Model Validation & Course Summary

AE4320 System Identification of Aerospace Vehicles

Coen de Visser
Department of Control & Simulation
30-3-2017

Course Outline

- **Lecture 1: (dr.ir. Coen de Visser)**
  - Course goals and objectives
  - Introduction to System Identification
- **Lecture 2,3: (dr.ir. Daan Pool)**
  - System Identification Experiments
- **Lecture 4,5,6: (dr.ir. Daan Pool)**
  - Kalman filters
  - State estimation & Sensor Fusion
- **Lecture 7,8: (dr.ir. Erik-Jan van Kampen)**
  - Advanced identification approach: Neural networks
  - Advanced identification approach: Interval Analysis

Course Outline

- **Lecture 9,10: (dr.ir. Coen de Visser)**
  - Model structure selection
  - Model parameter estimation

- **Lecture 11,12: (dr.ir. Coen de Visser)**
  - Advanced identification approach: Multivariate B-Splines

- **Guest Lecture 13: (ir. Henry Tol)**
  - Active Flow Control using Multivariate B-Splines

- **Lecture 14: (dr.ir. Coen de Visser)**
  - Model validation, course conclusion

Goals of this Lecture

Questions that will be answered during this lecture:

1. *What is model validation and why is it important?*
2. *What are the different methods for model validation?*
3. *How should I decide if a model is ‘good’ or ‘bad’?*

4. *What assignments are available and which should I choose?*

SysID High Level Overview

Where we are now in the System Identification Cycle:

Experiment phase

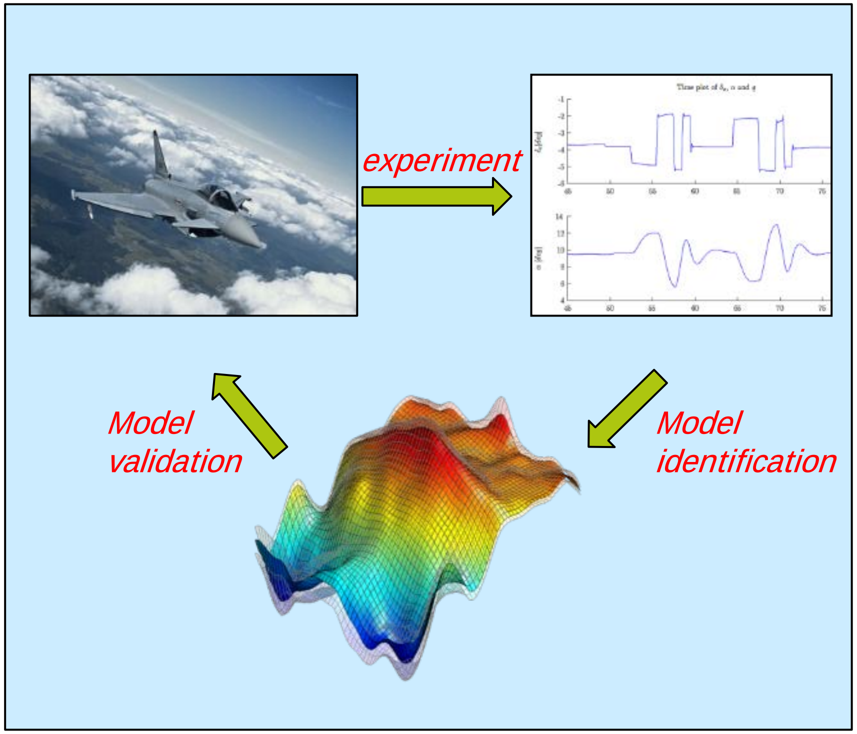
- Plant analysis
- Experiment design and execution
- Data logging and pre-processing

Model identification phase

- State estimation
- Model structure definition
- Parameter estimation

Model validation phase

- Model validation

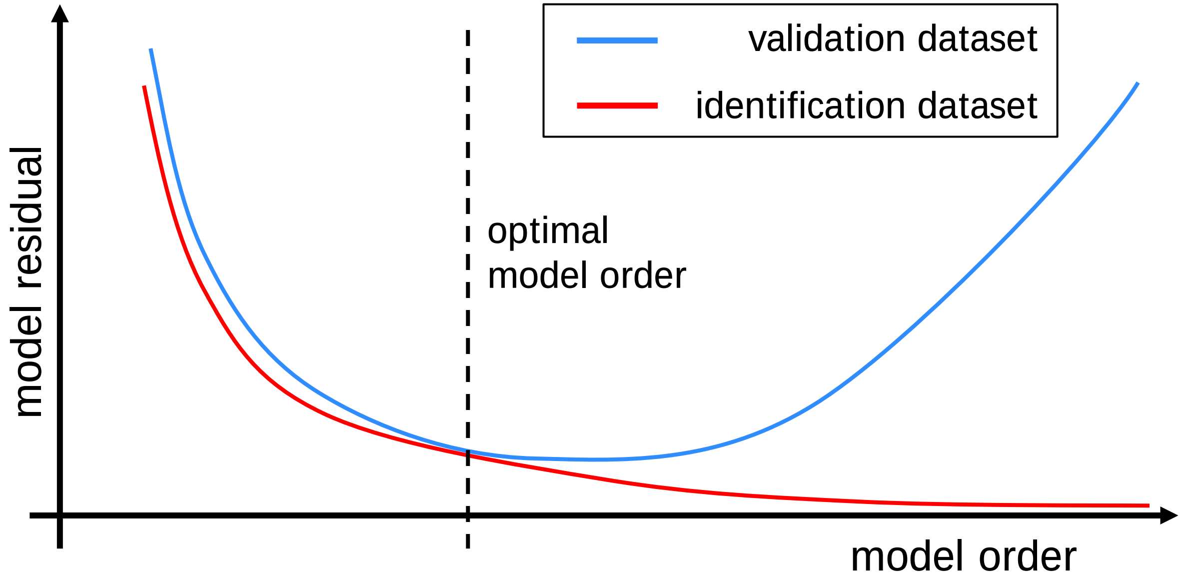


Model Validation

How do we validate a model?

Data **must** be split into a separate **identification** and **validation** datasets.

- Identification of a model performed using the identification dataset.
- The identified model is then tested against the validation dataset.



Model Validation

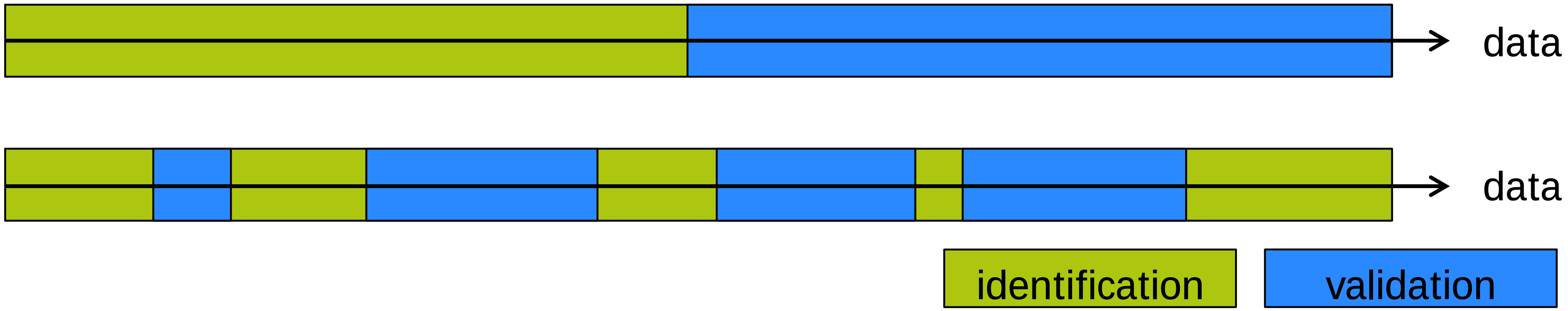
Identification & Validation datasets.

Splitting the dataset into identification and validation sets can be done in many different ways.

In general we want to have a **validation** dataset that is **as large as possible!**

In any case, try to divide into a identification set containing at least 50% of the data, and a validation set containing 50% of the data. Make sure the validation **set contains sufficient dynamics!**

There are many different strategies for splitting the dataset:



Model Validation

Methods for model validation

The following model validation methods can be used:

1. Analysis of model residuals: zero mean white noise?

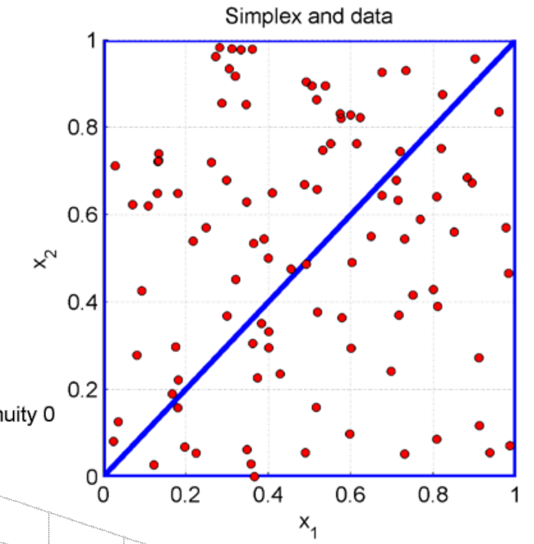
$$\varepsilon = p(x,\hat{\theta}) - Y$$

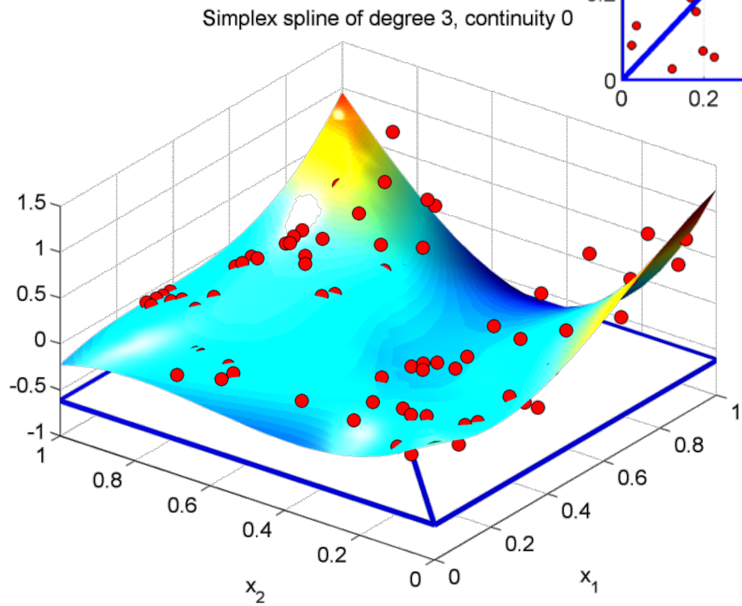
2. Parameter (co)variances

$$Cov\left\{\hat{\theta}\right\}$$

3. Spline specific:
   B-coefficient bounds

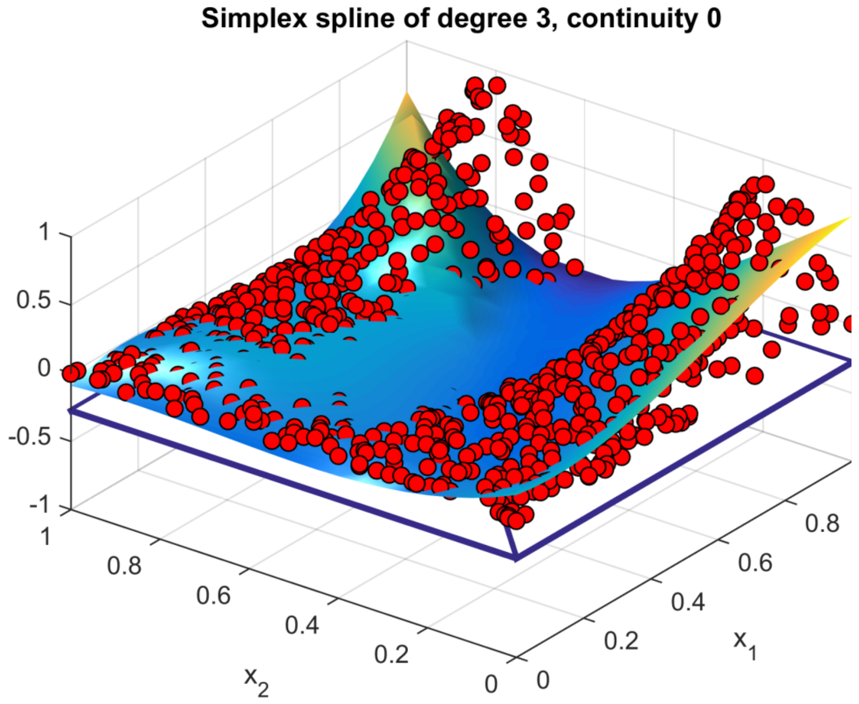
$$p(x) = \left\{\min(\hat{c}), \max(\hat{c})\right\}$$

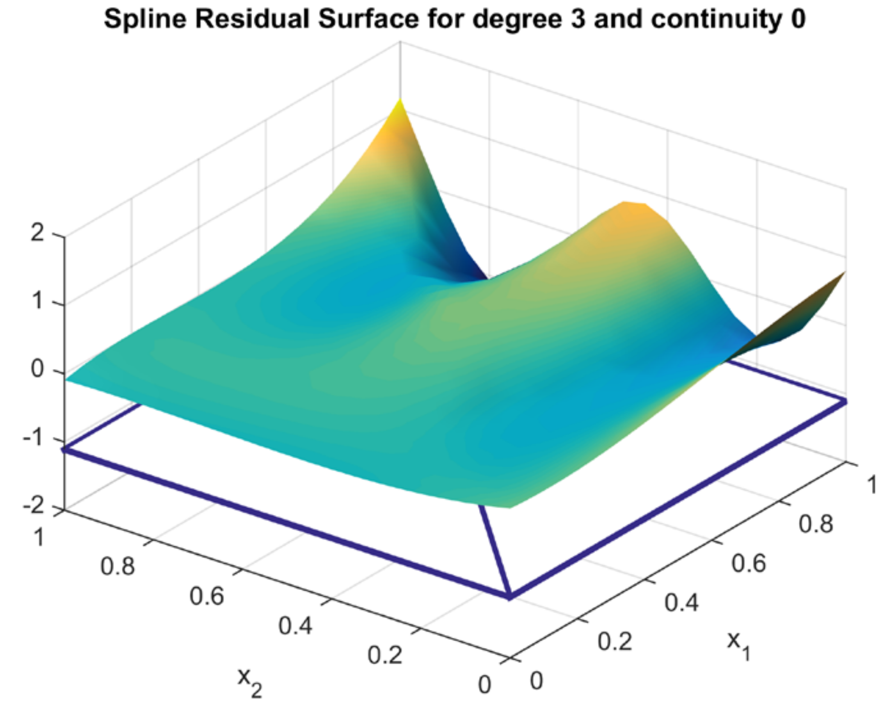




Simplex Spline Model Validation

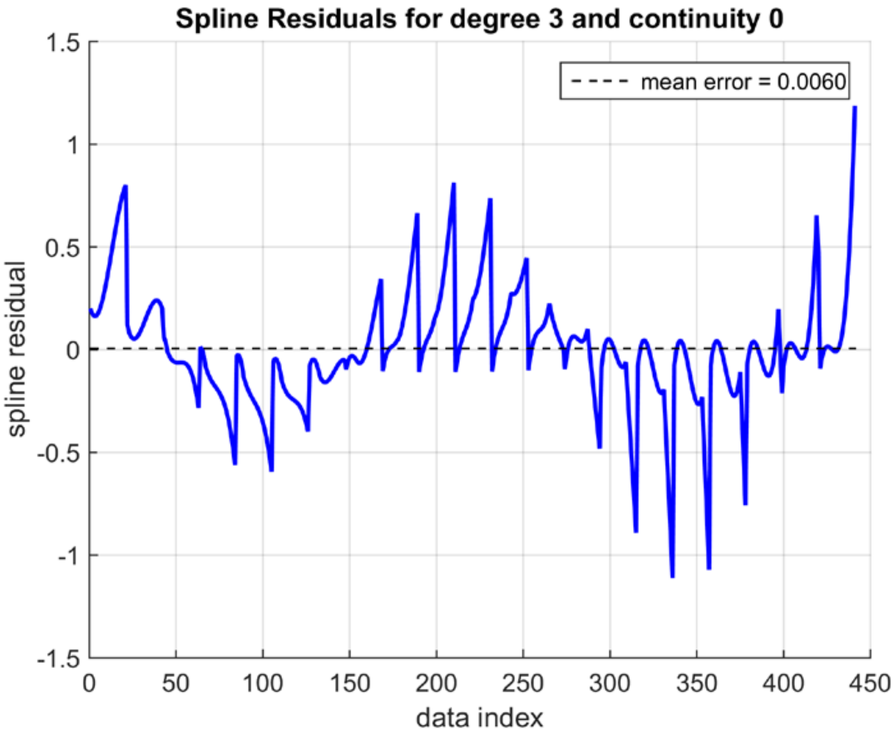
Model residual analysis: $$\varepsilon = p(x,\hat{\theta}) - Y$$
$$= s_0^3(x) - Y$$

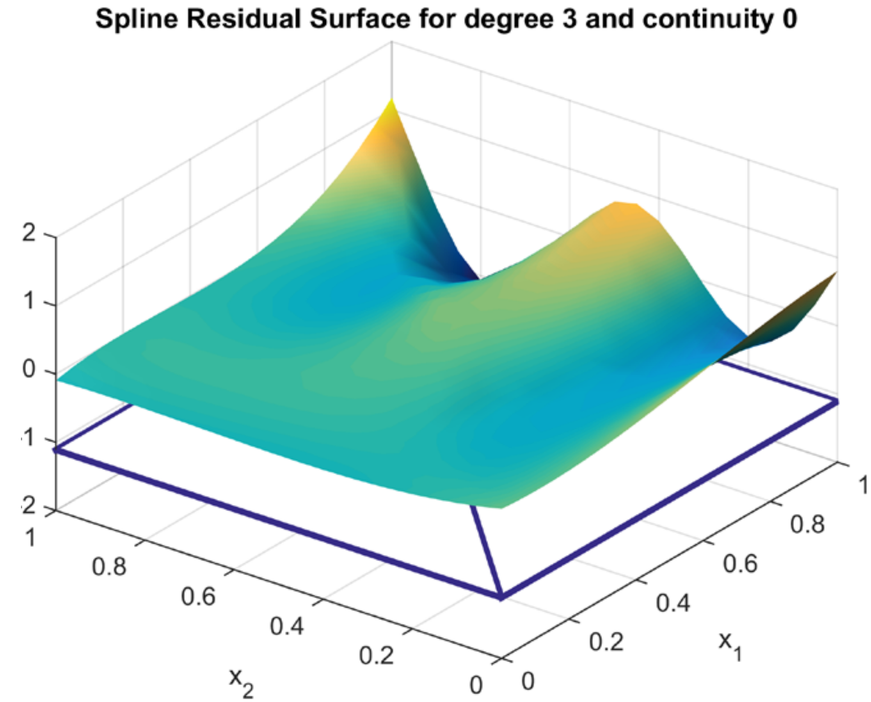




Simplex Spline Model Validation

Model residual analysis: Zero mean uncorrelated residual?



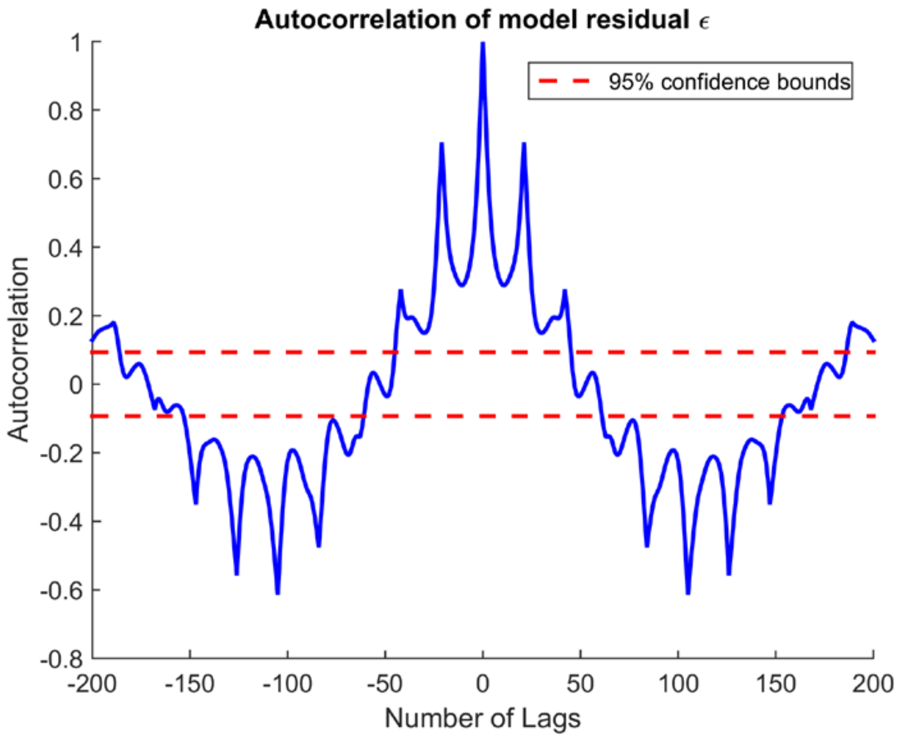


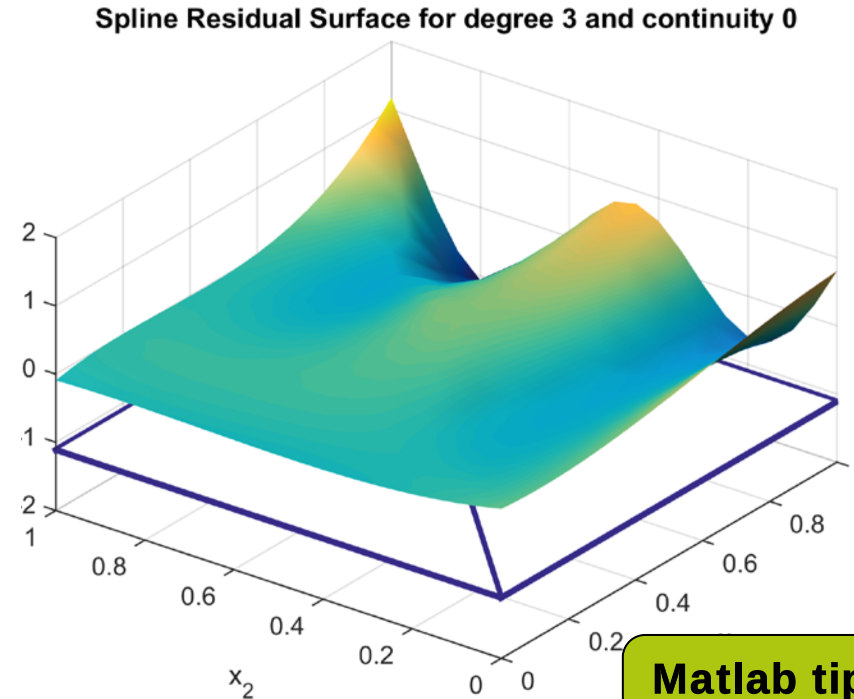
Simplex Spline Model Validation

Model residual analysis: Whiteness of residual

Calculated using autocorrelation function: $\gamma(l) = \sum_{i=-N}^{N} \varepsilon(i)\varepsilon(i+l)$

95% confidence bounds: $conf = 1.96 / \sqrt{N}$



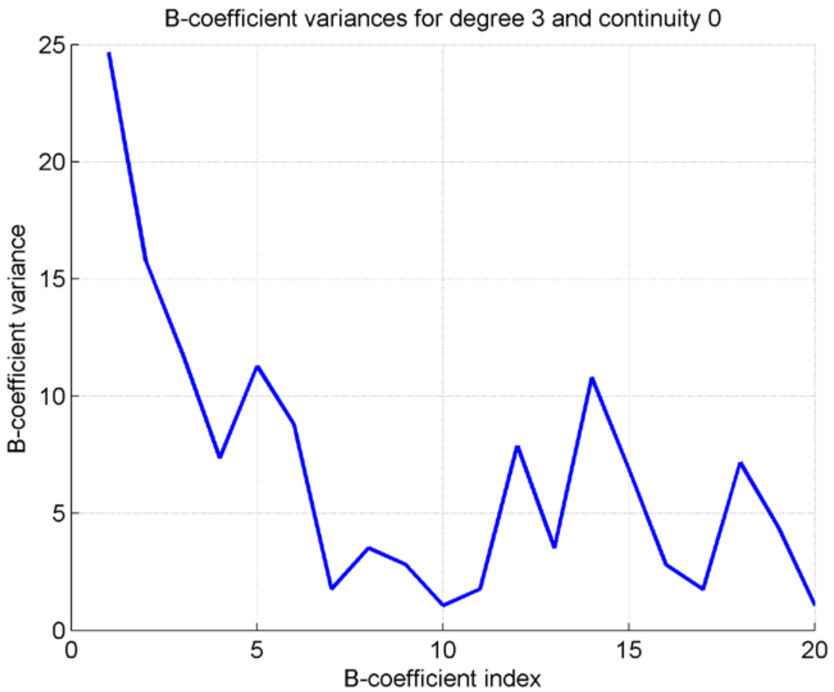


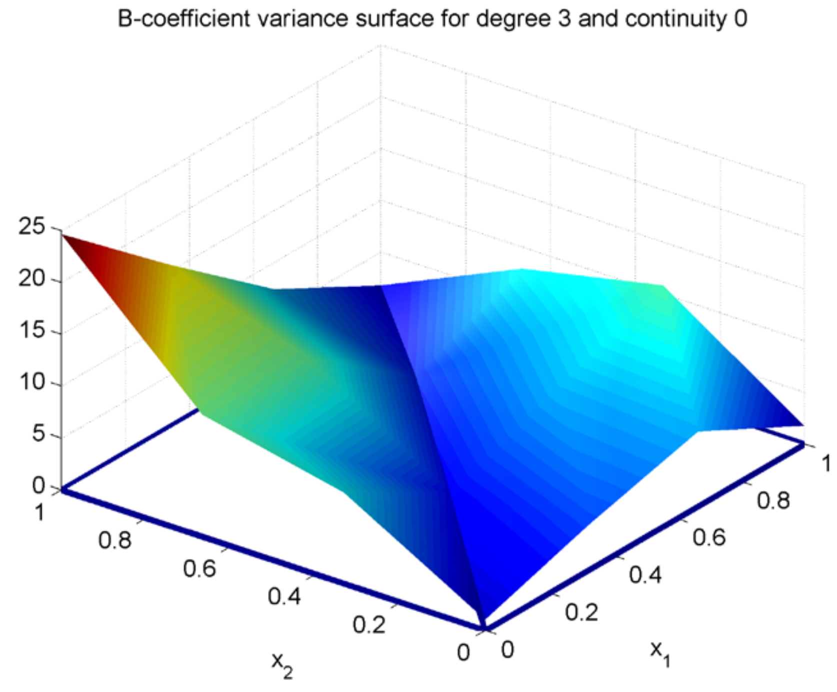
Matlab tip:
use xcorr

Simplex Spline Model Validation

Statistical analysis

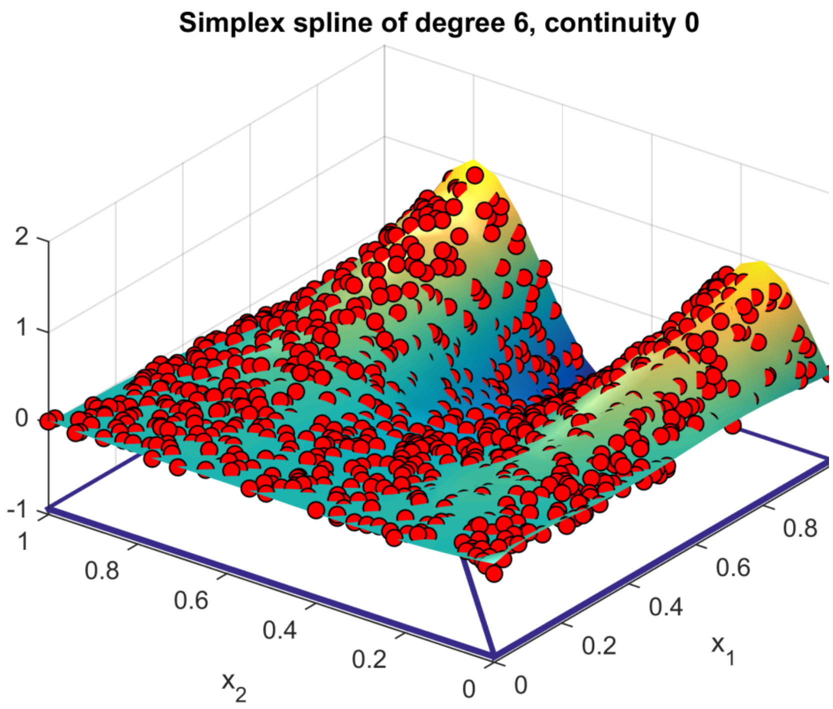
B-coefficient variances are given by: $Cov\{\hat{c}\}$

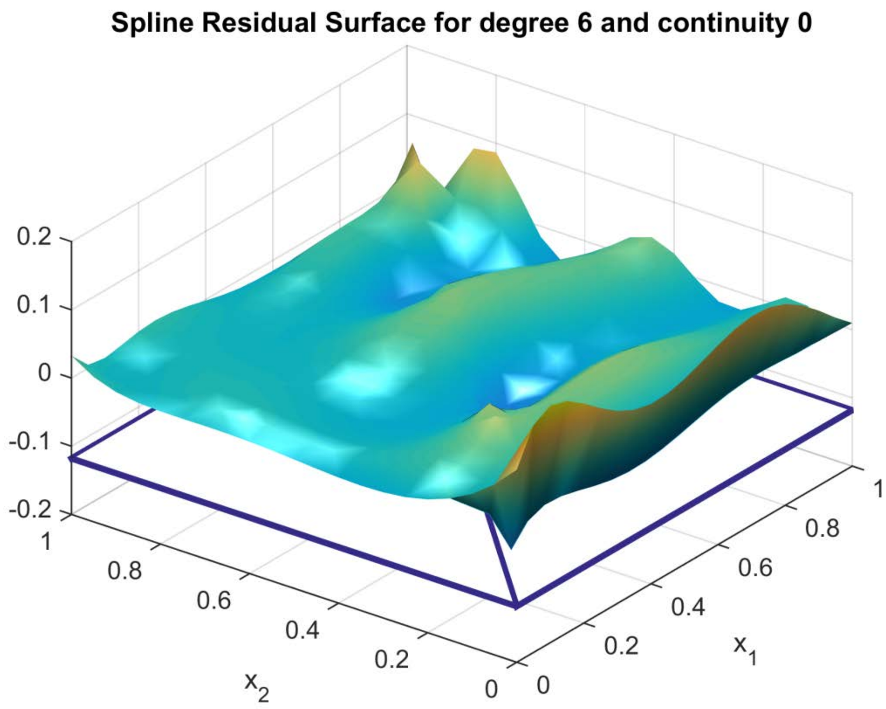




Simplex Spline Model Validation

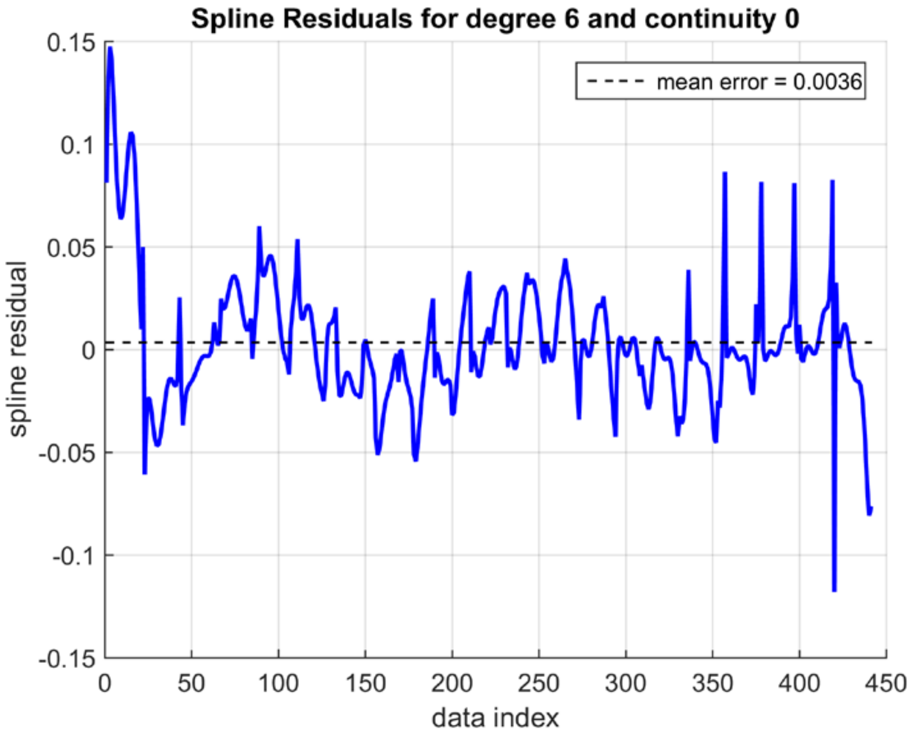
High order spline model residual analysis: $\varepsilon = s_0^6(x) - Y$

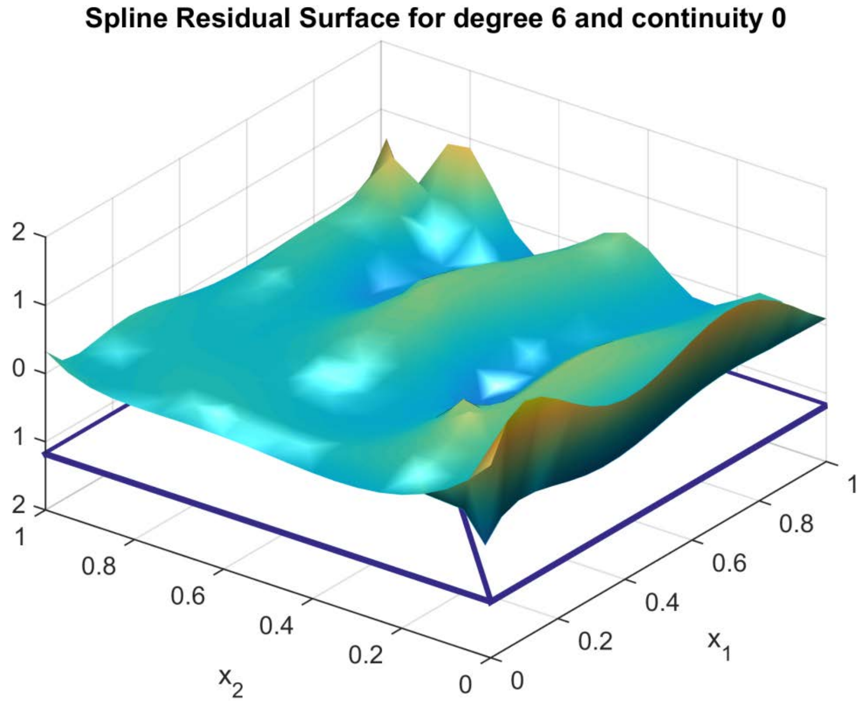




Simplex Spline Model Validation

Model residual analysis: Zero mean uncorrelated residual?



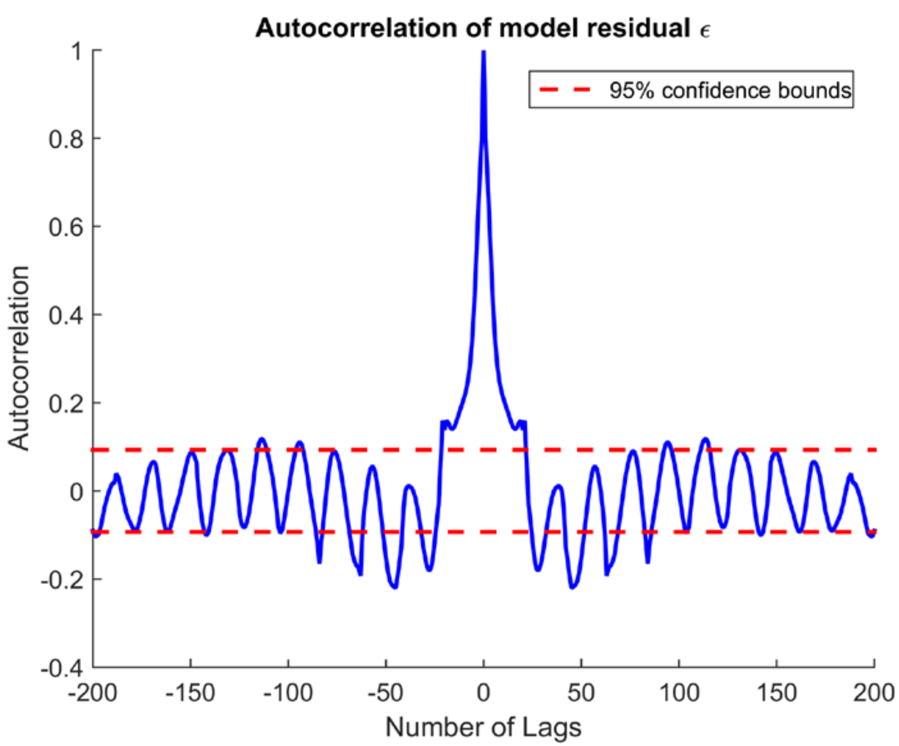


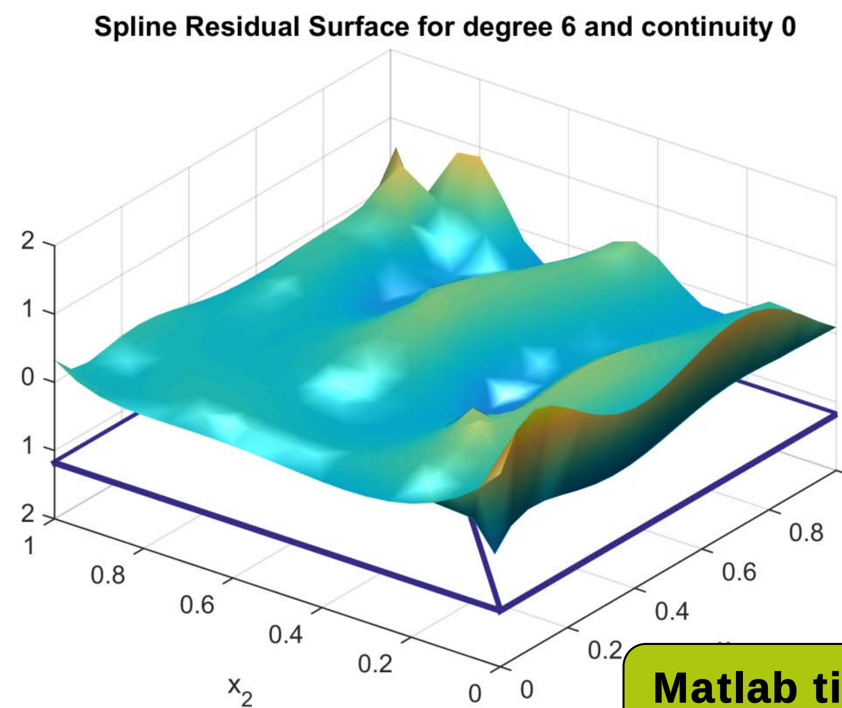
Simplex Spline Model Validation

Model residual analysis: Whiteness of residual

Calculated using autocorrelation function: $\gamma(l) = \sum_{i=-N}^{N} \varepsilon(i)\varepsilon(i+l)$

95% confidence bounds: $conf = 1.96 / \sqrt{N}$



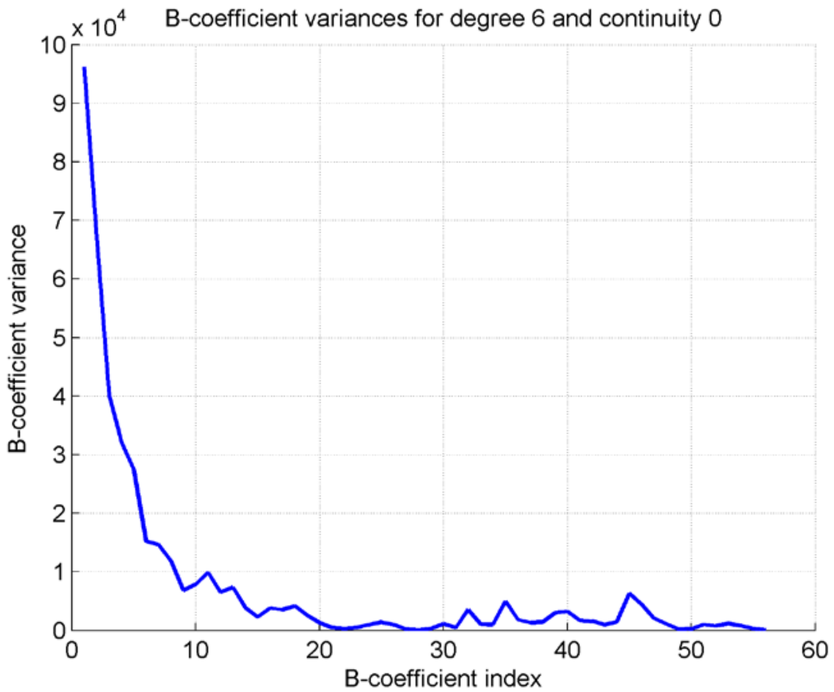


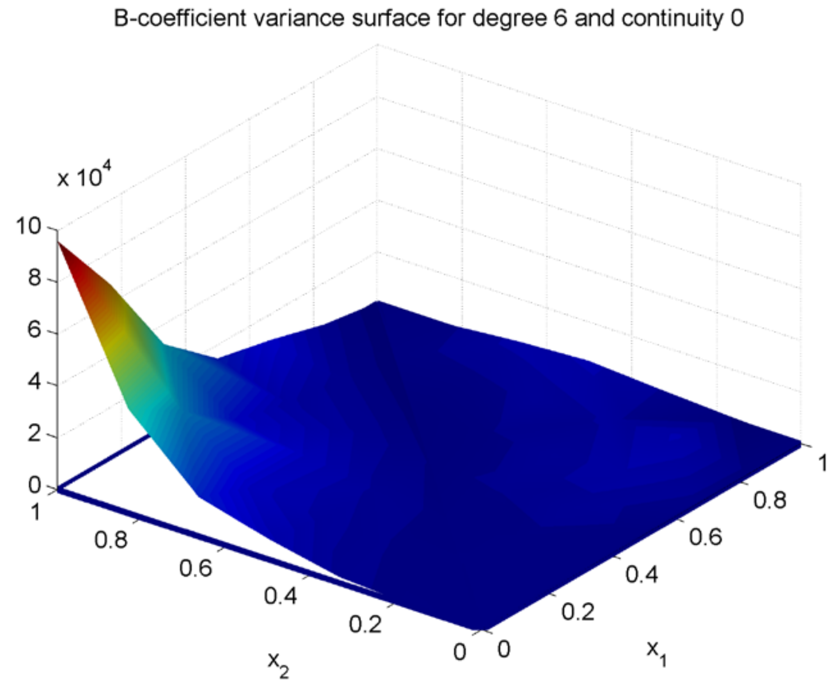
Matlab tip:
use xcorr

Simplex Spline Model Validation

B-coefficient variance surfaces

B-coefficient variances are given by: $Cov\{\hat{c}\}$

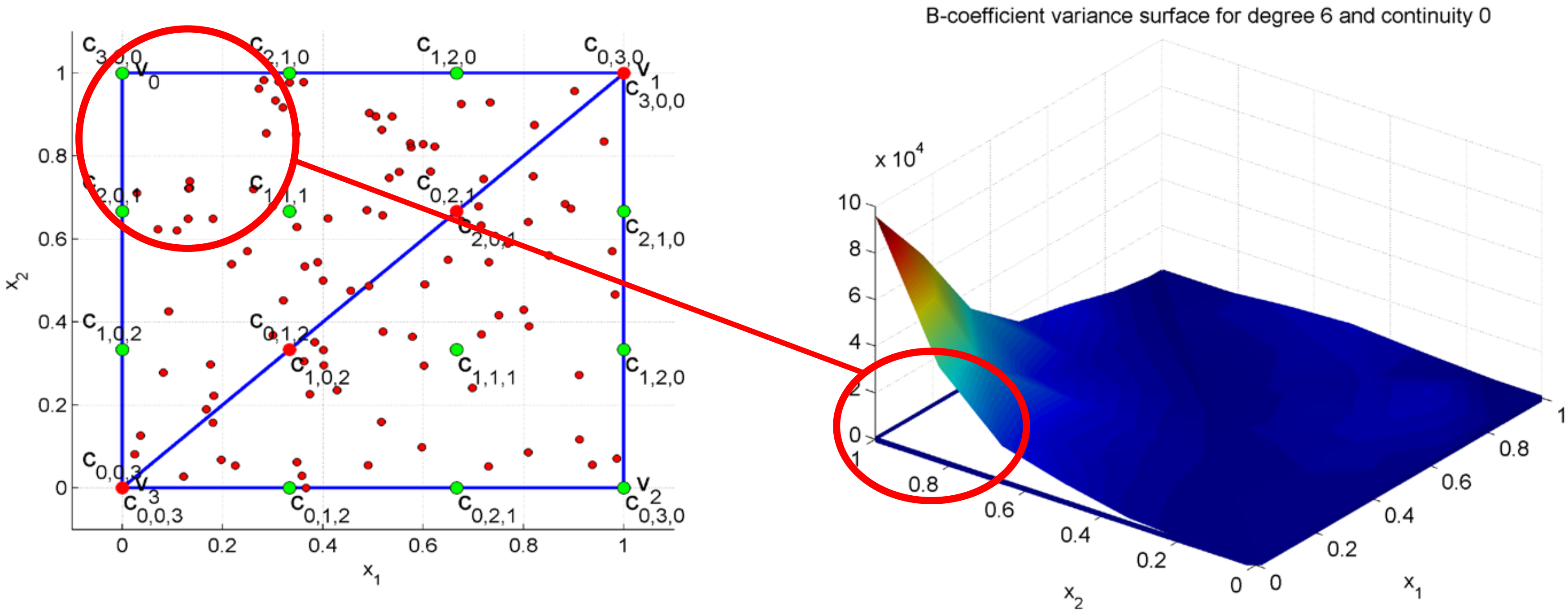




Simplex Spline Model Validation

B-coefficient variance surfaces

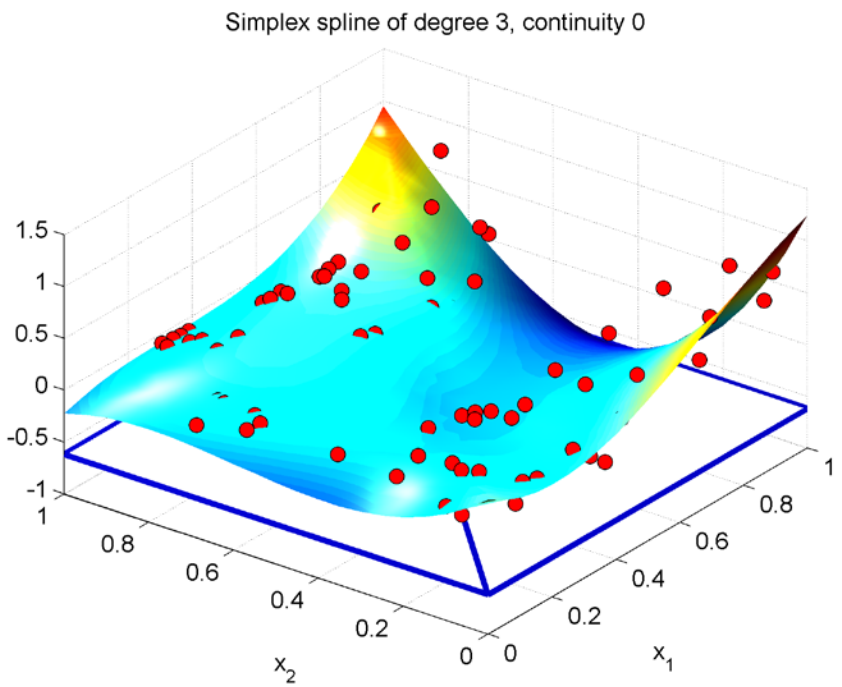
B-coefficient variance surfaces provide a mechanism to pinpoint local estimator failure to specific parts of the model. This in turn may point to inadequate local data coverage/conditioning, or incorrect model structures.

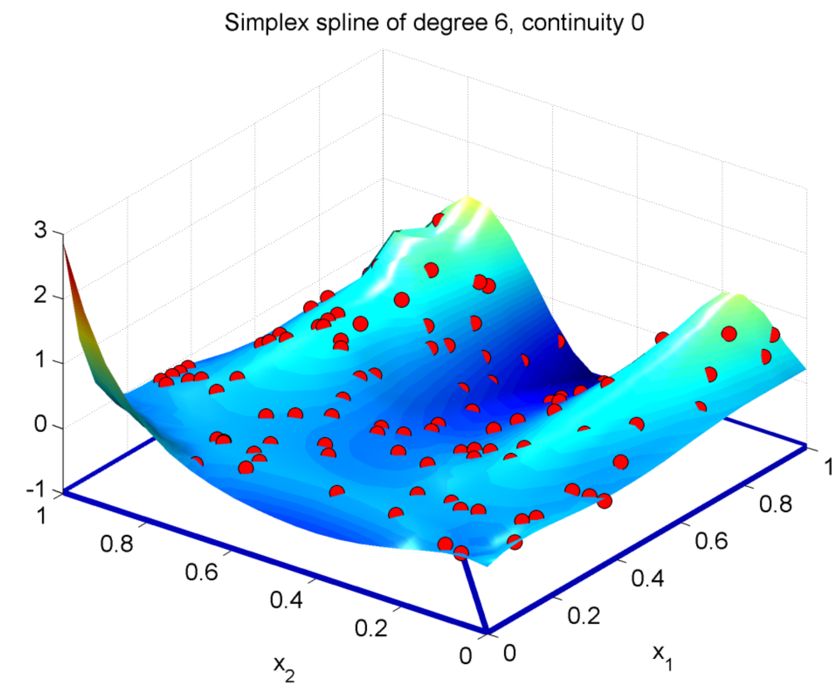


Simplex Spline Model Validation

B-coefficient bounds

B-form polynomials are bounded by B-coefficient values: $p(x) \in \{\min(\hat{c}), \max(\hat{c})\}$

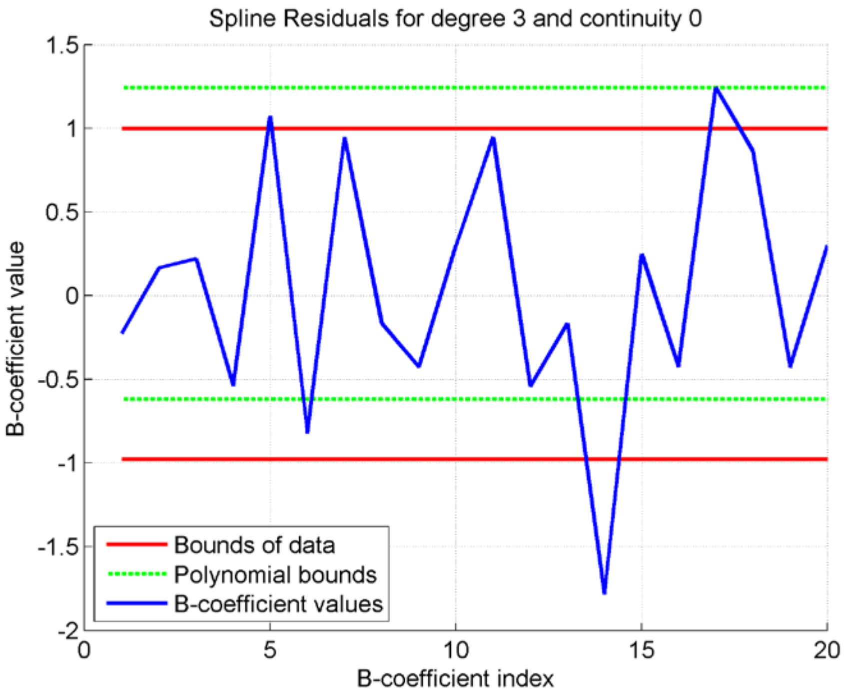


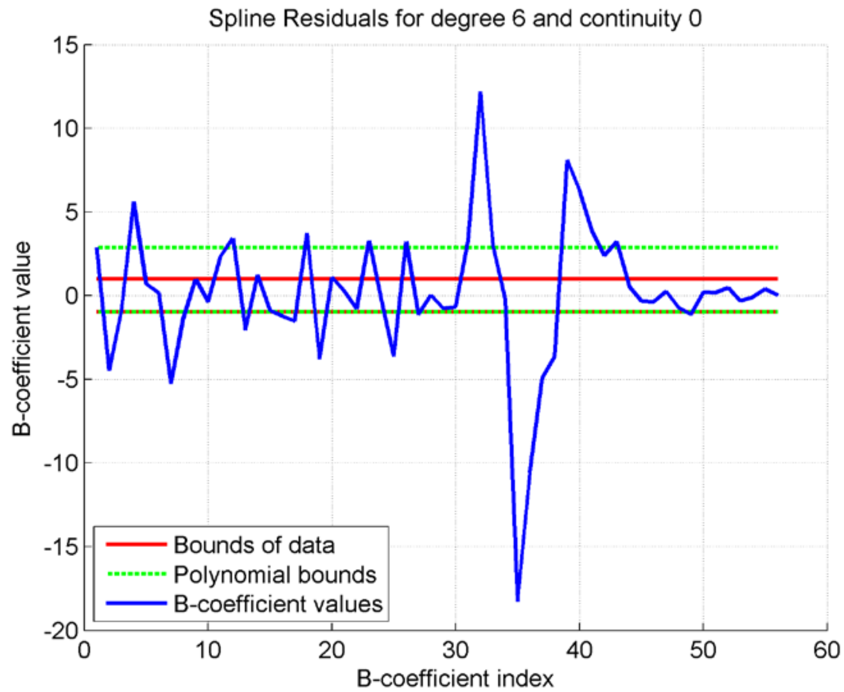


Simplex Spline Model Validation

B-coefficient bounds

B-form polynomials are bounded by B-coefficient values: $p(x) \in \left\{ \min(\hat{c}), \max(\hat{c}) \right\}$

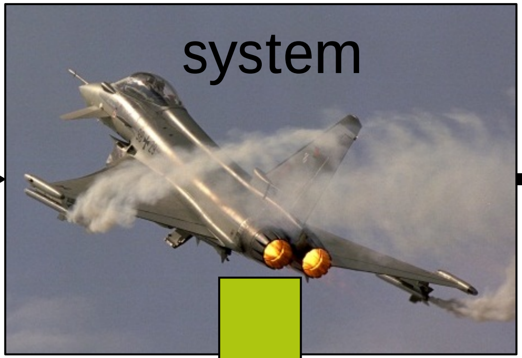


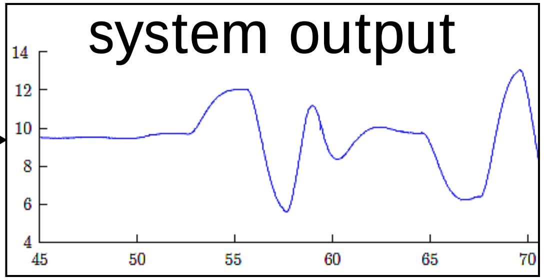


Completing the SysID Cycle…

Main course goal:

To provide tools and theory for creating and validating models of dynamic systems.

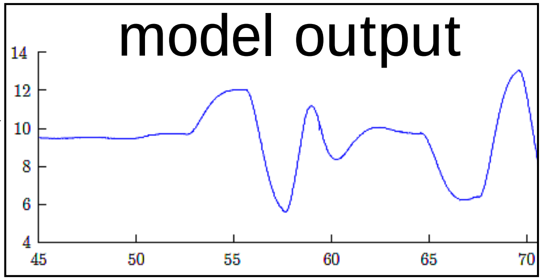




?

$$Y(s) = \frac{H(s)}{U(s)}$$

$$\dot{x}(t) = f(x,u,t)$$
$$y(t) = h(x,u,t)$$



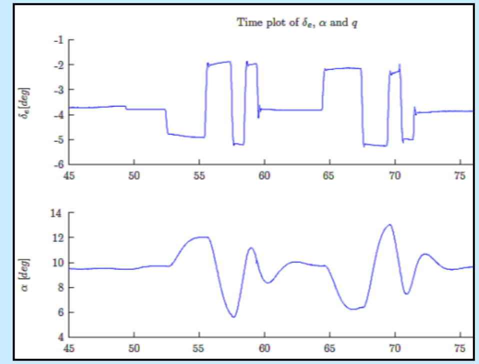
Completing the SysID Cycle…

The System Identification Cycle

plant

experiment

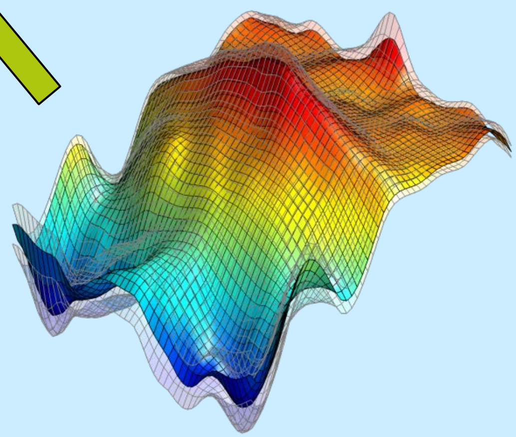
- *design*
- *execution*



data

Model identification

- *state reconstruction*
- *model structure definition*
- *parameter estimation*

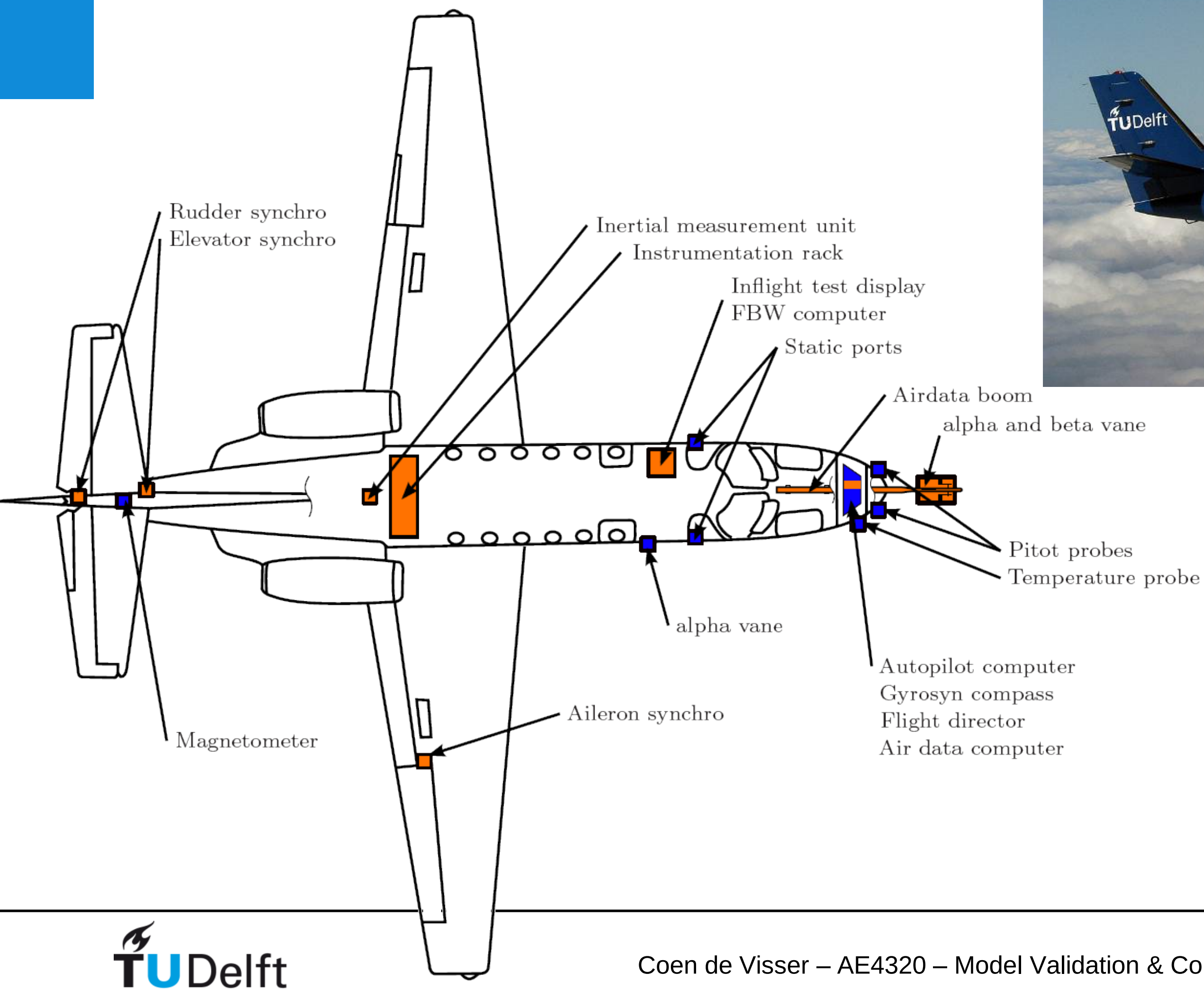


model

Model validation

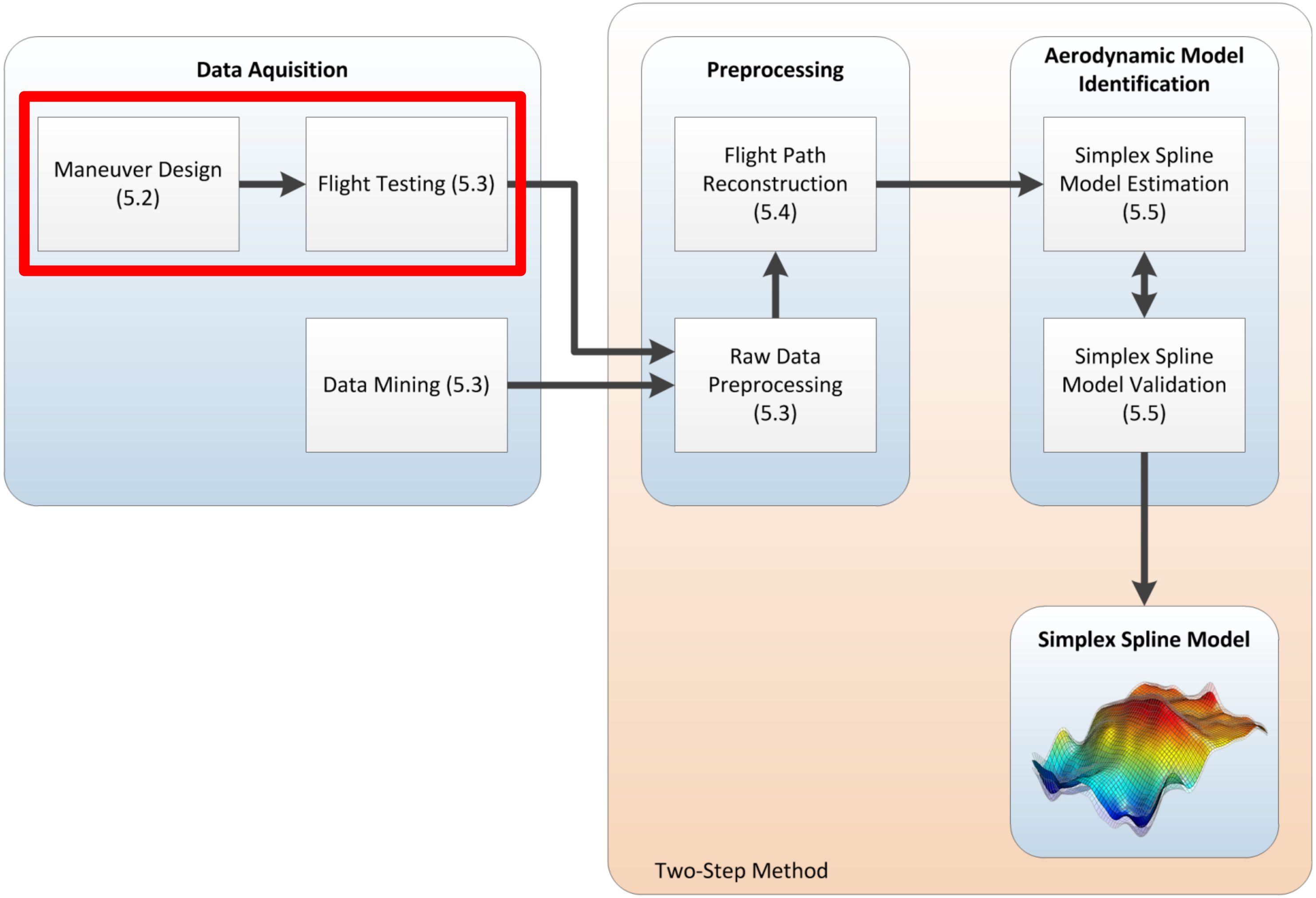
- *residual analysis*
- *statistical analysis*

Citation II Aero-Model Identification



Citation II Aero-Model Identification

The procedure for AMI

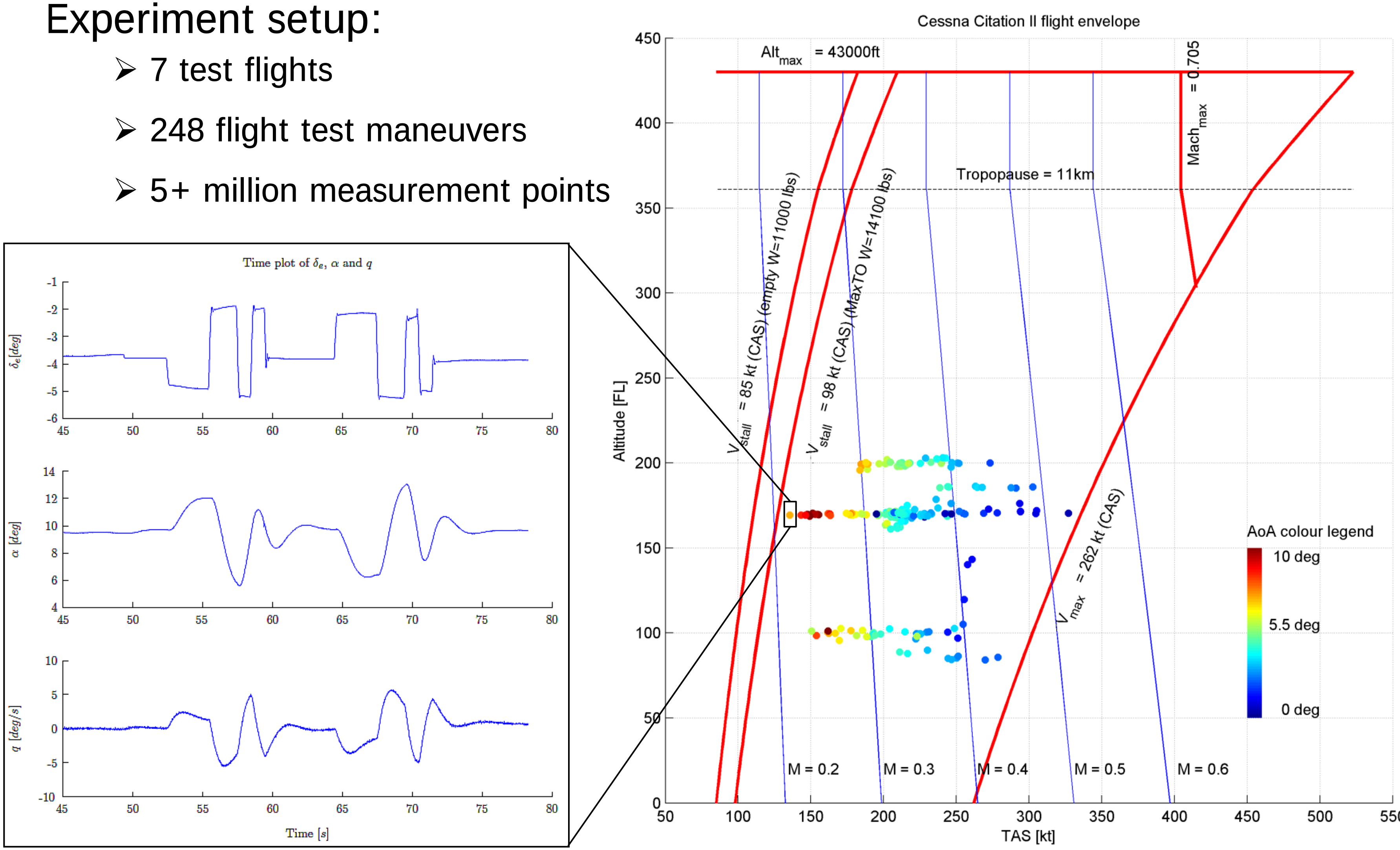


Citation II AMI: Experiments

Citation II AMI: Experiments

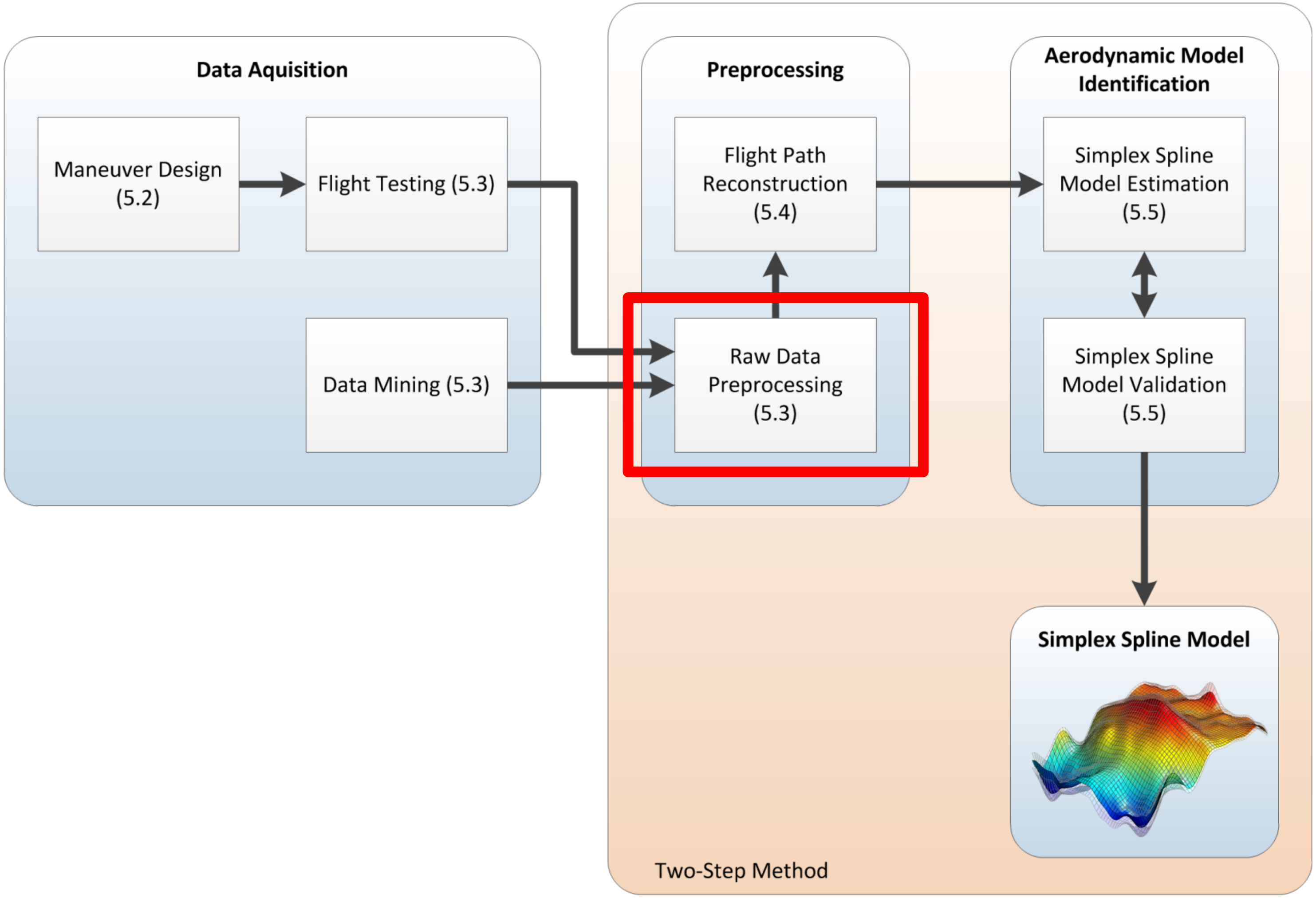
Experiment setup:

- 7 test flights
- 248 flight test maneuvers
- 5+ million measurement points



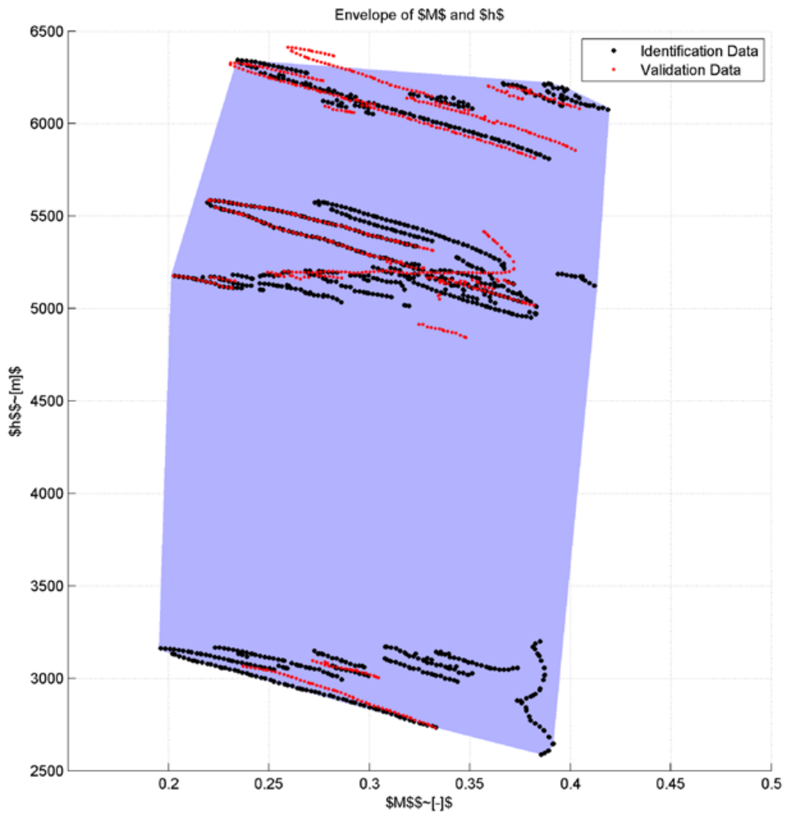
Citation II Aero-Model Identification

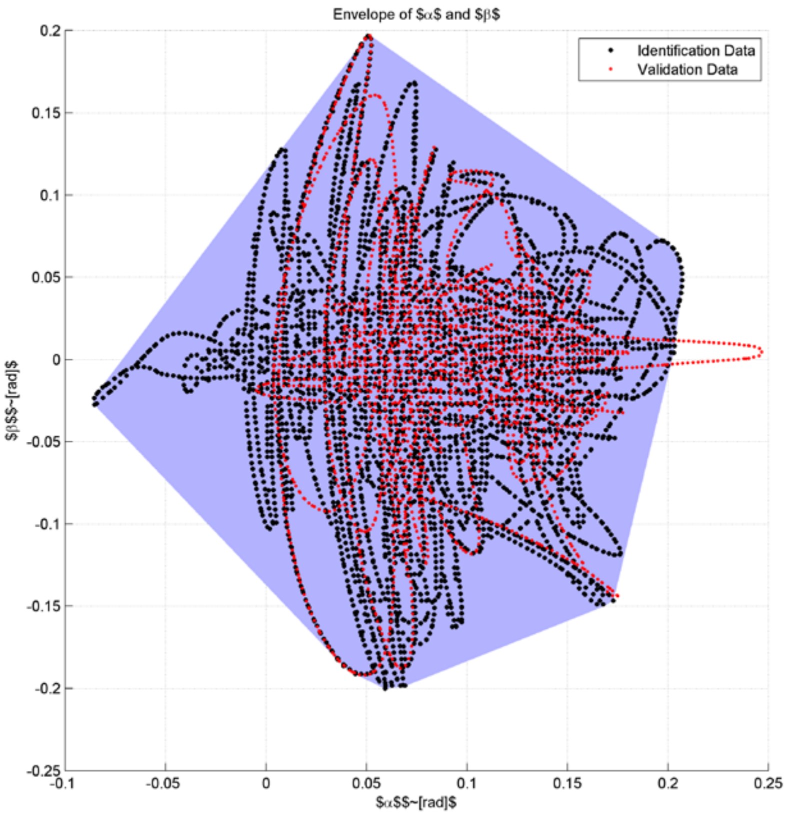
The procedure for AMI



Citation II AMI: Data Processing

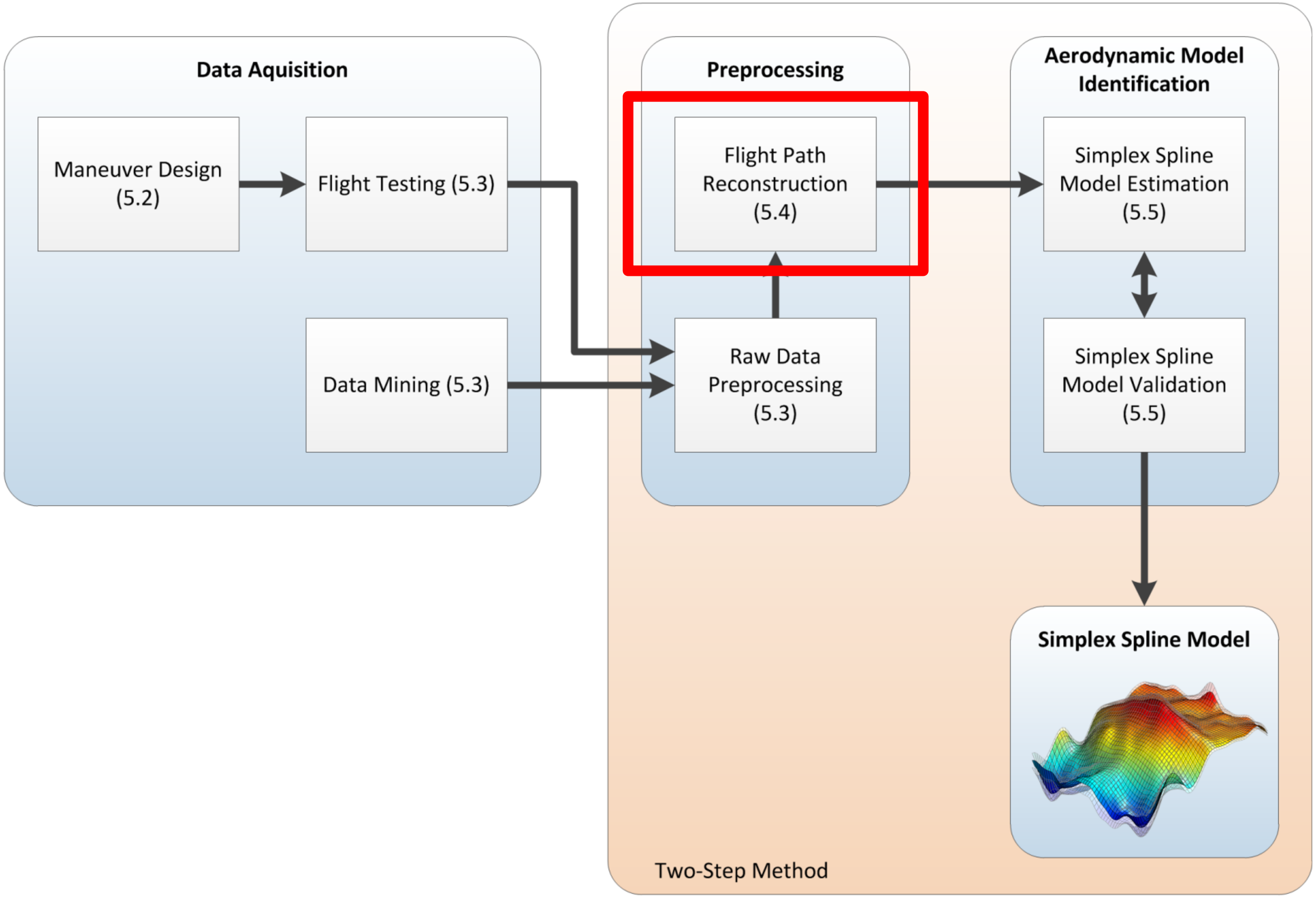
Envelope of the Flight Data





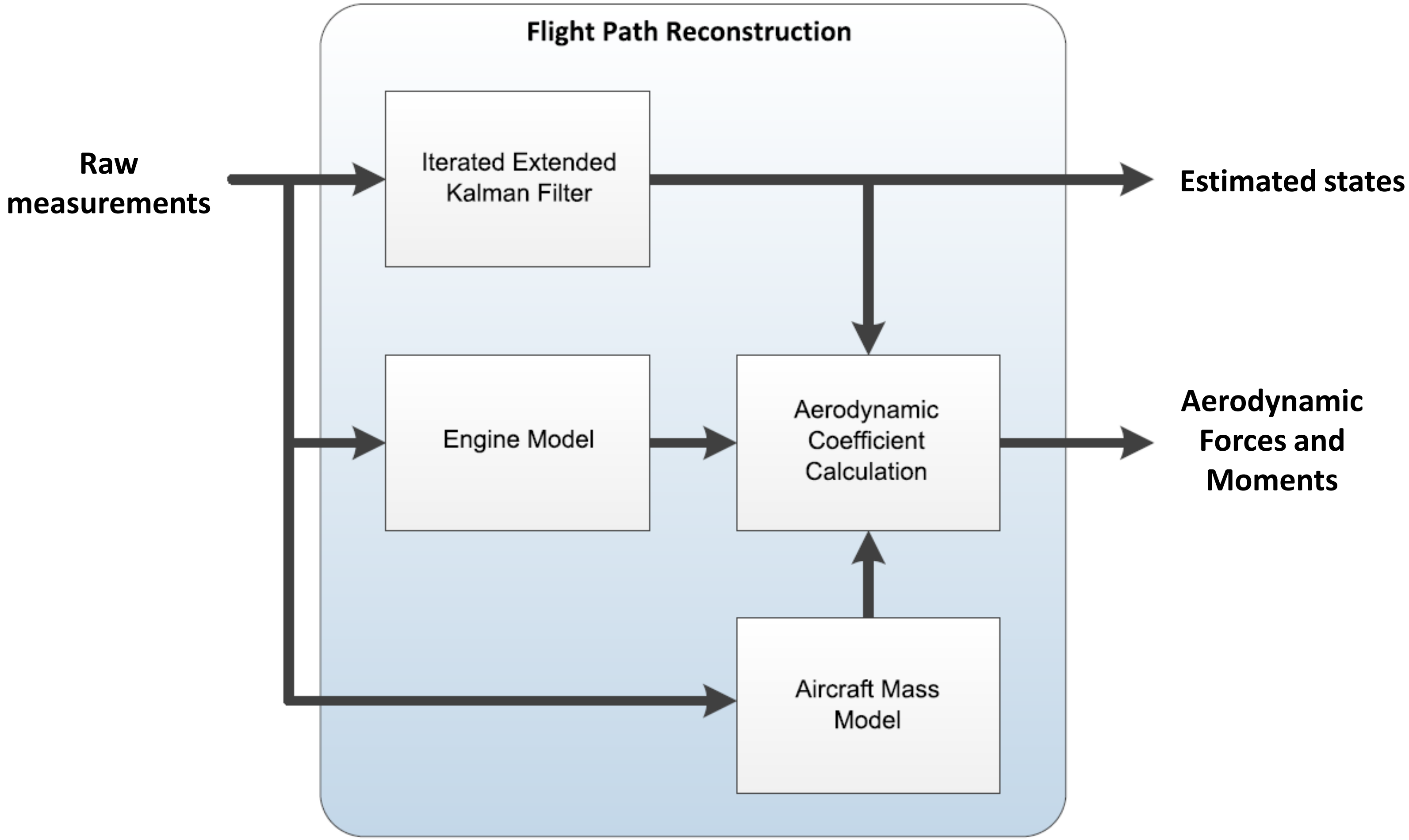
Citation II AMI: Flight path reconstruction

The procedure for AMI



Citation II AMI: Flight path reconstruction

Flight path reconstruction using IEKF



Citation II AMI: Flight path reconstruction

Define (observable) augmented state vector:

$$\mathrm{x}_m = \begin{bmatrix} u & v & w & \phi & \theta & \psi & z & \lambda_x & \lambda_y & \lambda_z & \lambda_p & \lambda_q & \lambda_r & C_{\alpha_{up}} \end{bmatrix}$$

Define state dynamics:

$$\begin{aligned}
\dot{\mathrm{x}}_m &= f(\mathbf{x}_m(t), \mathbf{u}_m(t), \mathbf{w}(t), t) \\
&= \begin{bmatrix}
A_x + \lambda_x - g\sin(\theta) - w(\lambda_q + q) + v(\lambda_r + r) \\
A_y + \lambda_y + w(\lambda_p + p) - u(\lambda_r + r) + g\cos(\theta)\sin(\phi) \\
A_z + \lambda_z - v(\lambda_p + p) + u(\lambda_q + q) + g\cos(\phi)\cos(\theta) \\
\lambda_p + p + (\lambda_r + r)\cos(\phi)\tan(\theta) + (\lambda_q + q)\sin(\phi)\tan(\theta) \\
(\lambda_q + q)\cos(\phi) - (\lambda_r + r)\sin(\phi) \\
(\lambda_r + r)\cos(\phi)\sec(\theta) + (\lambda_q + q)\sin(\phi)\sec(\theta) \\
w\cos(\phi)\cos(\theta) - u\sin(\theta) + v\cos(\theta)\sin(\phi) \\
\mathbf{0}_{7\times 1}
\end{bmatrix}
\end{aligned}$$

Citation II AMI: Flight path reconstruction

Define observation vector:

$$\mathbf{y} = \begin{bmatrix} \phi & \theta & \psi & V & \alpha_v & \beta_v & z \end{bmatrix}^\top$$

Define observation equations:

$$\begin{aligned}
\mathbf{y}_m &= h(\mathbf{x}_m(t), \mathbf{u}_m(t), \mathbf{v}(t), t) \\
&= \begin{bmatrix}
\phi \\
\theta \\
\psi \\
\sqrt{u^2 + v^2 + w^2} \\
(1 + C_{\alpha_{up}}) \arctan\left(\frac{w}{u}\right) + \frac{(q+\lambda_q)x_{v_\alpha}}{\sqrt{u^2+v^2+w^2}} + \\
\arctan\left(\frac{v}{\sqrt{u^2+w^2}}\right) - \frac{(r+\lambda_r)x_{v_\beta}}{\sqrt{u^2+v^2+w^2}} + \frac{(p+\lambda_p)z_{v_\beta}}{\sqrt{u^2+v^2+w^2}} \\
h - h_0
\end{bmatrix}
\end{aligned}$$

Citation II AMI: Flight path reconstruction

1. One-step ahead prediction: $\underline{\hat{x}}_{k+1,k} = \underline{\hat{x}}_{k,k} + \int_{t_k}^{t_{k+1}} \underline{f}\left(\underline{\hat{x}}_{k,k}, \underline{u}_k^*, t\right) dt$

2. Calculate Jacobians: $F_x(\bullet) = \frac{\partial}{\partial \underline{x}} f\left(\underline{x}(t), \underline{u}(t), t\right),\ H_x(\bullet) = \frac{\partial}{\partial \underline{x}} h\left(\underline{x}(t), \underline{u}(t), t\right)$

3. Discretize state transition & input matrices: $\Phi_{k+1,k}(\bullet),\ \Gamma_{k+1,k}(\bullet)$

4. Cov. matrix of state pred. error: $P_{k+1,k}(\bullet) = \Phi_{k+1,k}(\bullet) P_{k,k} \Phi_{k+1,k}^T(\bullet) + \Gamma_{k+1,k}(\bullet) Q_{d,k}(\bullet) \Gamma_{k+1,k}^T(\bullet)$

$k = 1,2,\ldots N$

$i = 1,2,\ldots l$

IEKF iterative part (see next sheet)

8. Cov. matrix of state estimation error:

$$P_{k+1,k+1}\left(\underline{\eta}_l\right) = \left[I_n - K_{k+1}\left(\underline{\eta}_l\right) H_x\left(\underline{\eta}_l\right)\right] P_{k+1,k}\left(\underline{\eta}_l\right) \left[I_n - K_{k+1}\left(\underline{\eta}_l\right) H_x\left(\underline{\eta}_l\right)\right]^T + K_{k+1}\left(\underline{\eta}_l\right) R_{k+1} K_{k+1}^T\left(\underline{\eta}_l\right)$$

Citation II AMI: Flight path reconstruction

IEKF iterative part

$i = 1,2,\ldots l$

5. Measurement equation Jacobian recalculation: $H_x\left(\underline{\eta}_i\right) = \frac{\partial}{\partial \underline{x}} h\left(\underline{\eta}_i, \underline{u}(t), t\right)$

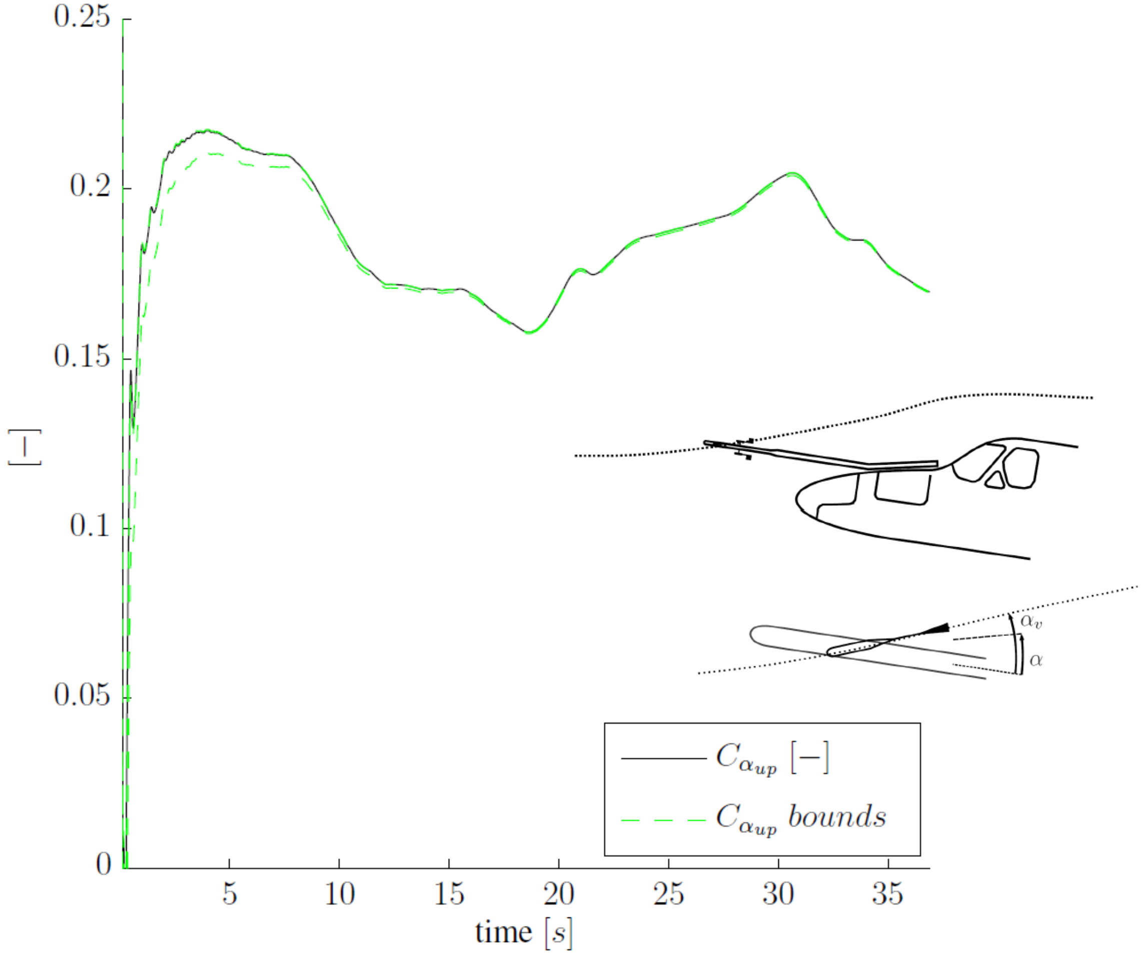
6. Kalman gain recalculation: $K_{k+1}\left(\underline{\eta}_i\right) = P_{k+1,k}(\bullet) H_x^T\left(\underline{\eta}_i\right)\left[H_x\left(\underline{\eta}_i\right) P_{k+1,k}(\bullet) H_x^T\left(\underline{\eta}_i\right) + R_{k+1}\right]^{-1}$

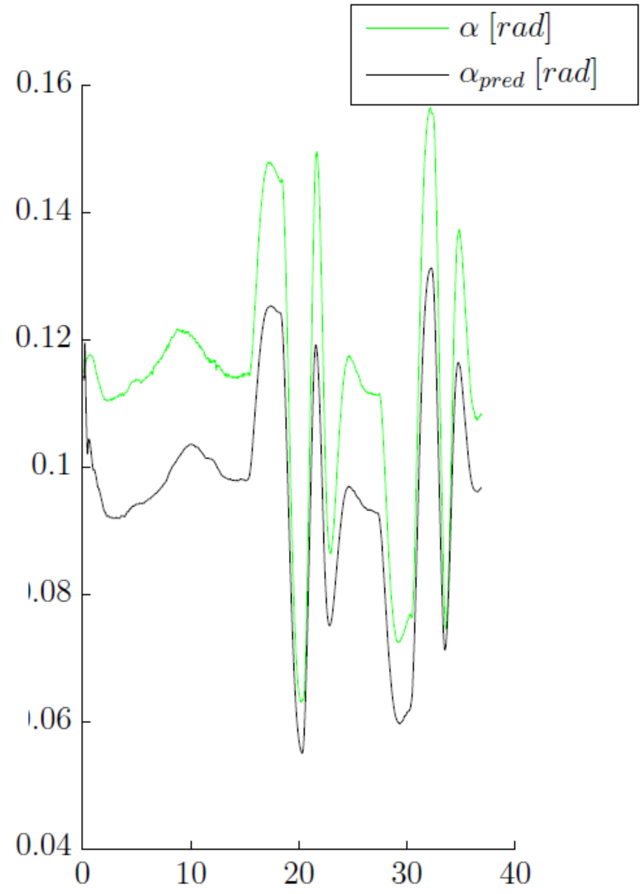
7. Measurement update, update state estimate : $\underline{\eta}_{i+1} = \underline{\hat{x}}_{k+1,k} + K_{k+1}\left(\underline{\eta}_i\right)\left(\underline{z}_{k+1} - \underline{h}\left(\underline{\eta}_i, \underline{u}_{k+1}\right) - H_x\left(\underline{\eta}_i\right)\left(\underline{\hat{x}}_{k+1,k} - \underline{\eta}_i\right)\right)$

$\underline{\hat{x}}_{k+1,k+1} = \underline{\eta}_l$

Citation II AMI: Flight path reconstruction

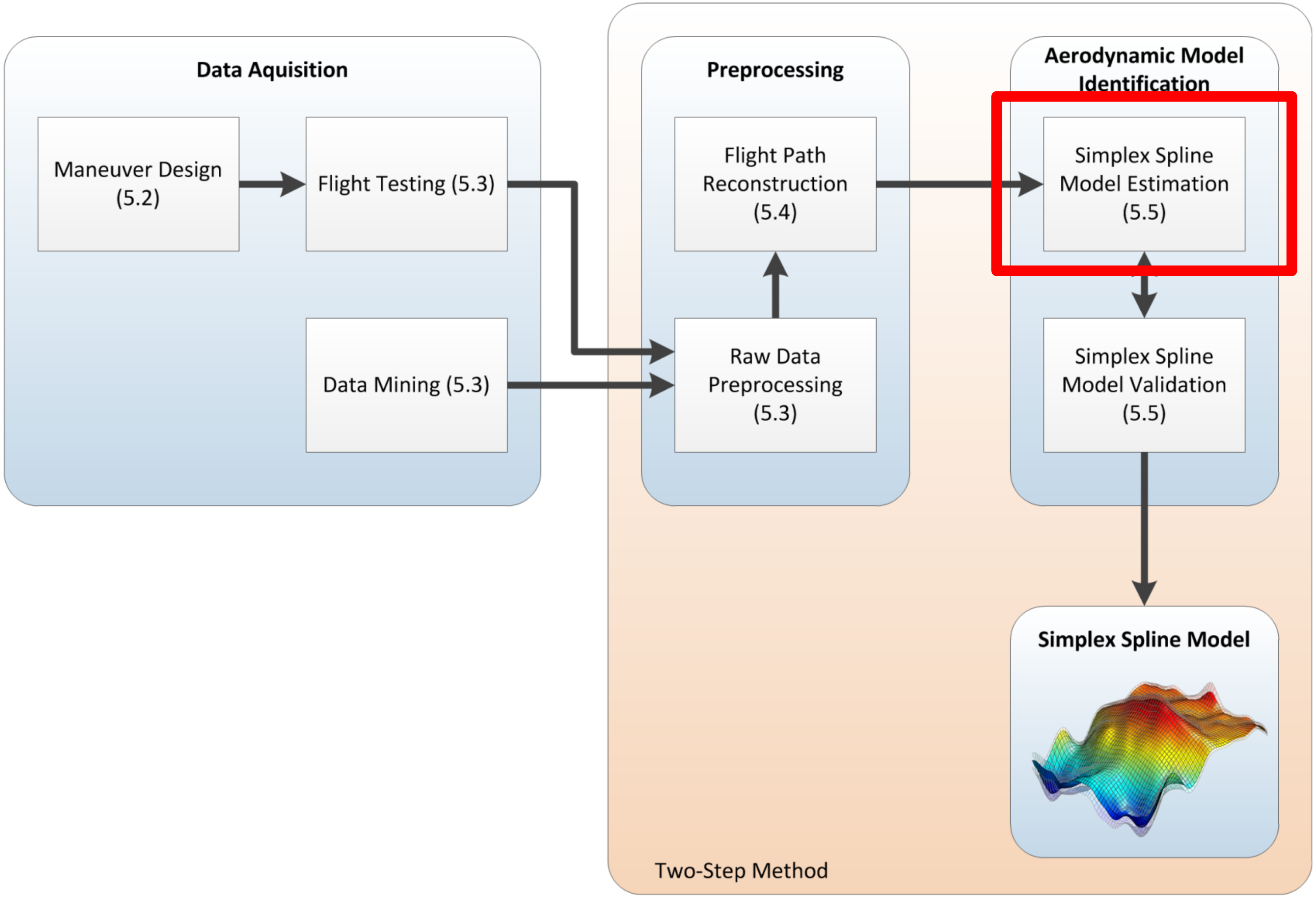
Example: estimation of upwash coefficient





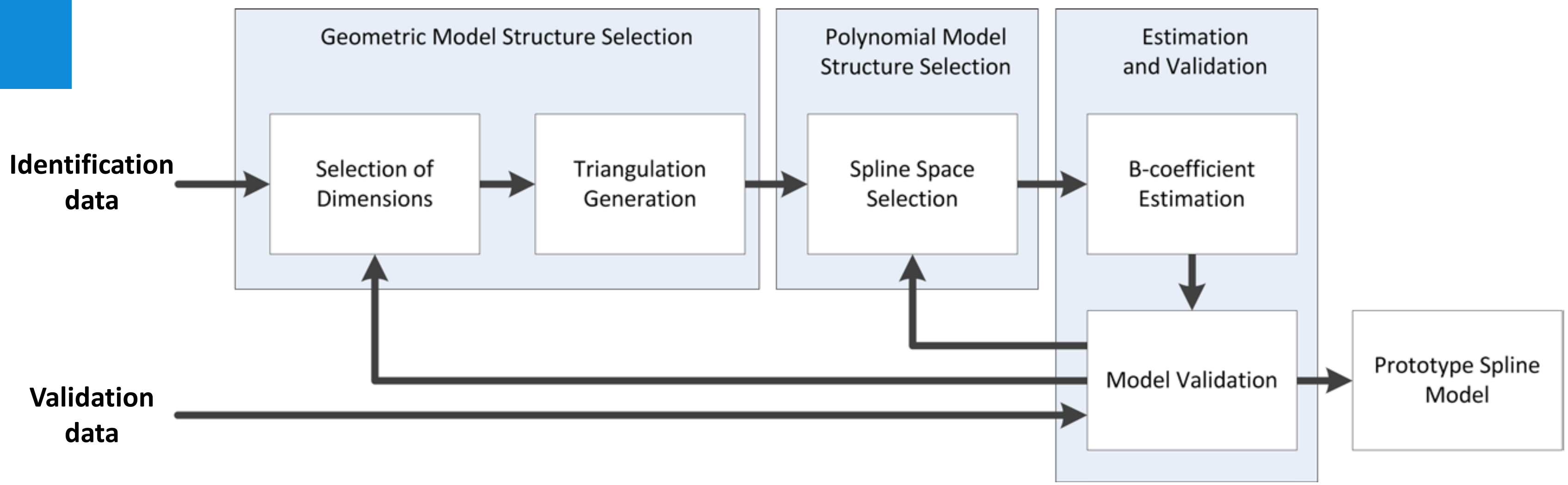
Citation II AMI: Flight path reconstruction

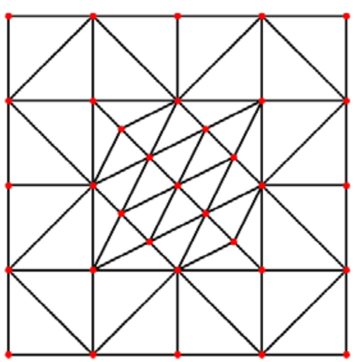
The procedure for AMI

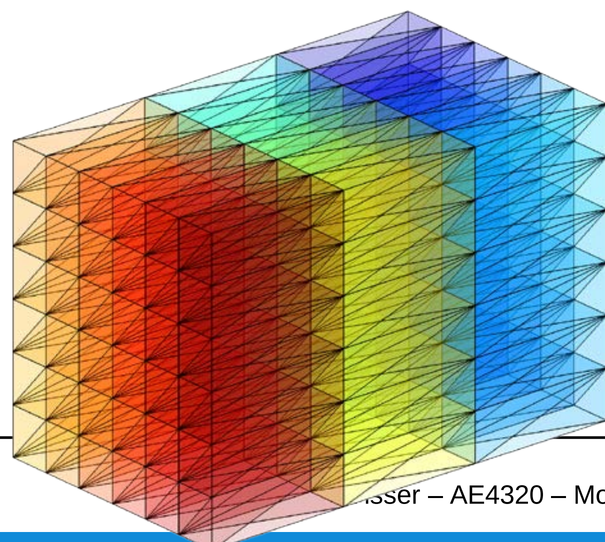


Citation II AMI: Model structure selection

Simplex Spline Model Estimation







$$p(x,c) = B^d(x) \cdot c$$

$$\hat{c}_{OLS} = \arg\min \left[\frac{1}{2} (Y - B \cdot c)^T (Y - B \cdot c) \right]$$

$$\text{subject to} \quad H \cdot c = 0$$

Citation II AMI: Model structure selection

Model structure selection procedure for modeling C_m

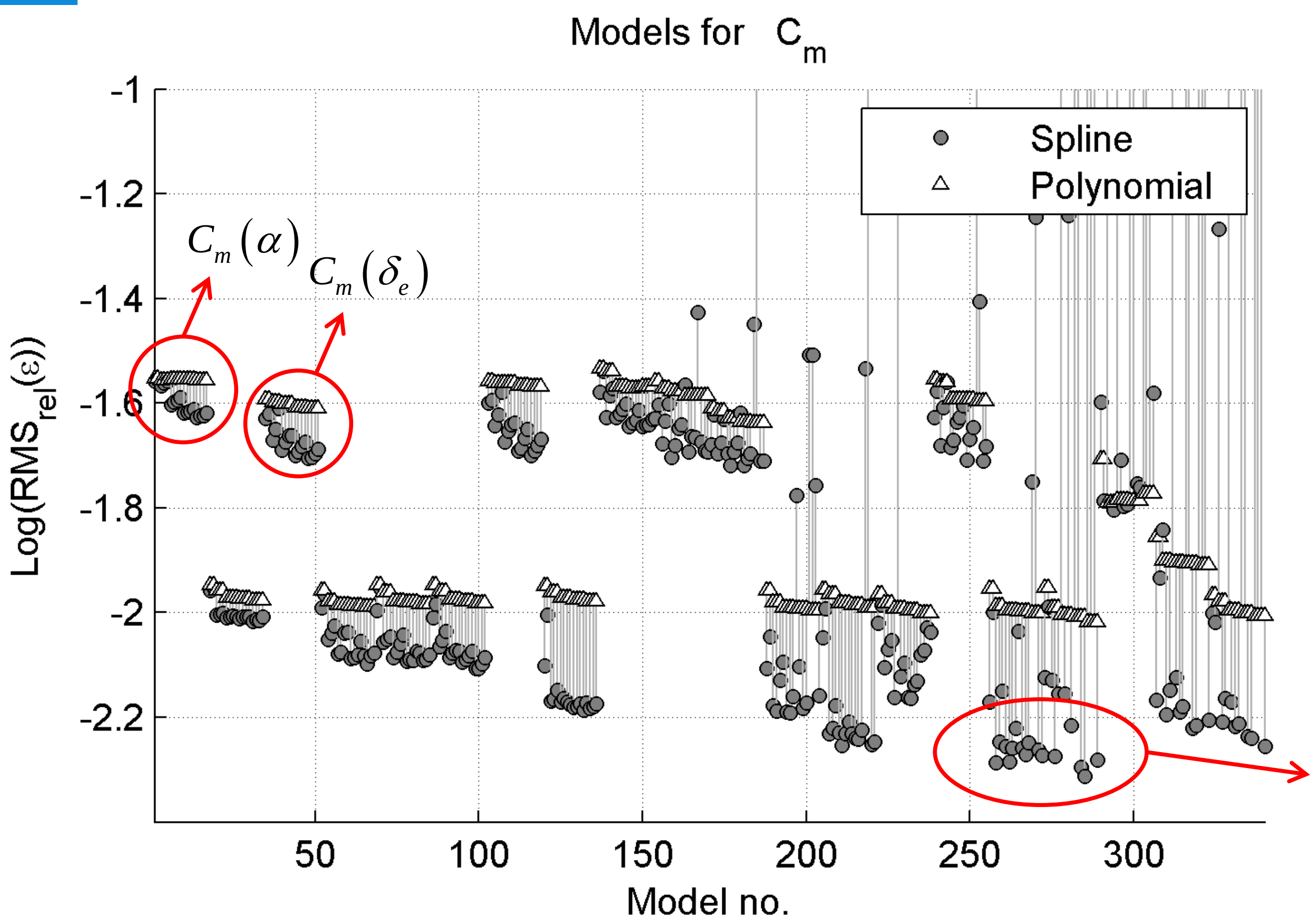
Model structure selection procedure

1. Based on expert knowledge, define sets of candidate dimensions
2. Create spline functions for each set of candidate dimensions
3. Calculate performance and select optimal candidate dimension set

	C_m
1	δ_e
2	(α,δ_e), $(\hat{q},\delta_e)$
3	$(\alpha,\hat{q},\delta_e)$, (α,M,δ_e), (α,β,δ_e), (α,δ_e,T_e), (α,M,δ_e), $(\hat{q},\delta_e,T_e)$, $(\alpha,\dot{\alpha},\delta_e)$, $(\dot{\alpha},\delta_e,T_e)$
4	$(\alpha,\hat{q},\delta_e,T_e)$, $(\alpha,\hat{q},M,\delta_e)$, $(\alpha,\dot{\alpha},\delta_e,T_e)$, $(\alpha,\dot{\alpha},M,\delta_e)$, (α,M,δ_e,T_e), $(\alpha,\beta,M,\delta_e)$, $(\alpha,\beta,\hat{q},\delta_e)$, $(\alpha,\beta,\dot{\alpha},\delta_e)$
5	$(\alpha,\beta,\hat{q},\delta_e,T_e)$, $(\alpha,\beta,\hat{q},M,\delta_e)$, $(\alpha,\beta,\dot{\alpha},\delta_e,T_e)$, $(\alpha,\beta,\dot{\alpha},M,\delta_e)$, $(\alpha,\hat{q},M,\delta_e,T_e)$, $(\alpha,\dot{\alpha},M,\delta_e,T_e)$

Citation II AMI: Model structure selection

Prototype spline models for pitching moment coefficient C_m



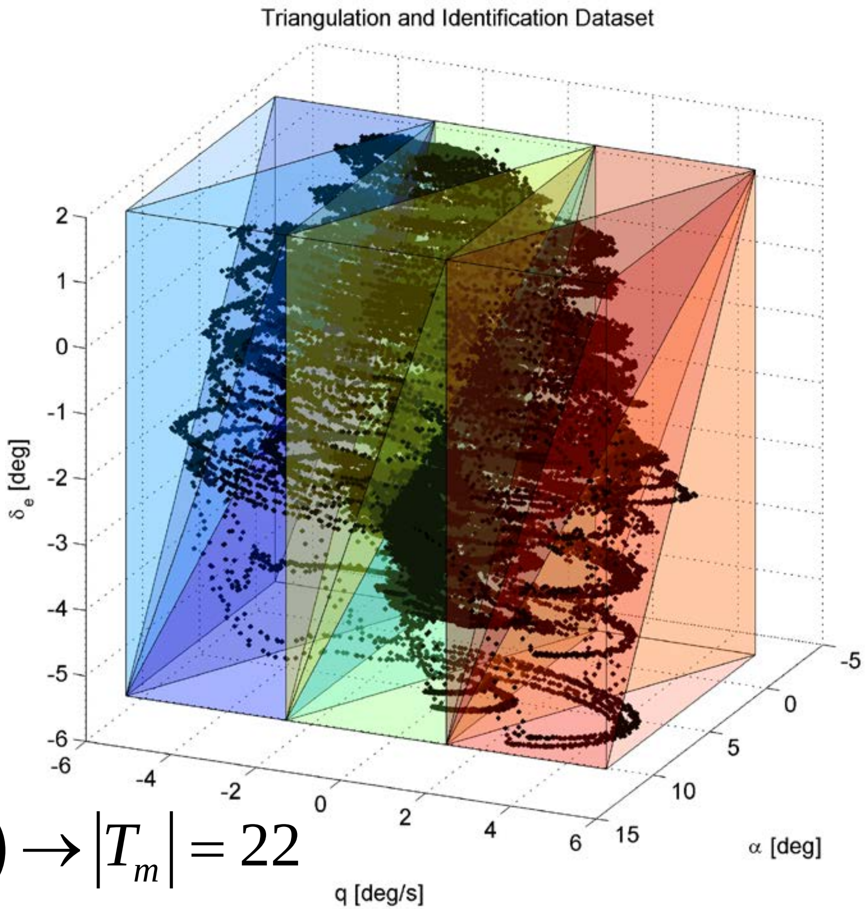
r	Spline spaces
-1	$\mathcal{S}_1^{-1}, \mathcal{S}_2^{-1}, \mathcal{S}_3^{-1}, \mathcal{S}_4^{-1}, \mathcal{S}_5^{-1}$
0	$\mathcal{S}_1^{0}, \mathcal{S}_2^{0}, \mathcal{S}_3^{0}, \mathcal{S}_4^{0}, \mathcal{S}_5^{0}$
1	$\mathcal{S}_2^{1}, \mathcal{S}_3^{1}, \mathcal{S}_4^{1}, \mathcal{S}_5^{1}$

Citation II AMI: Model structure selection

Spline model for pitching moment coefficient C_m

Selected model structure:

1. Splines defined on candidate dimension set (α, M, q, δ_e) produced the best results.
2. Spline function defined on 22 simplices (pentachorons) produced the best results (not pictured!).
3. Spline function of degree 4 and continuity order 2 produced the best results after model validation.

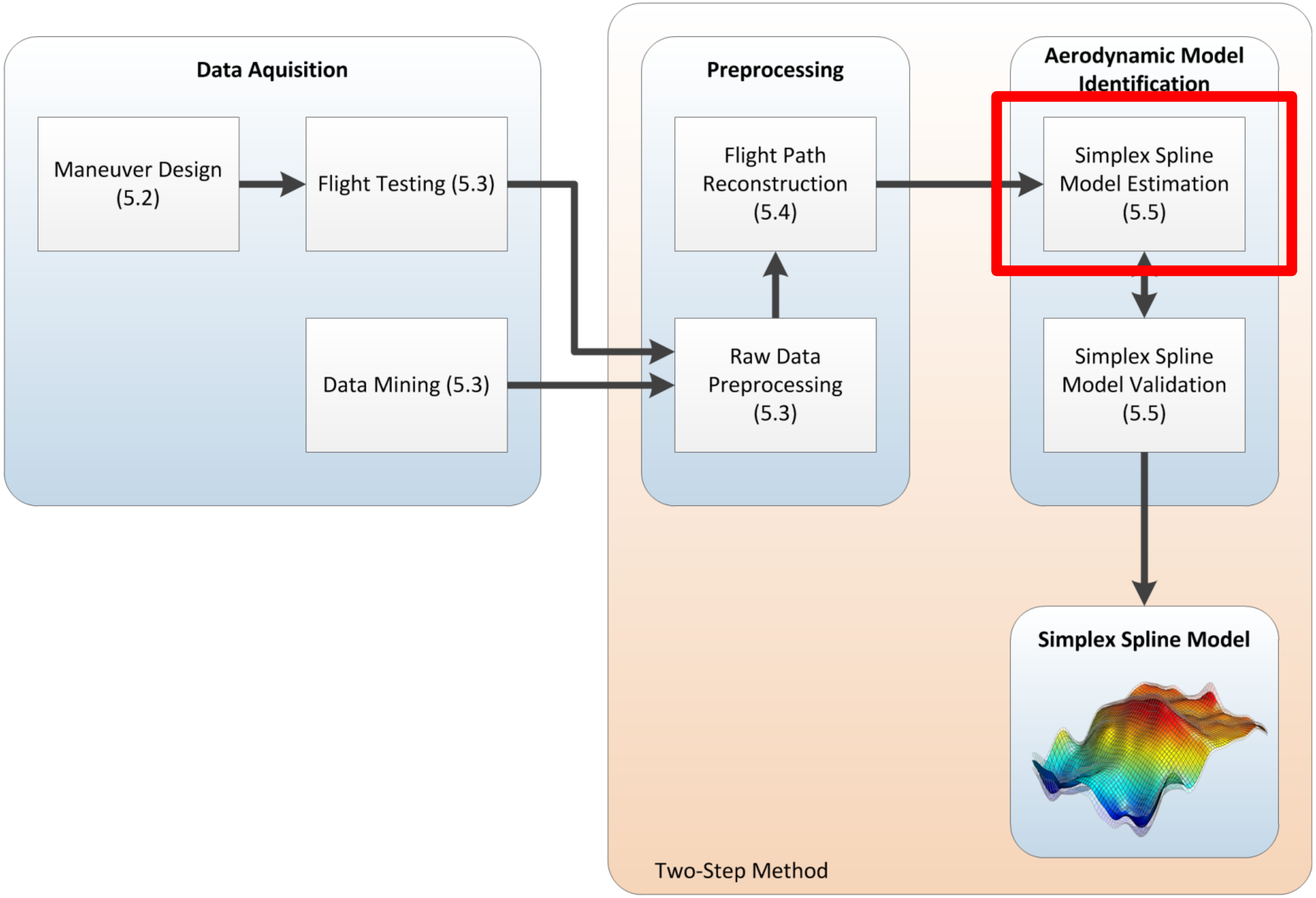


Final model structure selection:

$$s(\alpha, M, q, \delta_e) = B^4(x) \cdot c \in S_4^2(T_m) \rightarrow |T_m| = 22$$

Citation II AMI: Parameter estimation

The procedure for AMI



Citation II AMI: Parameter estimation

Equality constrained OLS estimation

The OLS cost function using the global regression matrix B and the global vector of B-coefficients c are defined for $B^4(x) \cdot c \in S_4^2(T_m)$:

$$J(c) = \frac{1}{2}\left(Y - B \cdot c\right)^T \left(Y - B \cdot c\right)$$

In this case we want to enforce 2nd order smoothness between spline pieces using the smoothness constraints $H \cdot c = 0$. Note that H is rank-deficient (i.e. $\mathrm{rank}\, H < \mathrm{row}\, H$) in general!

The **constrained** cost function minimum can be found as follows:

$$\hat{c} = \arg\min \left[\frac{1}{2}\left(Y - B \cdot c\right)^T \left(Y - B \cdot c\right) \right], \quad \text{subject to } \; H \cdot c = 0$$

This equality constrained optimization problem can be solved using **Lagrange Multiplier methods**.

Citation II AMI: Parameter estimation

Equality constrained OLS estimation

The OLS B-coefficient / Lagrangian estimator with Moore-Penrose pseudo-inverse is:

$$\begin{bmatrix} \hat{c} \\ \hat{\lambda} \end{bmatrix} = \begin{bmatrix} B^T \cdot B & H^T \\ H & 0 \end{bmatrix}^+ \cdot \begin{bmatrix} B^T \cdot Y \\ 0 \end{bmatrix}$$

This is equivalent to:

$$\begin{bmatrix} \hat{c} \\ \hat{\lambda} \end{bmatrix} = \begin{bmatrix} C_1 & C_2 \\ C_3 & C_4 \end{bmatrix} \cdot \begin{bmatrix} B^T \cdot Y \\ 0 \end{bmatrix}$$

The constrained OLS B-coefficient estimator is:

$$\hat{c} = C_1 \cdot B^T \cdot Y$$

With statistics:

$$Cov\{\hat{c}\} = C_1, \qquad Var\{\hat{c}\} = diag(C_1)$$

Citation II AMI: Parameter estimation

Equality constrained OLS estimation

We may also use a **much more efficient iterative solver**:

First Iteration: $$\hat{c}^{(1)} = \left(2B^T \cdot B + \frac{1}{\varepsilon} H^T \cdot H\right)^{-1} \cdot \left(2B^T \cdot Y - H^T \cdot \hat{\lambda}^{(0)}\right), \qquad 0 < \varepsilon \leq 1$$

Next Iteration: $$\hat{c}^{(k+1)} = \left(2B^T \cdot B + \frac{1}{\varepsilon} H^T \cdot H\right)^{-1} \cdot 2B^T \cdot B \cdot \hat{c}^{(k)}, \qquad 0 < \varepsilon \leq 1$$

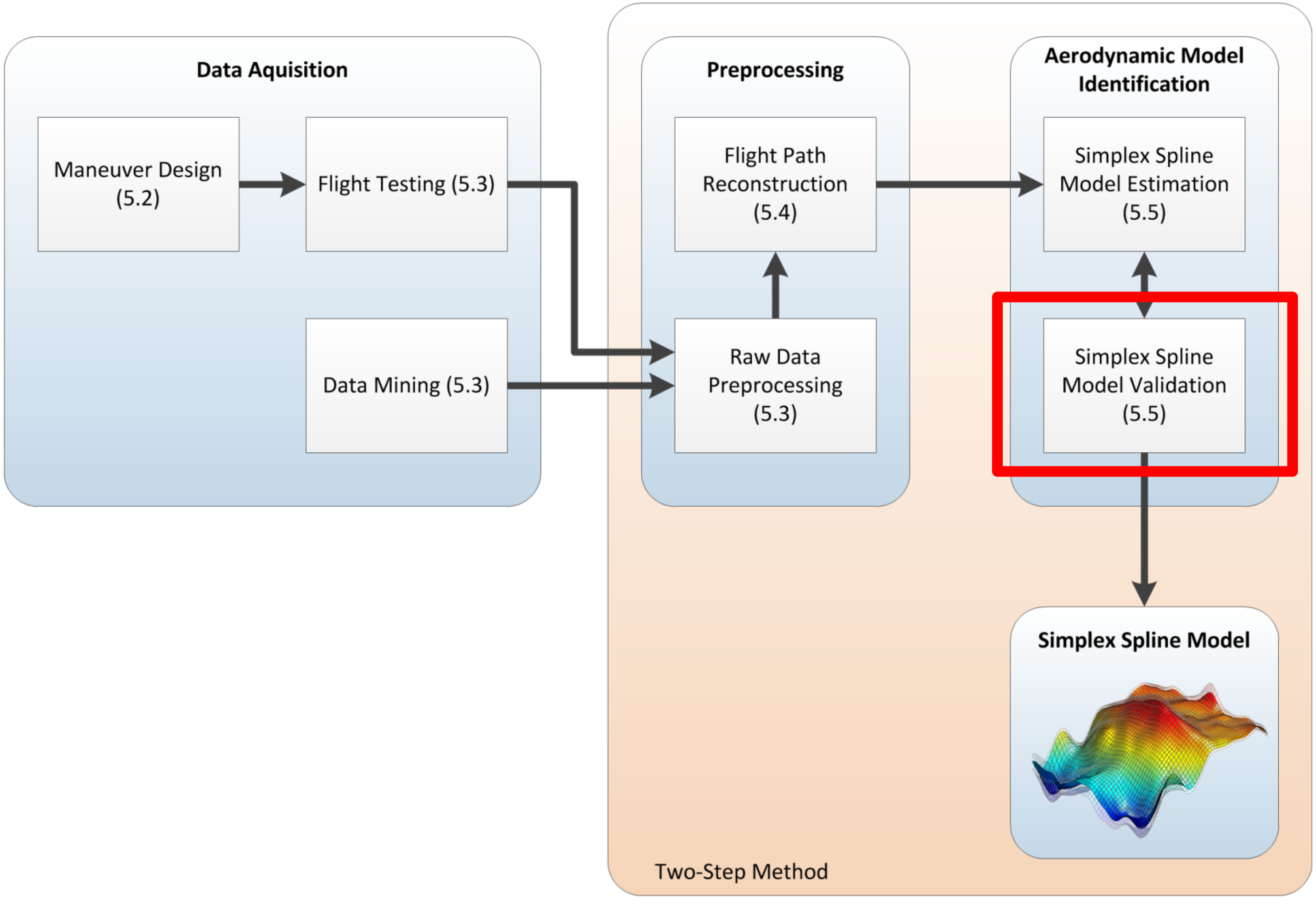
With ε a small number (e.g. 10^{-6}), and with $\hat{\lambda}^{(0)}$ an initial estimate for the Lagrange multipliers (e.g. $\hat{\lambda}^{(0)} = 0$).

The B-coefficient covariance matrix is approximated by:

$$Cov(\hat{c}) \approx \left(2B^T \cdot B + \frac{1}{\varepsilon} H^T \cdot H\right)^{-1}$$

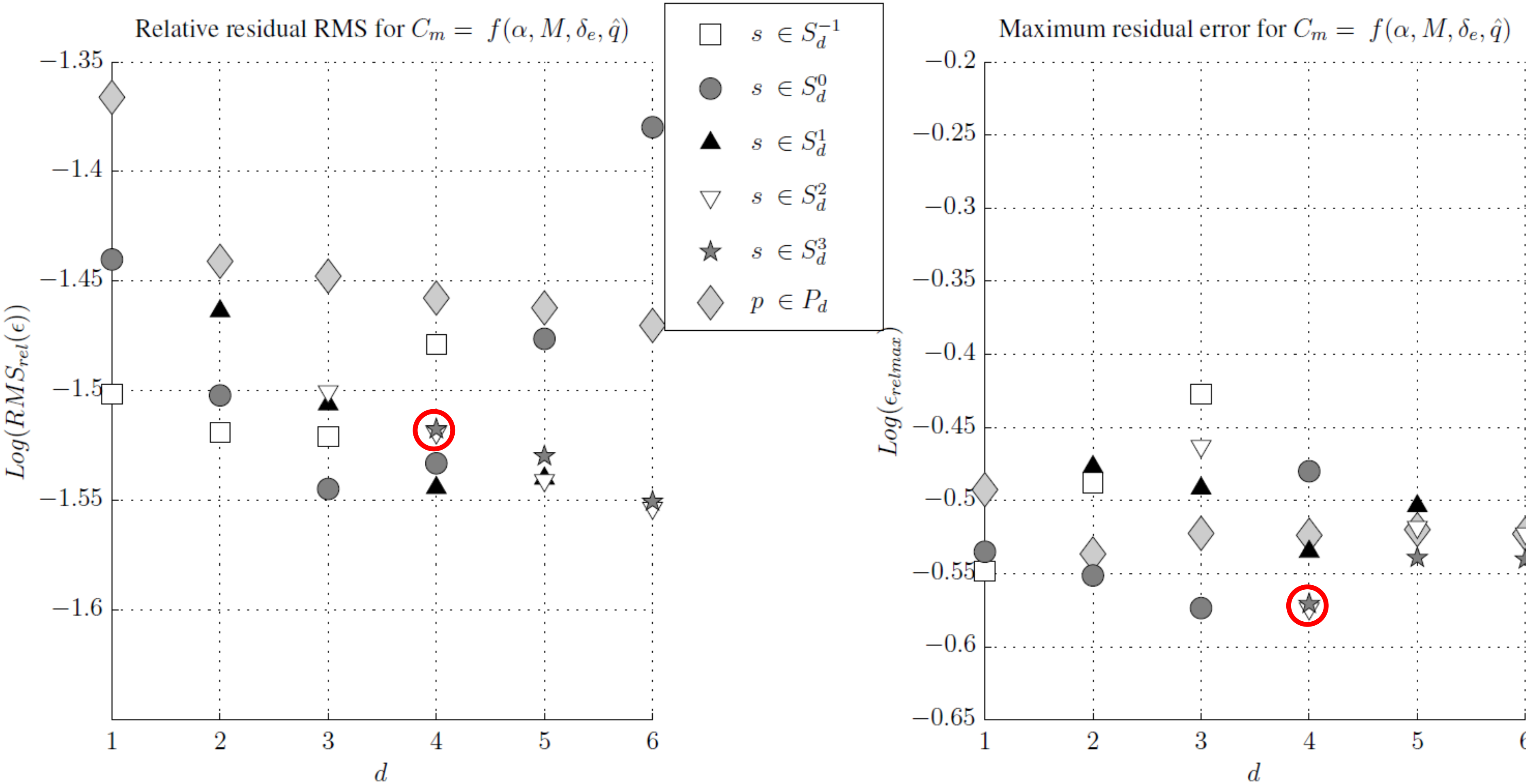
Citation II AMI: Flight path reconstruction

The procedure for AMI



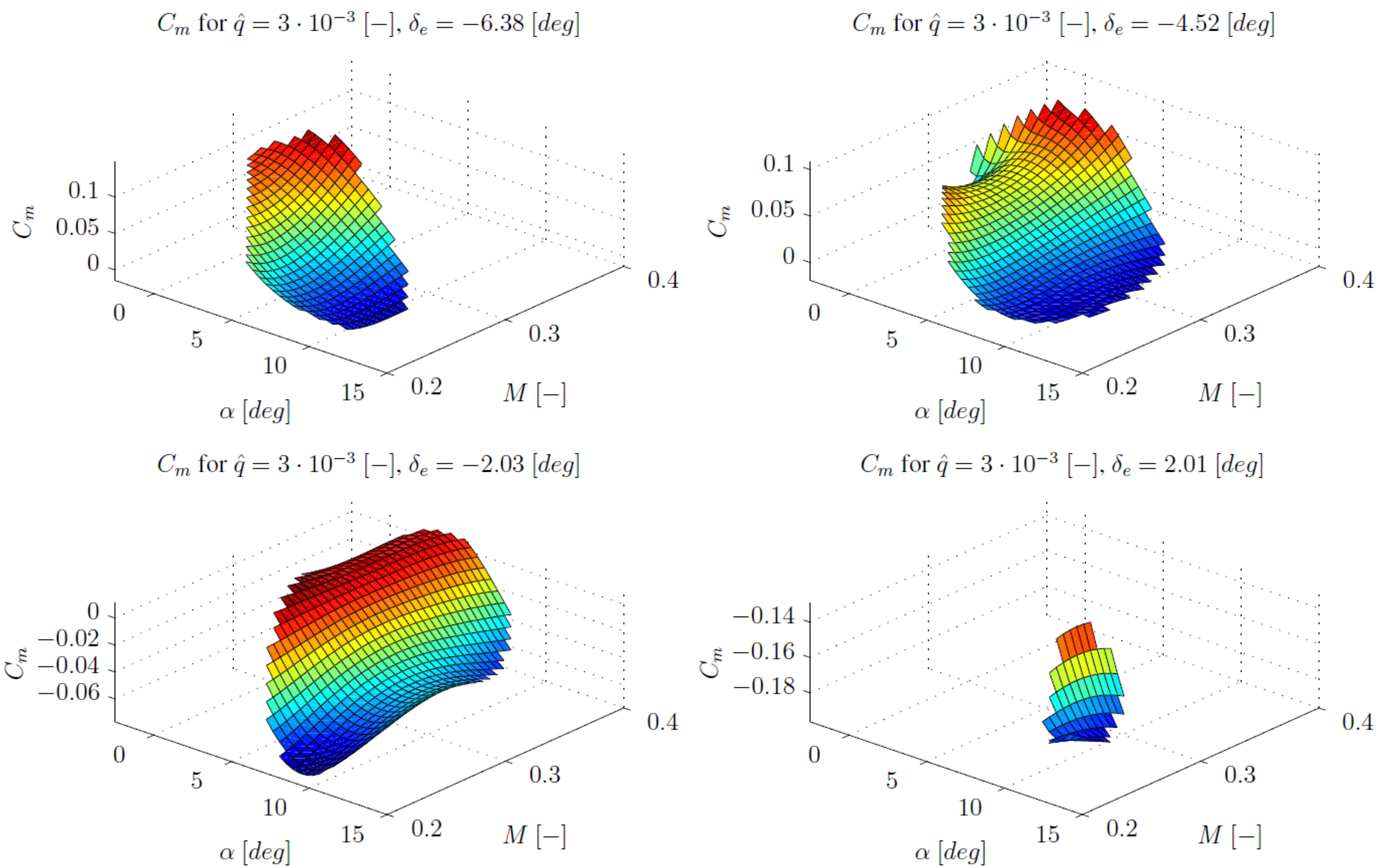
Citation II Aero-Model Validation: C_m

Spline model for Pitching Moment Coefficient

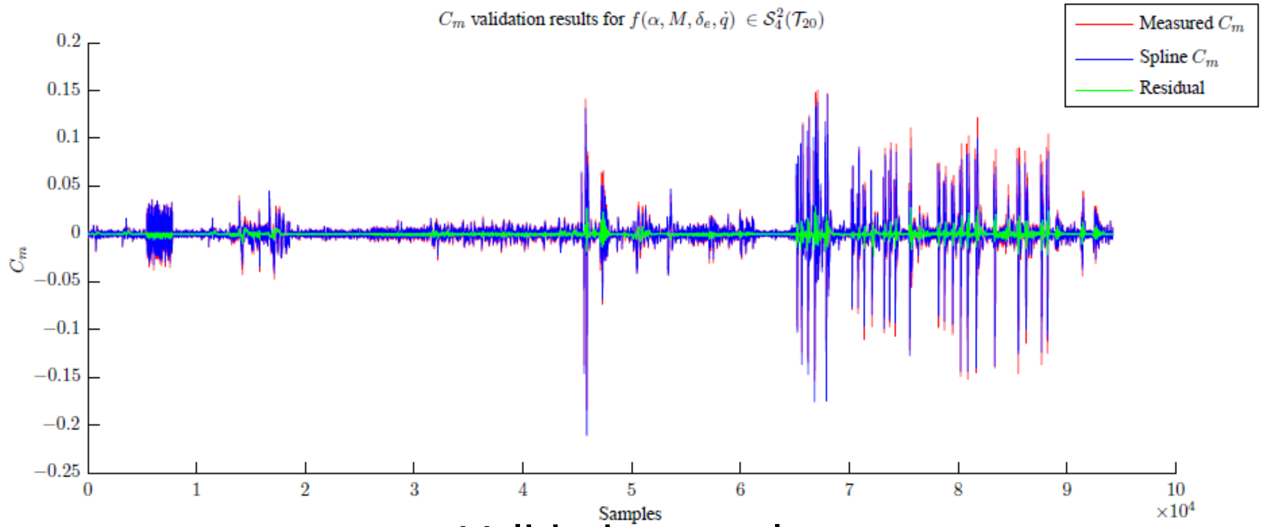


Citation II Aero-Model Validation: C_m

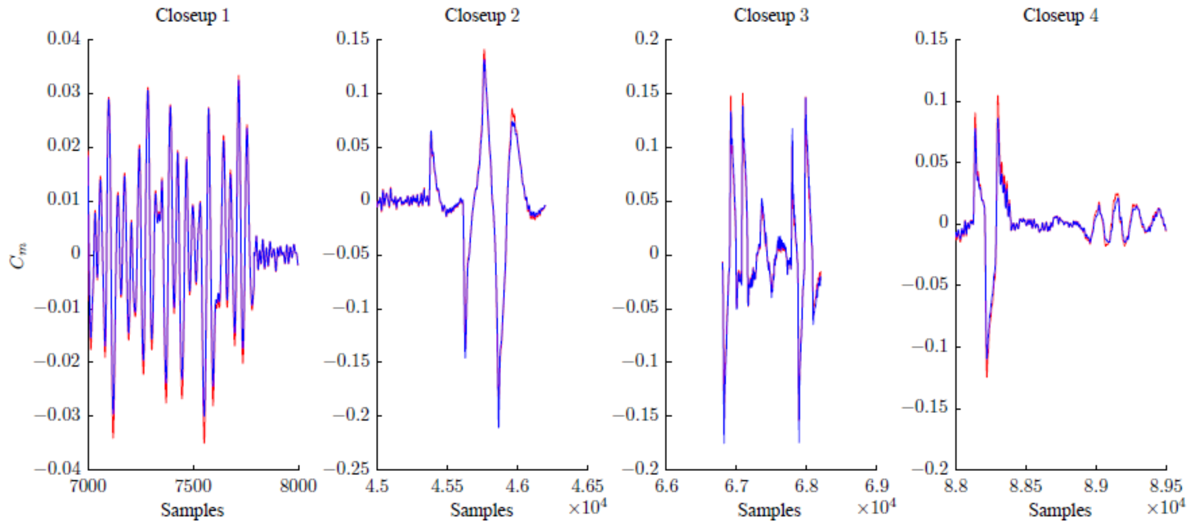
Spline model for Pitching Moment Coefficient (slices from 4D model)



Citation II Aero-Model Validation: C_m



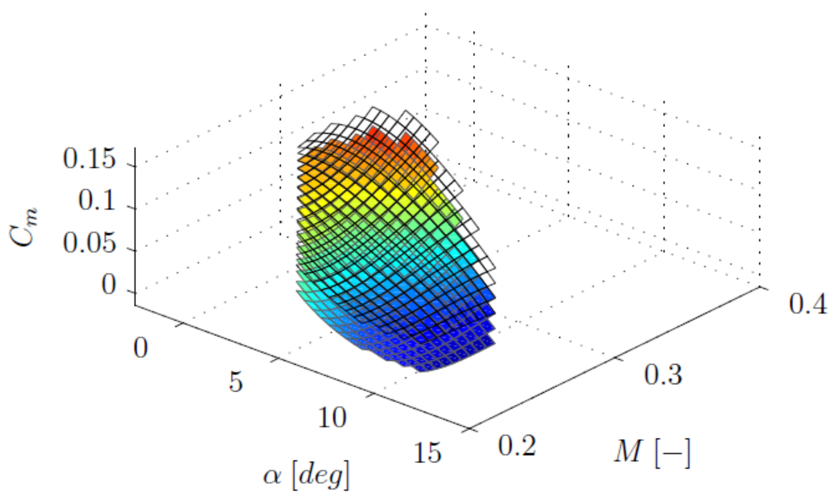
Validation results



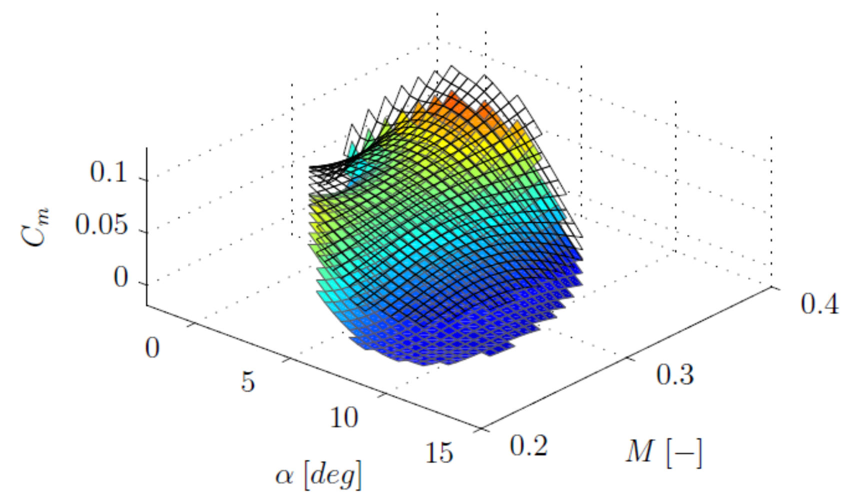
Citation II Aero-Model Validation: C_m

Spline model with 97% confidence bounds from model residuals

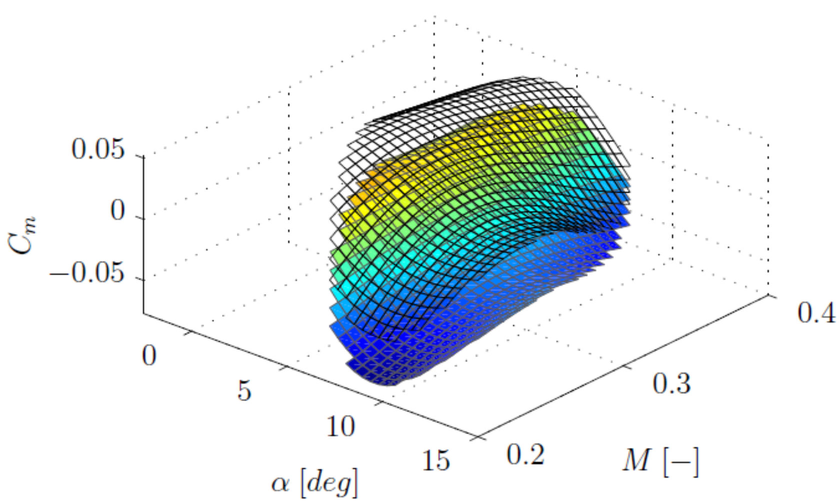
6σ Conf. Model for C_m for $\hat{q} = 3 \cdot 10^{-3}$ $[-]$, $\delta_e = -6.38$ $[deg]$



6σ Conf. Model for C_m for $\hat{q} = 3 \cdot 10^{-3}$ $[-]$, $\delta_e = -4.52$ $[deg]$



6σ Conf. Model for C_m for $\hat{q} = 3 \cdot 10^{-3}$ $[-]$, $\delta_e = -2.03$ $[deg]$



confidence companion spline model:

$$s_c = \sum_{|\kappa|=d} \tau_\kappa^{t_j} B_\kappa^d(\mathbf{b}), \;\; j = 1, 2, \ldots, J,$$

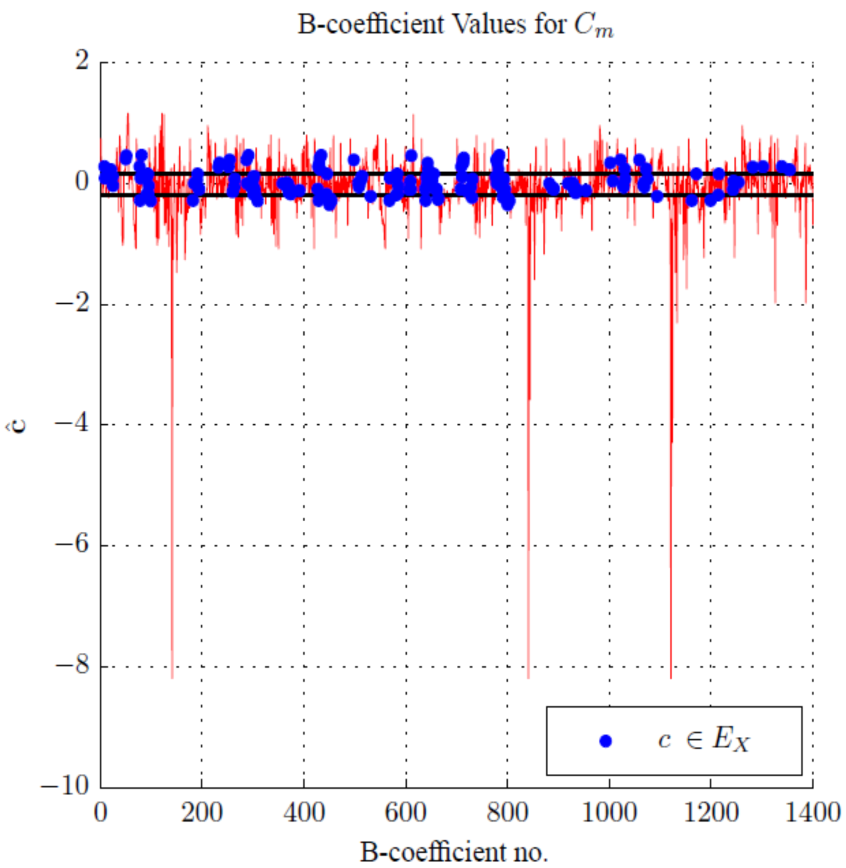
with coefficients:

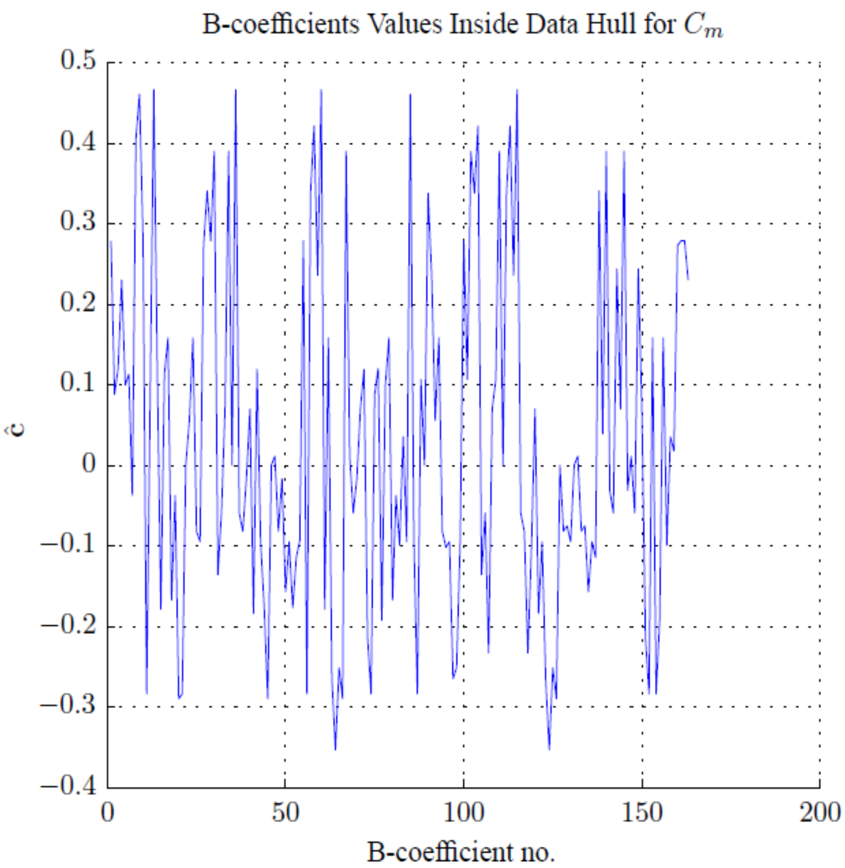
$$\tau_\kappa^{t_j} = t_{\frac{\alpha}{2p}, N-p} \sqrt{Var(\hat{c}_\kappa^{t_j})}$$

with $t_{\frac{\alpha}{2p},N-p}$ the student-T the Student's t distribution with N − p degrees of freedom, significance level $\frac{\alpha}{2p}$, and with p = J$\hat{d}$.

Citation II Aero-Model Validation: C_m

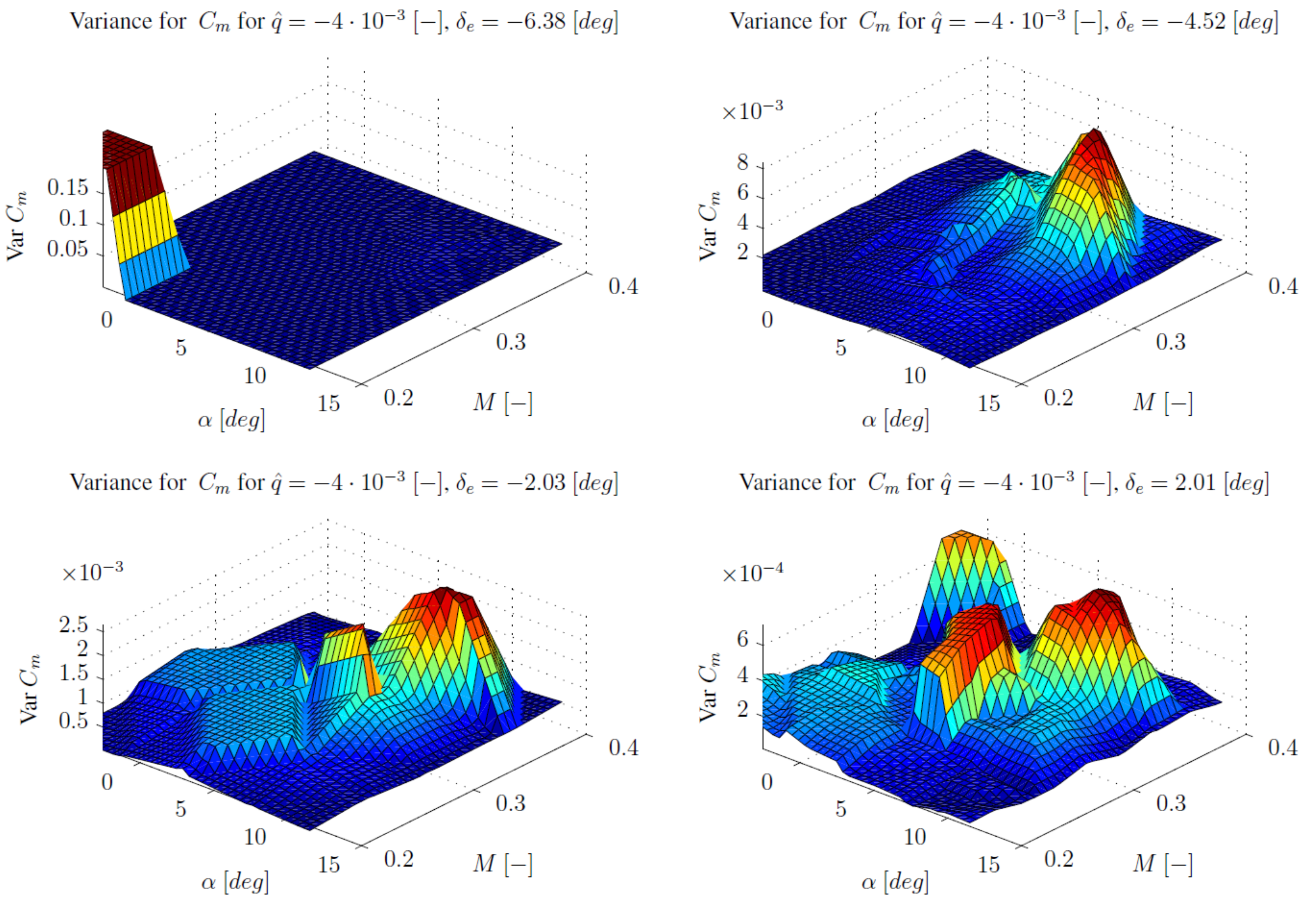
Stability analysis B-coefficient bounds





Citation II Aero-Model Validation: C_m

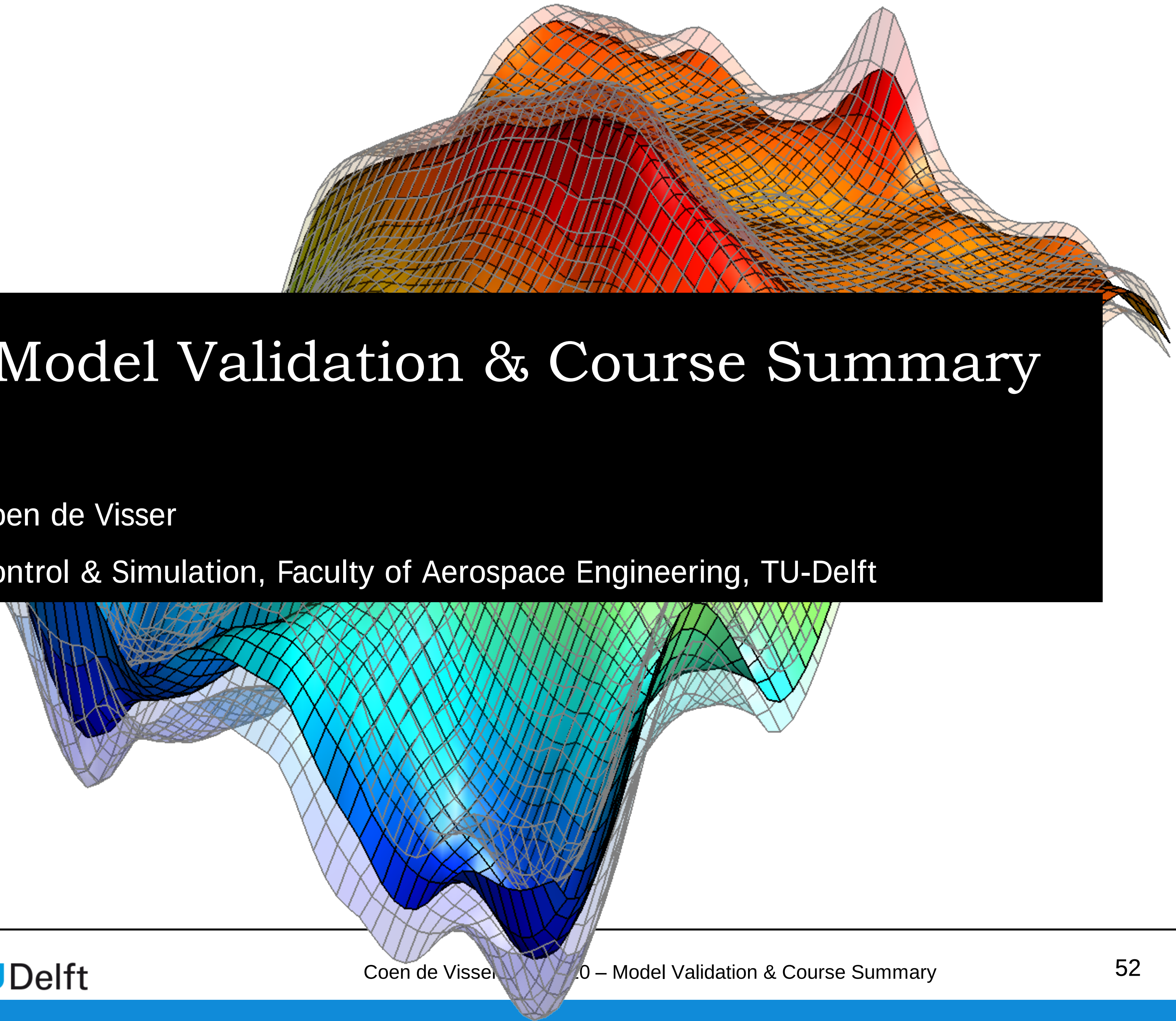
Variance surfaces of spline model



Citation II Aero-Model Validation: C_m

Final results for all spline models:

	C_X	C_Y	C_Z	C_l	C_m	C_n
Dimensions	(α, M, δ_e)	$(\beta, \delta_r, \hat{r})$	$(\alpha, M, \delta_e, \hat{q})$	$(\beta, \delta_a, \dot{p})$	$(\alpha, M, \delta_e, \hat{q})$	$(M, \beta, \delta_r, \hat{r})$
Triangulation	$\mathcal{T}_6$	$\mathcal{T}_6$	$\mathcal{T}_{22}$	$\mathcal{T}_6$	$\mathcal{T}_{22}$	$\mathcal{T}_{22}$
Spline Space	$\mathcal{S}_4^2$	$\mathcal{S}_3^1$	$\mathcal{S}_4^2$	$\mathcal{S}_4^2$	$\mathcal{S}_4^2$	$\mathcal{S}_4^2$
$\text{RMS}_{rel}\ \epsilon$	4.79%	7.18%	3.6%	1.73%	3.03%	2.55%
ϵ_{relmax}	30.61%	37.91%	20.90%	22.32%	26.68%	28.76%
R^2	87.07%	41.59%	90.78%	56.34%	64.44%	63.99%

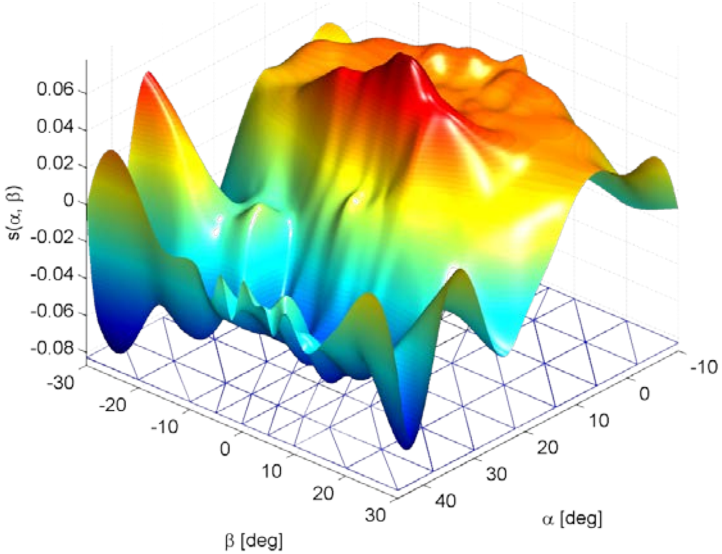


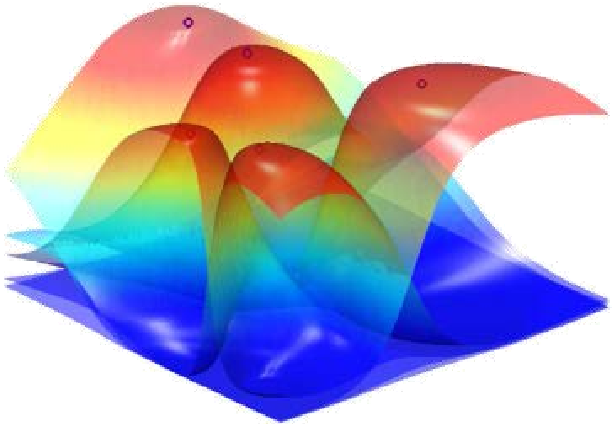
Model Validation & Course Summary

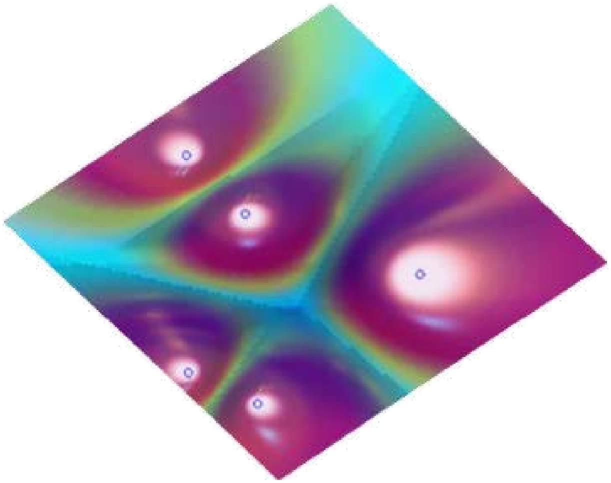
Coen de Visser

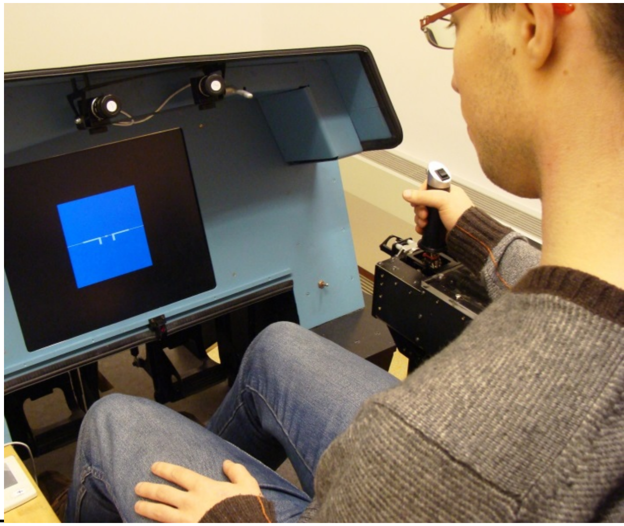
Control & Simulation, Faculty of Aerospace Engineering, TU-Delft

Description of Assignments









Assignment 1: Kalman Filter

Assignment:

This assignment concerns the aircraft aerodynamic model identification using flight test data with the two-step approach learnt in the lecture.

Accurate estimation of aircraft position, airspeed body components and attitude with a GPS/IMU/Airdata integrated system with wind will be performed in the first step of the two-step approach applying the Extended or Iterated Extended Kalman Filters EKF/IEKF).

After this, a least squares parameter estimator for a standard polynomial model will be developed.

Finally, the estimated model will be validated using a subset of the test data.

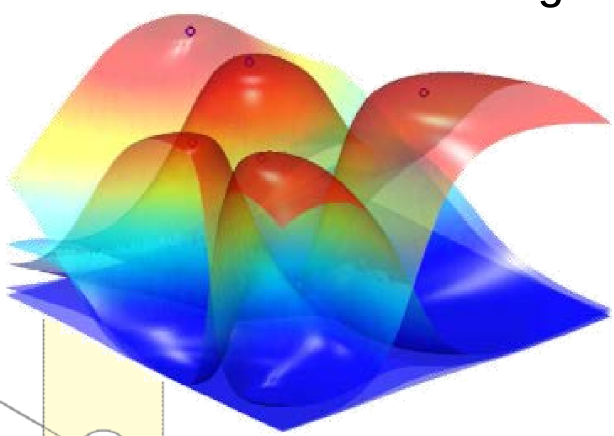
Assignment 2: Neural Networks

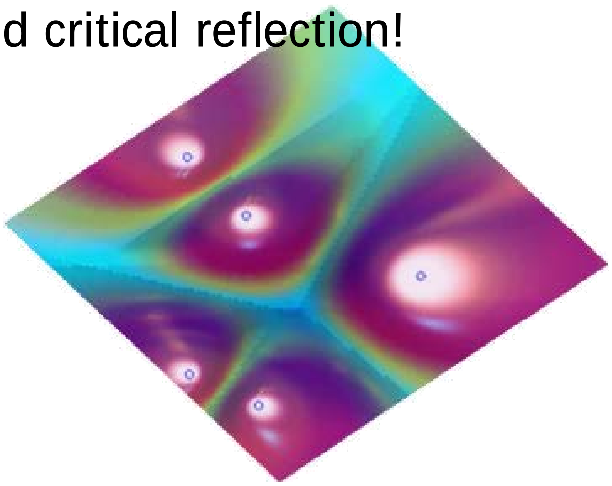
Assignment:

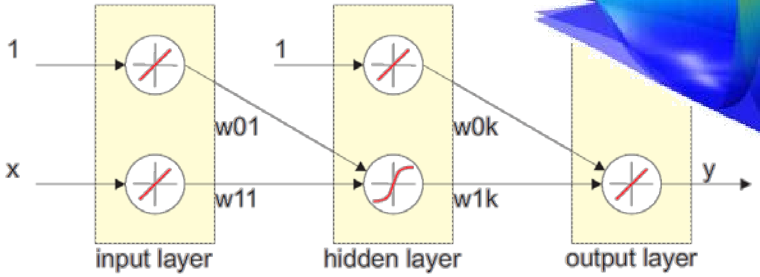
This assignment concerns the creation of a neural network model using a highly nonlinear and noisy wind tunnel dataset of the F-16.

Both feed-forward and RBF networks are used to model the dataset.

Students will be rewarded for creative thinking and critical reflection!







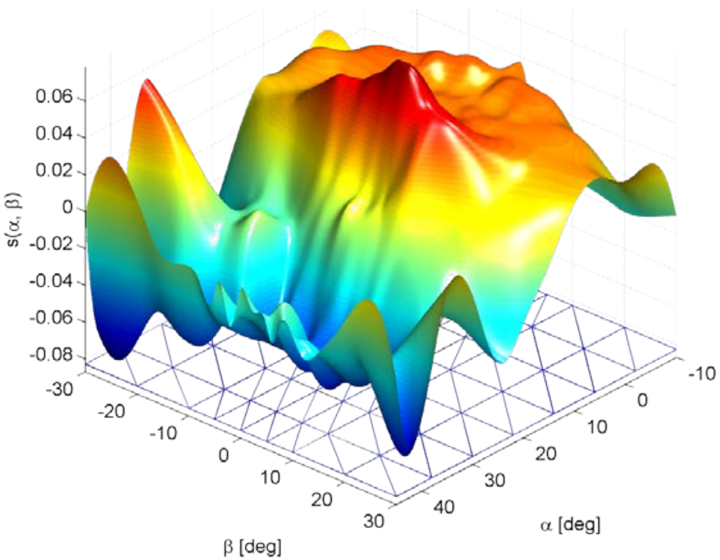
Assignment 3: Multivariate Splines

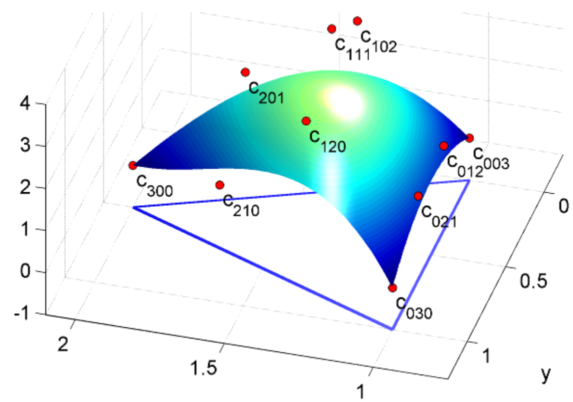
Assignment:

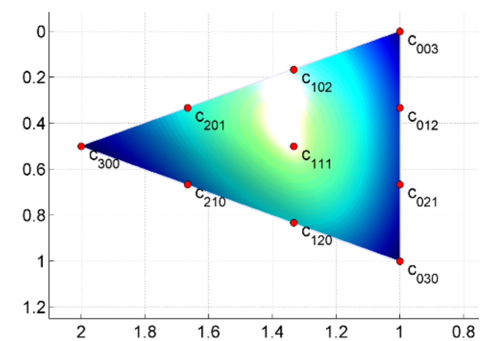
This assignment focusses on a promising new class of function approximator: the multivariate simplex B-spline.

The assignment concerns the creation of a multivariate simplex B-spline model using a highly nonlinear and noisy wind tunnel dataset of the F-16.

Students will be rewarded for creative thinking and critical reflection!







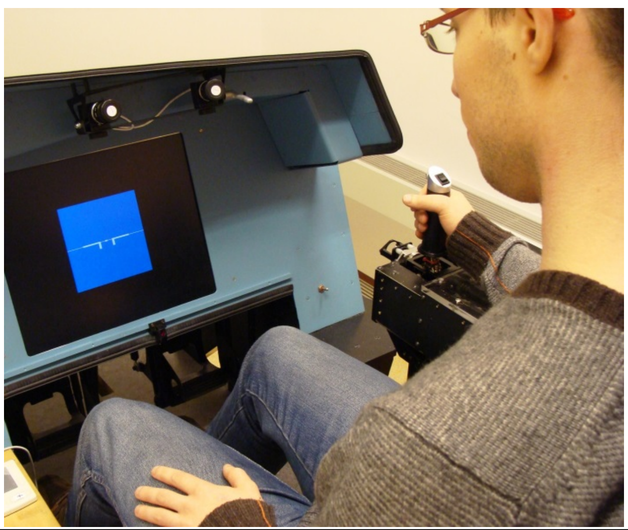
Assignment 4: Pilot Identification

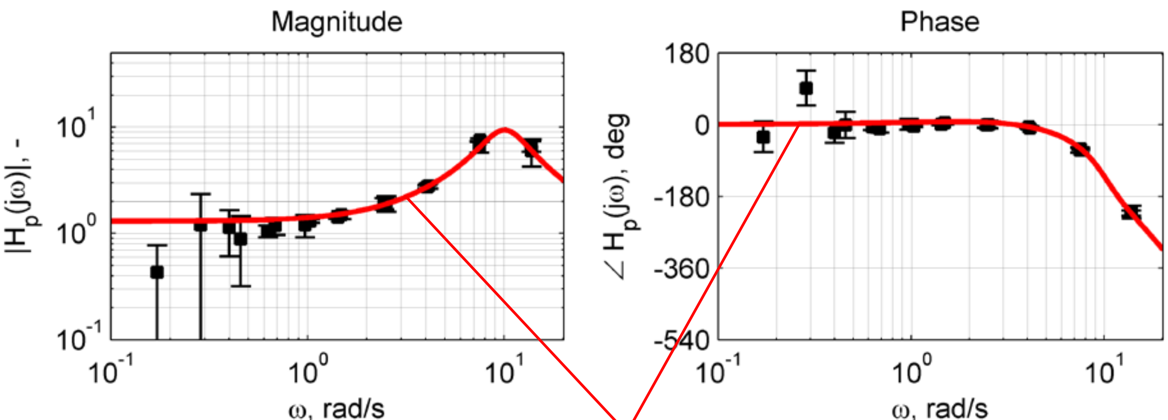
Assignment:

This assignment concerns the identification of unknown pilot dynamics using black-box system identification techniques.

You are asked to select the best possible model for describing the pilot dynamics and fit this model to your provided dataset.

Students will be rewarded for creative thinking and critical reflection on their selected model!





$$H_p(j\omega) = K_p \frac{T_L j\omega + 1}{T_I j\omega + 1} e^{-j\omega\tau_p} \frac{\omega_{nm}^2}{(j\omega)^2 + 2\xi_{nm}\omega_{nm} j\omega + \omega_{nm}^2}$$