Prof. Lovera Marco

ICT FOR CONTROL SYSTEMS ENGINEERING

Milano - 2014/2015

POLITECNICO DI MILANO

Model Identification: *Comparison of Black-Box methods to identify the attitude dynamics of a quadrotor helicopter*

Russo Alessio, Savaia Gianluca

School of Industrial and Information Engineering

Politecnico di milano

https://github.com/rssalessio/ictproject

PROJECT GOAL

GOAL

- Identify the attitude dynamics of a quadrotor helicopter using classical PEM methods.
- Compare the results obtained with those obtained using subspace identification.

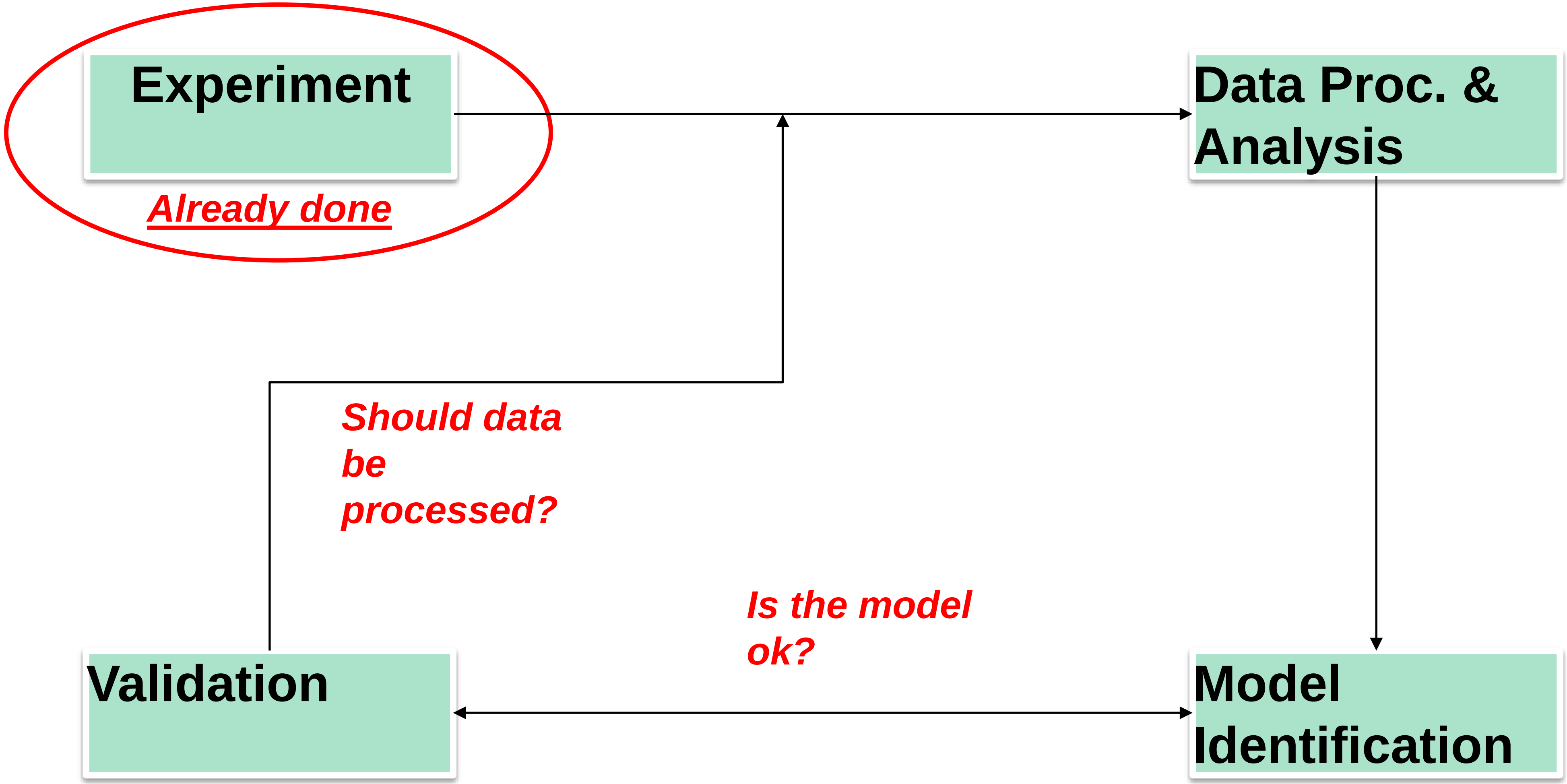
DATA

- Model identified with subspace methods.
- Set of data measured during three experiments $\{(\boldsymbol{x},\boldsymbol{y})_i, i = 1,2,3\}$

TOOLS

- Matlab
- Github: https://github.com/rssalessio/ictproject **(code available)**

IDENTIFICATION CYCLE



EXPERIMENT

EXPERIMENT

- Three experiments, done in the following manner for $t \in [0, T]$:
  - For $t \in [0, T_1]$ the system is closed loop, with an unknown regulator and a certain reference $r(t)$.
  - For $t \in [T_1 + 0.2, T_2]$ the system is in open loop, where $u(t)$ is a PRBS (Pseudo random binary source) signal.
  - For $t \in [T_2 + 0.2, T]$ the system is again in closed loop.

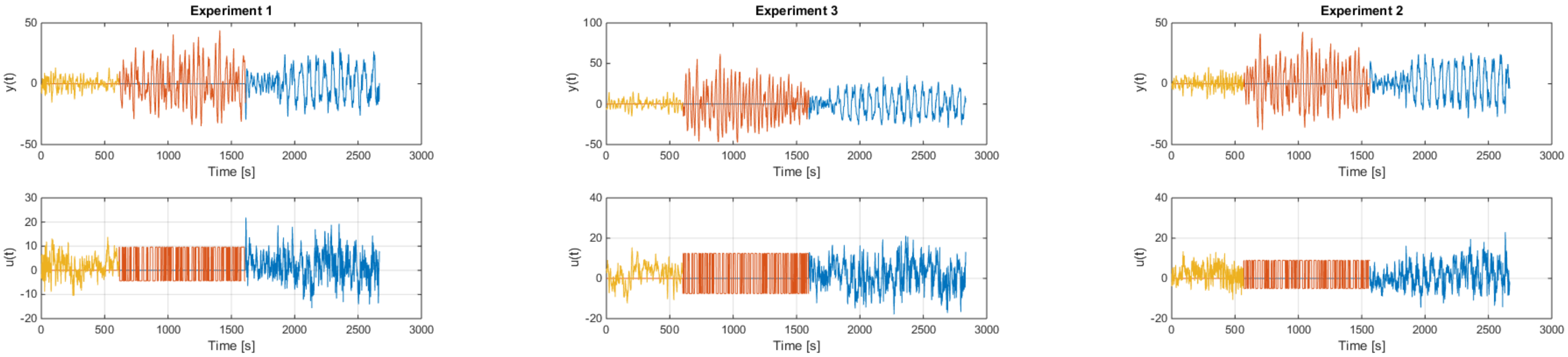
- Sampling time of input($u(t)$)/output($y(t)$) $T_s = 0.2\ [s] \Rightarrow f_s = 5\ [Hz]$, can see dynamics up to $2.5[Hz]\ \equiv 15.7\ [\frac{rad}{sec}]$.

- Experiment data is collected in a set $\{(\boldsymbol{x}, \boldsymbol{y})_i\,, i = 1,2,3\}$, where i is the i-th experiment.

- $(\boldsymbol{x}, \boldsymbol{y})_i = \{(\boldsymbol{x}, \boldsymbol{y}\,)_{ij}, j = 1,2,3\}$, where $j = 1$ represents data for $t \in [0, T_1]$, etc… .

$(\boldsymbol{x}, \boldsymbol{y})_{i2}$ is the open loop data

EXPERIMENT DATA



Experiments are quite similar, though the third one has some differences. For identification purposes we'll see that it's better to consider the full set of an experiment than just a small set of it because of the number of data, despite the fact that we have 3 different stationary signals. Deterministic trend is more influent.

DATA PROCESSING AND ANALYSIS

When analysing data for black box modelling there are several steps to consider:

1. Is there any trend on the data (constant or linear) ? ⇒ Detrend (not advisable if the system contains an integrator -> we lose dynamics). In our case there is no linear trend, so no we lose no dynamics.

2. Analyse the covariance and spectrum of the input signal to understand the level of excitation
   1. We prefer signal that behave like WN to improve (identifiability) the estimate of a parametric model

3. Estimate the impulse response and frequency response to gain:
   1. Insight into the system dynamics
   2. Estimation of the input dead time (time delay of the input)

Input Covariance – Spectrum Analysis 1/5

Analysis of the covariance gives insight on the level of excitation of the input signal: for example we can run the Anderson Whiteness Test to understand if the input signal behaves like a random white noise.

Also the rank of the correlation matrix can give some info, but most of the time has a persistence of excitation very high:

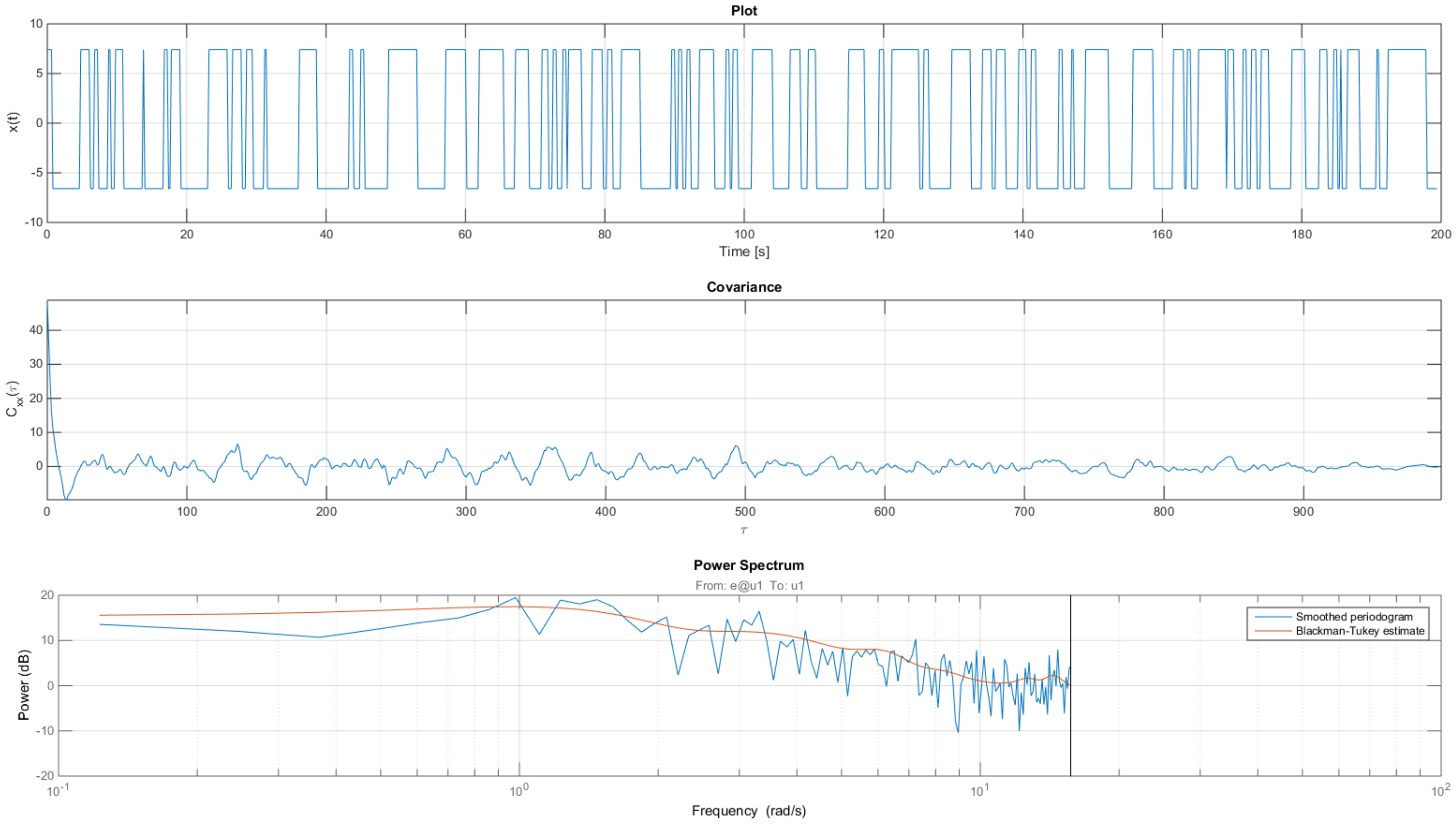
- Matlab command pexcit always returned 50 (degree of persistence of excitation, its calculated based on min(n/3, 50) where n=rank(Ru))

Input Covariance – Spectrum Analysis 2/5

The following table summarise the results of the Anderson Test, with confidence 90%, considering all samples of each set.

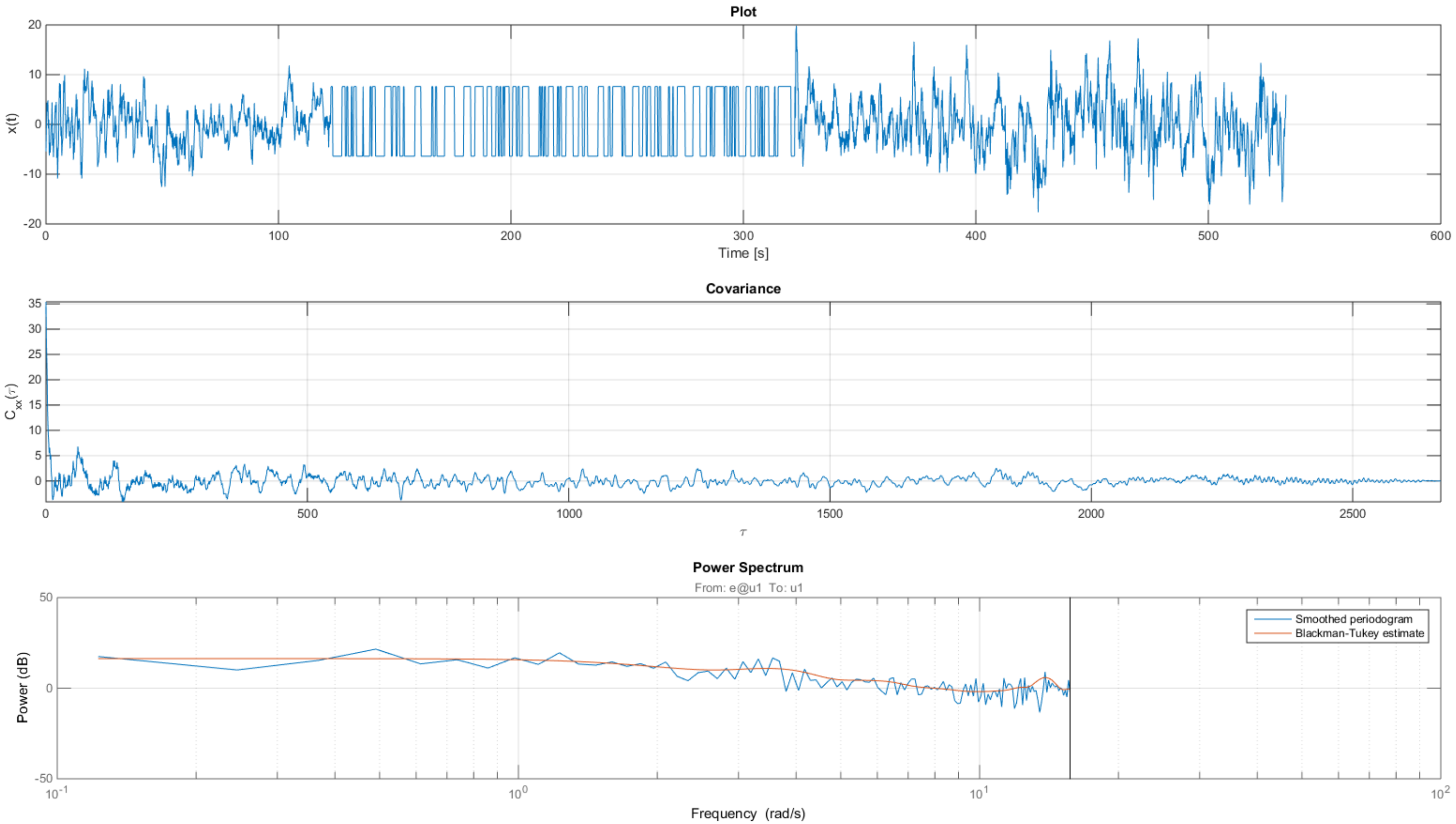
Ratio of violation	**1st Exp**	**2nd Exp**	**3rd Exp**
Full data	0.2245	0.2343	**0.2231**
1st portion	0.3154	0.4293	0.4110
2nd portion	**0.1785**	0.3049	0.2995
3rd portion	0.4483	0.7414	0.9138

Input Covariance – Spectrum Analysis 3/5



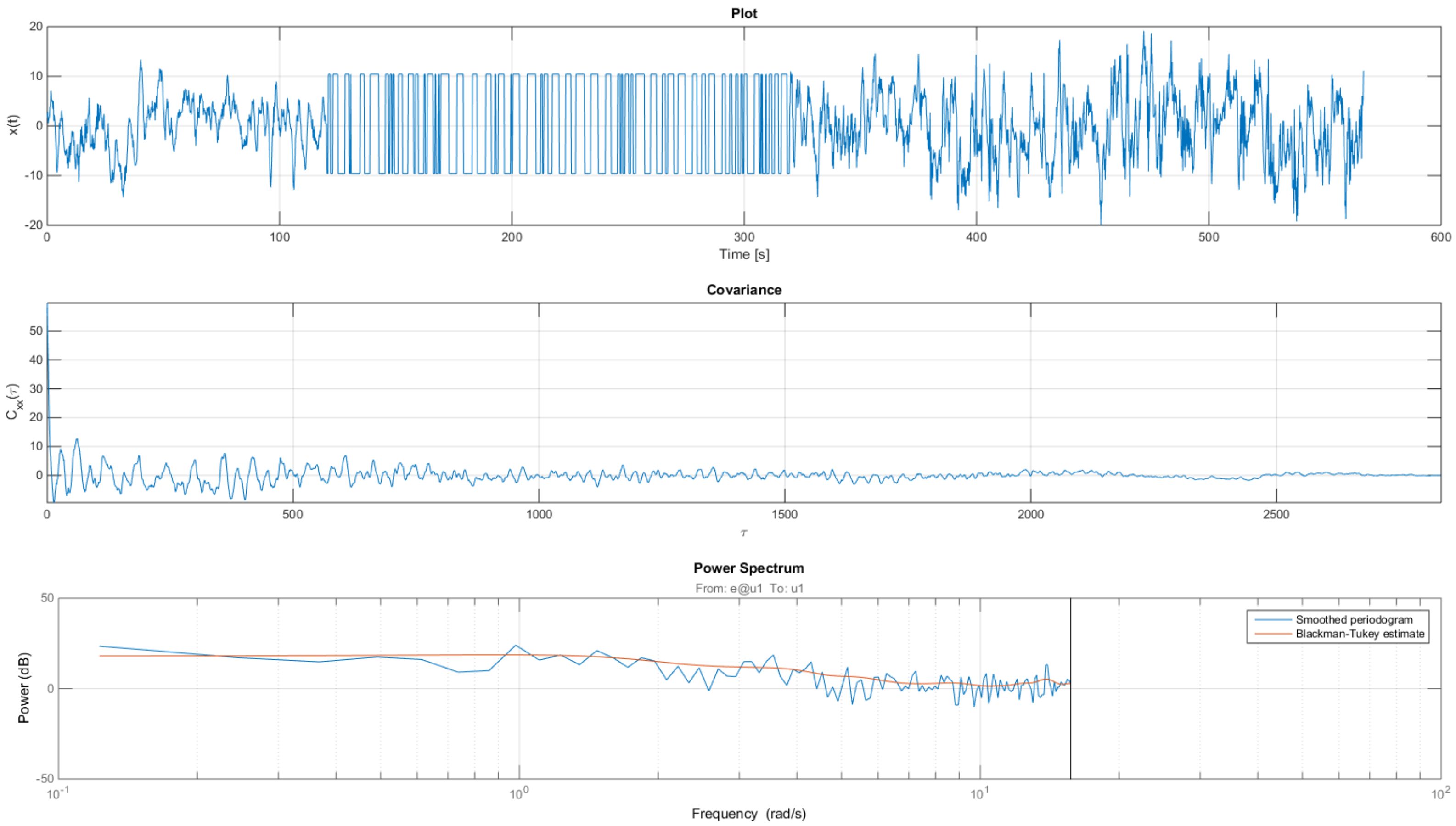
1rd experiment, 2nd data set

Input Covariance – Spectrum Analysis 4/5



1rd experiment, full data set

Input Covariance – Spectrum Analysis 5/5



3rd experiment, full data set

IMPULSE RESPONSE 1/3

Main two approaches to estimate the impulse response are:

1. Identify a FIR model
2. Make use of correlation analysis

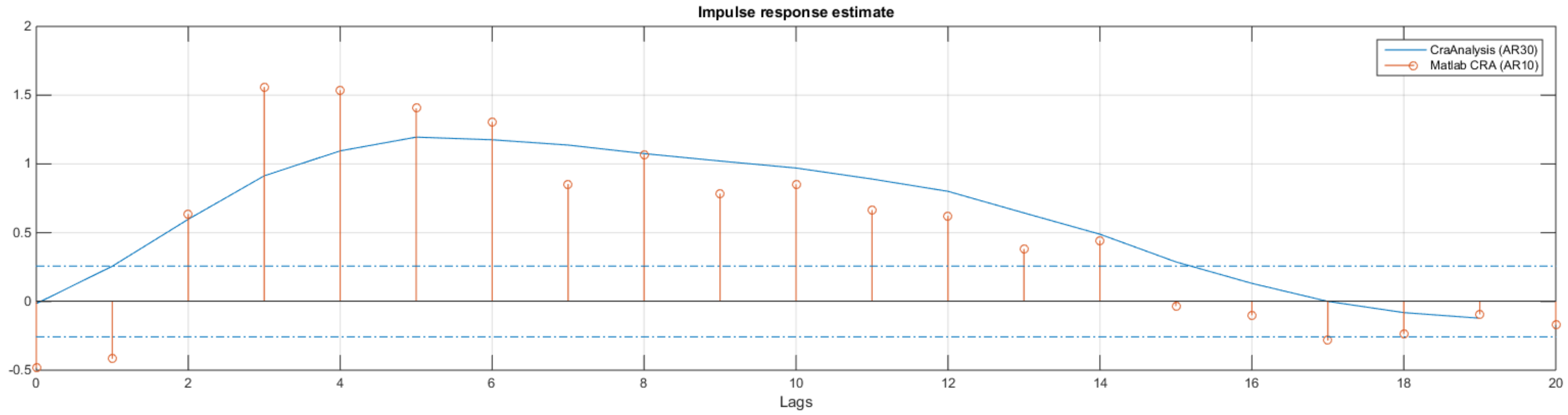
From this we can make a rough estimation of the impulse response, and together with the measured data, try to estimate the input delay (dead time).

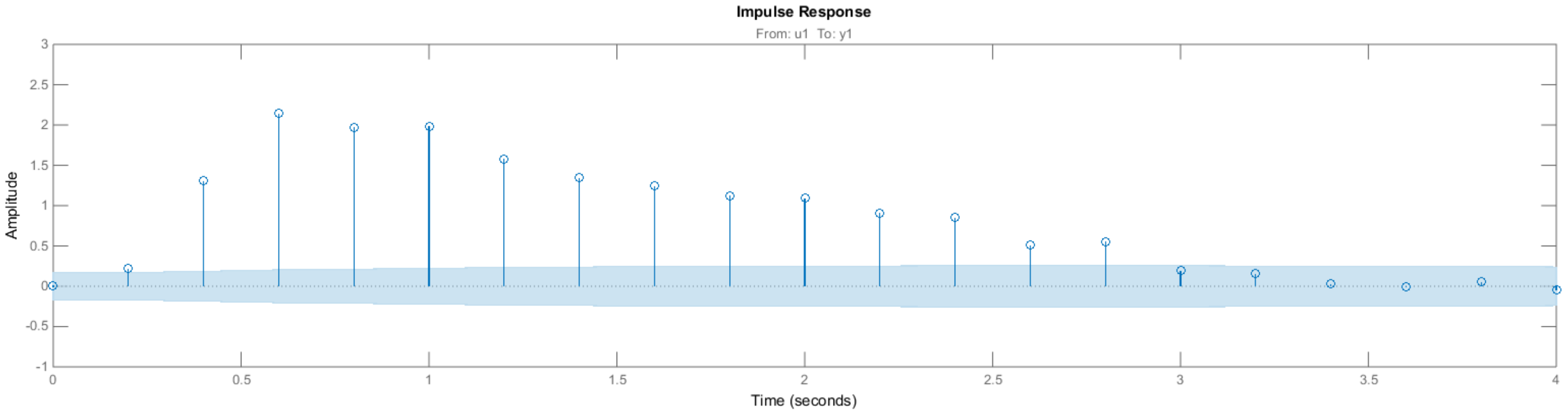
The correlation analysis works in the following way:
$y(n) = \sum_{k=1}^{\infty} h(n-k)u(k)$, suppose $u(n) = A(z)e(n), e \sim wn(0, \lambda^2)$ and $\exists A^{-1}(z)$ causal filter, then consider $\hat{y} = A^{-1}(z)y, \hat{u} = A^{-1}(z)u$ (we still have the same impulse response if we apply the filter to both the signals):

$$R_{yu}(\tau) = E[y(n)u(n-\tau)] = \sum_k h(k)E[u(n-k)u(n-\tau)] = \sum_k h(k)R_u(\tau) = \lambda h(k)$$

$$\Rightarrow h(\tau) = \frac{R_{yu}(\tau)}{\lambda}$$

IMPULSE RESPONSE 2/3 – Estimation





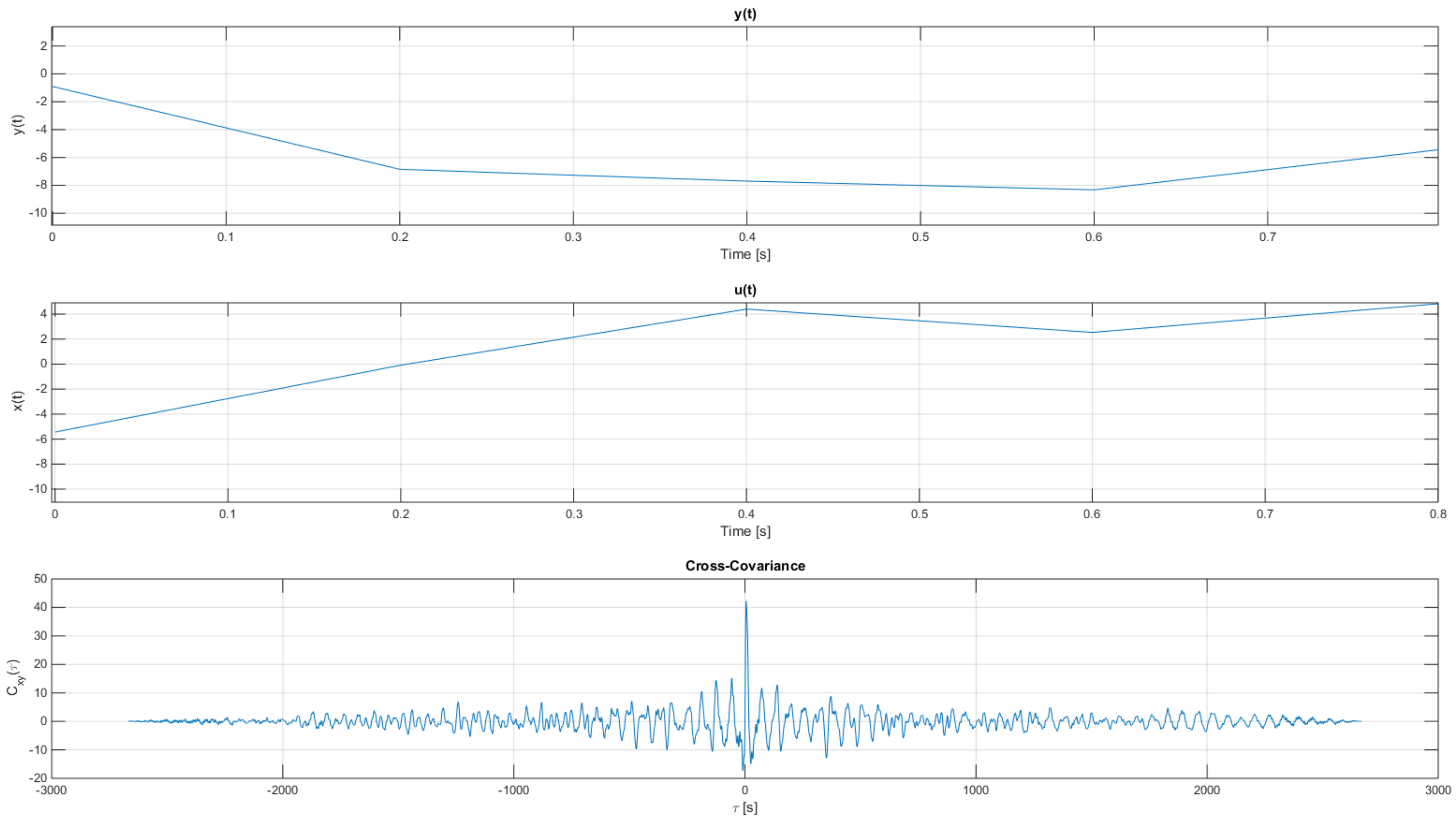
First image is based on correlation analysis.

Second one is based on the identification of a fir model up to n=20.

Seems like that
A positive input
Is delayed of 2-3 lags (See confidence region)

Tested on all the datasets, quite similar behaviour.

IMPULSE RESPONSE 3/3 – Data Analysis



Effect of u >0 after 2-3 Ts = 0.4-0.6 sec

Model identification

MODEL IDENTIFICATION - PEM

Based on PEM approach ⇒ minimize a loss function $l(\bullet)$ (or a distance) function of $\epsilon(t) = y(t) - \hat{y}(t)$,where $\hat{y}$ is a function that predicts the value of $y(t)$ based on past values of y, u.

If $\hat{y}$ fully describes the output ⇒ $y(t) = \hat{y}(t) + e(t),\ e \sim wn(0, \lambda^2) \Rightarrow \epsilon \sim wn(0, \lambda^2)$.

How to obtain $\hat{y}$? In general we can describe a linear system in the following way:

$$y(t) = G(z^{-1}, \theta^0)u(t-k) + H(z^{-1}, \theta^0)e(t) \quad e \sim wn(0, \lambda^2), \theta \in \Re^n$$

Where G, H are rational functions of $z^{-1} = e^{-\frac{d}{dt}}$, effectively defining then a lag operator, i.e:

$$z^{-1} f(t) = e^{-\frac{d}{dt}} f(t) = f(t-1)$$

Parametrised in θ.
Then:

$$H(z^{-1}, \theta^0)^{-1} y(t) = H(z^{-1}, \theta^0)^{-1} G(z^{-1}, \theta^0) u(t-k) + e(t)$$
$$y(t) = (1 - H(z^{-1}, \theta^0)^{-1}) y(t) + H(z^{-1}, \theta^0)^{-1} G(z^{-1}, \theta^0) u(t-k) + e(t)$$
$$\Rightarrow \hat{y}(t) = (1 - H(z^{-1}, \theta^0)^{-1}) y(t) + H(z^{-1}, \theta^0)^{-1} G(z^{-1}, \theta^0) u(t-k)$$

MODEL IDENTIFICATION – WHICH MODEL

We don't know G,H of the true system, neither its parametrisation, then how is $\hat{y}$ built up in reality? (denote with G^0, H^0 the transfer functions of the true system, then:

- Define the numerator and denominator degrees of G, H
- Define the parametrization of these rational functions

Based on this we get various models: ARX, ARMAX, OE, BJ, where the first one has the simplest dynamics, whilst the BJ model has the most complex dynamics.

Then we choose θ that gives the lowest values of $l\big(\epsilon(t)\big) \Rightarrow min_{\theta} l(y(t) - \hat{y}(t, \theta))$ is the problem to solve.

In this project we are interested in simulation. In fact we want to find G s.t. $Gu(t-k) \sim y(t)$How to address this problem with the minimization problem?

MODEL IDENTIFICATION – WHICH MODEL

$$min_{\theta} l(y(t) - \hat{y}(t,\theta)) = min_{\theta} l\left(H^{-1}y(t) - H^{-1}Gu(t-k)\right) = min_{\theta} l(H^{-1}(y(t) - Gu(t-k)))$$

If we pose H=1 then we obtain a problem that address our project's request (an OE model)

How good may be the the estimated G?

1. Consider that y,u are correlated, there is a feedback
2. We are also trying to estimate a noise (measurement noise, etc…) with the input u

Though we obtain the best G s.t. $Gu(t-k) \sim y(t)$.

Now, how to choose the cost function?

MODEL IDENTIFICATION – LOSS FUNCTION

1. $l(\epsilon) = E[\epsilon(t)^2] = E[\left(y(t) - \hat{y}(t)\right)^2]$ mean square error
2. $l(\epsilon) = ||\epsilon(t)||_{L^2} = ||y(t) - \hat{y}(t)||_{L^2} = \int_0^T \left(y(t) - \hat{y}(t)\right)^2 dt$ L^2norm
3. $l(\epsilon) = sup_t\,|\epsilon(t)| = sup_t|y(t) - \hat{y}(t)|$ norm of the uniform convergence
4. $l(\epsilon) = mfit\left(\epsilon(t)\right) = \frac{||y(t)-\hat{y}(t)||_2}{||y(t)-Ey(t)||_2}$ matlab fit function

For sure the 3rd cost function is the most inappropriate, since it tries to minimise the maximum error, which may be given by a noise peak.
Minimising the 4th or the 1st is the same since we are using the Euclidean norm, which minimise the variance of the error. Again, we may have peaks given by the noise, though if we have enough points may have no influence => global convergence.
The L^2 norm is simply the first cost function.

MODEL IDENTIFICATION – LOSS FUNCTION

$\bar{J}(\theta) = min_{\theta} l\left(y(t) - Gu(t-k)\right) = E\left[\left(y(t) - Gu(t-k)\right)^{2}\right] \approx \frac{1}{T}\sum_{t=1}\left(y(t) - G(z^{-1},\theta)u(t-k)\right)^{2} = J(\theta)$

(and for the asymptotic pem theory we have uniform convergence for $N \rightarrow \infty$) .

Unfortunately it's a non linear problem, so we need iterative methods to solve the minimisation problem which is a derivation problem.

How do we analyse the fact that we are using data measured in a closed loop system? Is there a way to consider The experiment an open loop experiment? We don't know either the reference signal or the controller.

It is known, from the frequency analysis of PEM, that a fully modeled noise transfer function will give better results in closed loop experiments when compared to OE models (where best means $G \sim G^0$), whilst OE models are the best ones in OL experiments. In fact, from the identification experiments, BJ models had worse results compared to OE models.

Moreover, we can assume that the quadrotor noise plus the measurement noise is little compared to the input signal.

Thus we can assume the experiment to have u and e uncorrelated => open loop experiment.

Consider the following expression:

$E[\epsilon(t)^2] = E\left[\left\{H^{-1}\left(y(t) - Gu(t-k)\right)\right\}^2\right] = E\left[\left\{H^{-1}\left((G^0 - G)u(t-k) + H^0 e(t)\right)\right\}^2\right]$, if $E[ue] \approx 0$ then the input signal And the error are uncorrelated, and:

$E[\epsilon(t)^2] = E[\{H^{-1}(G^0 - G)u(t-k)\}^2] + E[\{H^{-1}H^0 e(t)\}^2]$, using the Parseval theorem:

$$2\pi E[\epsilon(t)^2] = \int_{-\pi}^{\pi} \frac{\left|G^0\left(e^{j\omega},\theta^0\right) - G\left(e^{j\omega},\theta\right)\right|^2}{|H(e^{j\omega},\theta)|^2} \phi_u d\omega + \int_{-\pi}^{\pi} \frac{\left|H^0\left(e^{j\omega},\theta^0\right)\right|^2}{|H(e^{j\omega},\theta)|^2} \lambda^2 d\omega$$

If H=1 the minimisation of $E[\epsilon(t)^2]$ is given by the minimisation of $\left|G^0\left(e^{j\omega},\theta^0\right) - G\left(e^{j\omega},\theta\right)\right|^2$. For this reason the OE model yields better results.

Can we apply a filter on data to improve the results? From the previous formula the frequency weighting is the Following:

$$\Delta = \frac{\phi_u}{|H(e^{j\omega}, \theta)|^2}$$

If we apply a filter $L(z)$ on the data a factor L appears on the expression multiplying both terms. Notice that L is s.t. $L^2 = \Delta$.

Unfortunately, it's difficult to obtain a filter expression for ϕ_u, therefore it's easier to use a filter L s.t. $L = H$. But, Notice that the spectrum of u is quite flat, therefore it's not a problem.

In this way all models converge to the OE ones.

MODEL VALIDATION

MODEL VALIDATION

In order to validate a model, it is necessary to compare it with another set of data (y, u), since using again the training set would lead to non affordable (biased) results.

Unfortunately, as it is easy to see, all three sets are quite similar; because of this we cannot rely validation process only and further analysis are needed, even though it is a first step which allows us to have a impression.

As discussed in previous slides, there are many cost function we can make use of but in this project we have chosen the minimization of the simulation error $E\left[\left(y(t) - \hat{y}(t)\right)^2\right]$.

The validation will be performed in this way:

1. *Choose a model class*
2. *Identify a model from one dataset using different order of complexity and keeping best fit*
3. *Validate data on the other two dataset and write down results*
4. *Repeat point 2. and 3. picking another dataset for training and the others for validation*
5. *Repeat from point 1. with another model class*

MODEL VALIDATION - ARX

*Training on 1 exp
A(z): 1 -1.9551 0.64593 0.70572 -0.41605 0.025549
B(z): 0 0 0.27464 -0.26442
Validation on 1 exp: 68.191 Validation on 2 exp: 63.8589 Validation on 3 exp: 86.9643

*Training on 2 exp
A(z): 1 -1.8858 0.50166 0.76105 -0.37101
B(z): 0 0 0.28013 -0.26904
Validation on 1 exp: 68.4208 Validation on 2 exp: 63.2204 Validation on 3 exp: 88.7234

*Training on 3 exp
A(z): 1 -1.8037 0.2948 0.93228 -0.41768
B(z): 0 0 0.17739 -0.033072 -0.1383
Validation on 1 exp: 67.6965 Validation on 2 exp: 62.7143 Validation on 3 exp: 85.5182

MODEL VALIDATION - OE

*Training on 1 exp
A(z): 1 -2.0798 1.5923 -1.0809 0.85391 -0.2781
B(z): 0 0 -0.17999 0.65694 -0.47025
Validation on 1 exp: 54.6866 Validation on 2 exp: 49.5952 Validation on 3 exp: 74.6829

*Training on 2 exp
A(z): 1 -2.046 1.4758 -0.99924 0.89773 -0.32148
B(z): 0 0 -0.19542 0.68029 -0.47878
Validation on 1 exp: 54.9233 Validation on 2 exp: 49.3283 Validation on 3 exp: 75.5403

*Training on 3 exp
A(z): 1 -3.7358 5.1579 -3.0089 0.49021 0.096966
B(z): 0 0 -0.077889 0.61883 -1.3542 1.1687 -0.35517
Validation on 1 exp: 58.1653 Validation on 2 exp: 49.5579 Validation on 3 exp: 62.7157

MODEL VALIDATION

OE performs better than **ARX** models as expected from the previous theoretical discussion.

ARMAX and **BJ** give exactly the same results as **OE** models using simulation focus. This is due to the fact that Matlab uses a 2 step identification, where in the first steps assume the T.F. from e to y to be 1, to identify G. Then The T.F. from e to y is identified. Therefore we obtain the OE model for G.

ARMAX and **BJ** give different results using prediction focus, but even with high complexity models they don't get as close as to the OE model. This might be a clue of being an open loop experiment since for closed loop data Detailed $e\ to\ y$ T.F.s will give a better result (because of the compromise).

A deeper analysis can be done with a Matlab script we wrote (compareModels) which shows graphically the properties of the two models, as:

- Simulated output compared to actual data, Spectra analysis
- Absolute error: mean, variance and max
- Covariance of simulation error, Anderson Test

Both models pass Anderson Whiteness Test but **OE** has a better fitting and less error variance.
See attached Matlab scripts for more detailed info.

MODEL VALIDATION – DYNAMICS

Another way to discriminate among different models is to look at their dynamic behaviour.

Here we will analyze:

- Poles and Zero
- Step response

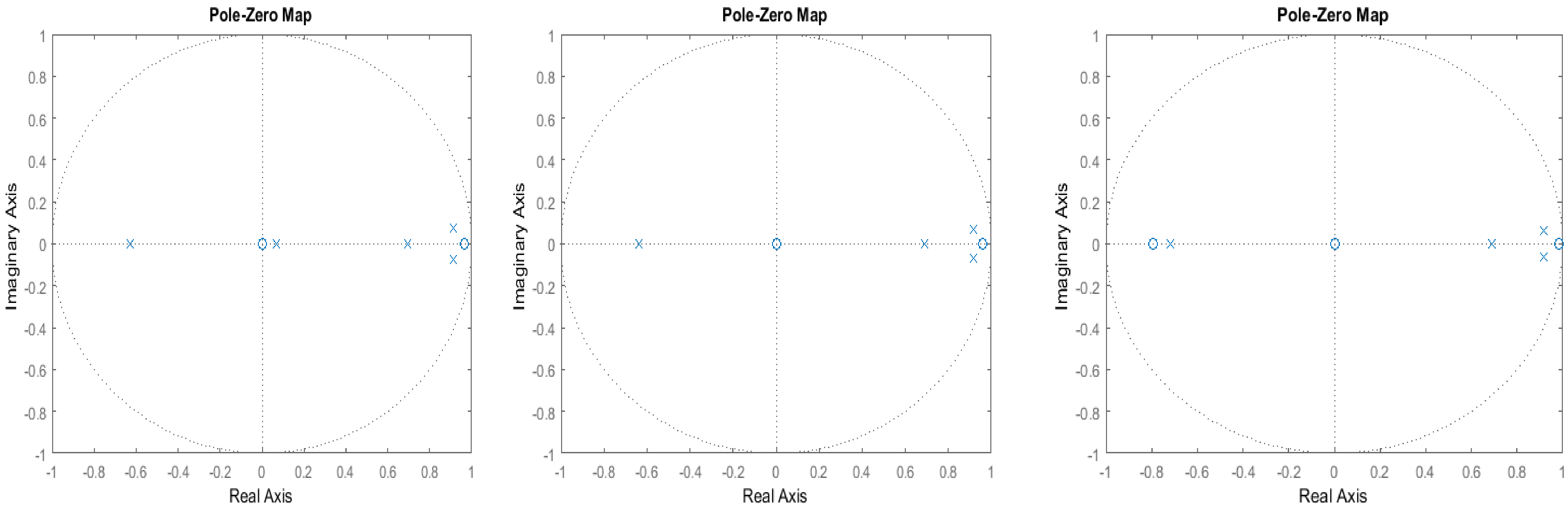
trying to understand which dynamic best suit the actual system.

We will also attempt to synthesize new models by shaping the pole-zero plot.

Each model class is trained using the three datasets with simulation focus, leading to three different models for each class. Model classes tested: ARX, OE.

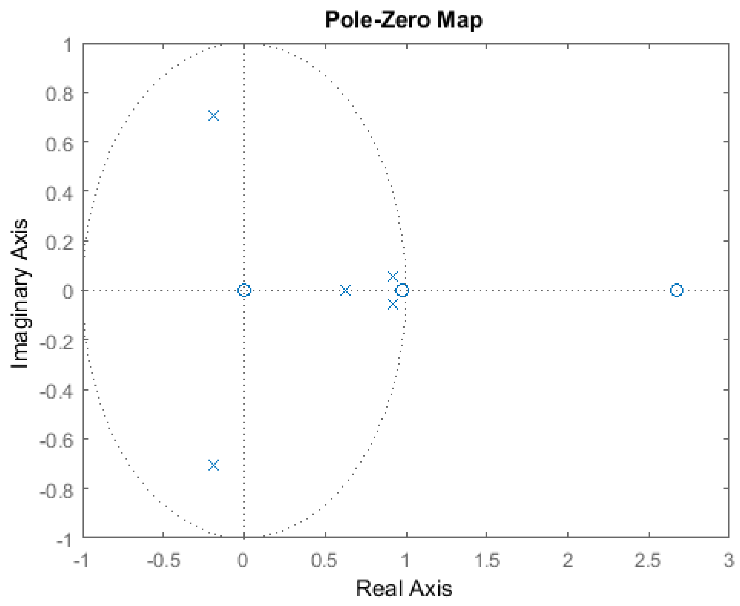
MODEL VALIDATION – DYNAMICS

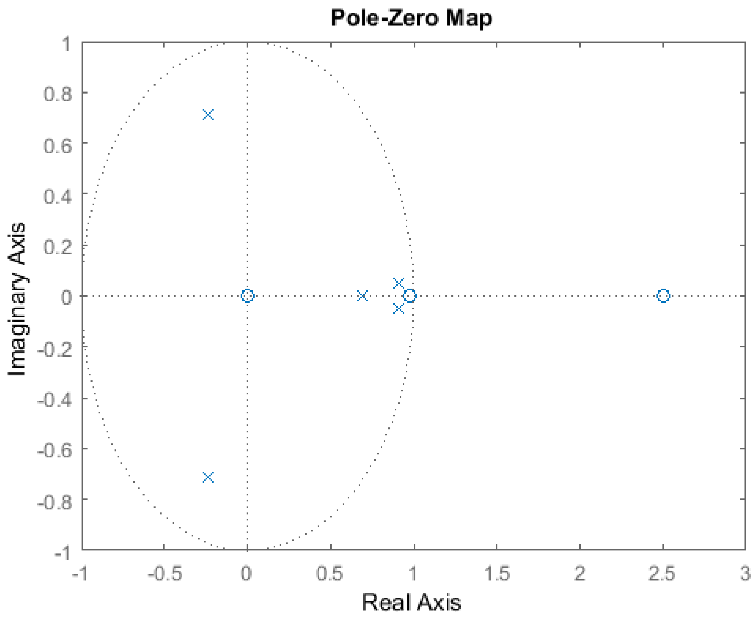
ARX MODELS

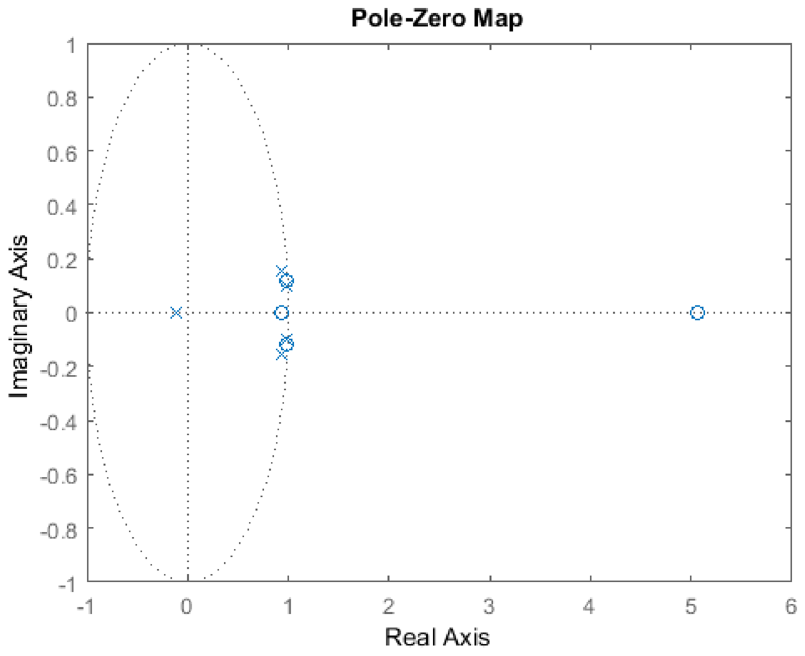


MODEL VALIDATION – DYNAMICS

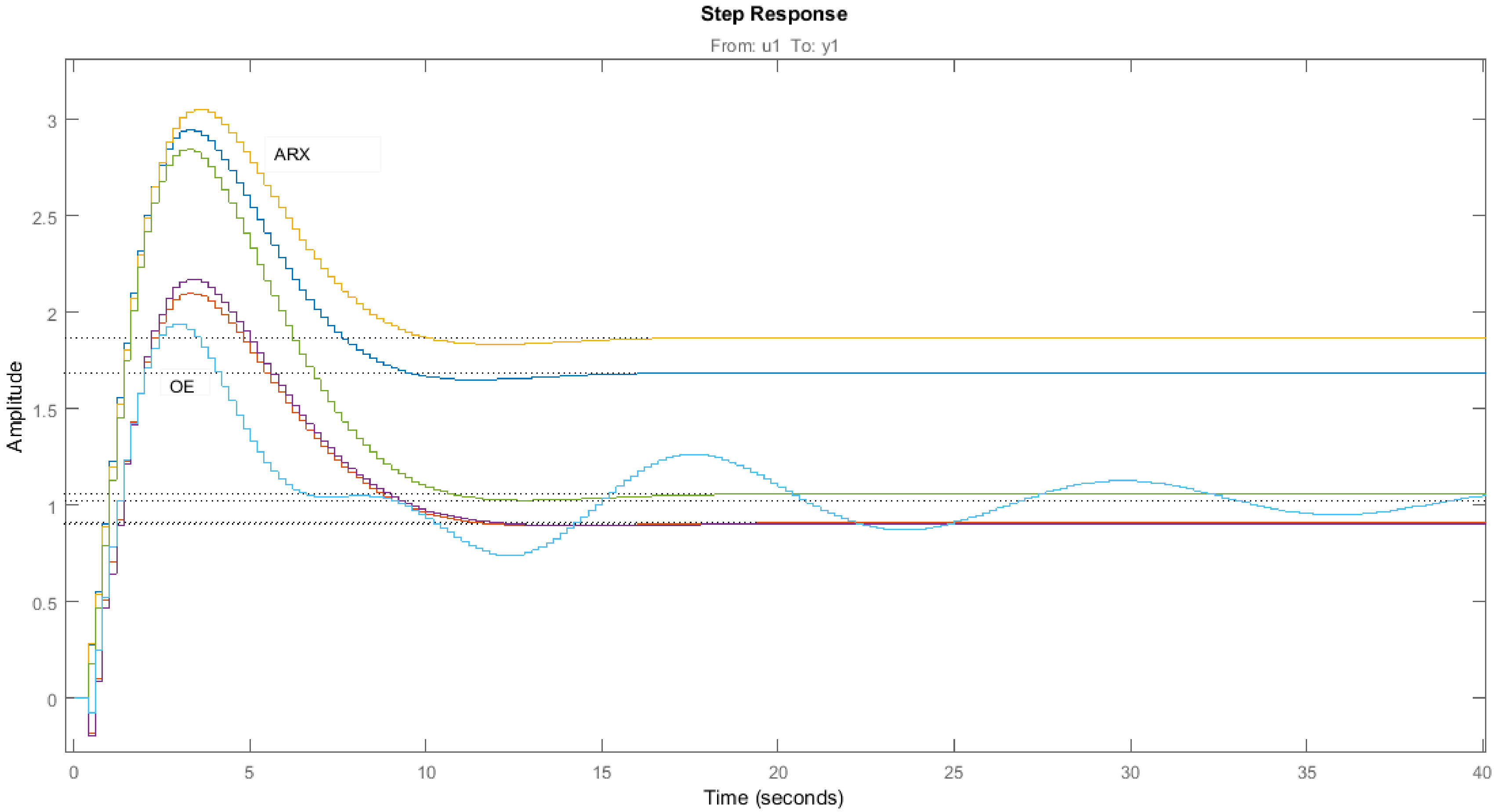
OE MODELS







MODEL VALIDATION – DYNAMICS



MODEL VALIDATION – DYNAMICS

Looking at the step responses we can evaluate (roughly) the gain of the plant and use this piece of information to discriminate some models.

The third **OE** model has a quite long settling time which make us think that this model is not very reliable (indeed the third dataset gives very different result in every identification).

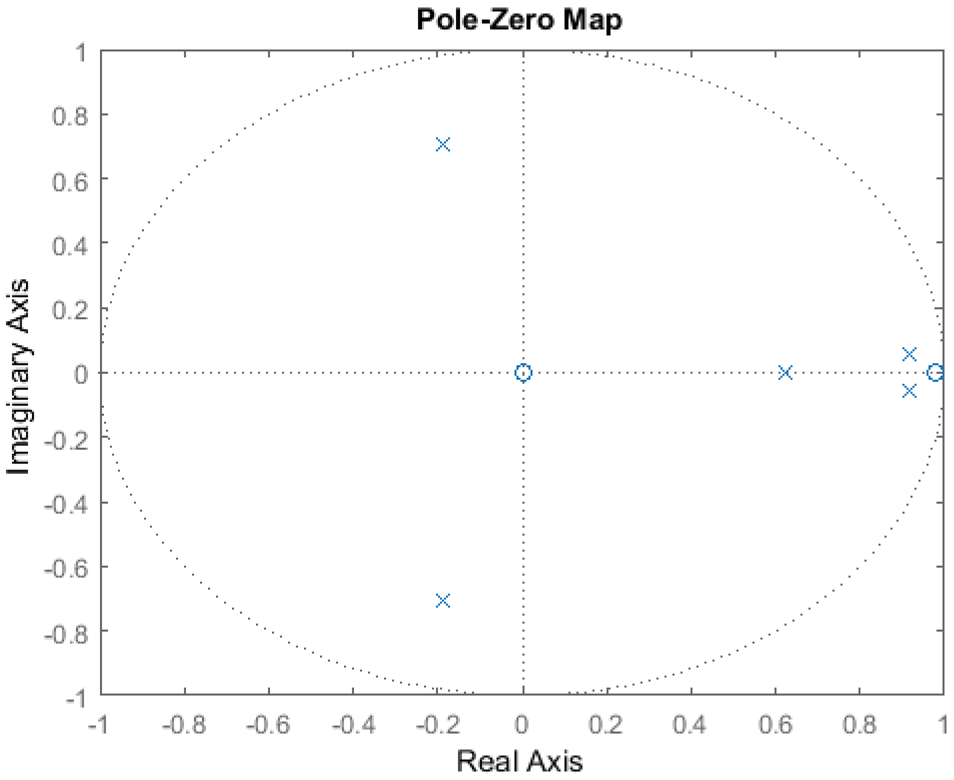
Overshoot is more significant (in percentage) in **OE** models rather than **ARX** ones.

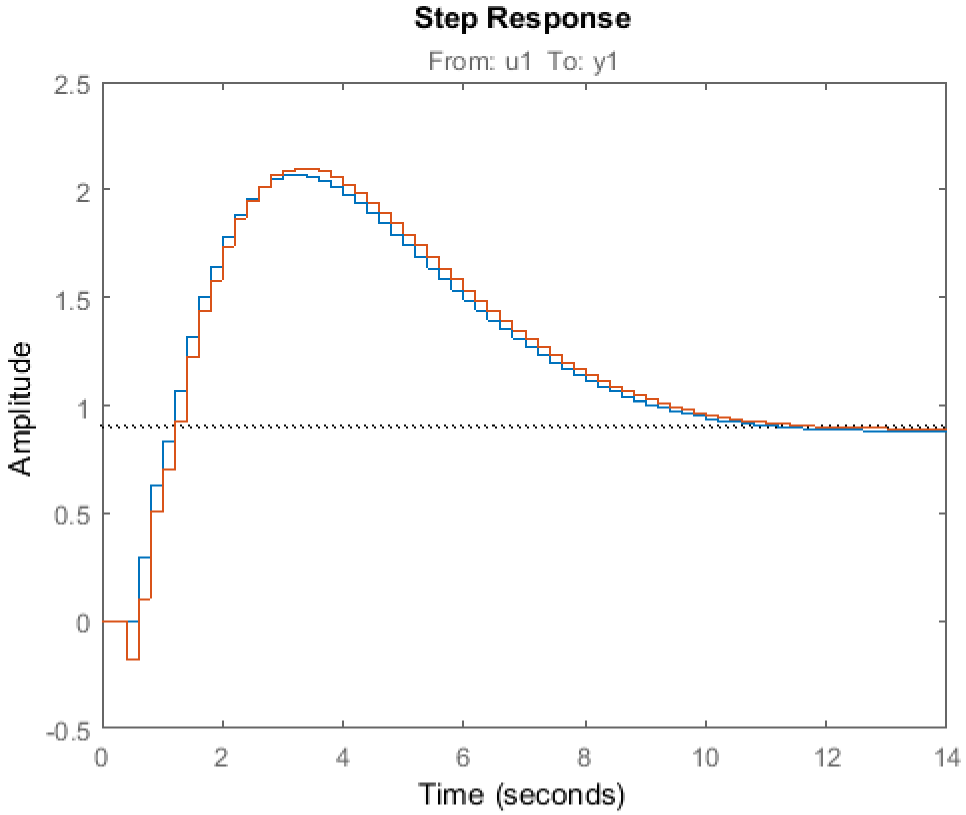
Looking at zero-poles plot we can see that they are not very different, indeed in both **ARX** and **OE** we have the same set of zero and poles around $z = 1$.

The first and second **ARX** models are equal since we can cancel a zero in the origin and the pole close to it.

The first two **OE** models are very similar and they both have a zero outside the unit circle, which makes us think that an overparametrization might have occurred; in order to verify this option we can cancel that zero and see wheter or not the dynamics is somehow affected.

MODEL VALIDATION – DYNAMICS





Step response leads to the same results and even the «compareModels» analysis tells us there is little difference in terms of absolute error.

MODEL VALIDATION – DYNAMICS

The best way to constrast overparametrization is not shaping directly the zero-pole plot, but to perform another identification using a lower order.

We have then modified the identification script in order to see how much the variance of the simulation error varies among different orders. If there is little difference we will accept a worse performance in respect to a lower order.

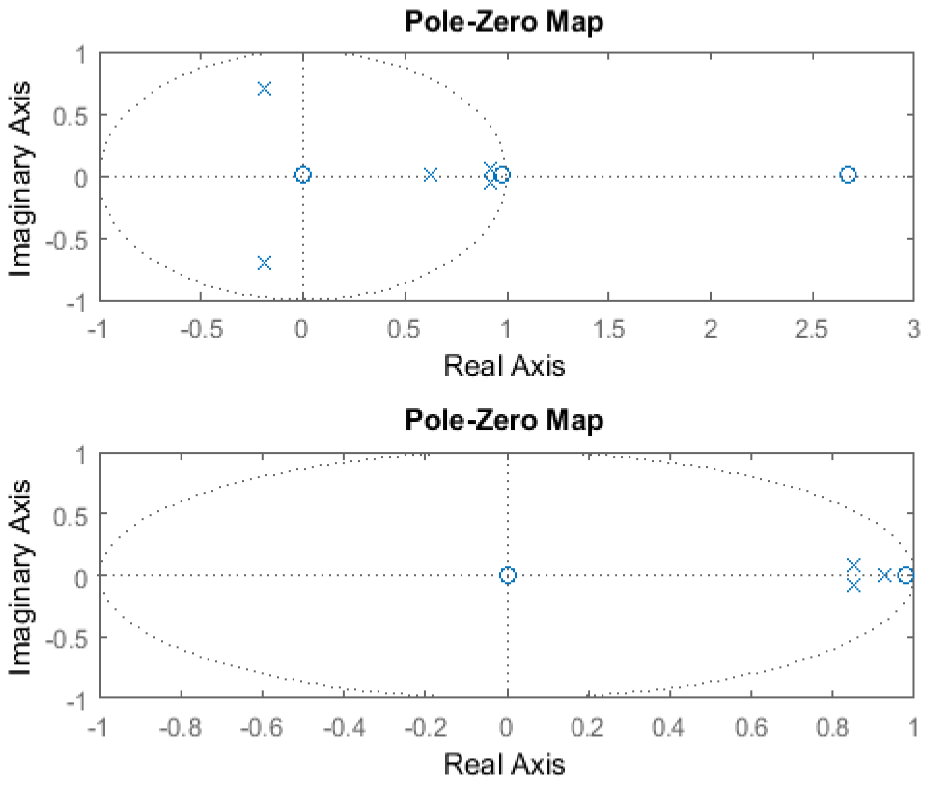
A very good compromise in terms of performances has been found in the model **OE33** which has always a zero outside the unit circle though.

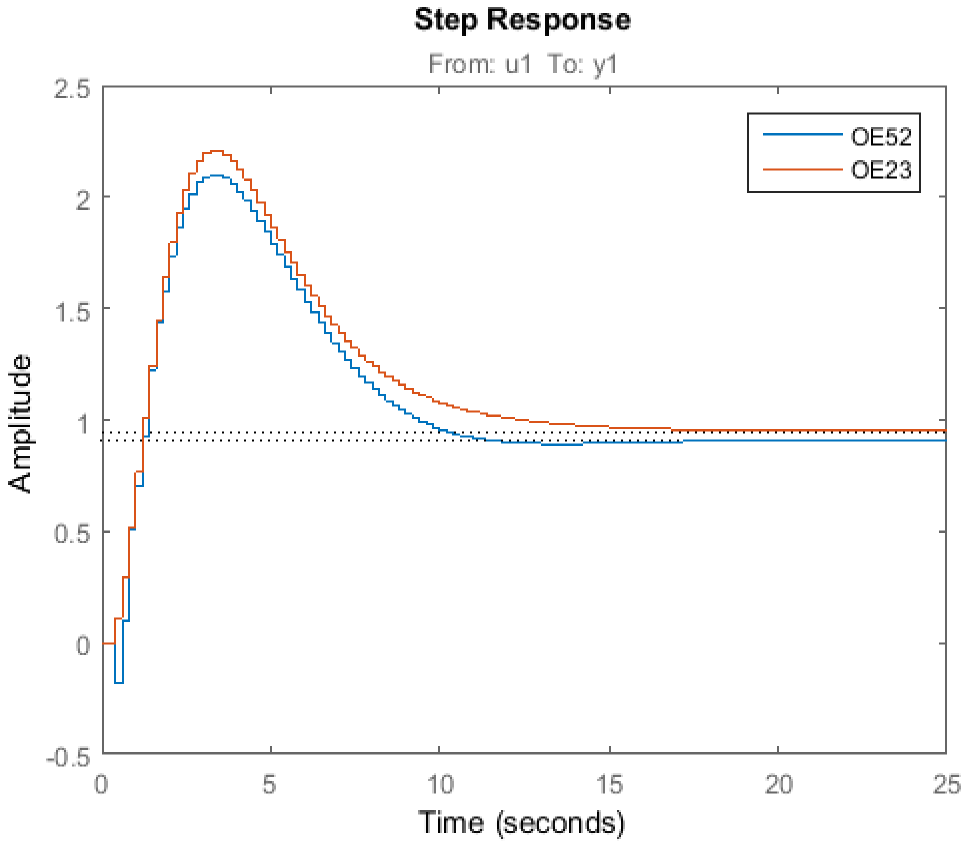
We have then tried both to lower down the order of the identification and to cancel «by hand» that zero: the best is the former which still passes the Anderson Test.

In the next slide we show the comparison between the more complex **OE35** and this **OE23**:

A(z):	1	-2.6222	2.2976	-0.67315
B(z):	0	0	0.11213	-0.10996

MODEL VALIDATION – DYNAMICS





«compareModels» tells us the variance increases of one point only.

CONCLUSION

We finally conclude by comparing our models with the one provided as reference (state-space).

The state space model is much more complex since it is a 5th order system (5 poles and 5 zeros) whilst our final choice is a 3rd order (3 poles and 2 zeros).

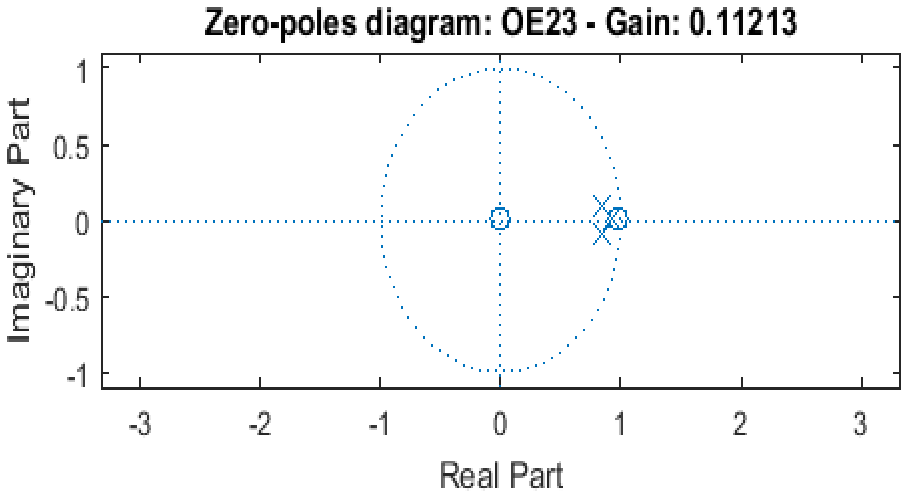
On the other hand it has a better fitting of the data, though the difference in terms of absolute error is minimal.

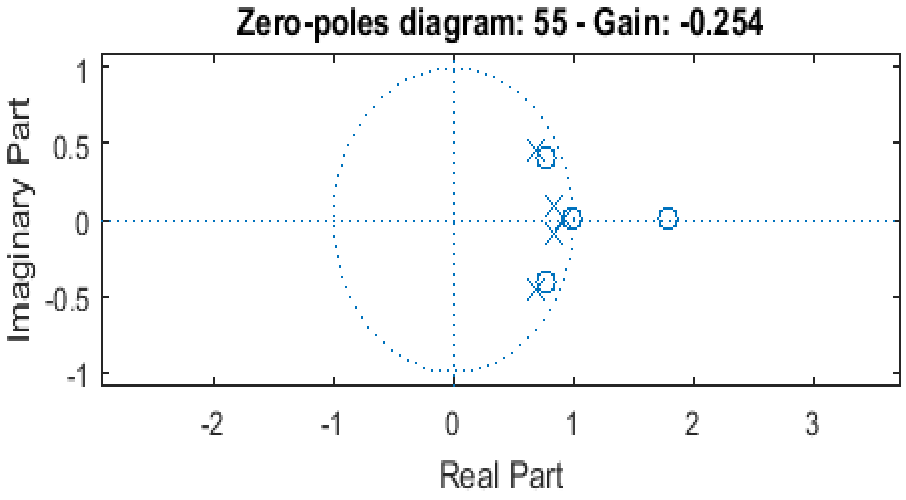
Also in terms of Error-to-Signal Ratio the state space is slightly better:

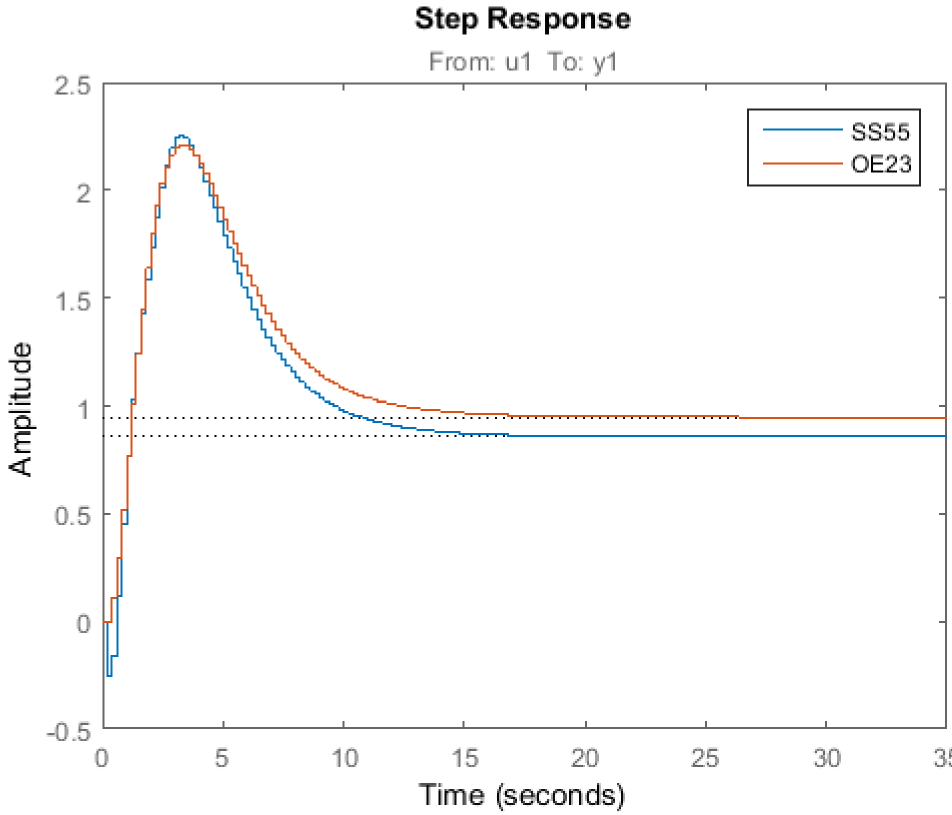
- SS55: 36,69%
- OE35: 37,87%
- OE23: 38,85%

In the next slides we show some graphical comparison of the two models.

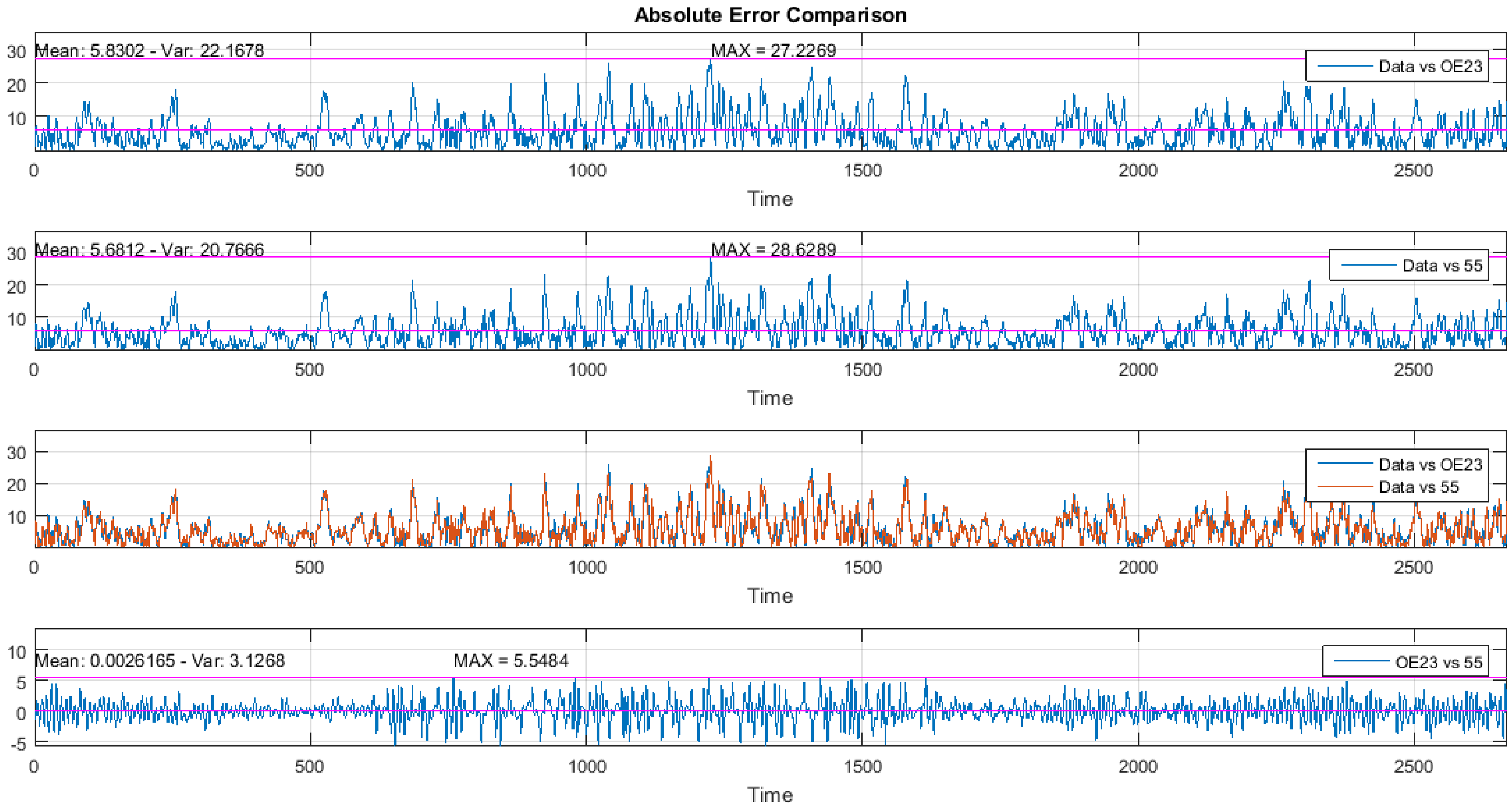
CONCLUSION


Zero-poles diagram: OE23 - Gain: 0.11213
Imaginary Part
Real Part


Zero-poles diagram: 55 - Gain: -0.254
Imaginary Part
Real Part


Step Response
From: u1 To: y1
Amplitude
Time (seconds)
SS55
OE23

CONCLUSION


Absolute Error Comparison
Mean: 5.8302 - Var: 22.1678
MAX = 27.2269
Data vs OE23
Time
Mean: 5.6812 - Var: 20.7666
MAX = 28.6289
Data vs 55
Time
Data vs OE23
Data vs 55
Time
Mean: 0.0026165 - Var: 3.1268
MAX = 5.5484
OE23 vs 55
Time

OTHER IDEAS

OTHER IDEAS

1. Non linear identification: tried to use matlab toolbox: Bad results
2. Interpolation of data, sampling at higher frequency, in order to obtain more training data: Bad results
3. Best result in terms of minimum variance found setting the input delay to 1: 5^{th} order system with poles and zeros condensed in a small area, suggesting a bad identification. Moreover the input delay should be higher than 1.

Something more to try:

1. Instrumental variable method
2. New experiments including impulse and step response
3. Compare results with a physical model