# Advanced Model Based Control

## Group assignment - Group 3

Matthieu Fontaine     Karl Gudmundson     Xiyu Fu

26 January 2018

# Outline

# Outline

# System Identification

## Identification of a physical model

- Using a priori information about the system during the identification step is crucial. We want to create a 'grey' box model.

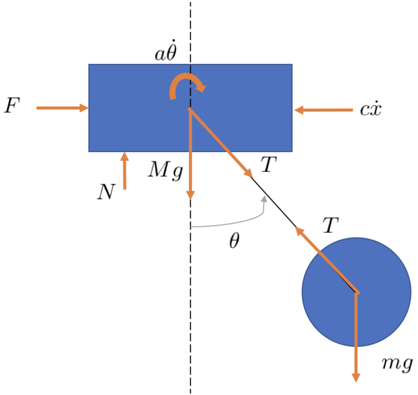



Figure: Physical model of the overhead crane

- General equations of the system are derived using (simple) multibody dynamics and kinematics

$$(M+m)\ddot{x} = mL(\dot{\theta})^2 \sin\theta - mL\ddot{\theta}\cos\theta + F - c\dot{x}$$

$$mL^2\ddot{\theta} = -mgL\sin\theta - mL\ddot{x}\cos\theta - a\dot{\theta}$$

- The second equation describes the relationship between the input speed $u$ and the angle (output $y$). However, it's a nonlinear relationship. In practice the system dynamics is approximated by a linear model.
- In what range is this model "valid"? The angle $\theta$ has to be small.

# System Identification

## Identification of a physical model

- The model can be linearized using Taylor series around $\theta = 0$

$$mL^2\ddot{\theta} = -mgL\theta - mL\ddot{x}\theta - a\dot{\theta}$$

- Based on this second-order ODE, the transfer function between the speed of the cart and the angle of the mass is established:

$$\frac{\Theta(s)}{\dot{X}(s)} = G_{\text{mass}}(s) = \frac{-mLs}{mL^2s^2 + as + mgL}$$

- The desired transfer function between the $x$-position of the mass and the speed of the cart is deduced using Taylor expansion:

$$x_{mass} = x + L\sin\theta \approx x + L\theta$$

- This leads to a transfer function that is made of the sum of two ingredients; a pure integrator and $G_{\text{mass}}$ :

$$\frac{X_{mass}(s)}{\dot{X}(s)} = \frac{X(s) + L\Theta(s)}{\dot{X}(s)} = \frac{as + mgL}{mL^2s^3 + as^2 + mgLs}$$

# System Identification

## Identification of a physical model

- Bode plot of the transfer function between the speed of the cart and the $x-$position of the mass. The different physical values are measured on the test bank. ($L = 0.35$ [m], $m = 0.7$ [kg], $g = 9.81$ [m/s·s], $a = 0.08$ [N·s/rad])

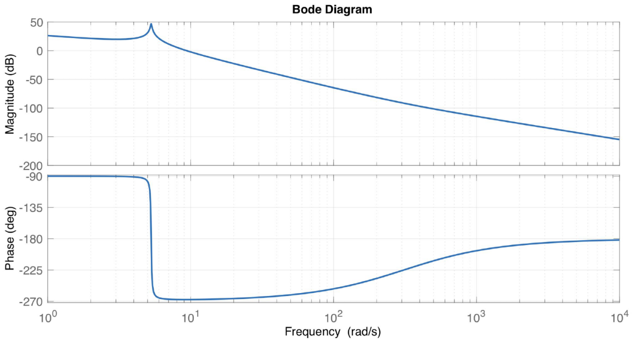



Figure: Bode plot of the physical model of the overhead crane

# Outline

# System Identification

## Experiments

- Collecting information about the system is required to derive a non-parametric model.
- To do this, one can measure the system response while the system is in normal operation.
- One must choose an appropriate excitation signal. This means an excitation signal that optimizes a given criterion and that takes a priori knowledge into account.
    a. The bandwidth of the system lies inside a frequency range of 0.1 and 10 Hertz, as seen in the physical model.
    b. A good frequency resolution (0.01 Hertz) is required because of the resonance.
- But what type of excitation signal is needed?

- A multisine excitation is the best signal one can choose, since it has the following properties:
  a. Low crest factor (equal to 1.45).
  b. Low time factor (equal to 1).
  c. Ability to generate an arbitrary power spectrum which is required to keep the signal-to-noise ratio constant and to have a constant relative error.
  d. No leakage.
- The error in the phase and the amplitude of the FRF is proportional to

$$\epsilon = \frac{1}{\sqrt{n}}(S/N)^{-1}_{\text{output}}$$

# Outline

## System Identification

- Identification of a physical model
- Experiments
- Non-parametric FRF
- Parametric identification
- Model validation
- Uncertainty model
- Noise and non-linearities

## Controller Design

- $H_\infty$ loop shaping

## Trajectory tracking

- Discrete controller
- Controller validation

## Iterative Learning control

- Formulation of ILC
- Implementation

## Conclusion

# System Identification

## Non-parametric FRF

- Input and output frequency bins are disturbed by stochastic errors.
- The exact transfer function is given by:

$$G(k) = \frac{Y(k)}{U(k)}$$

- The empirical transfer function is given by:

$$G_m(k) = \frac{Y_m(k)}{U_m(k)} = G(k)\frac{1 + \frac{N_y(k)}{Y_{(}k}}{1 + \frac{N_u(k)}{U_{(}k}}$$

- To reduce the effect of stochastic errors, one can consider averaging techniques. In our case, we have used a sequence of 4 spectral input-output sets of data.
- These sets of data have been obtained by applying a DFT on 4 non overlapping time records.

- One can use either linear estimators (least square estimator) or nonlinear estimators.
- However when it is possible to synchronize the measurements:

$$\hat{G}_{EV} = \frac{1/M \sum_{i=1}^{M} Y_{m,i}(k)}{1/M \sum_{i=1}^{M} U_{m,i}(k)}$$

- This actually corresponds to the ML (maximum-likelihood) estimator. This estimator has nice properties. Especially since its mean square error is made of only a nonsystematic error term:

$$MSE(G\hat{(}k)) = b^2(\hat{G}(k)) + \text{var}(\hat{G}(k))$$

# System Identification

## Non-parametric FRF

- Using the ML estimator for 4 synchronized non-overlapping measurements, one can produce the following FRF using the *nonparam_ident()* function from the LC toolbox:

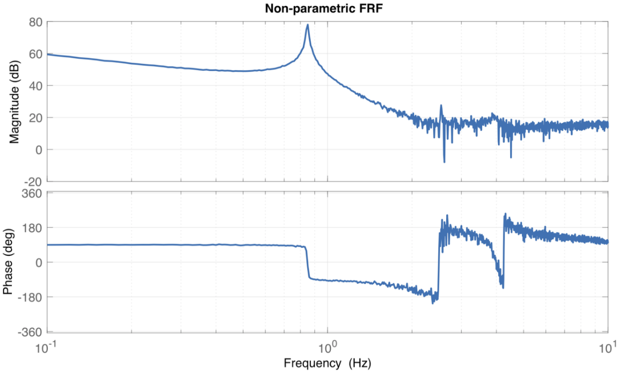



Figure: Bode plot of the non-parametric estimation using MLE.

# Outline

- Frequency-domain identification has some advantages compared to time-domain identification.
- Two approaches are possible: deterministic or stochastic. Here we are going to consider the deterministic approach.
- A general model of the transfer function in the $\Omega$-domain is given by:

$$G(\Omega, \boldsymbol{\theta}) = \frac{B(\Omega, \boldsymbol{\theta})}{A(\Omega, \boldsymbol{\theta})} = \frac{\sum_{i=0}^{n_b} b_{n_b - i} \Omega^i}{\sum_{i=0}^{n_a} a_{n_a - i} \Omega^i}$$

- In the equation above, $\boldsymbol{\theta}$ is a vector containing all the unknown coefficients of the parametric transfer function. Knowing the order of the system thanks to the physical model, we obtain:

$$G_{\text{param}}(s) = \frac{b_1 s + b_0}{a_3 s^3 + a_2 s^2 + a_1 s} \approx \frac{as + mgL}{mL^2 s^3 + as^2 + mgLs}$$

- Therefore the $\boldsymbol{\theta}$ vector is in our case given by:

$$\boldsymbol{\theta} = \begin{bmatrix} b_0 & b_1 & a_1 & a_2 & a_3 \end{bmatrix}$$

- We have to determine 5 unknown coefficients (or equivalently 4 unknown coefficients)

$$\boldsymbol{\theta} = \begin{bmatrix} \widetilde{b}_0 & \widetilde{b}_1 & \widetilde{a}_1 & \widetilde{a}_2 & 1 \end{bmatrix}$$

- How do we do that? By fitting the non-parametric model to a parametric model according to an optimal criterion.
- The optimal criterion that we are going to use here the non-linear least square criterion:

$$V_{NLS}(\boldsymbol{\theta}) = \frac{1}{2} \sum_{k=1}^{N} |G_m(k) - G(\Omega, \boldsymbol{\theta})|^2$$

- For the overhead crane, we are going to apply that method using the *param_ident()* function from LC toolbox.
- However, we are going to neglect the high frequency zero computed, since it is located at a frequency much higher than the bandwidth of the system. It does not play any role in this controller.
- Doing this will also simplify the work for the ILC, which is discussed later.
- During the model validation we are going to show why it can be safely removed.
- We eventually end up with the following parametric model:

$$G_{par} = \frac{-16604}{s^3 + 0.0611s^2 + 28.39s}$$

# System Identification

## Parametric identification

- The Bode plot of the parametric estimate and its comparison with the non-parametric estimate.
- The next step is model validation.

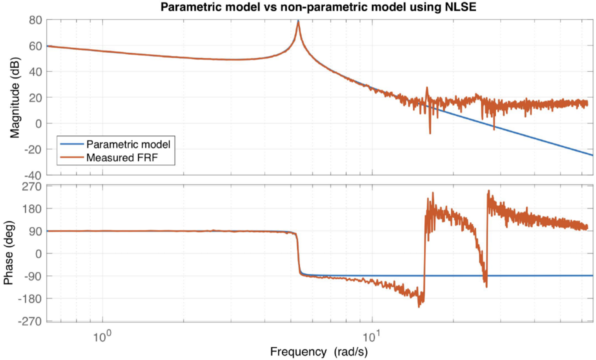



Figure: Bode plot of the parametric estimation using NLSE.

# Outline

# System Identification

## Model validation

- The last part of the system identification process is crucial.
- Confront the model with as much information as possible.
- First a physical check of the parameters is a good idea:

$$G_{\text{param}}(s) = \frac{b_1 s + b_0}{a_3 s^3 + a_2 s^2 + a_1 s} \approx \frac{as + mgL}{mL^2 s^3 + as^2 + mgLs}$$

| | Physical quantities | Parametric model |
|---|---|---|
| $a_1$ | $g/L = 28.03$ | 28.39 |
| $a_2$ | $a/mL^2 = 0.08$ | 0.0611 |
| $a_3$ | 1 | 1 |
| $b_0$ | $g/L \cdot \text{Gain} = -\ 5606$ | -16604 |
| $b_1$ | $a/mL^2 = 0.08$ | 0 (neglected) |
| $\omega_{res}$ | 5.29 [rad/s] | 5.33 [rad/s] |
| $\xi_{res}$ | 0.00857 | 0.00601 |

# System Identification

## Model validation

- The pole-zero map comparison of the parametric and the physical model is meaningful:

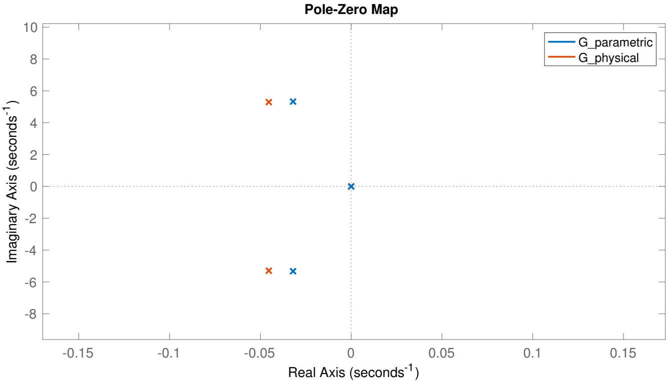



Figure: Pole-zero map for the two models.

# System Identification

## Model validation

- Graphical check of the frequency response functions of the physical and parametric models:

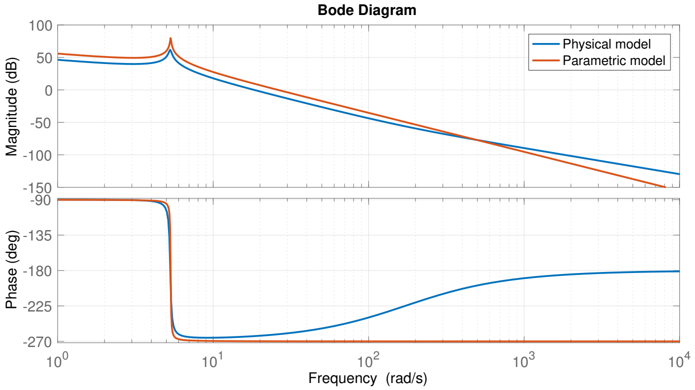



Figure: Bode plot for the two models.

# System Identification

## Model validation

- Another way to validate the model is empirical parameter confidence intervals.
- Obtain a set of 10 estimates of each parameter and compute the mean and variance of everyone.
- The first parameter is located inside the confidence interval. We can validate it.

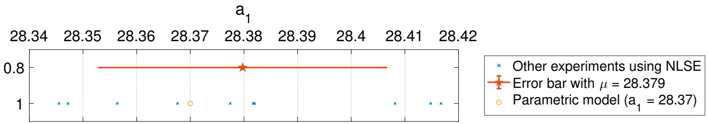



Figure: Confidence interval for $a_1$

# System Identification

## Model validation

- For the parameter $a_2$, the parametric model is accurate because the parameter is in the middle of the interval and really close to the mean value. We can validate $a_2$.

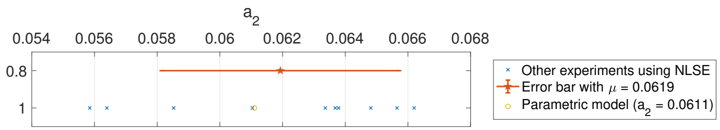



Figure: Confidence interval for $a_2$

# System Identification

## Model validation

- For the parameter $b_0$, the parametric model is also good since the parameter is within the confidence interval. Even if it's not very next to the mean value, the error is acceptable (deviation of 0.5 %). We can validate this parameter.

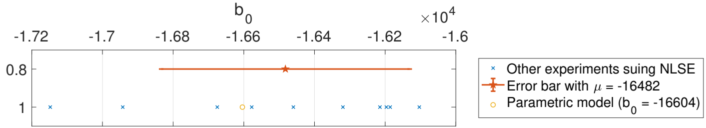



Figure: Confidence interval for $b_0$

# System Identification

## Model validation

- This parameter is a bit special because the confidence interval contains 0. And even worse, the center of the confidence interval is really close to 0.
- Removing this parameter is possible and it leads to the reduced model we have used so far.
- This parameter defines a high frequency zero that is hard to estimate due to high frequency measurement noise.

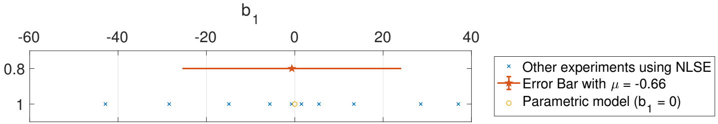



Figure: Confidence interval for $b_1$

# System Identification

## Model validation

- The last test performed to validate the model is a time domain check.

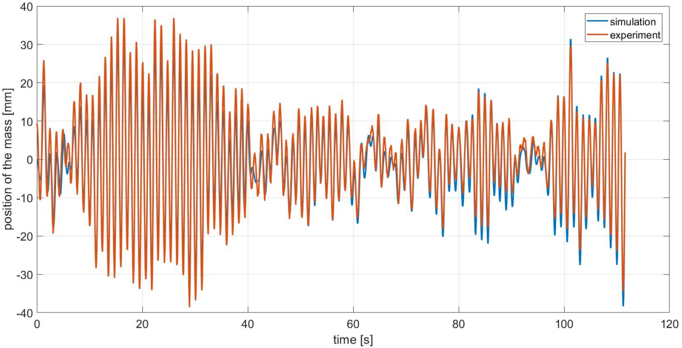



Figure: Experiment results and the simulation results from the identified model

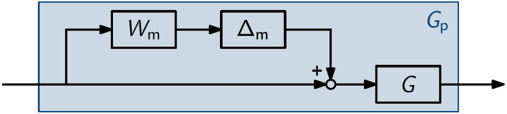
# Outline

- Uncertainty is dealt with in 3 steps:
  1. Determining an uncertainty set $\Pi$ for multiplicative uncertainty.
  2. Check **robust stability** to ensure that the system remains stable even with subject to uncertainty.
  3. Check **robust performance**. If RS is satisfied, determine whether the performance specifications are met for all plants in the uncertainty set.
- For this purpose several notations are introduced: $\Pi_m$, $G(s)$ and $G_p(s)$.
- Notice here that we are considering **parametric uncertainty**.

# System Identification

## Uncertainty model

- The multiplicative model describes unstructured uncertainty using this relationship: $\Pi_m = G_p(s) = G_s \cdot (1 + W_m(s)\Delta_m(s))$
- The $\Delta_m(s)$ is any stable transfer function which at each frequency is less than or equal to one in magnitude.
- $W_m(s)$ is a function of $s$ that constitutes an upper bound for the multiplicative uncertainty.



- The procedure for computing $G_p(s)$ in both cases requires to know:
  1. The nominal plant $G(s)$. Here chosen as the parametric model:

$$G_{par} = \frac{-16604}{s^3 + 0.0611s^2 + 28.39s}$$

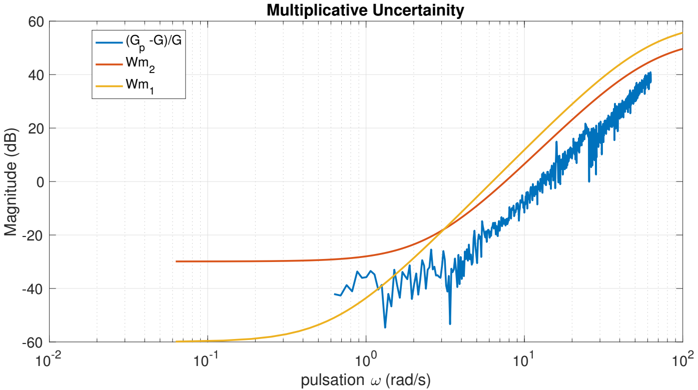
  2. Determining $W_m(s)$ with respect to the following criteria:

$$|W_m(j\omega)| \geq \max \left| \frac{G_p(j\omega) - G(j\omega)}{G(j\omega)} \right|$$

- Notice that $G_p(s)$ is chosen as the the non-parametric model

- Let's investigate two weighting functions for multiplicative uncertainty:
- The second weighting function is given by :

$$W_{2,m}(s) = \frac{K(s^3 + 3\tau_1 s^2 + 3\tau_1^2 s + \tau_1^3)}{s^3 + 3\tau_2 s^2 + 3\tau_2^2 s + \tau_2^3}$$

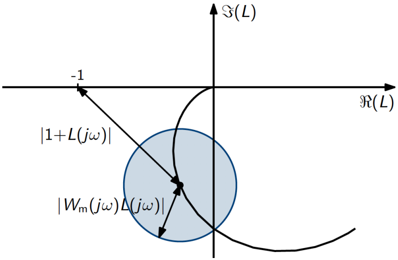


# System Identification

## Uncertainty model

- The system is RS if the uncertain loop transfer function $L_p$ does not encircle -1.

$$L_p = G_p K = GK(1 + W_m \Delta_m)$$



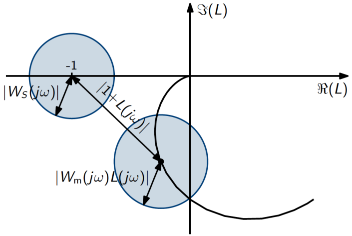

- Another condition for robust stability can then be derived and used easily:

$$\|W_m T\|_\infty < 1$$

# System Identification

## Uncertainty model

- The system is RP if the uncertain loop transfer function $L_p$ does not cross the circle made by $W_S$ and centered at -1.




- Another condition for robust performance can then be derived and used easily:

$$\| |W_s(j\omega)S(j\omega)| + |W_m(j\omega)T(j\omega)| \|_\infty < 1$$

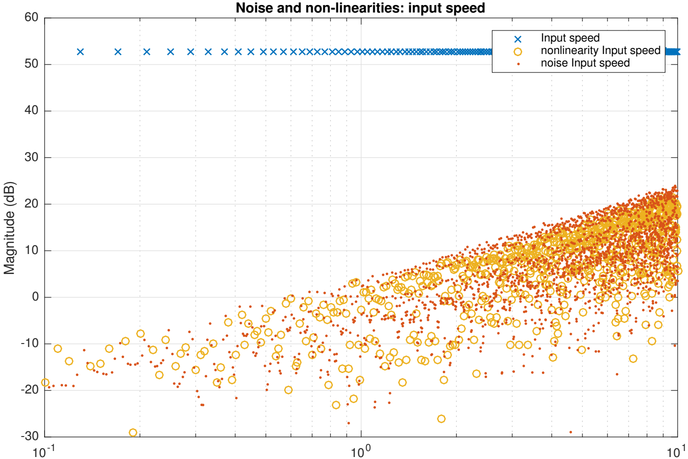
# Outline

- We are assuming a linear model
- How exact is this assumption?
- To know this, we have to check the non-linear behavior and the noise contributions of the system
- To measure the noise and the non-linearities an odd-odd-multisine was used as excitation signal. This means that:
  - Only every fourth frequency has an amplitude different from zero
  - At different lines the output will contain different parts of the input
- 3 periods are measured at the same time, which is enough to be able to measure linear contribution, noise and non-linear distortions separately.

# System Identification

## Noise and non-linearities

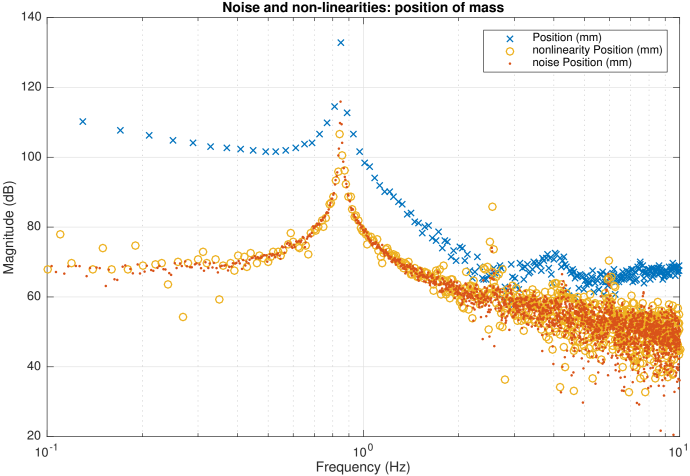
- There exists non-linearities and noise even for the input, although small
- Higher noise level and more non-linear distortions at a higher frequency



# System Identification

## Noise and non-linearities

- Most interesting part: position of mass



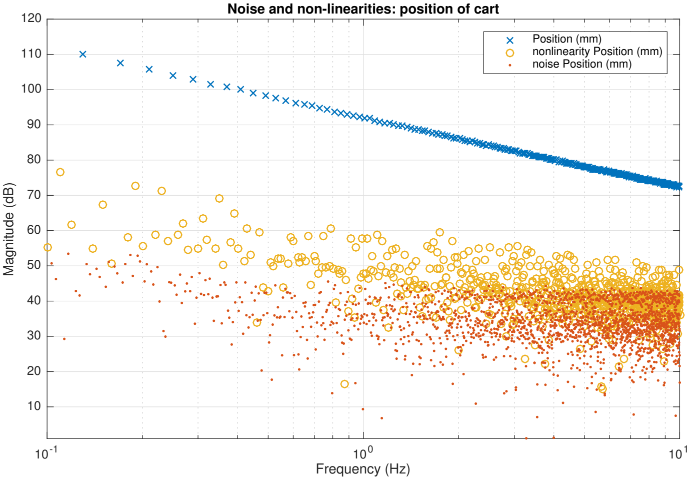
- Reflections:

1. Noise level and non-linear distortions have $20 - 40$ dB lower magnitude than the linear contribution, except at higher frequencies. A consequence of bigger impact by noise and non-linearities at higher frequencies, as seen in the figure of the input speed.
2. At higher frequencies the noise and nonlinearities gets more diverse.
3. Our system can be seen as approximately linear at low frequencies. The non-linearities do not have a significant impact until higher frequencies.

# System Identification

## Noise and non-linearities

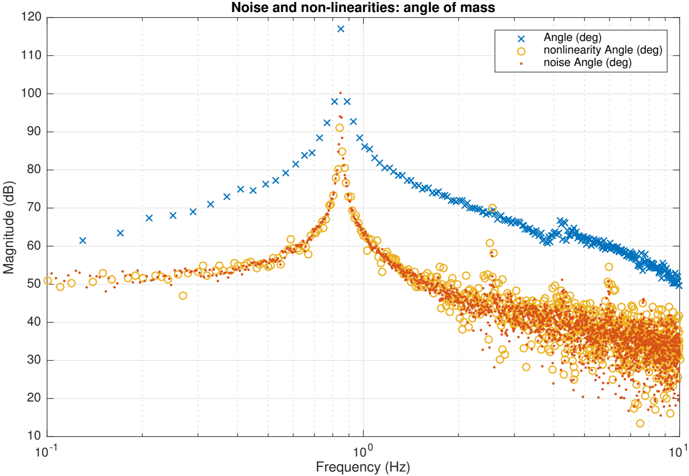
- Corresponding figure for position of cart



# System Identification

## Noise and non-linearities

- Corresponding figure for angle of mass



# Outline

# Controller design

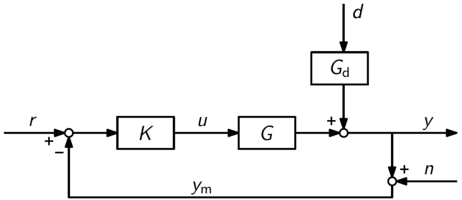
## $H_\infty$ loop shaping

- We can define the objective to be achieved by the controller:
  1. **Large bandwidth**. The bandwidth (in terms of the sensitivity function) has to be maximized such that the settling time of the controller in response to an input is minimal.
- We can define constraints too:
  1. The controller is designed using $H_\infty$ **loop shaping paradigm**.
  2. **Physical constraints** of the system that limit the performance (Bode sensitivity integral,...)
  3. **No steady state error** ensured trough $\lim_{t\to\infty} e(t) = 0$
  4. Additional constraint on the **maximal overshoot** to be limited. Trade-off between settling time and overshoot.
  5. The controller should be **stable and robust** when it's subject to uncertainty. (More on this later on)

# Controller design

## $H_\infty$ loop shaping

- To formulate the problem in a system theory fashion, we need to design a feedback linear controller $K$ according to the $H_\infty$ paradigm




- The following relationships hold

$$y = Tr + SG_d d + Tn$$

- with $S$ and $T$, the sensitivity and complementary sensitivity

$$S = \frac{1}{1+GK} \qquad T = \frac{GK}{1+GK}$$

# Controller design

## $H_\infty$ loop shaping

- The objective is to maximize the bandwidth:

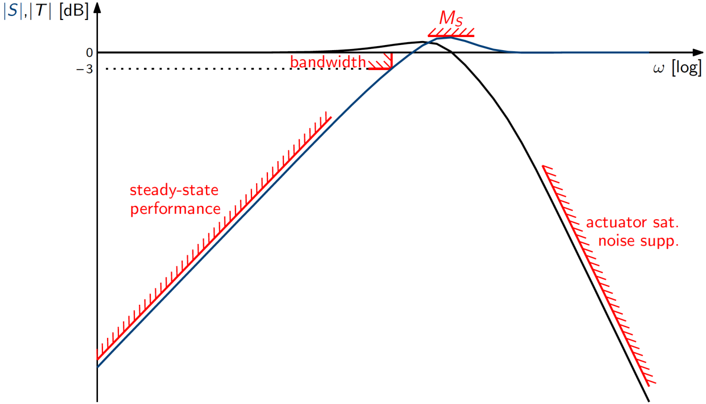



Figure: Sensitivity and complementary sensitivity functions

# Controller design

## $H_\infty$ loop shaping

- So far the objectives and constraints are a bit fuzzy. Let's precise them for this plant in particular by using the *LC Toolbox* tools.
  - First design choice : **Order of LF weight**.
  - Second design choice : **Frequency cutoff of LF weight**.
  - Third design choice : **Order of HF weight**.
  - Fourth design choice : **Frequency cutoff of HF weight**.
  - Fifth design choice : **MS Dc weight**.
  - Sixth design choice : **MT Dc weight**.

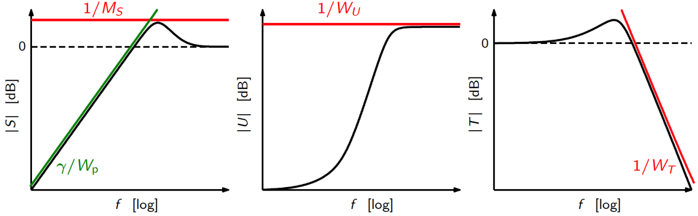



Figure: Weights for bounding the sensitivity and complementary sensitivity functions

# Controller design

## $H_\infty$ loop shaping

- Based on the trade-off defined on the previous slide and on numerous experiments, a controller has been designed:

| $W_S$ | $M_S$ | $W_T$ | $M_T$ | $W_U$ |
|---|---|---|---|---|
| $f_{co} = 1.55$ [Hz] | 4 [dB] | $f_{co} = 2$ [Hz] | 2 [dB] | 6 [dB] |
| 2nd order | / | 3rd order | / | / |
| -100 [db] | / | -60 [dB] | / | / |

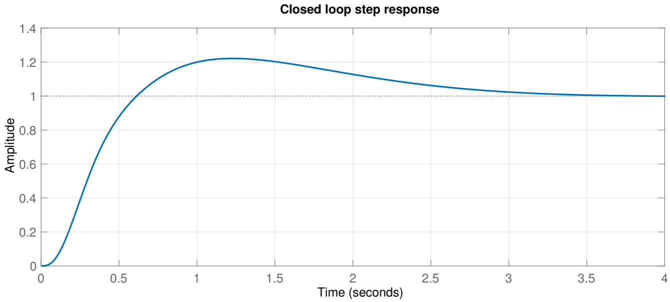



Figure: Closed loop step response using both controllers

# Controller design

## $H_\infty$ loop shaping

- Cleaned (high frequency poles and zeros as wells as near pole-zero couples are removed) version of the controller. Notice the low frequency integrator.

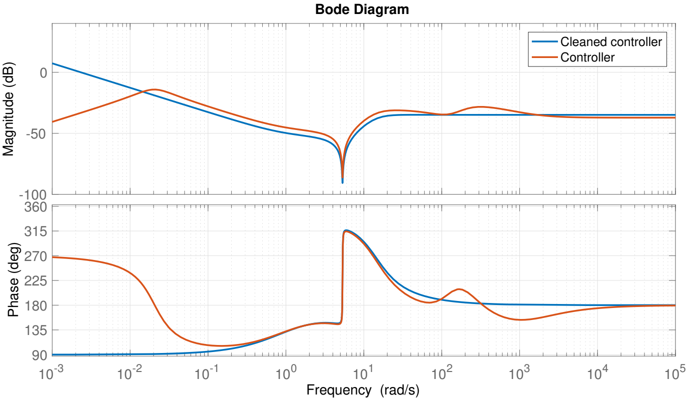



Figure: Cleaned version of the controller.

# Controller design

## $H_\infty$ loop shaping

- Sensitivity function of the controller. Notice the 40 [dB] per decade slope. The bandwidth is equal to 1.7 [rad/s]

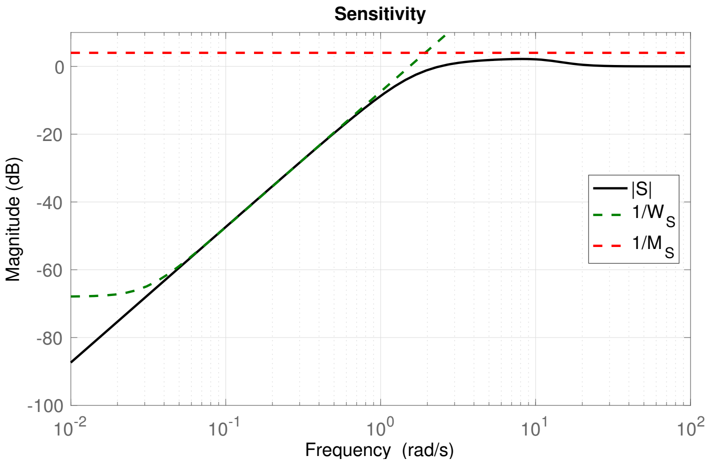



Figure: Sensitivity function $S$

# Controller design

## $H_\infty$ loop shaping

- Complementary sensitivity function of the controller.

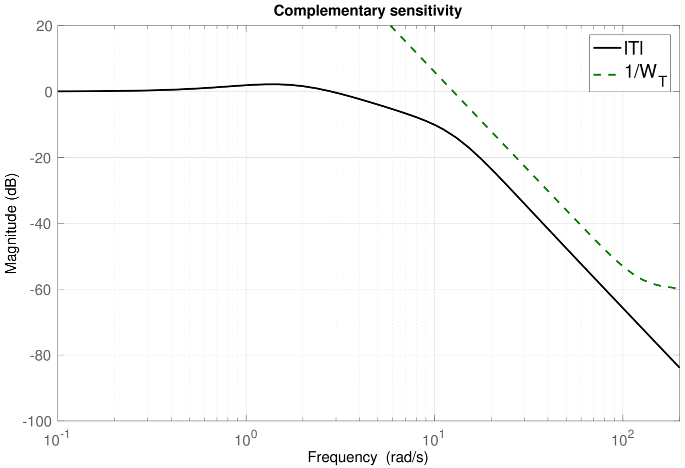
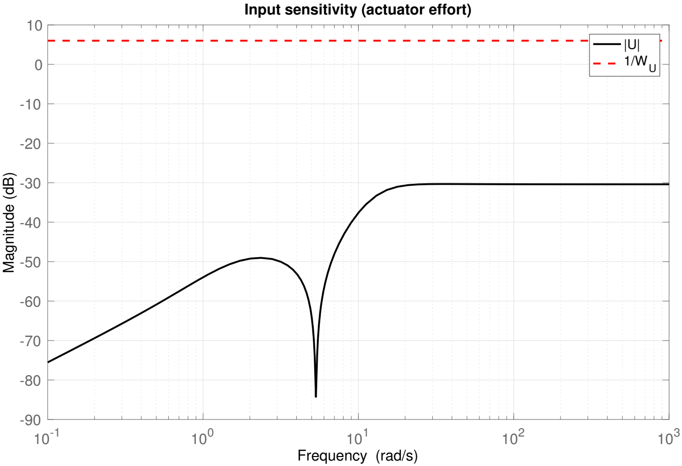



Figure: Complementary Sensitivity function $T$

# Controller design

## $H_\infty$ loop shaping

- Input sensitivity $U$. Here saturation should not occur because of the strict upper bound for the input sensitivity.



# Controller design

## $H_\infty$ loop shaping

- Let's investigate the robust stability and performance of this controller.
- For that purpose, reuse the criteria defined earlier
- A sufficient condition for robust stability

$$\|W_m(j\omega)T(j\omega)\|_\infty < 1$$

- A sufficient condition for robust performance

$$\||W_s(j\omega)S(j\omega)| + |W_m(j\omega)T(j\omega)|\|_\infty < 1$$

- In other words, check if the bode plots of these 2 quantities are never larger than 0 [dB]

# Controller design

## $H_\infty$ loop shaping

- Let's investigate the robust stability first using $W_{m,2}(s)$ as weighting function

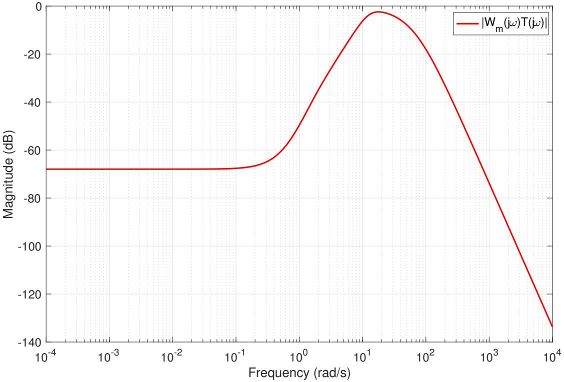



Figure: RS for the aforementioned controller

# Controller design

## $H_\infty$ loop shaping

- Given that the sufficient condition is met for robust stability, $L_p$ should not encircle -1. We can conlude that the first controller is robustly stable.

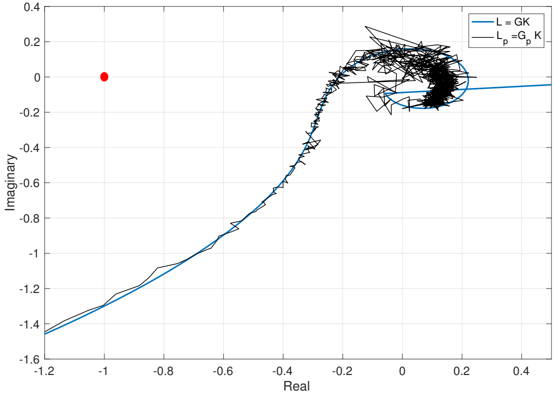



Figure: RS for the aforementioned controller

# Controller design

## $H_\infty$ loop shaping

- Let's investigate the robust performance using $W_{m,2}(s)$ as weighting function. The controller is rather conservative.

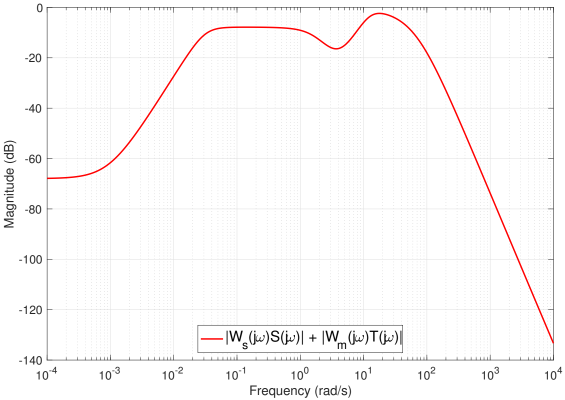



Figure: RP for the aforementioned controller

# Outline

# Trajectory tracking

## Discrete controller

- To test the controller it first needs to be made discrete. For that purpose one can use the *Tustin* method provided by MATLAB with a sampling period of 0.001 [s].

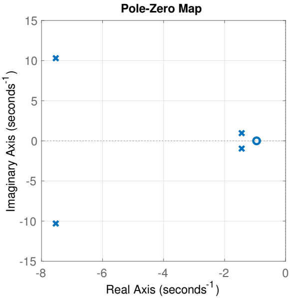



Figure: Poles-zeros map of the continuous close-loop system

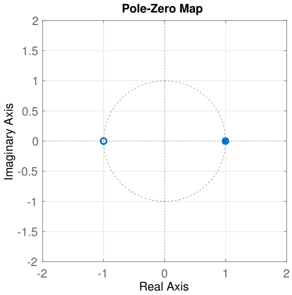



Figure: Poles-zeros map of the discrete close-loop system

# Trajectory tracking

## Discrete controller

- A Simulink model can be built to compare the results obtained in simulation and on the real system.

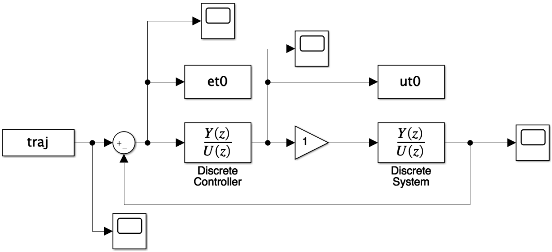



Figure: Simulink model used as reference

- A smooth trajectory can be generated using the *polytraj()* function.
- Trajectories are 30 [cm] long and have a settling time between that can be set between 3 and 10 [s].

# Outline

# Trajectory tracking

## Controller validation

- A trajectory function with a settling time of 3 [s] and a length of 300 [mm].

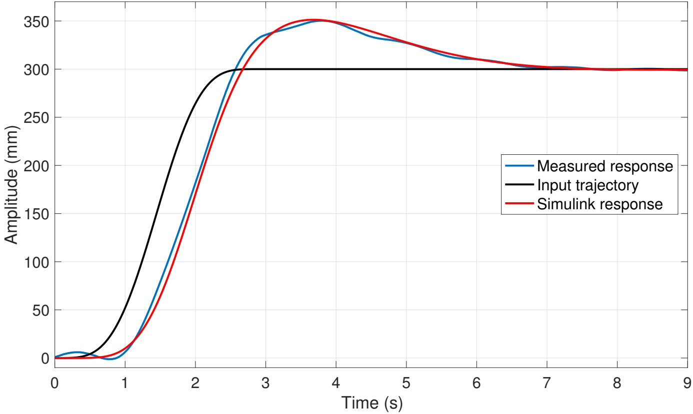



Figure: Comparison between real system and simulation

# Trajectory tracking

## Controller validation

- A faster trajectory (1 [s] and length of 140 [mm]) can also be tested. Notice the slight non-minimum phase behavior.

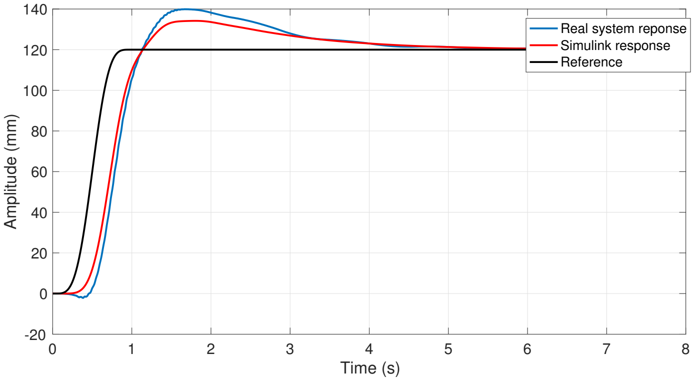



Figure: Comparison between real system and simulation.

# Outline

- At this point one would like to improve the performance of the controller. Especially the overshoot needs to be attenuated.
- How? By iterative learning control (ILC).
- The idea of ILC is that the performance of a system that executes the same task multiple times can be improved by learning form the previous executions (iterations).
- For improving the performance the error signals from previous iterations are used.

# Iterative Learning control

## Formulation of ILC

- The objective is to obtain for $N$, the number of iterations of the ILC:

$$\lim_{N \to \infty} y_d(k) - y(k) = 0 \qquad \forall k$$

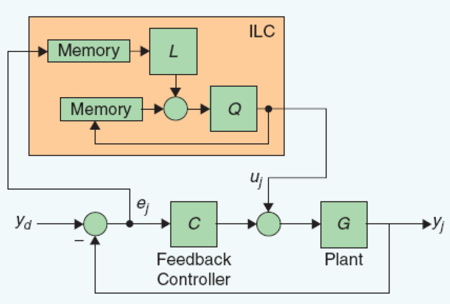



Figure: ILC Block scheme with parallel arrangement

# Iterative Learning control

## Formulation of ILC

- A widely used ILC learning algorithm is given by

$$u_{j+1}(k) = Q(q)\Big[u_j(k) + L(q)e_j(k+1)\Big]$$

- $j$ is the iteration index, $k$ is the time index, $Q(q)$ and $L(q)$ are defined as Q-filter and learning filter, respectively.
- One would like a stable and with good transient behavior ILC algorithm.

$$\|e_\infty - e_{j+1}\| \leq \gamma \|e_\infty - e_j\|$$

- A sufficient condition for asymptotic stability and monotonic convergence ($|\gamma| < 1$) in case of causal $Q(q)$ and $L(q)$ filters is

$$\|Q(z)(1 - zL(z)P(z)\|_\infty < 1$$

# Outline

- The implementation is done using :
    1. A $Q$-low pass 4th order Butterworth filter with a cut-off frequency below the resonance frequency of the overhead crane. Here it is chosen as 0.7 [Hz].
    2. A plant inversion method for a minimum-phase system.
    3. A serial arrangement structure is chosen such that the system is directly fed with $y_d$ and the output of the ILC (computed offline).
- The ILC equation is therefore:

$$u_{j+1}(k) = Q(q)\left[u_j(k) + \hat{P}(q)^{-1} e_j(k+1)\right]$$

- Notice that $Q(q)$ and $L(q)$ are causal.

# Iterative Learning control

## Implementation

- First we would like to check the stability and transient behavior of the ILC using the criterion:

$$\|Q(z)(1 - zL(z)P(z)\|_\infty = 0.0313 < 1$$

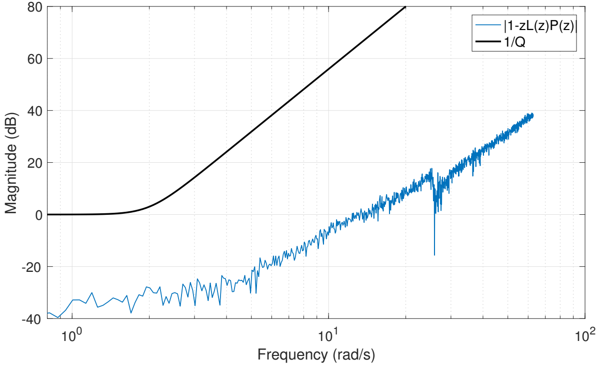



Figure: ILC stability criterion is respected. Monotonic convergence is ensured too.

# Iterative Learning control

## Implementation

- The evolution of the error for the first and second iteration are given in this figure. The tracking error is significantly reduced.

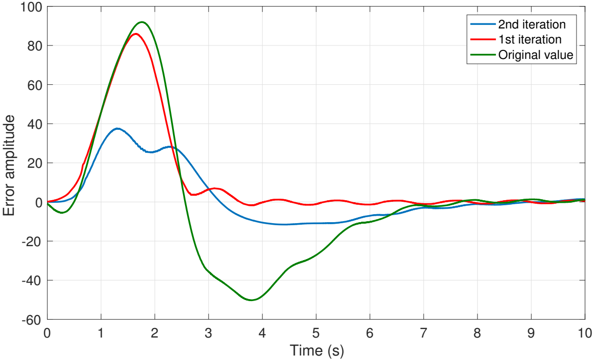



Figure: Tracking error using ILC for the first and second iteration

# Iterative Learning control

## Implementation

- The step response shows an improvement in terms of settling time and reduction of the overshoot. By iterating even further even better results can be obtained.

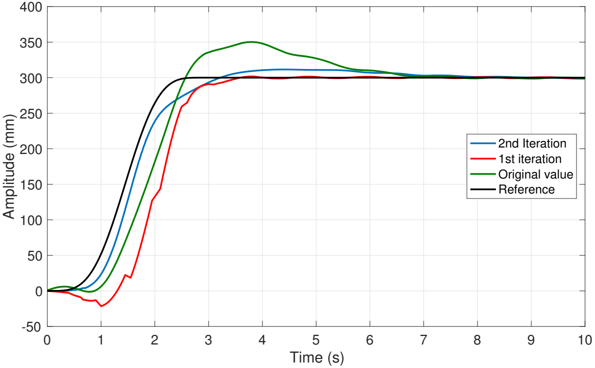



Figure: Time domain response for a smooth step for 1st and 2nd iteration

# Iterative Learning control

## Implementation

- After 7 iterations one eventually obtain the following error. The tracking error is considerably reduced compared the original error.

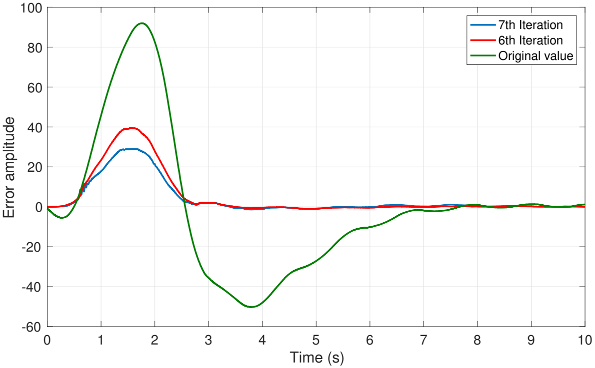



Figure: Tracking error using ILC for the first and second iteration

# Iterative Learning control

## Implementation

- The smooth step response is improved. The overshoot is gone and the settling time is much lower.

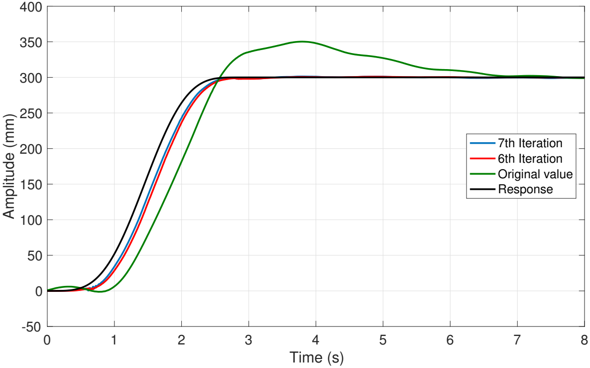



Figure: Time domain response for a smooth step for 1st and 2nd iteration

# Iterative Learning control

## Implementation

- One can eventually compare the evolution of the $L_2$ norm and $L_\infty$ norm of the tracking error for 7 iterations.

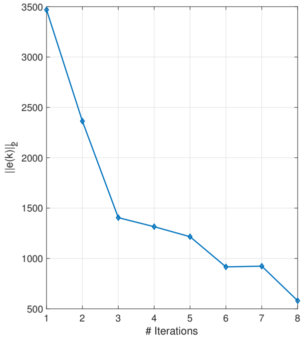



Figure: Evolution of the $L_2$ norm of the error

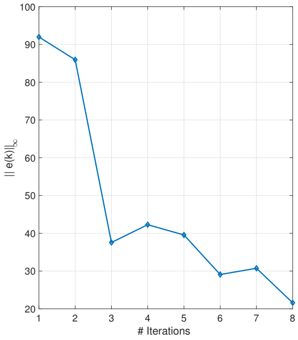



Figure: Evolution of the $L_\infty$ norm of the error

# Conclusion

- What has been done?
- In this project, an overhead crane system has been identified using maximum likelihood estimation. The identified model was validated based on different criteria. Uncertainty and nonlinearity was also taken into account. Then a controller was designed based on the $H\infty$ paradigm. Using this closed loop system, a trajectory tracking task was successfully performed both in simulations and experiments. To further improve the performance, an ILC process was developed. The improvement was then validated in experiments.