Estimation and Learning in Aerospace Project

Model Identification and State Estimation for Multi-rotor UAV

Authors:

- Maria Alessandra De Luca
  - 10520091
- Adrian Burton Parisi
  - 10710691

Outline

- Objectives of the Project
- System Identification
  - Available Data
  - LTI Verification
  - Black box Model Identification
  - Transfer Function Order analysis
  - Estimated Model
  - Simulated vs Measured Comparison
  - Model Validation
  - Uncertainty analysis
- State Estimation
  - Problem Statement
  - Kalman Filter
  - Noise Covariance Matrices
  - Initial Values
  - Initial Results and Refinement
  - Final Results
  - Steady State Kalman Filter

Objectives

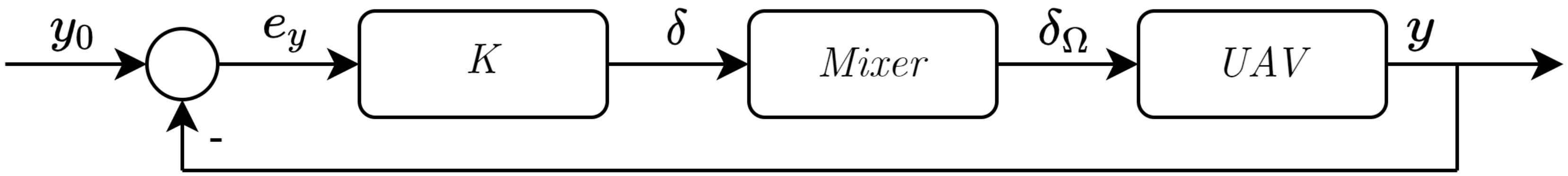
The goals for this project are two-fold:

1. System identification of closed-loop lateral dynamics
    1. Define and identify a model using given response data corresponding to 2 PRBS sequences.
    2. Assess the uncertainty of the proposed identified model
    3. Validate the model with an additional PRBS sequence

2. Lateral velocity estimation with a Kalman filter
    1. Design a Kalman filter with the previously proposed identified model
    2. Validate the performance of the designed filter with all available data

ANT-X

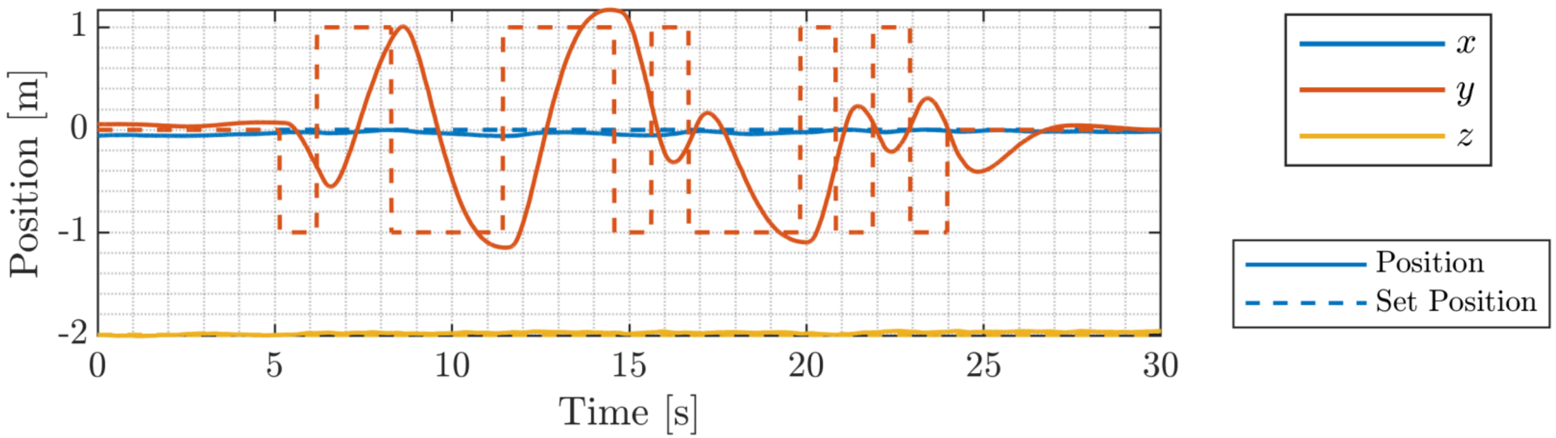
https://antx.it/

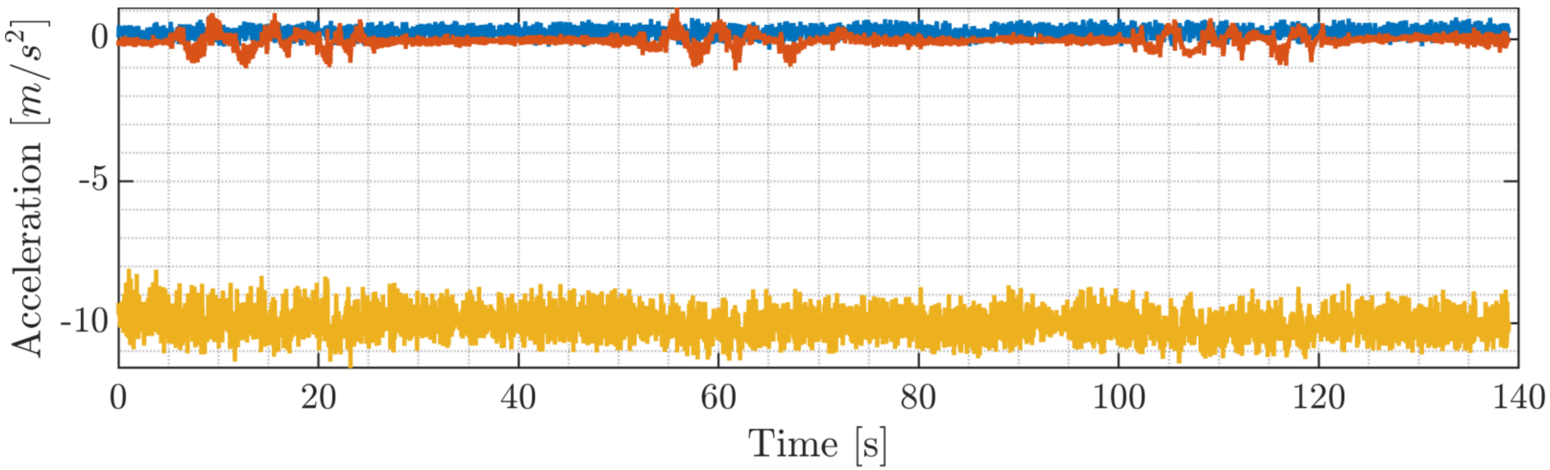


System Identification

Available Data

- NED Set Position
- NED Position
- Body Acceleration
- NED Velocity
- Time
  - 139 seconds of data
  - Sampling Frequency
    - 100Hz

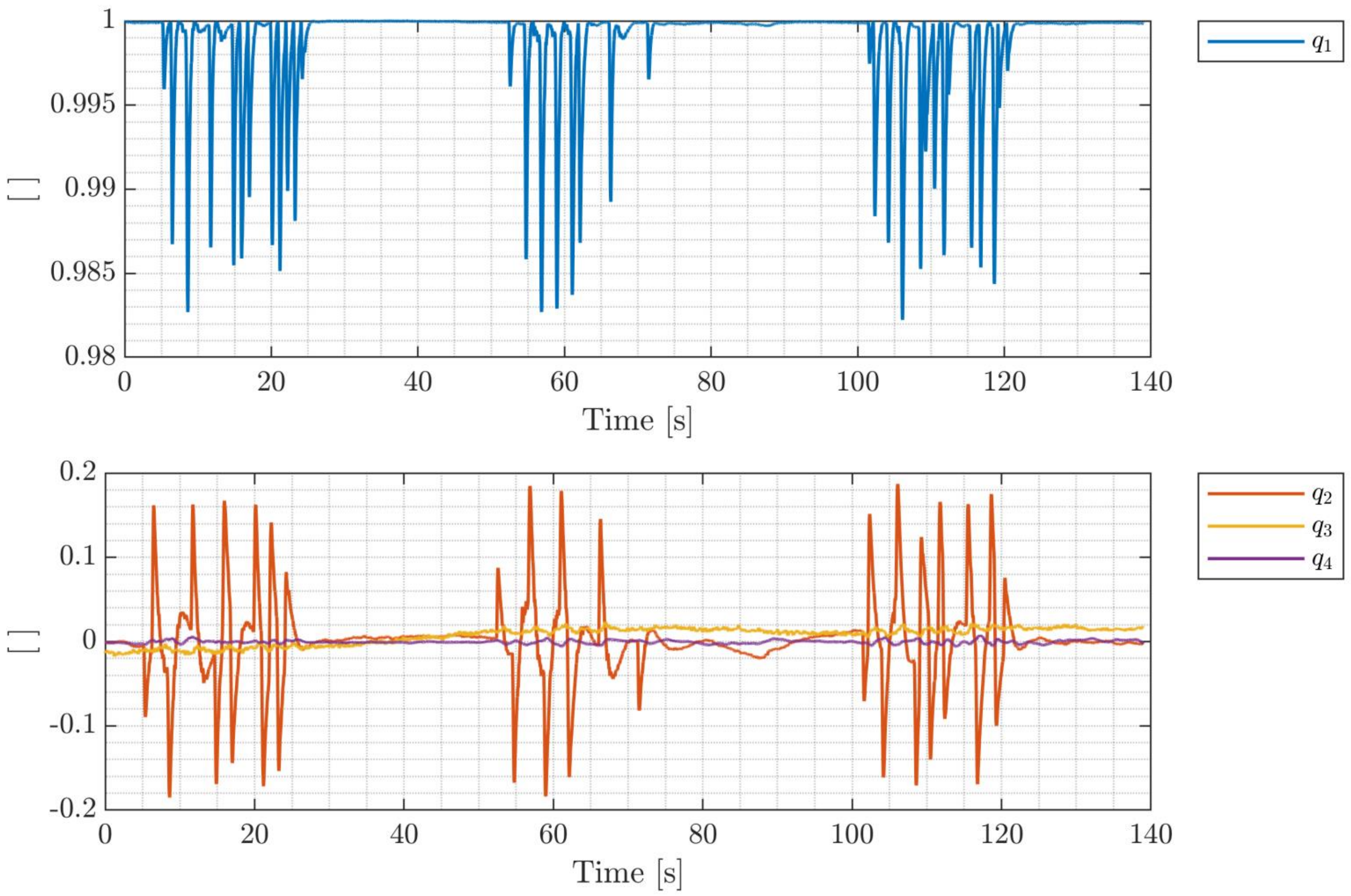




Available Data

- Scalar - 1st component

- Quat. Rotation Matrix is built.
  - Acceleration rotated Body to NED



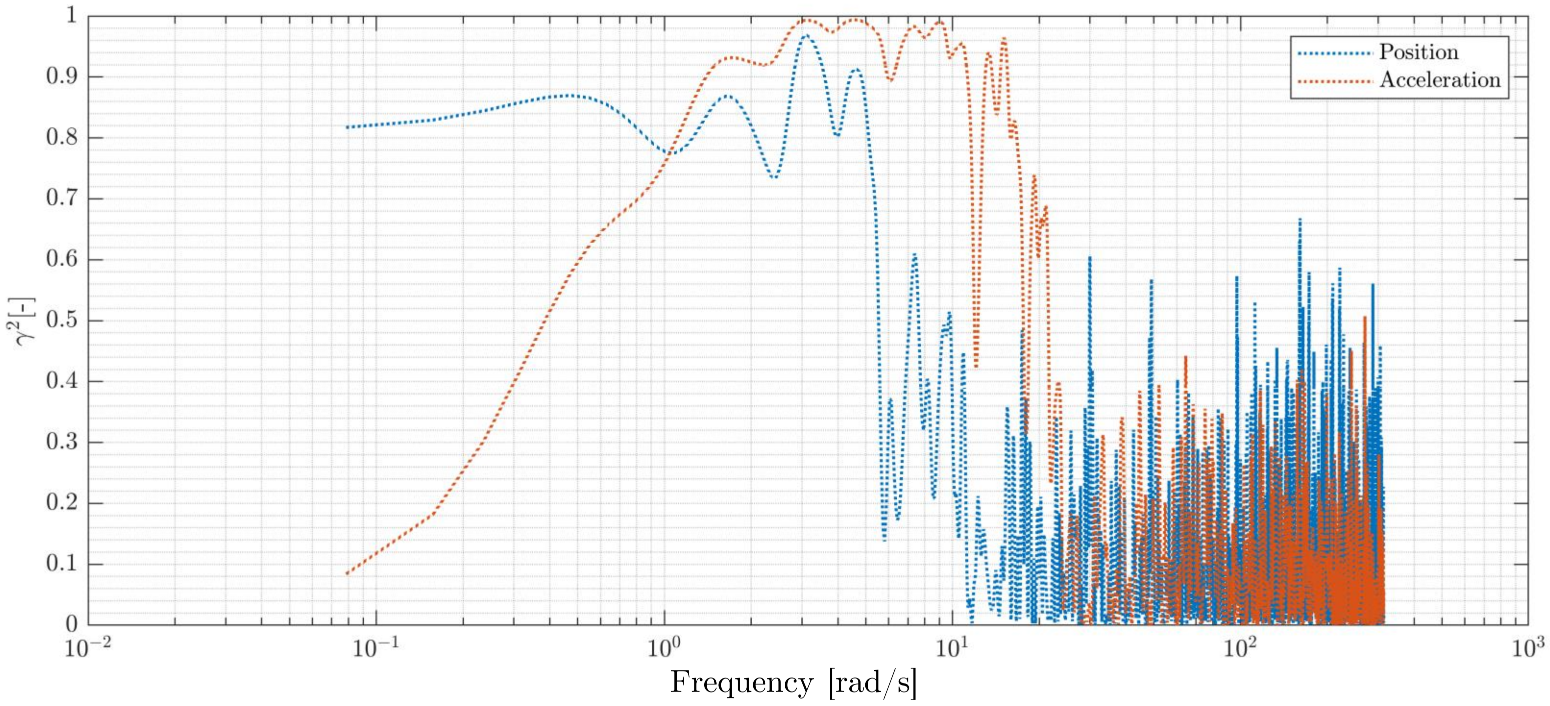
Black Box Model Identification

Assumption: LTI system

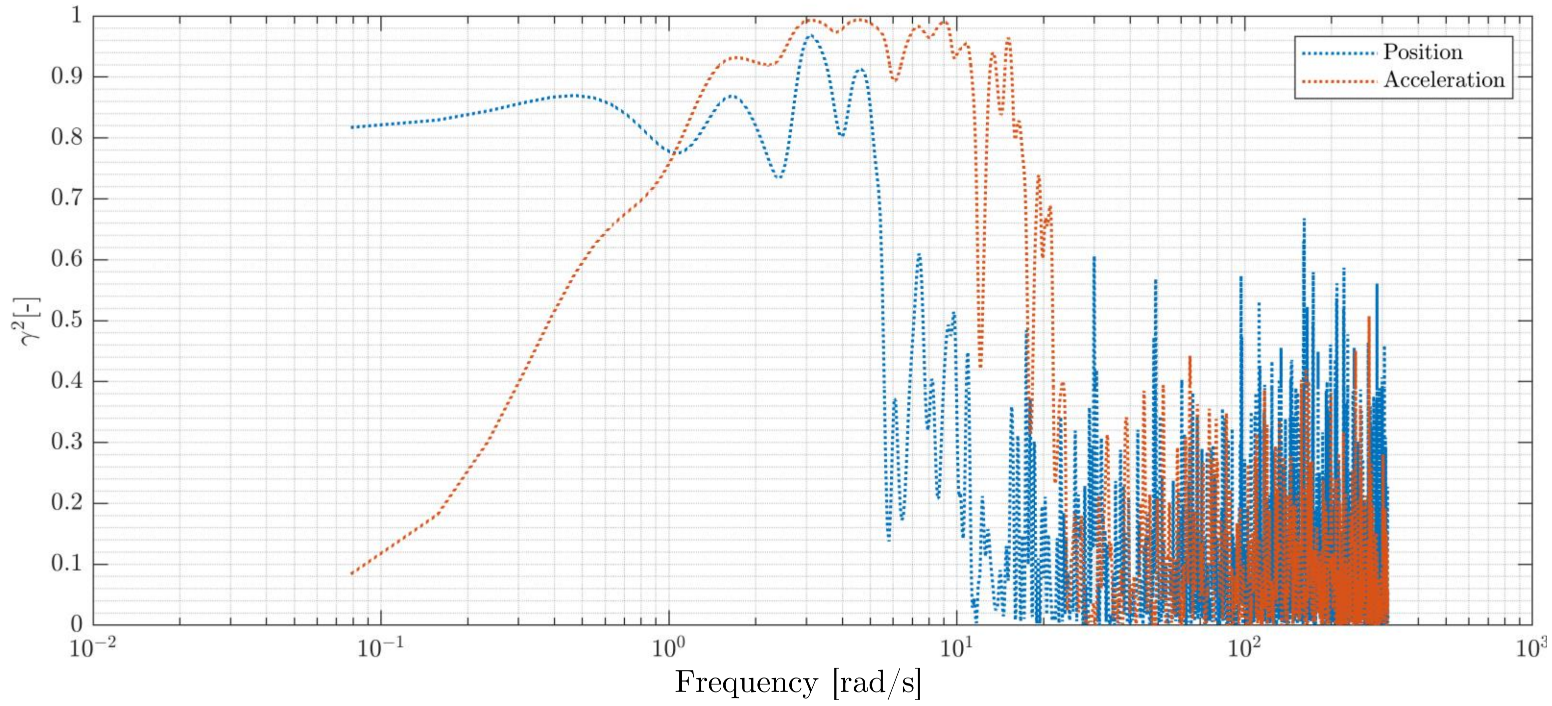
To verify it, the coherence function γ^2 is estimated and plotted ⟶ *WELCH METHOD*

- The original signal of length N is broken into segments of length `nsc = floor(N/4.5)`
- Overlapping fraction between segments: 50%
- Hamming Window

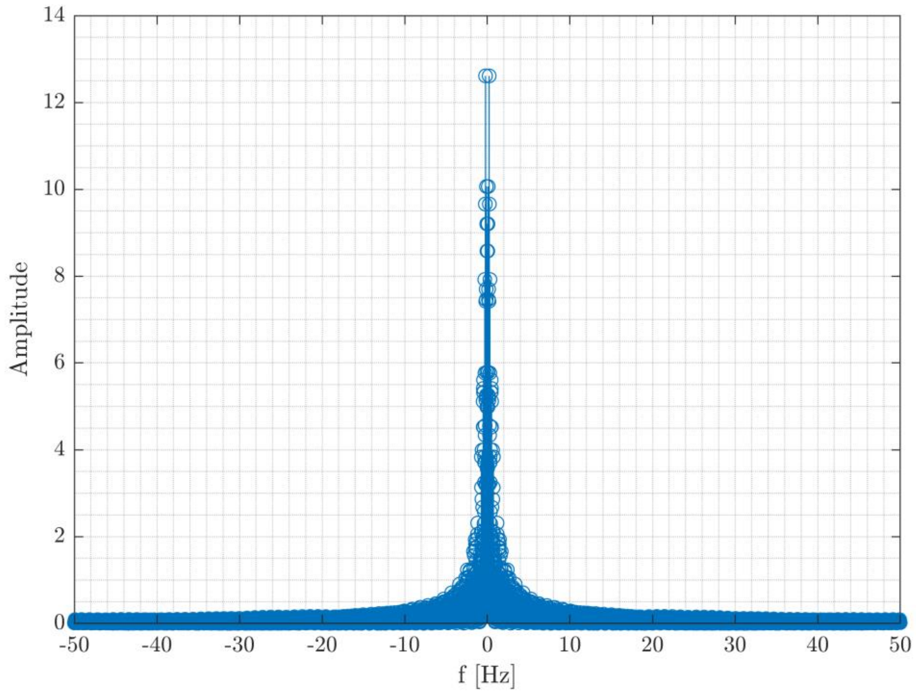
LTI Verification: Coherence Function



LTI Verification: Coherence Function



Input signal FFT



The γ^2 high frequency behaviour is justified

Black Box Model Identification

$$G(z) = \frac{\beta_0 z^m + \beta_1 z^{m-1} + \beta_2 z^{m-2} + \cdots + \beta_m}{z^n + \alpha_1 z^{n-1} + \alpha_2 z^{n-2} + \cdots + \alpha_n}$$

1. Model structure definition
   - Choose discrete model class
   - Different transfer functions are generated, with:
     - Numerator: imposed 0-th order for the position and expect 2nd order for the acceleration
     - Denominator: order gradually increased.
2. Identification procedure
   - Data corresponding to 2 PRBS sequences
   - For each transfer function couple, generate state space representation
     1. `tf({numerators}, denominator, sample_time)`

        → Discrete transfer model
     2. `sys = ss(tf(…))`

        → Convert the discrete transfer model to a discrete state space model
3. Estimation of parameters through `greyest`

Black Box Model Identification

4. Extract State Space:
   - Matrices
   - Parameters
   - Variances

5. Simulate discrete response using estimated parameters

6. Calculate FIT by comparing simulated and measured outputs

$$\mathrm{FIT} = \left[1 - \frac{\sum_{k=1}^{N} \left(y\left(k\right) - y_m\left(k\right)\right)^2}{\sum_{k=1}^{N} y_m\left(k\right)^2}\right] \times 100\%$$

Transfer Function Order Analysis

FIT – Position % | Acceleration %

Numerator Order

Denominator Order	0th | 0th	0th | 1st	0th | 2nd	0th | 3rd
1st	04.96% | 30.51%	04.62% | 28.86%	N/A	N/A
2nd	03.24% | 37.71%	54.61% | 93.81%	86.09% | 54.85%	N/A
3rd	03.31% | 37.71%	64.55% | 64.35%	99.26% | 91.42%	04.96% | 30.51%
4th	74.66% | 04.73%	76.42% | 73.98%	99.55% | 97.39%	95.60% | 93.24%

Estimated Model

- All initial values for the parameters were 1E-6
  - 0 caused errors within the generation of the discrete transfer functions

$$G_p(z) = \frac{\rho_5}{z^4 + \rho_1 z^3 + \rho_2 z^2 + \rho_3 z + \rho_4}$$

$$G_a(z) = \frac{\rho_6 z^2 + \rho_7 z + \rho_8}{z^4 + \rho_1 z^3 + \rho_2 z^2 + \rho_3 z + \rho_4}$$

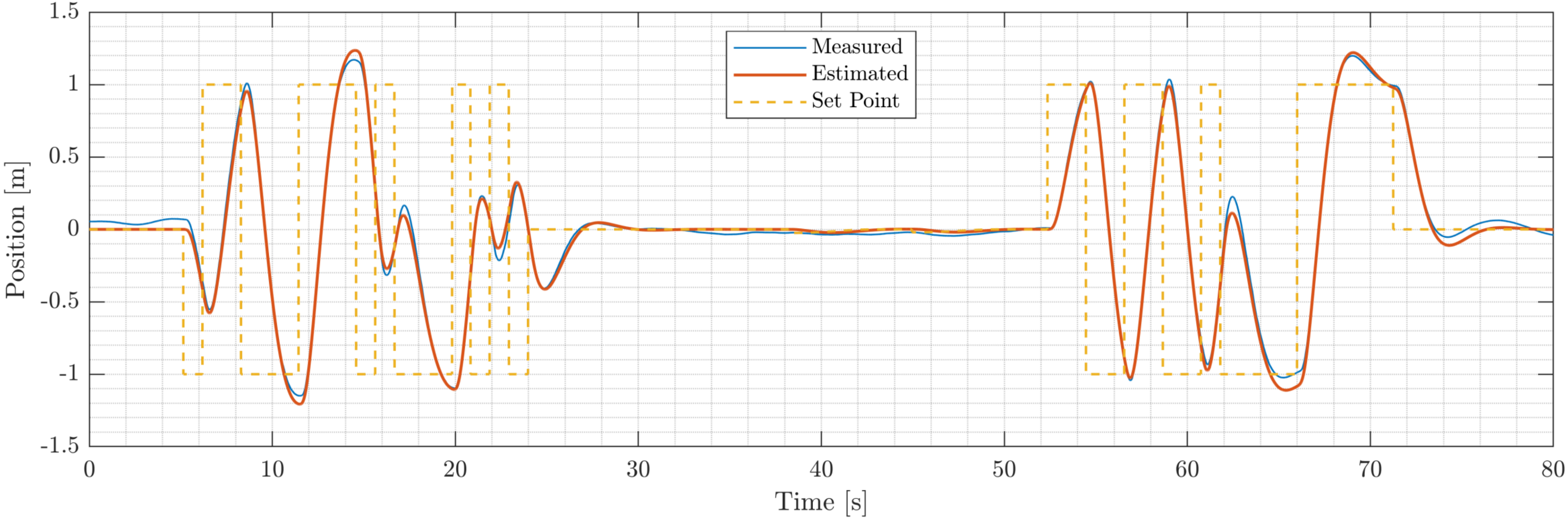
	Estimated Value	Std % [1]
ρ_1	-3.8599 ± 0.0016	0.0421
ρ_2	5.5895 ± 0.0048	0.0855
ρ_3	-3.5991 ± 0.0047	0.1303
ρ_4	0.8695 ± 0.0015	0.1764
ρ_5	(1.332 ± 0.014)×1E-6	1.0796
ρ_6	(-1.397 ± 0.0160)×1E-2	1.1481
ρ_7	(2.786 ± 0.0320)×1E-2	1.1483
ρ_8	(-1.389 ±0.016)×1E-2	1.1485

[1] A std % > 20 % implies the estimation is not significant

$$A = \begin{bmatrix} 3.8599 & -2.7947 & 1.7995 & -0.8695 \\ 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0.125 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

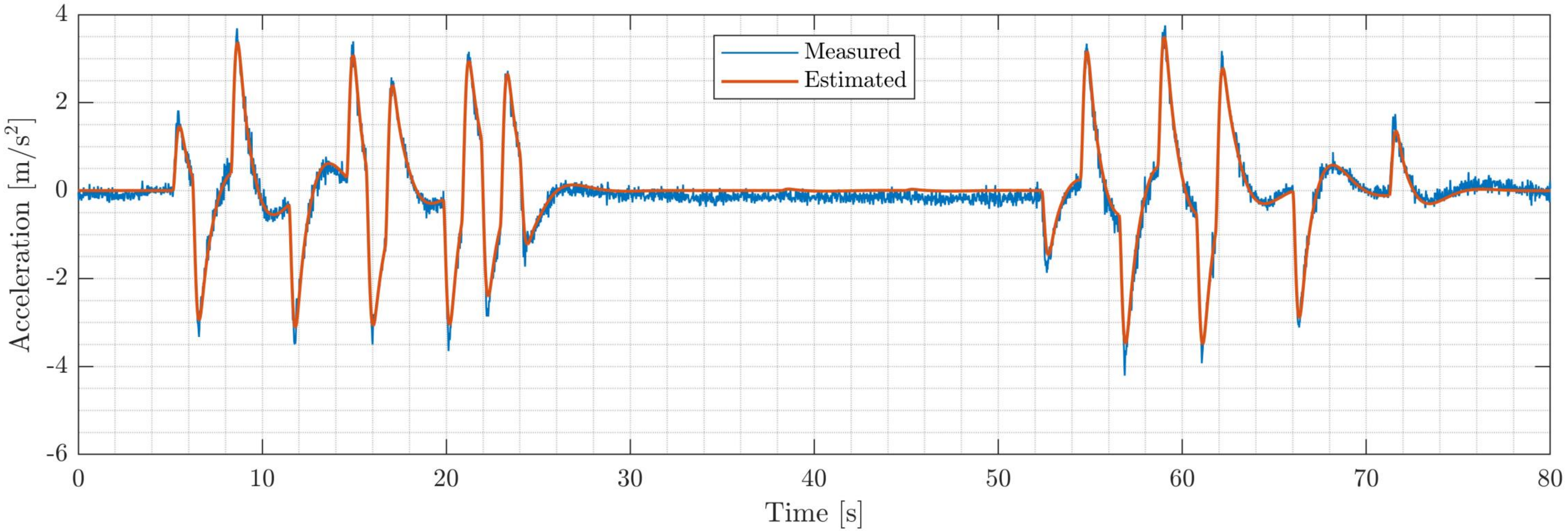
$$C = \begin{bmatrix} 0 & 0 & 0 & 1.065 \times 10^{-5} \\ 0 & -0.0559 & 0.1115 & -0.1112 \end{bmatrix} \quad D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Simulated vs Measured Comparison – Position



FIT = 99.55%

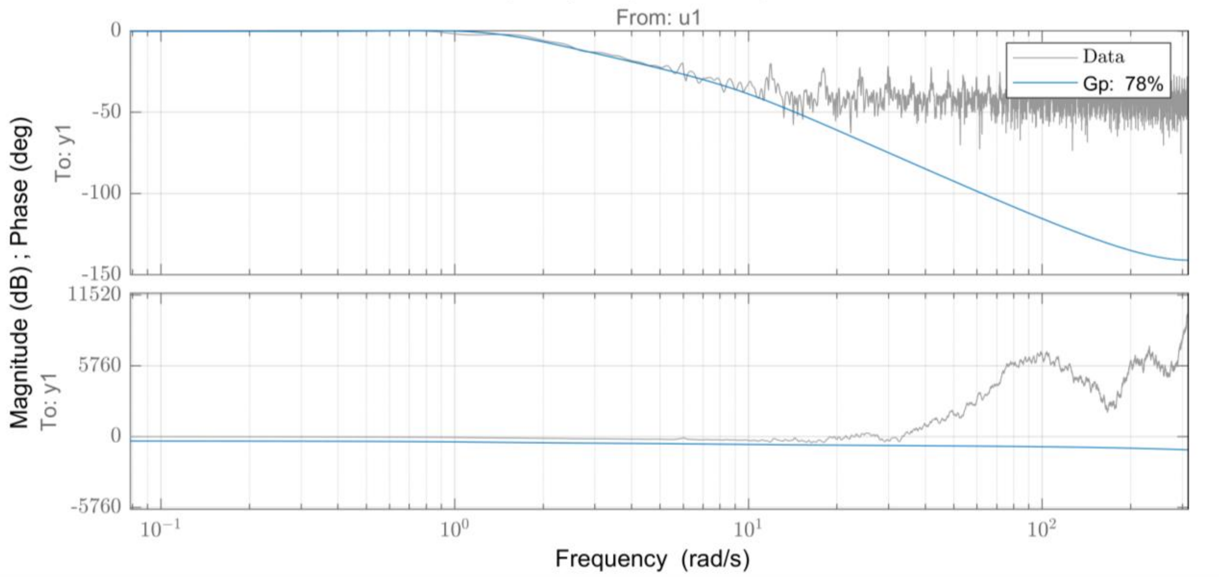
Simulated vs Measured Comparison – Acceleration



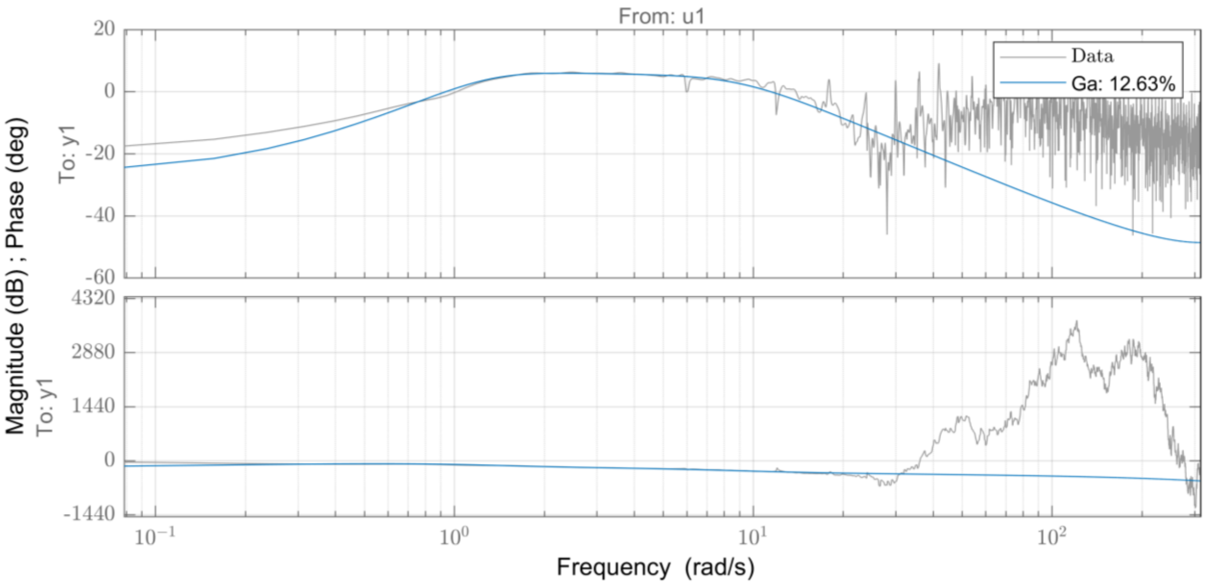
FIT = 97.39%

Simulated vs Measured Comparison - Frequency

TF Position / Pos set point

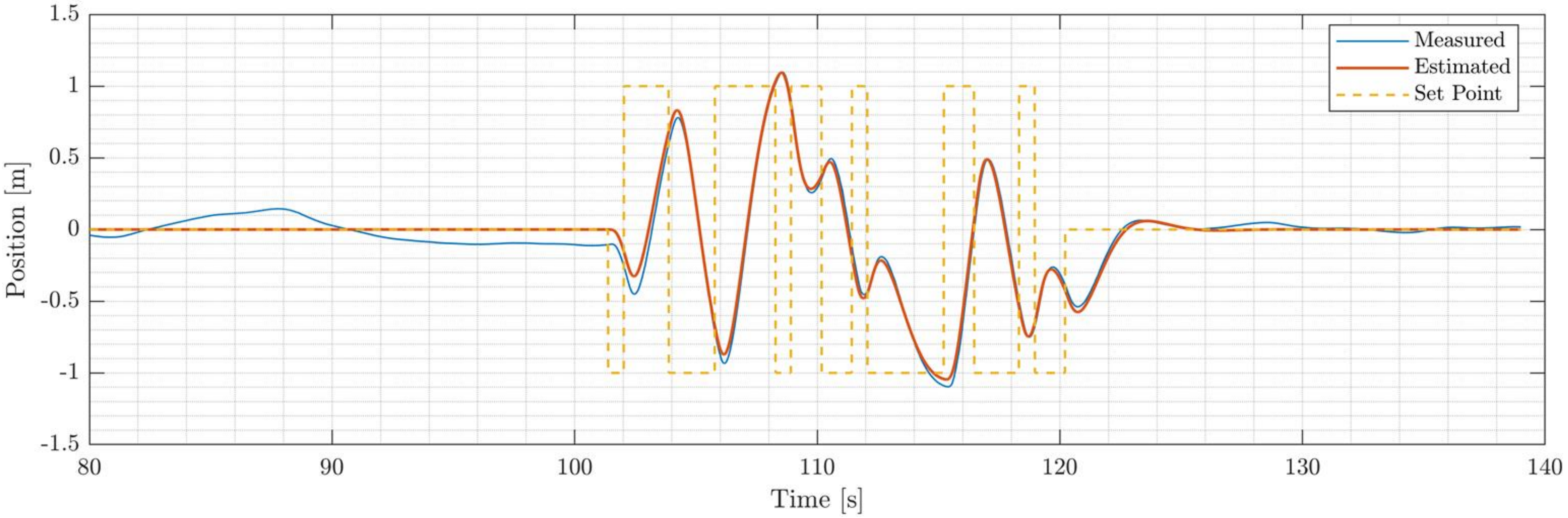


TF Acceleration / Pos set point



Model Validation - Position

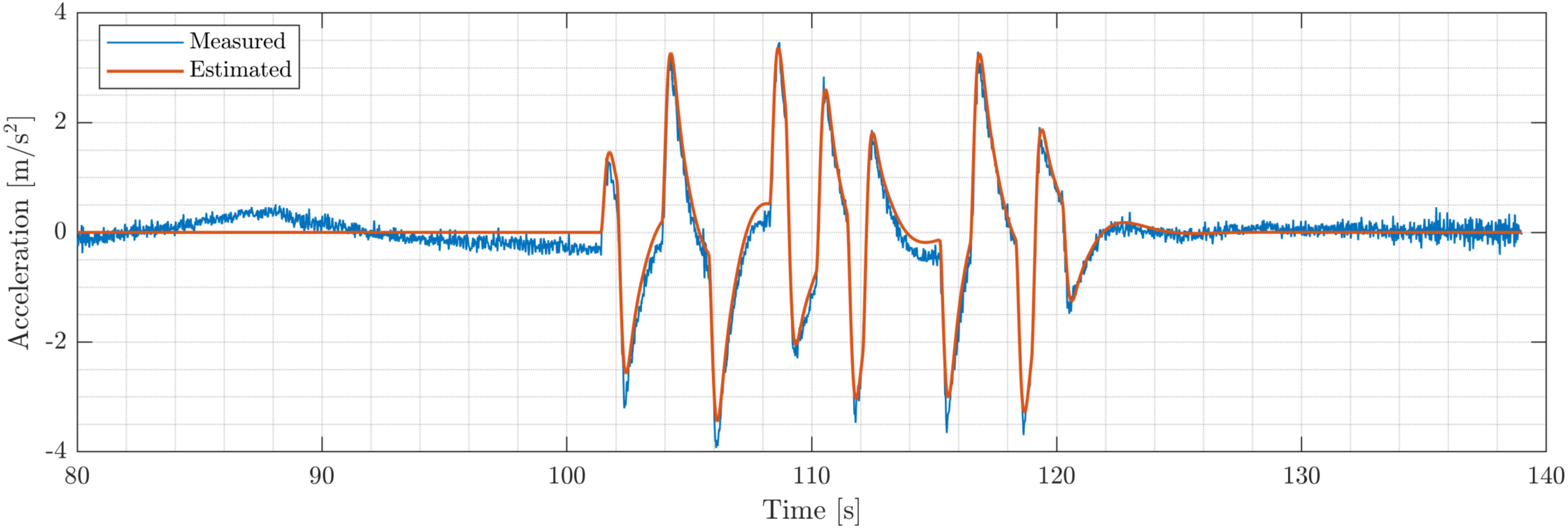
Validation dataset is the response to the 3rd PRBS



FIT = 96.62%

Model Validation - Acceleration

Validation dataset is the response to the 3rd PRBS



FIT = 95.05%

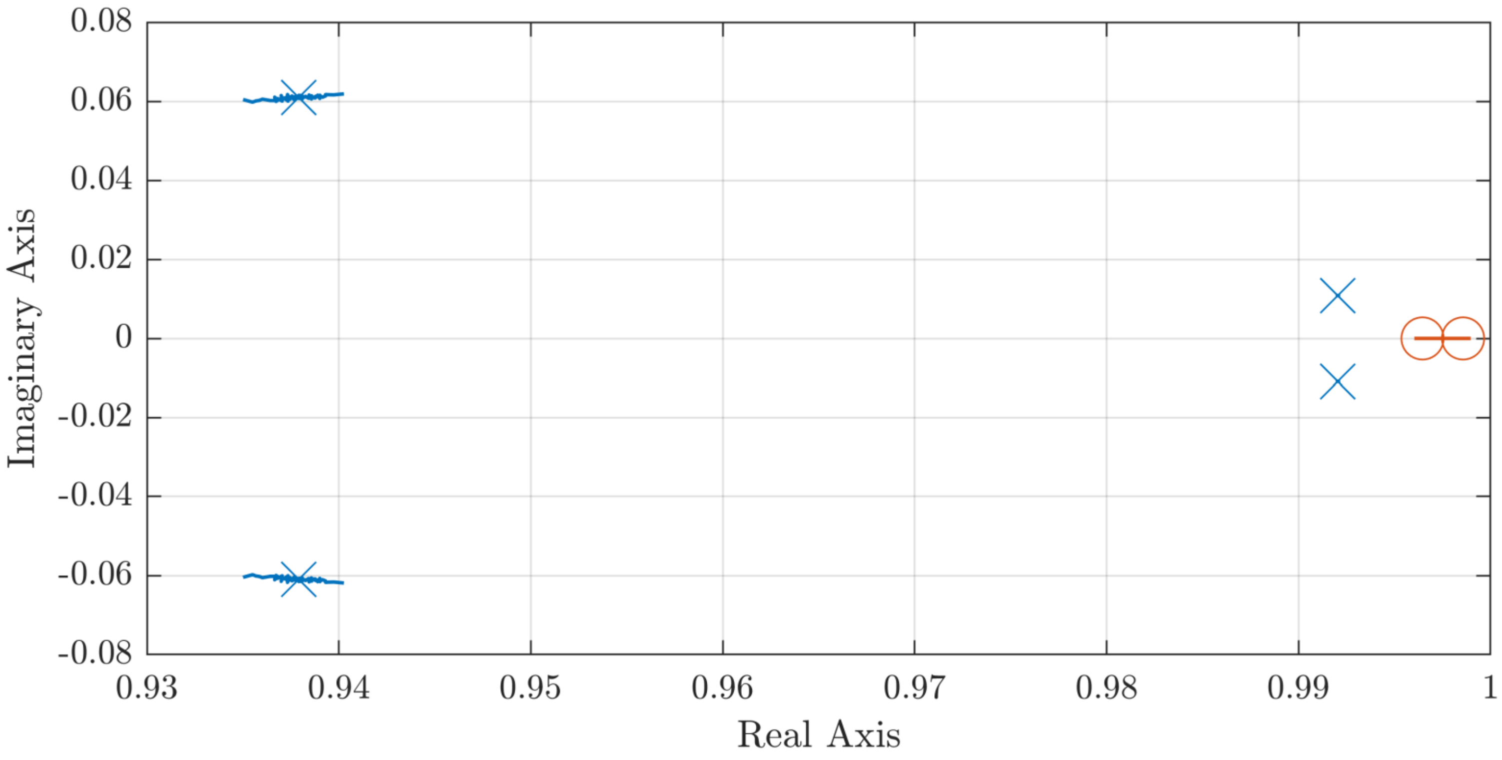
Uncertainty Analysis

- For each parameter, a Gaussian distribution is considered, with:
    - Mean value: the estimated parameter value
    - Variance: estimated in the identification procedure.

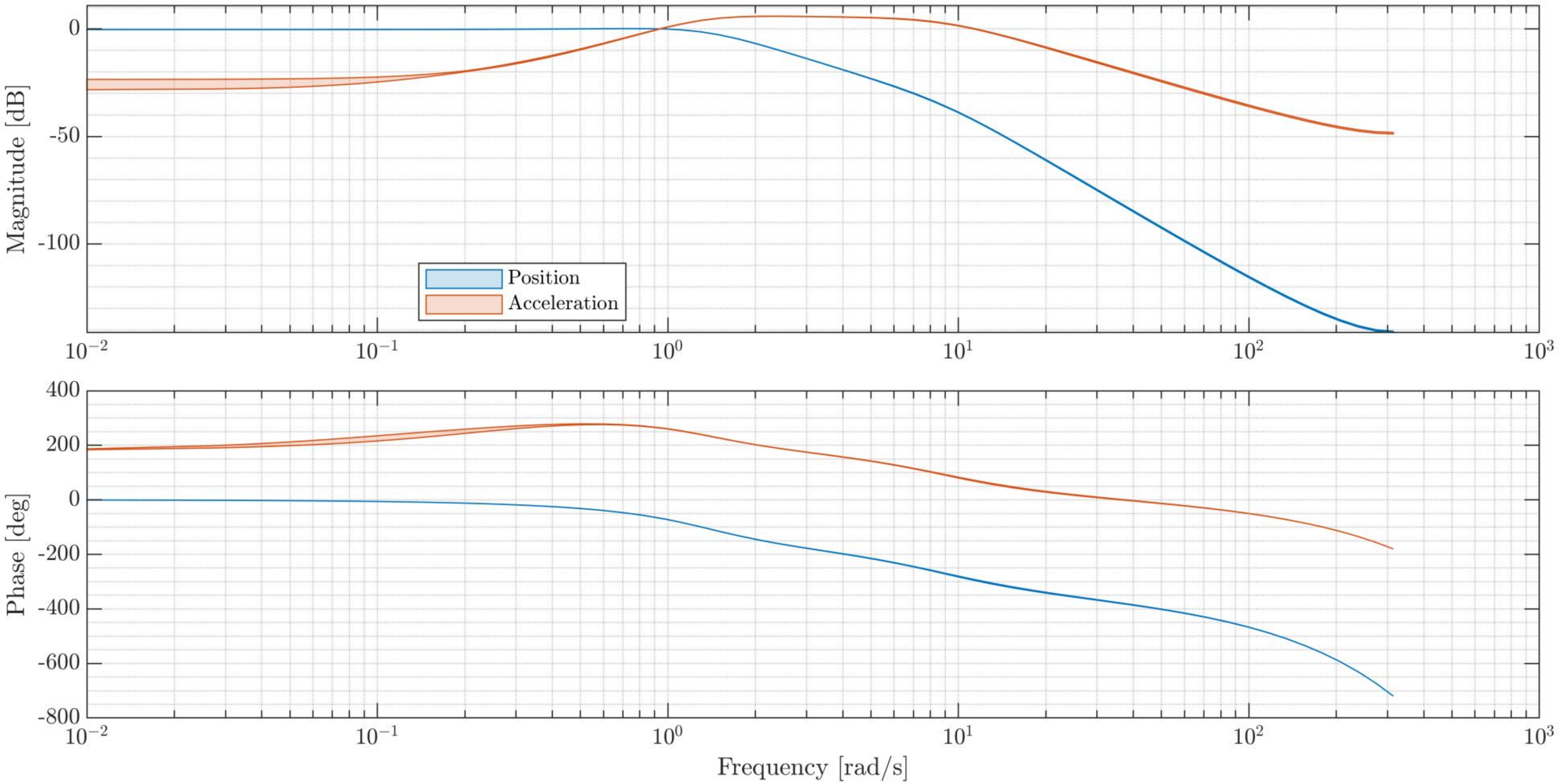
- From the same normal distribution, N random vectors are obtained through Matlab's `mvnrnd` function.
    - N = 500 perturbed models are therefore generated.

Uncertainty Analysis – Pole/Zero Map

- Common poles
  - Shared 4th order denominator
- Zeros only with acceleration
  - 2nd order numerator for acceleration
  - 0th order numerator for position



Uncertainty Analysis – Bode



State Estimation

Problem Statement

DT-DT problem

$$x(t+1) = A\,x(t) + B\,u(t) + w(t) \qquad x(1) = x_1$$
$$y(t) = C\,x(t) + v(t)$$

- v and w are DT white Gaussian noise processes : $w \sim G(0, W) \qquad v \sim G(0, V)$
- x_1 is a Gaussian random variable : $x_1 \sim G(0, P_1)$
- v , w and x_1 are independent.

Kalman Filter

- Kalman filter requires:
  - Measurement noise covariance V
  - Process noise covariance W
  - Initial values
- Works together with our SS model to improve the overall estimation

Prediction:

$$\hat{x}(N)(-) = A\,\hat{x}(N-1)(+) + Bu(N-1)$$

$$P(N)(-) = AP(N-1)(+)A^T + W$$

$$P_y(N) = CP(N)(-)C^T + V$$

$$K_f(N) = P(N)(-)\,C^T P_y(N)^{-1}$$

Correction:

$$e(N) = y(N) - C\,\hat{x}(N)(-)$$

$$\hat{x}(N)(+) = \hat{x}(N)(-) + K_f(N)\,e(N)$$

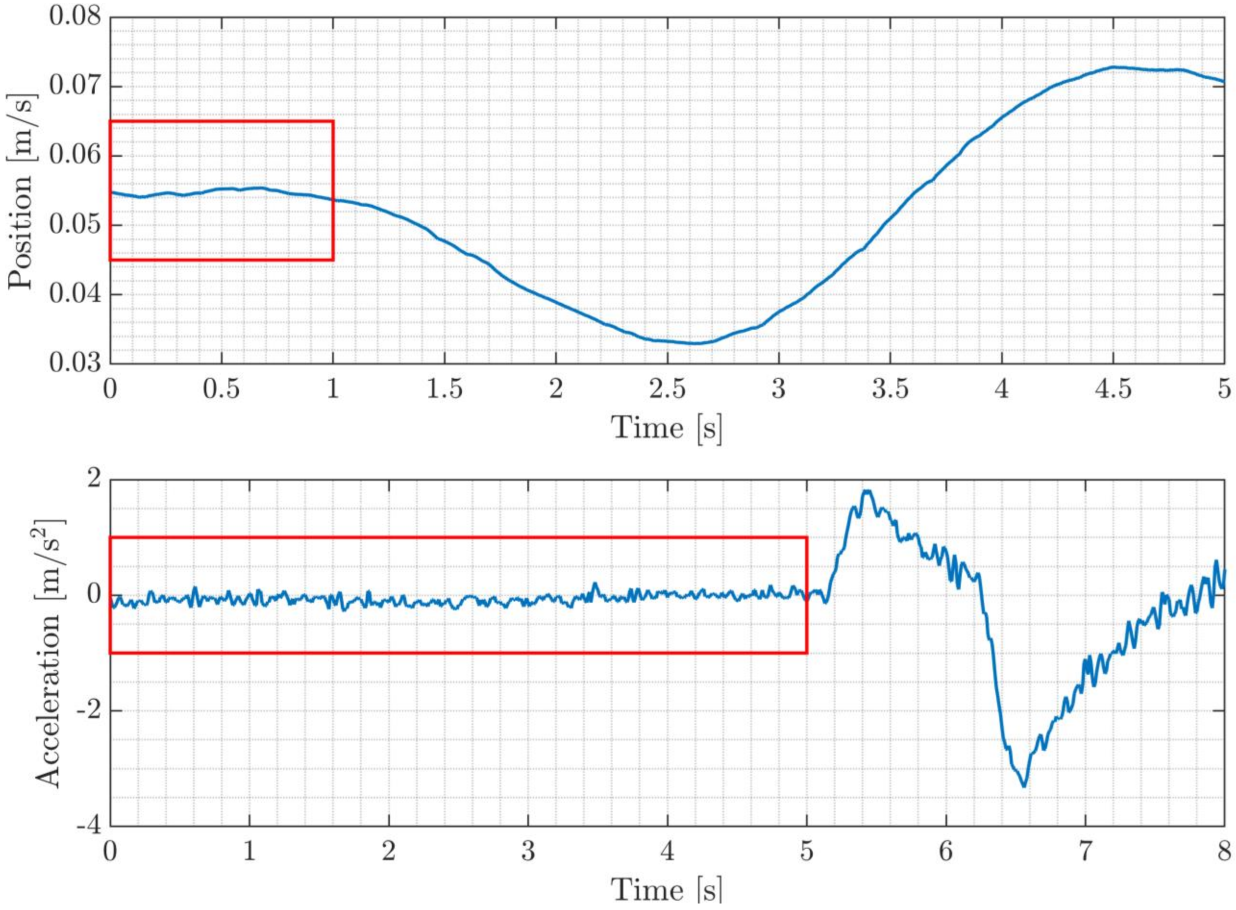
$$P(N)(+) = \left[I - K_f(N)\,C\right]P(N)(-)$$

$$\hat{y}(N) = C\hat{x}(N)(+)$$

Measurement Noise Covariance

- Used initial data prior to the excitations
- Different time spans due to additional effects affecting positional data
  - Taken segment where effect is minimal

$$\mathbf{V} = \begin{bmatrix} 1.83 \times 10^{-7} & 0 \\ 0 & 6.83 \times 10^{-3} \end{bmatrix}$$



Process Noise Covariance

$$\mathbf{W} = \mathbf{B}\mathrm{Q}\mathbf{B}^{\mathrm{T}}$$

- Goal: Achieve the best fit by tuning the Kalman Filter
  - Performed various Kalman runs with differing Q
    - Q = [0.001 0.01 0.1 1 10 100]
  - Compared the Kalman filtered against the measured values
    - Focused on the Fit for Acceleration as the Position Fit consistently is high 99.XX%
  - Optimal value found at $\mathrm{Q} = 100$

Initial Values

- Initial State vector:
  - Calculated via the acceleration measurements and the Output matrix
  - For the sake of generality, a random value up to 3σ is added.

  For each Q iteration: 15 different `rand` are used to generate 15 different states IC. The relative solutions are computed and the average is taken.

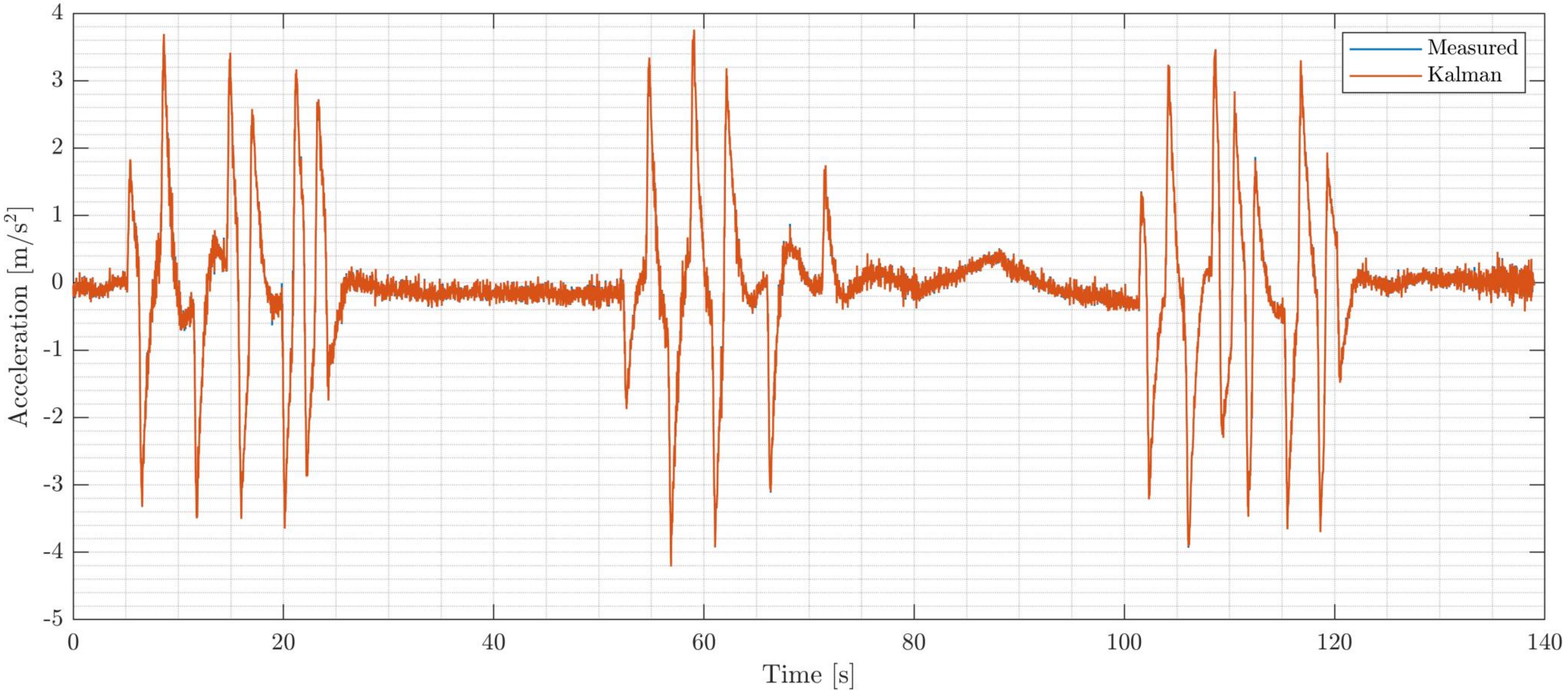
- Initial Predictor error covariance:
  - Arbitrarily put equal to $\mathbf{W}$

$$\hat{x}(1,:) = \mathbf{C}(2,:)^{-1} y(1,2) + 3\,\sigma_x\,\mathrm{rand}$$

$$\sigma_x(1) = \sqrt{\mathbf{P}(1)}$$

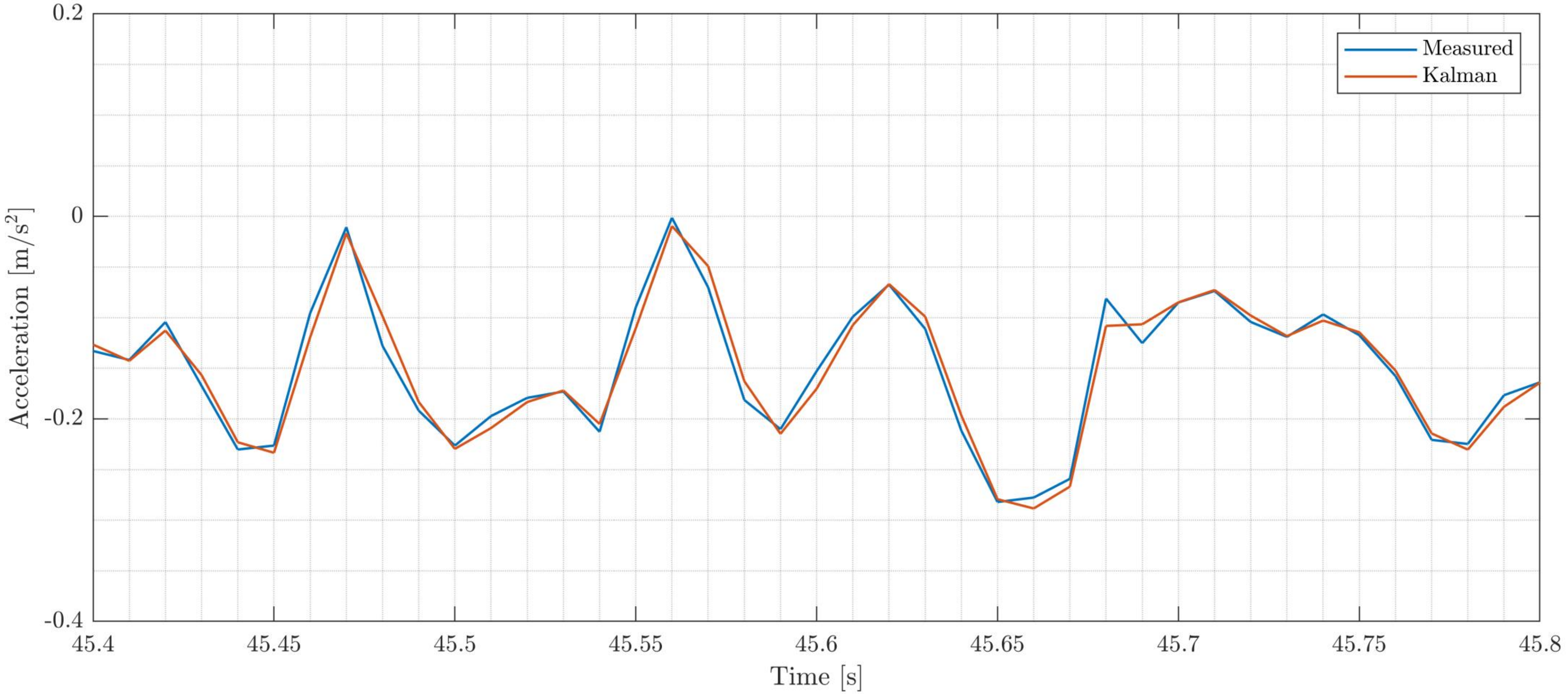
$$\mathbf{P}(1) = \mathbf{W}$$

Results - Acceleration



FIT = 99.12%

Results - Acceleration



There is over fitting therefore, we need to refine the value of Q

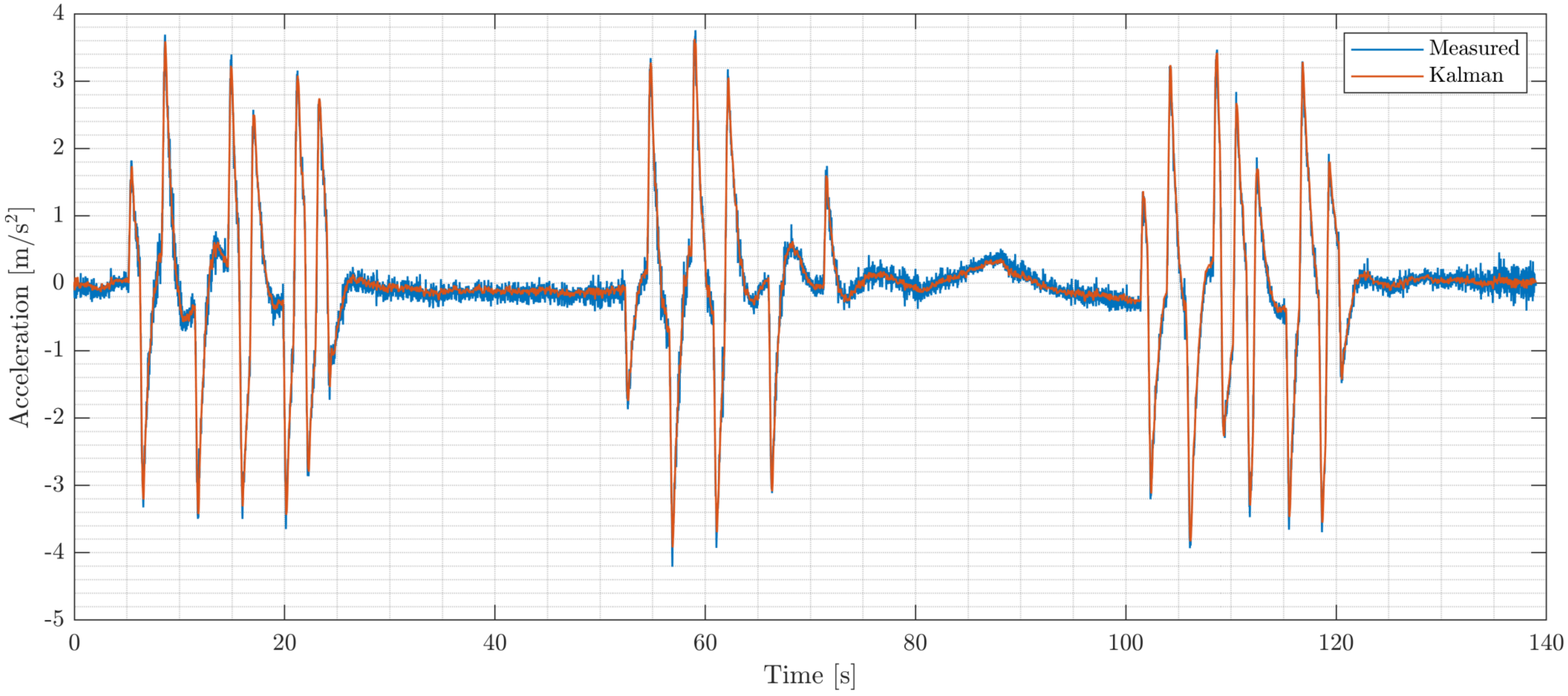
Results - Acceleration

- The lower the Q the more the estimation *lags* behind the real value.

- The larger the Q the more we *trust* the measurements and the less we *trust* the model; however too high of a Q results in a noisy estimation and risk of overfitting.
    - Which is what we previously found!

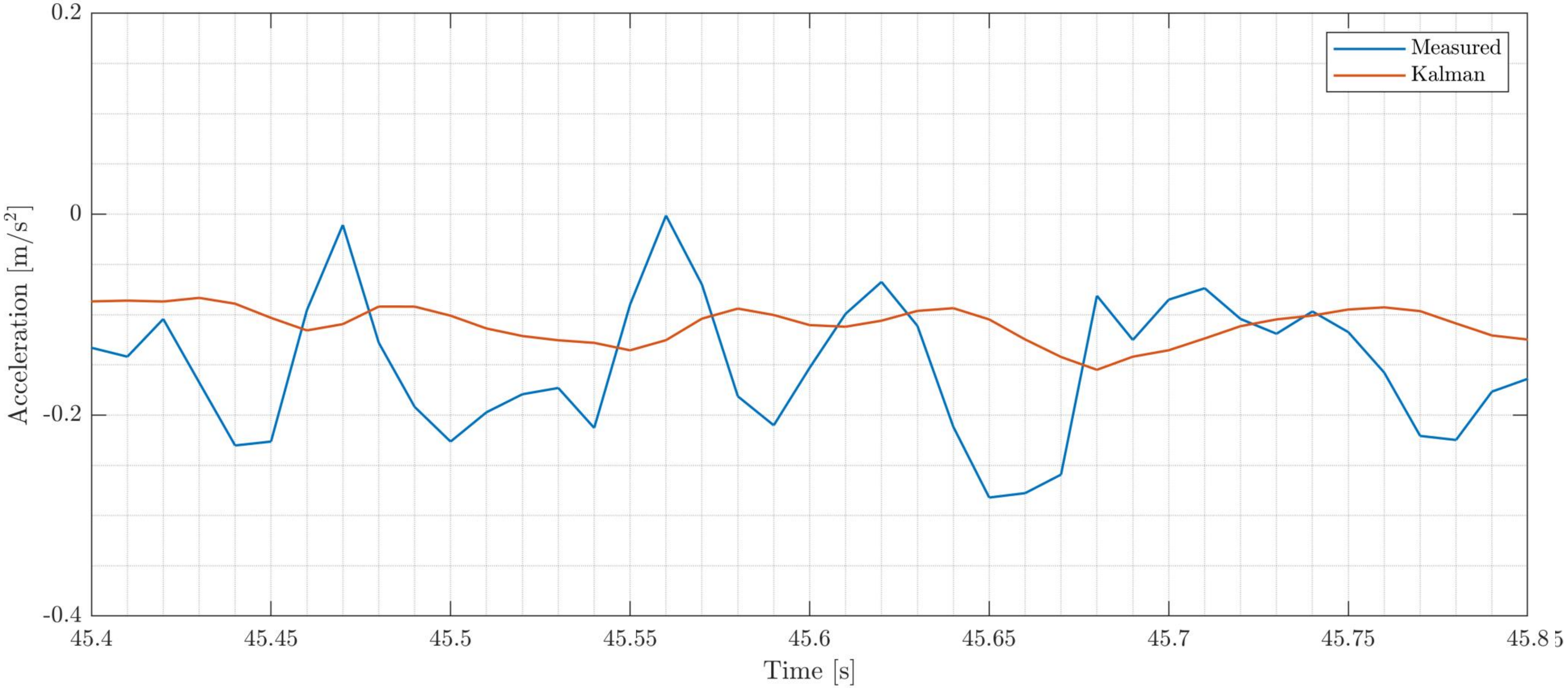
- Refine Q to achieve a compromise
    - After trial-and-error process, $Q = 0.03$ was found to both follow the real measurements and possess a reduction in noise.

Results - Acceleration



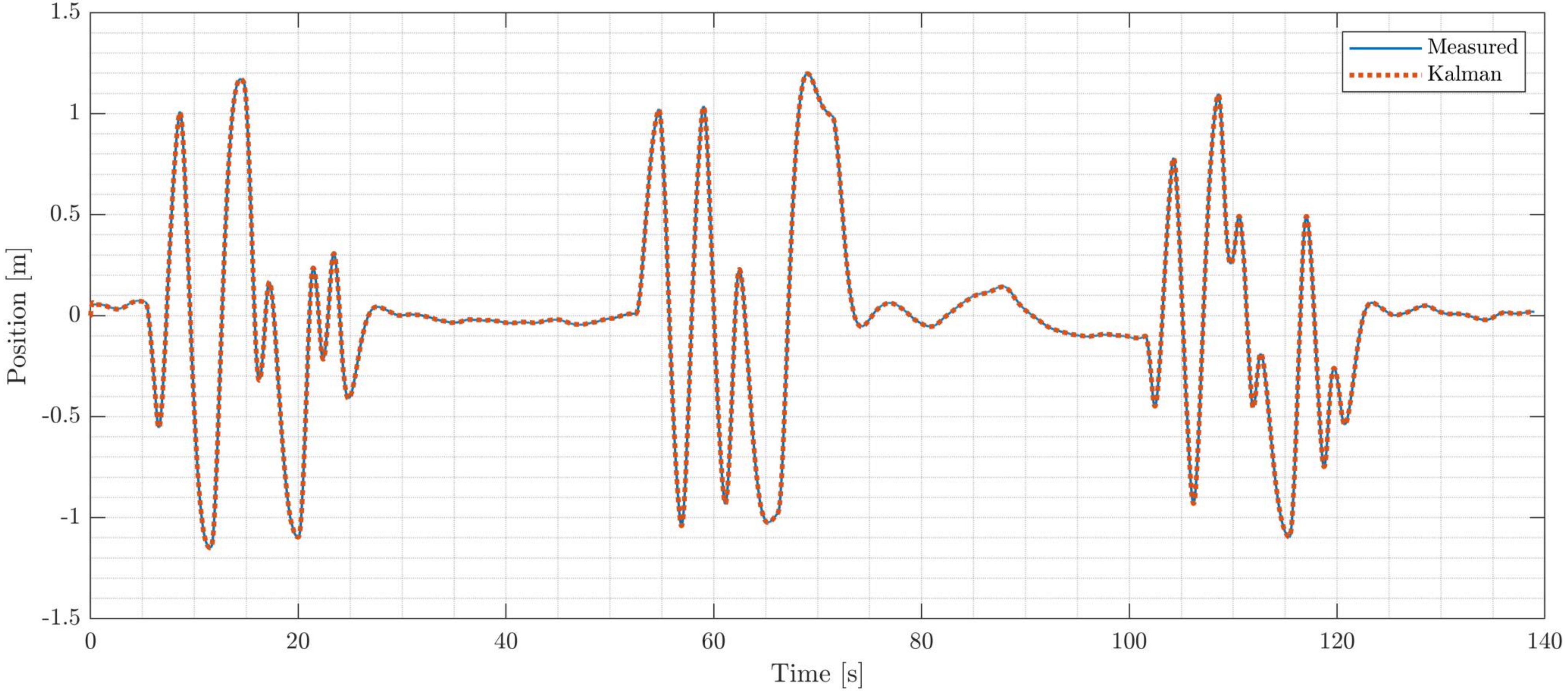
FIT = 98.75%

Results - Acceleration



Noise has left effect, some lagging is visible but is minimal

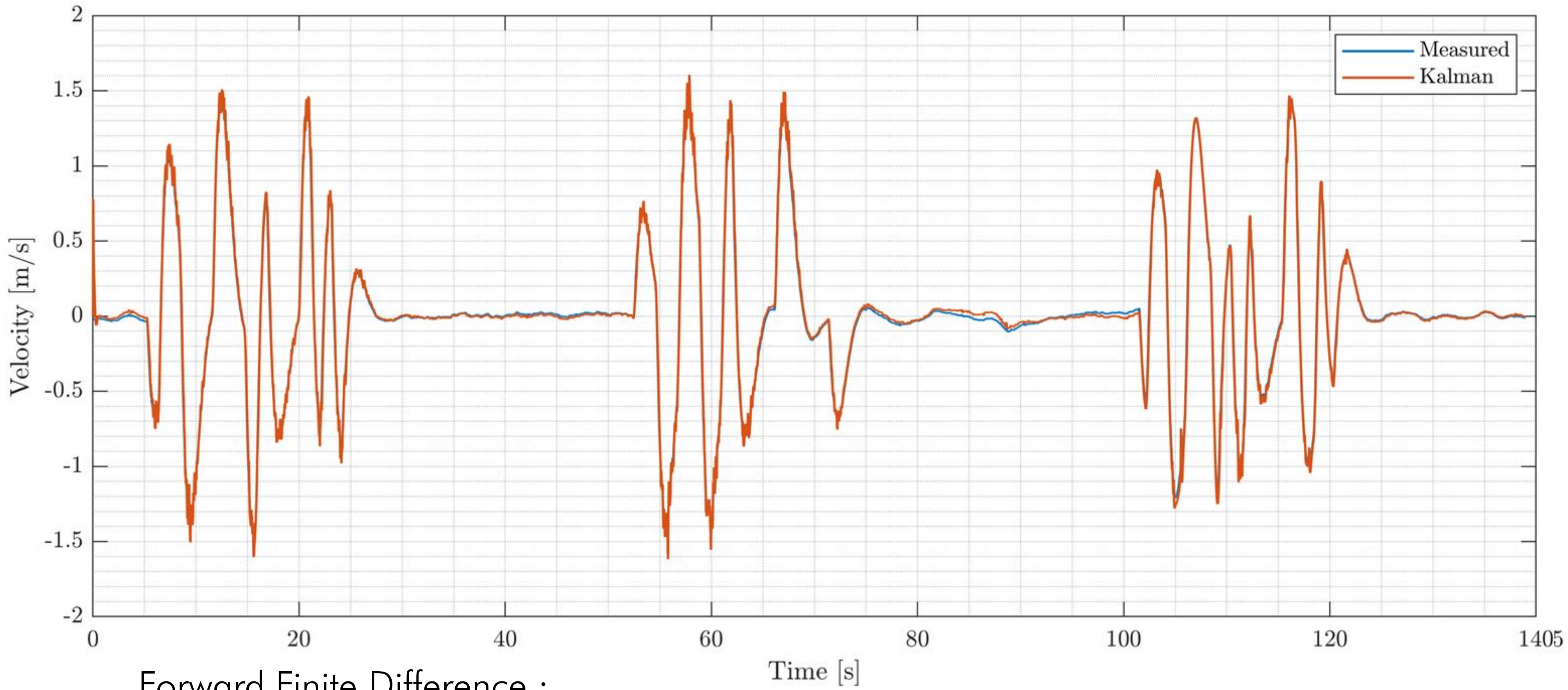
Results - Position



Almost perfect estimation

FIT = 99.9985%

Results - Velocity



Forward Finite Difference :

$$\hat{v}(N-1) = \frac{\hat{y}(N) - \hat{y}(N-1)}{T_s}$$

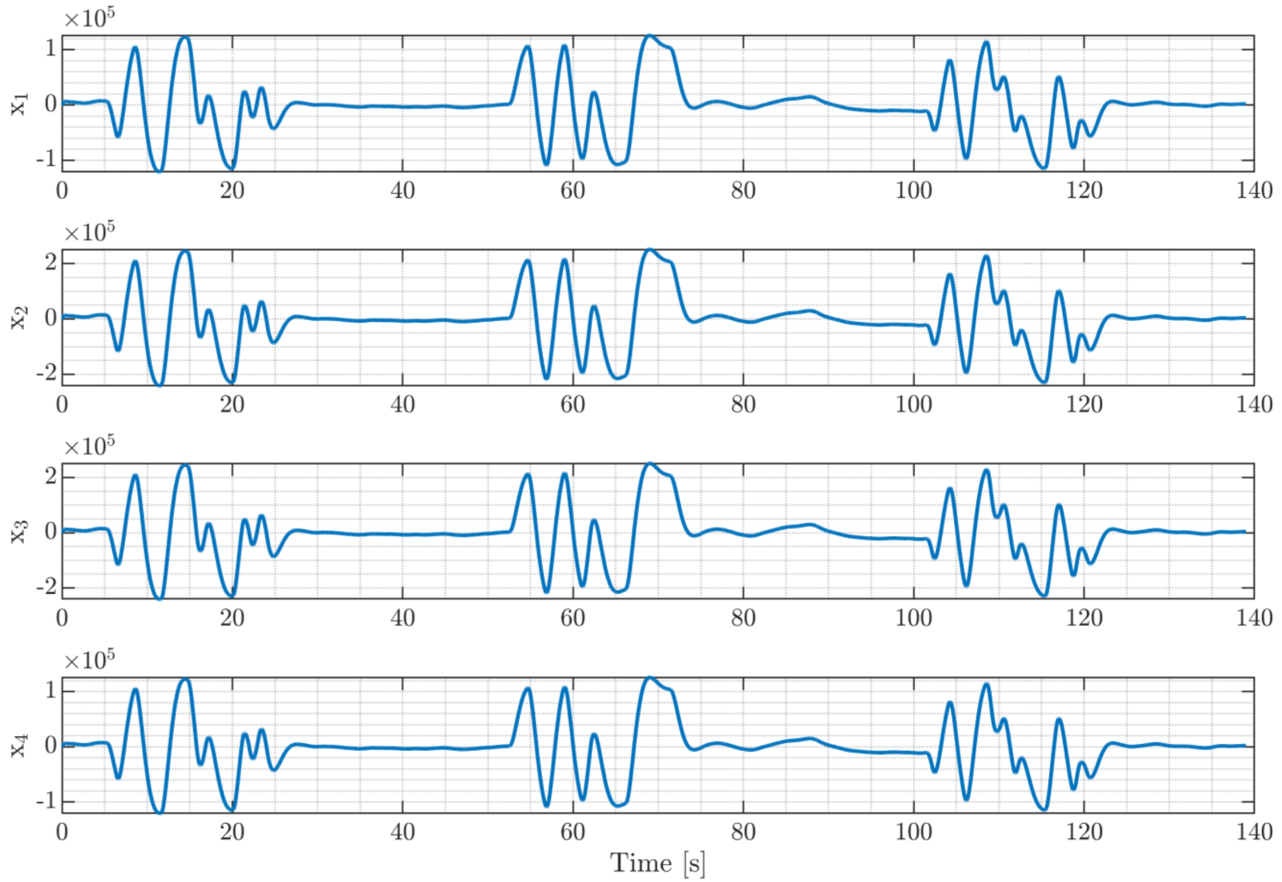
FIT $= 99.66\%$

Results - States

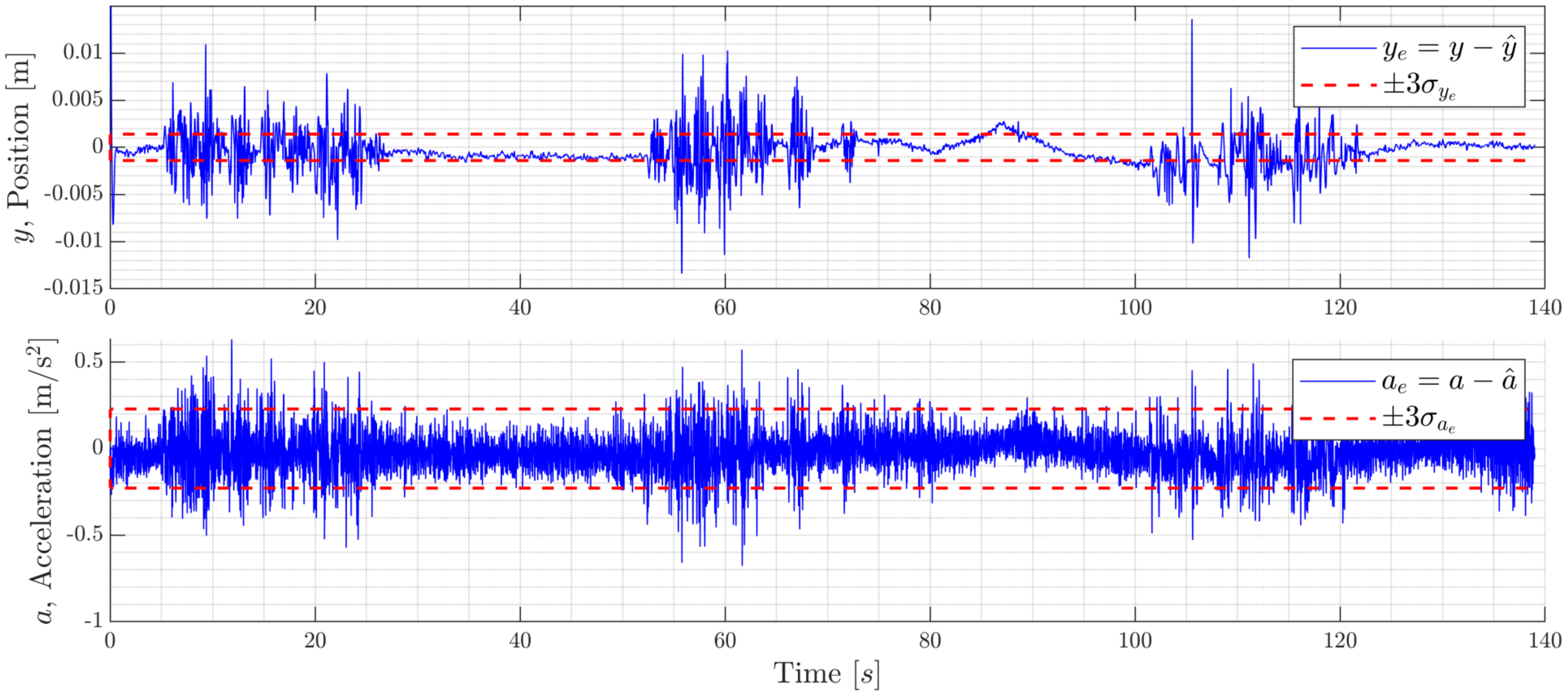
- Extraction of the states for the refined Q

- Little variance in the shape

- Some variance in the scaling comparing the different states



Results – Innovation and Confidence Interval



- Position
  - Unknown external inputs affecting the system
  - $\sigma_{y_e} = \sqrt{P_y(1,1)}$
- Acceleration
  - Large errors from noise around the estimated value
  - $\sigma_{a_e} = \sqrt{P_y(2,2)}$

Steady State Kalman Filter

Up to now the gain $K(N)$ has been considered time-varying.

However, if:

- (A, C) pair is observable
- (A, Gv) pair is reachable where $Gv\, Gv' = W$

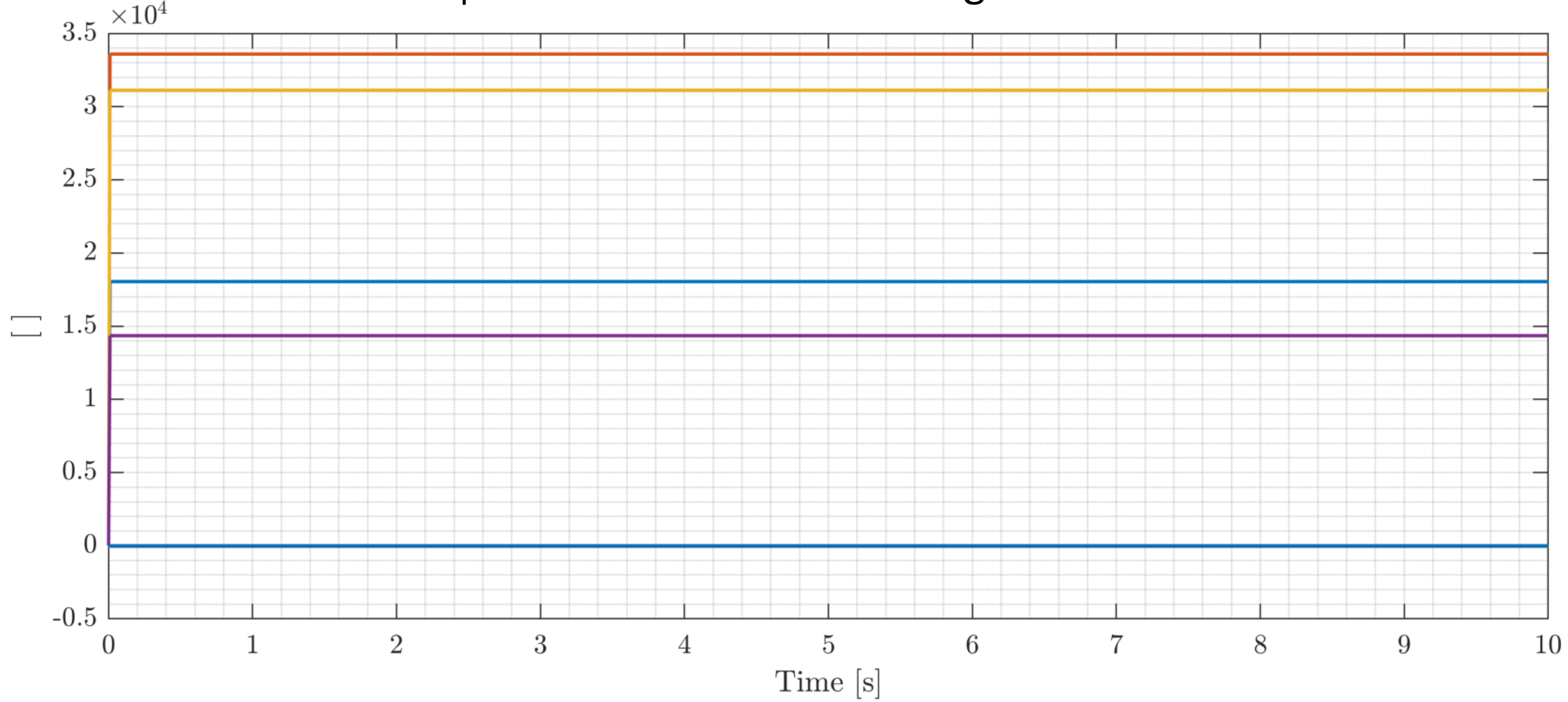
The system is asymptotically stable, $P(N)$ converges to a constant and so does the gain.

Therefore a SSKF can be implemented, allowing to reduce the overall computational effort.

Steady State Kalman Filter

- Both assumptions are verified in our case
  - Both *obsv* and *ctrb* matrices show rank 4
- Indeed it can be observed that $P(N)$ rapidly converges to a constant value

Components of the Kalman gain over time



Fin.