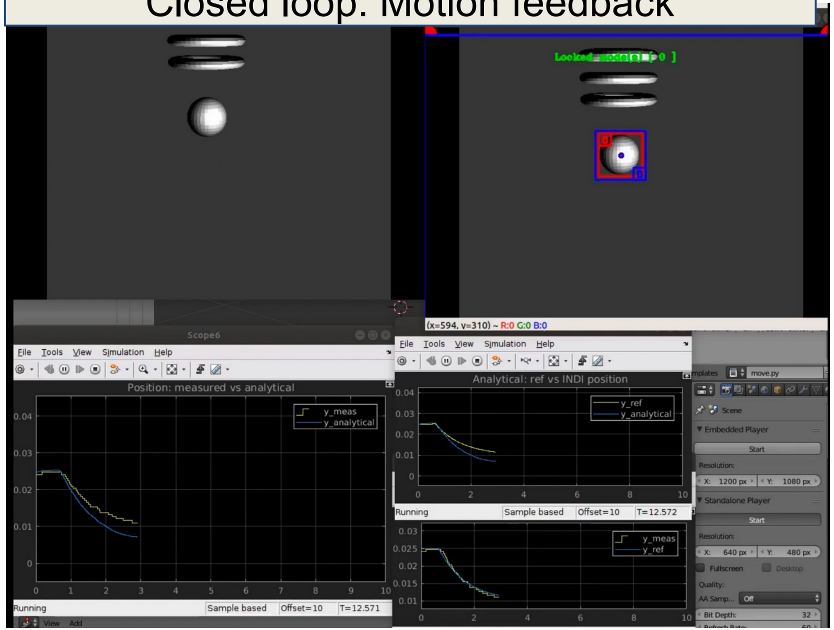
Visual tracking KCF-Kalman couple

Examples: actuator position feedback (left) and INDI with visual sensor feedback (right)

Freehand: Tracking actuator motion



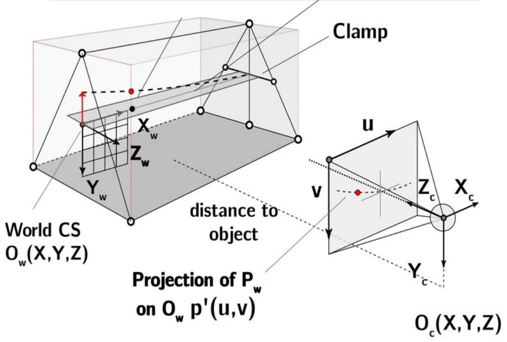
Closed loop: Motion feedback



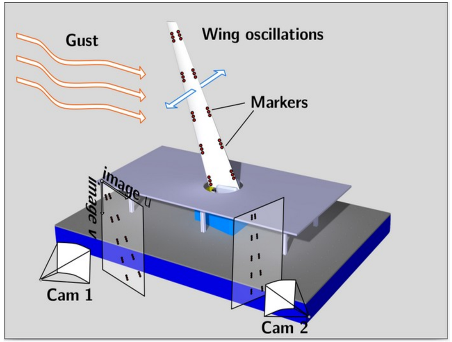
![TU Delft logo]

Experiments performed

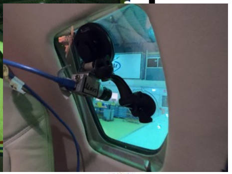
Single camera clamped beam:

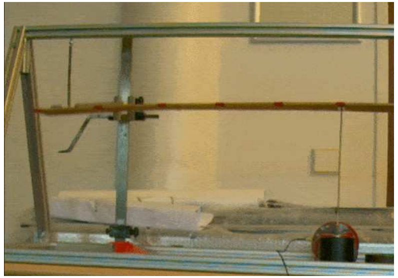


WT test stereo setup:



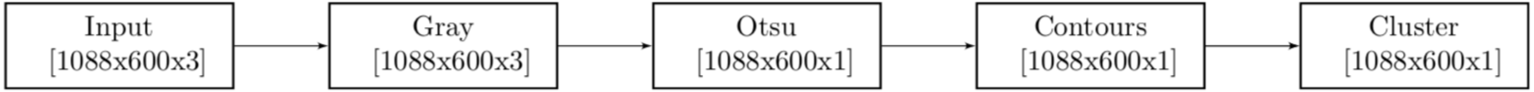
FT Cessna Citation

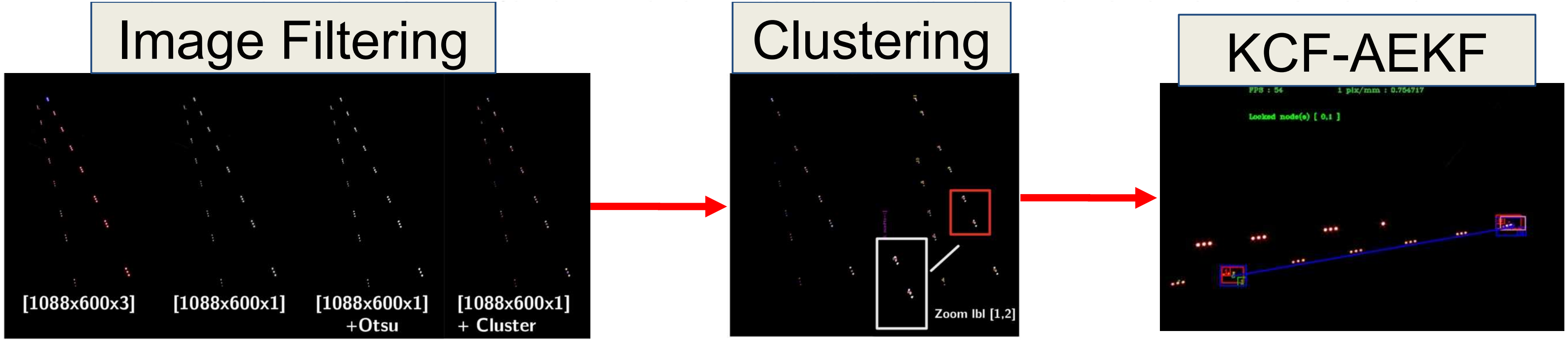




ML methodology: multistep filtering

- Adaptive Image filtering (Otsu, Parallel color-filter)
- Clustering with DBSCAN (Density Based Clustering)
- Approximate geometry FLANN (Fast Lib Approximate Nearest Neighbors)
- KCF+AEKF for tracking and learning system parameters

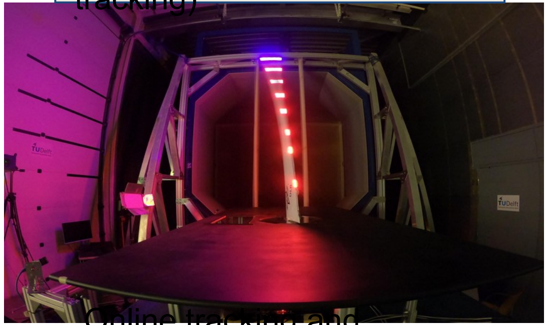




Adaptive Visual Tracking

adding dynamics (EKF) and adaptation (Parameter estimation)

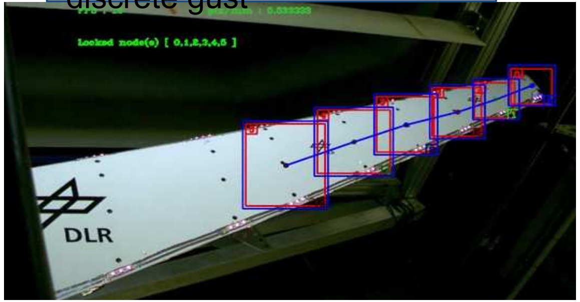
Experiment setup (low vis. tracking)



Machine learning auto labelling (Dbscan and partitioning)

Online tracking and parameter estimation with

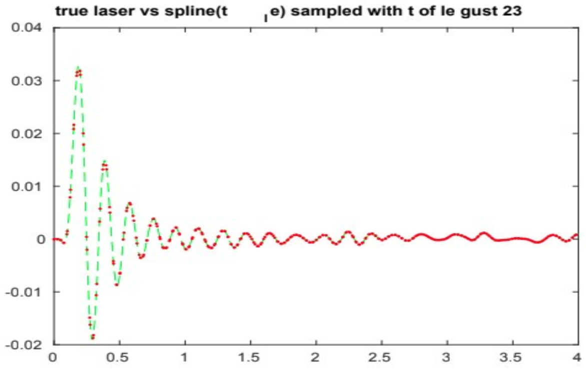
Tracking aeroelastic wing discrete gust



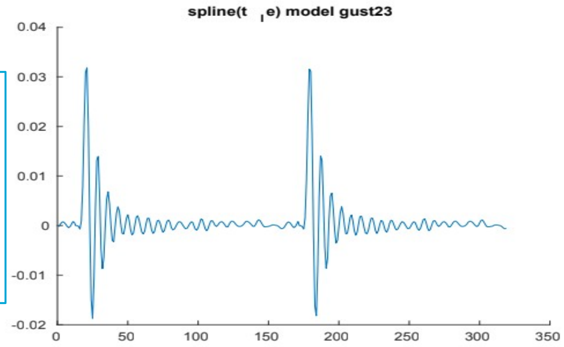
TU Delft

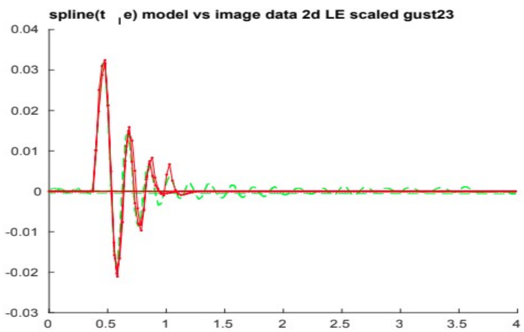
Adaptive Visual Tracking

Laser data sampled at non-uniform time step of camera:

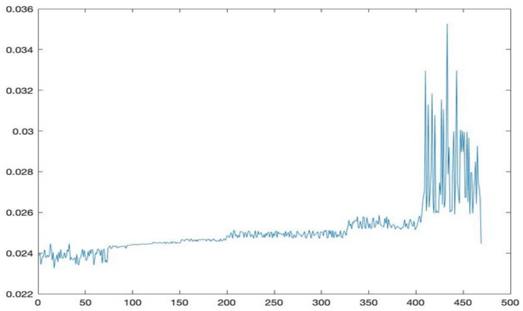


laser: 400Hz
Cam: +- 41hz





acc: 1-8mm



Research Vision: Visual tracking

Advantages:

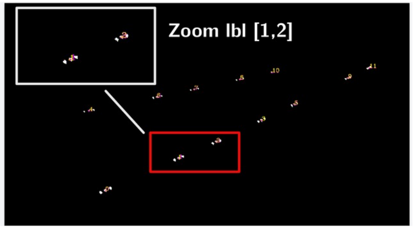
- No need for accelerometers (certification) just a camera
- Robust against lightning conditions and signal loss
- Tracker parameter interpreted as system state → suitable for distributed systems

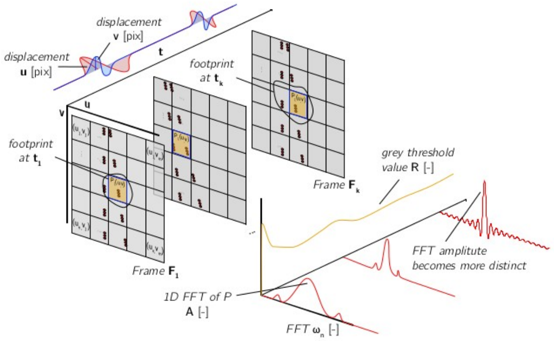
Limitations

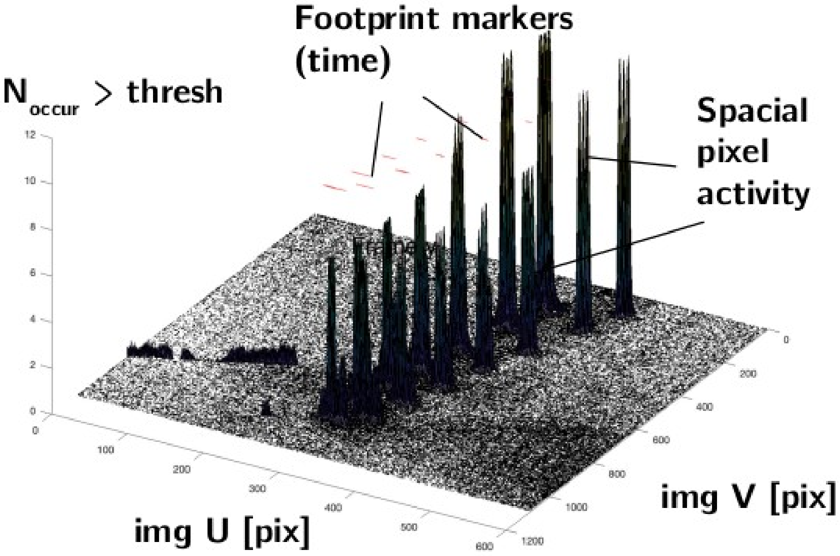
- Camera dependency (update rate)
- more camera views to get x,y,z coords

Establishing marker footprint across a sequence

- active pixel areas can be determined by SDFT (Sliding Discrete Fourier Transform)





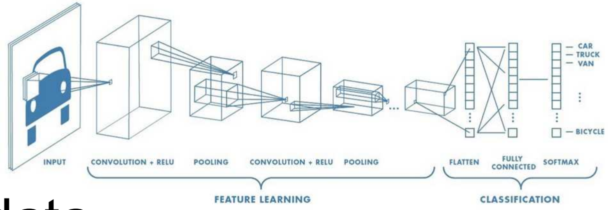


TU Delft

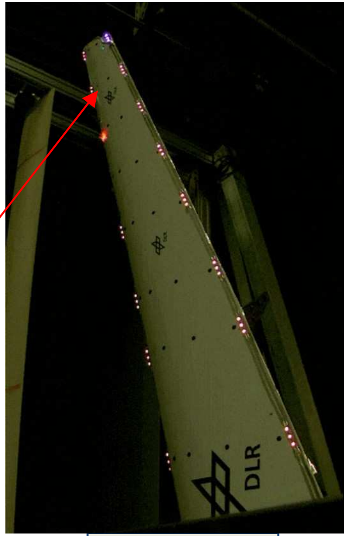
AI / Deep Learning

AI approach: Learning from raw data

- Loose markers use raw images
- Use DCNN (Deep Convolutional neural network) and learn to predict from data
- Large spline dataset with laser measurement as annotations

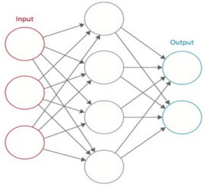


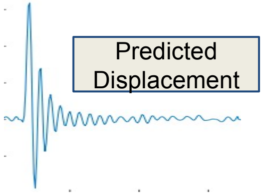
(source: towardsdatascience)



Input images

CNN/Resnet





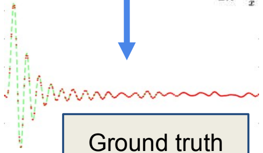
Predicted Displacement

Error metric

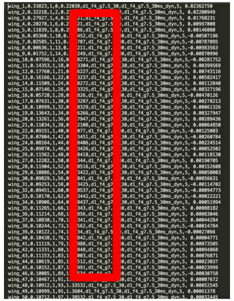
$$E = \frac{1}{2n}\sum_x \|(y(x) - y'(x))\|^2$$

Update weights

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} = -\eta o_i \delta_j$$

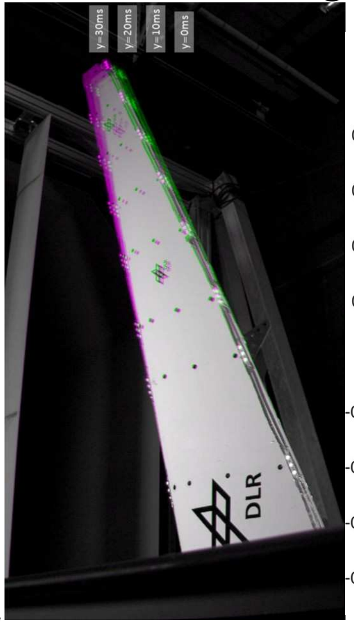


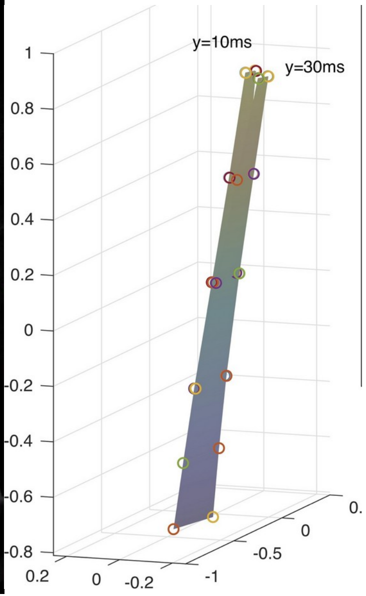
Ground truth



AI with DCNN approach: pretrained ResNet

- Dataset 15000+ images, 12 experimental conditions
- laser measurement (400 Hz) sampled non-uniform step +-41Hz, + geo-laser (static baseline) from 12 experimental

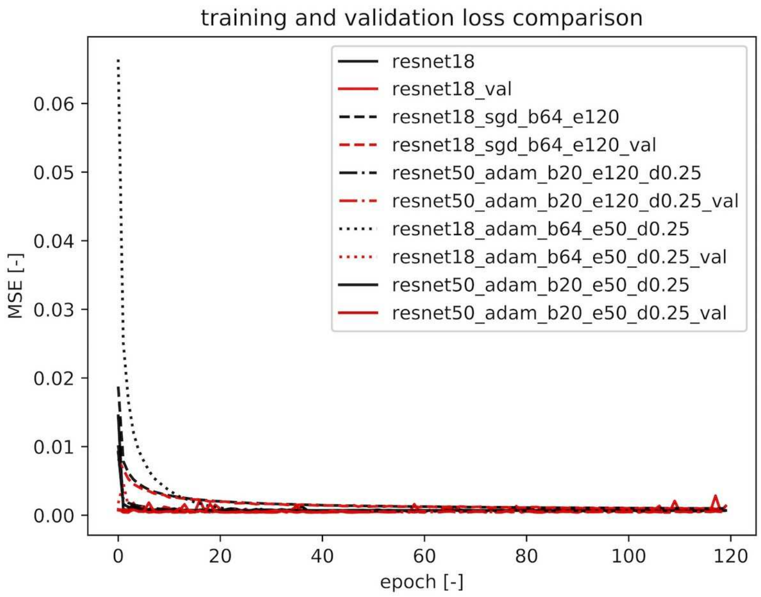


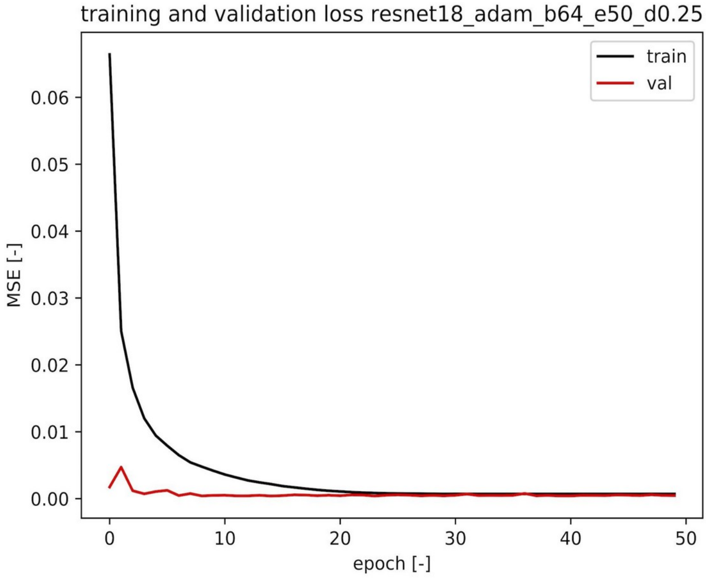


- Resnet (depth) 18/34/50/101 pretrained model (11.174M-48.513M) parameters
- Replaced last fully connected layer (512,1)
- Regression problem displacement dy normalized [-1,1]
- Adam/SGD + momentum optimizer
- 120 epoch 1e-4 learning rate

Training results with Resnet 18,34,50,101

- Resnet18 adam optimiser (lr=1e-4) good lightweight solution for real-time tracking
- Training time +-2hrs Nvidia 1050ti (Resnet18) 120 epochs

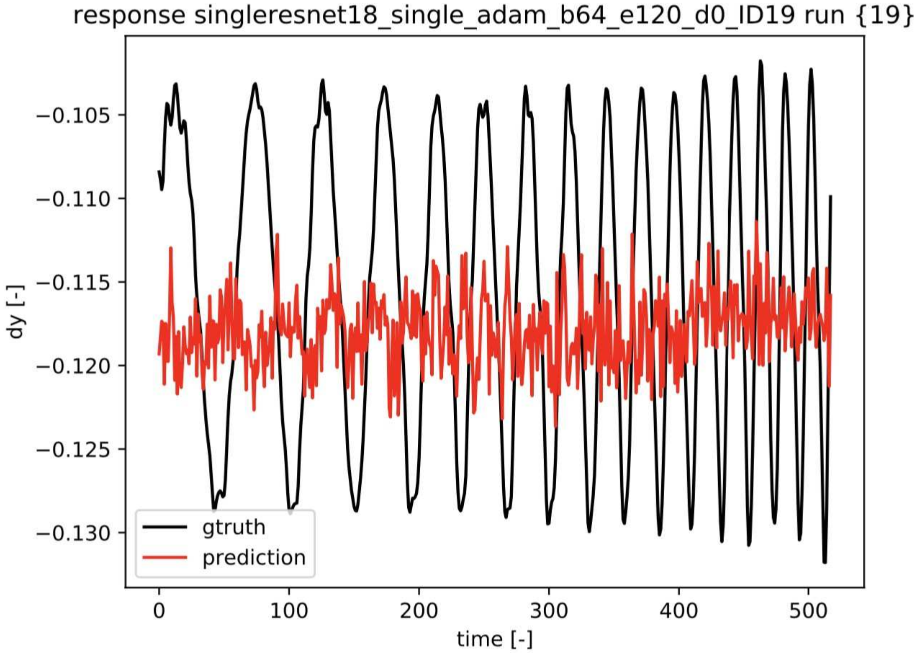




TUDelft

However inference not very successful

- Fails to predict response
- Possibly due classification-regression approach is better

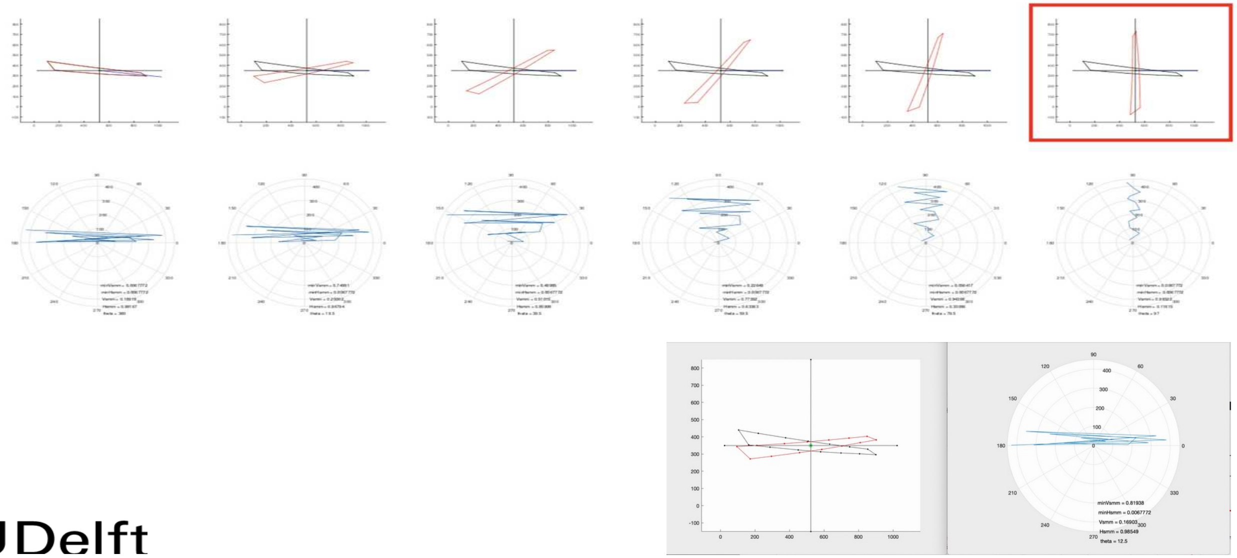


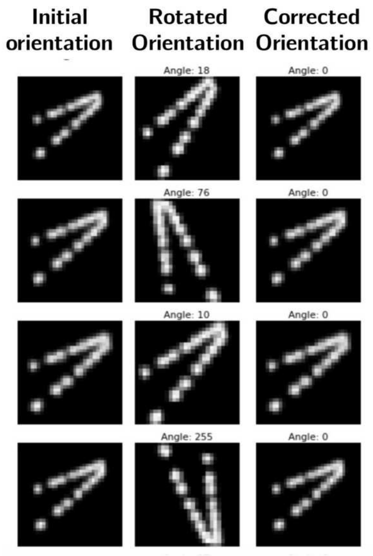
More successful symmetry detection

- Classification problem, detect angle rotation [0,360]
- MNIST network structure (d4nst/RotNet), input 28x28x1
- Also solvable with ML + geometric convolution

ML/ CV

AI/ DCNN

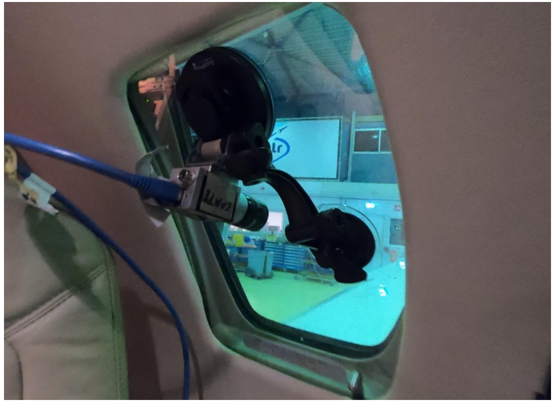


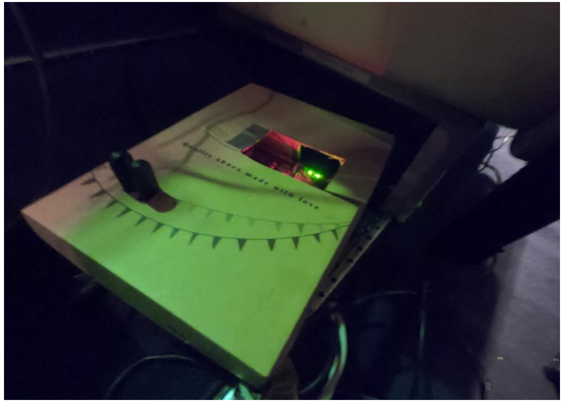


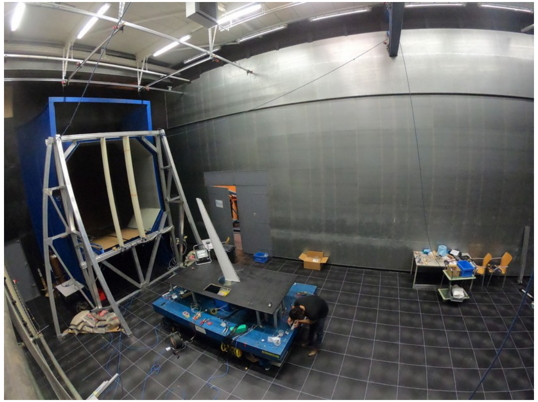
TU Delft

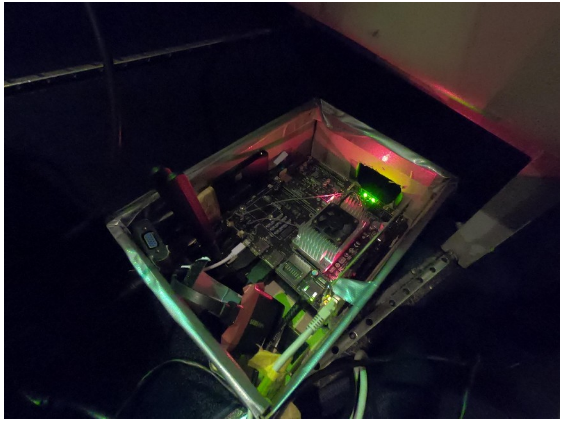
Experiments Wind Tunnel and Flight Test

Images flight and wind tunnel test









Flight test footage: during experiment

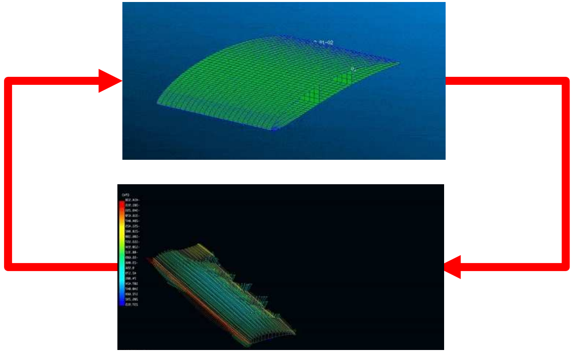
Flight test footage: landing

TU Delft

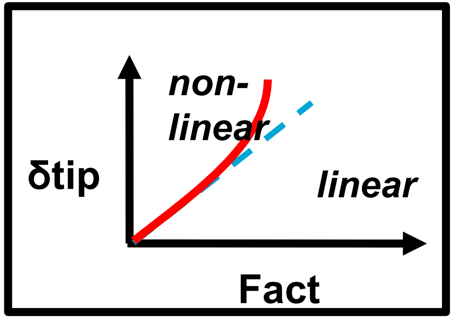
Future work / AI Applications
Aeroservoelasticity

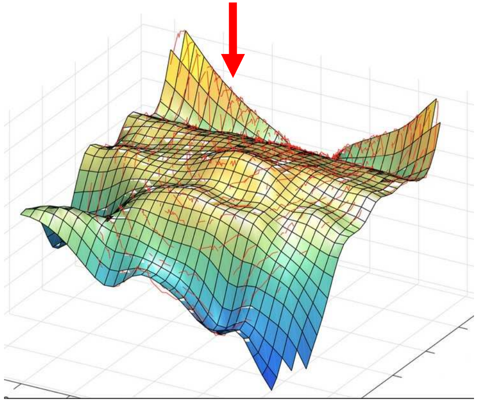
Multidimensional complex function approximation

aero load



shape





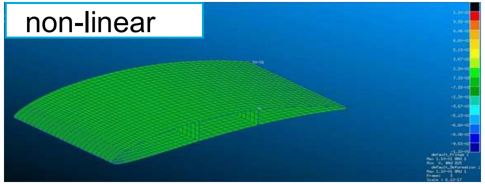
input → output

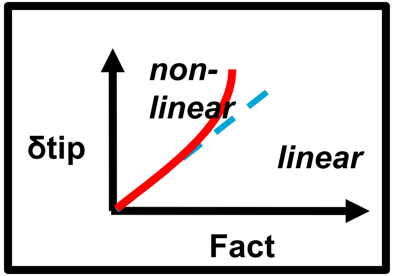
u x

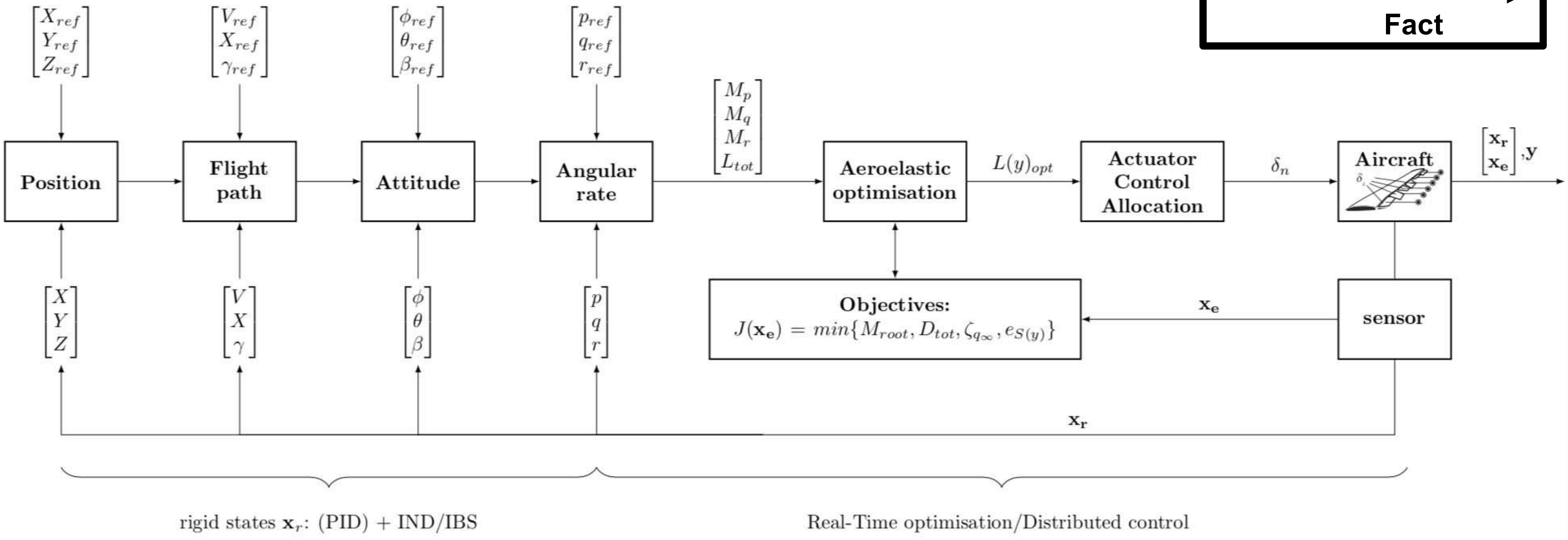
$f(x, u)$

TU Delft

Investigate RL to take care of non-linearity

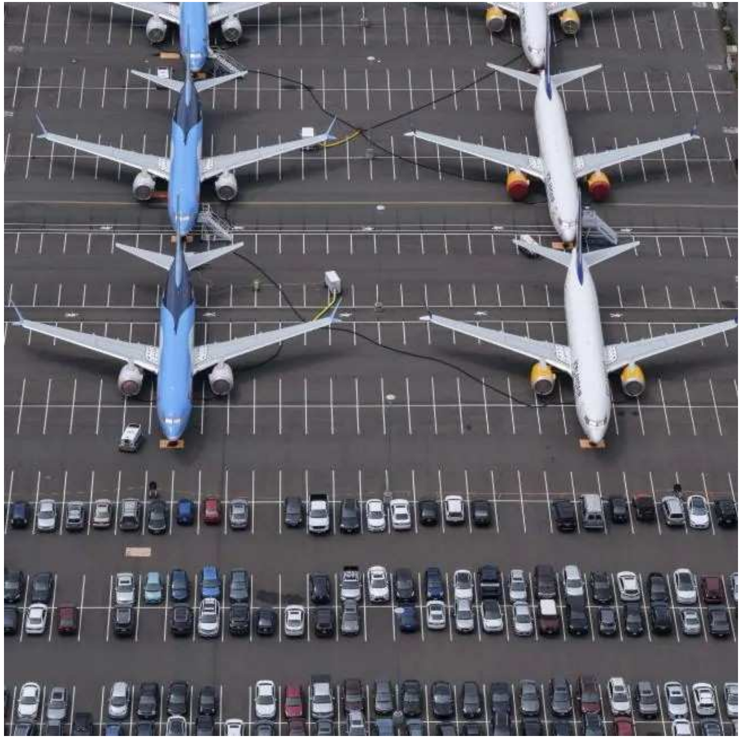






To AI or not to AI?

Grounding of 737-MAX



Picture: Stephen Brashear/Getty Images/AFP*Source:AFP*

Thank you