Project Report

Advanced Control Engineering II (MECH-M-2-RTV-RTV-ILV)

Crane Control

Mechatronics-Smart Technologies | Master

2nd semester

Lecturer: Dipl.-Ing. Dr. techn. Phillip Kronthaler

Group: MA-MECH-24-VZ

Authors: Jim Buso, Matthias Johann Hosp, Elias Karner

June 12, 2025

Contents

1 Introduction

The model chosen for this project is a crane, modeled with two carts and a simple pendulum, which is explained in more detail in Chapter 2. The goal is to implement trajectory planning, a Linear Quadratic Regulator (LQR) and an Extended Kalman Filter (EKF).

2 Modeling

The kinematic model of the crane can be interpreted as two carts on top of each other, as can be seen in 2.1. The base of the crane is equivalent with the bottom cart. The base deflects if the cart is accelerated and stops oscillating after a given time if the cart is stopped. Hence, it is modeled with a spring, that has the stiffness k_b, and damper d_b. The top cart is modeled with just a damper d_c, which is supposed to model the friction between the cart and the base. For the load, which is kinematically equivalent to a simple pendulum, no friction is assumed, and thus no damping is assumed.

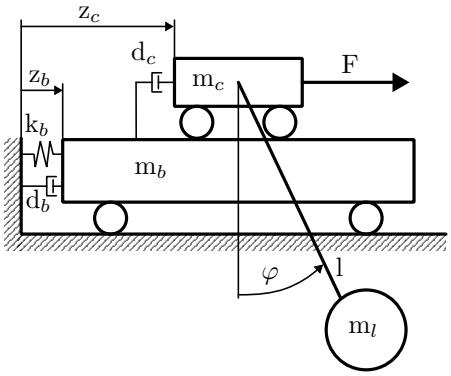


Figure 2.1: Model of the crane

The position of the cart was chosen as z_c, the position of the base as z_b, and the deflection of the load is described using φ. Those variables are also used as the generalized coordinates for the Lagrangian equations in order to derive the equations of motion. The way the generalized coordinates are chosen allows one to decouple the system, yielding the sketch of the cart shown in Figure 2.2.

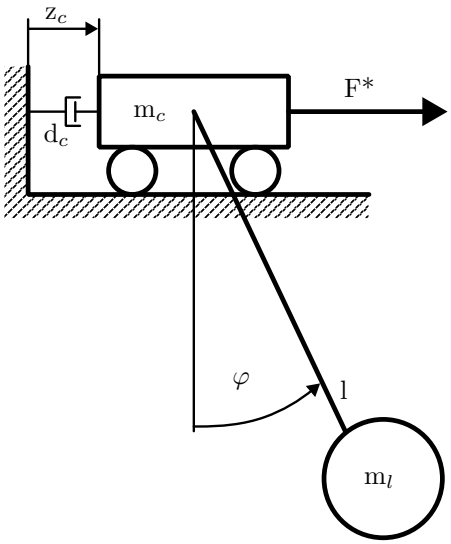


Figure 2.2: Decoupled cart

From this sketch the kinetic energy of the system T and the potential energy V can be calculated as shown in (1) with g as the gravitational acceleration. In addition, the generalized forces in the direction of z_c and φ can be calculated as in (2).

$$T = \frac{1}{2} m_c \dot{z}_c^2 + \frac{1}{2} m_l \left[(l\dot{\varphi} sin(\varphi))^2 + (l \cdot \dot{\varphi} cos(\varphi) + \dot{z}_c)^2 \right] \qquad V = -m_l g l cos(\varphi) \tag{1}$$

$$Q_{z_c} = F^* = F - d_c \left(\dot{z}_c - \dot{z}_b \right) \qquad Q_\varphi = 0 \tag{2}$$

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right) - \frac{\partial \mathcal{L}}{\partial q_i} = Q_i \qquad \text{with} \qquad \mathcal{L} = T - V \tag{3}$$

Using the Euler-Lagrange equations from (3), with $q_i = z_c, z_b$ and φ, the first two equations of motion (EOM) can be derived, yielding (4) and (5).

$$\ddot{z}_c(m_c + m_l) + \ddot{\varphi} m_l l cos(\varphi) - \dot{\varphi}^2 m_l l sin(\varphi) + d_c(\dot{z}_c + \dot{z}_b) - F = 0 \tag{4}$$

$$\ddot{\varphi} m_l l^2 + \ddot{z}_c m_l l cos(\varphi) + m_l g l sin(\varphi) = 0 \tag{5}$$

Since the base is now also decoupled form the rest of the system it can be calculated like a single mass oscillator, as shown in figure 2.3. Since the EOM of such a system is well known it can be directly formulated as shown in (6).

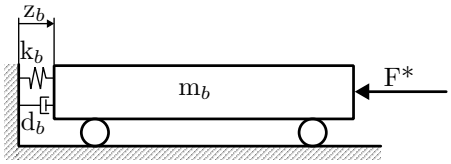


Figure 2.3: Decoupled base

$$\ddot{z}_b m_b + \dot{z}_b(d_b + d_c) - \dot{z}_c d_c + z_b k_b + F = 0 \tag{6}$$

With (4),(5) and (6) the non linear state space model can be constructed as seen in (7) with (8) as the output.

$$\dot{\vec{x}} = \begin{bmatrix} \dot{z}_c \\ \frac{-\dot{z}_c d_c + \dot{z}_b d_c + \dot{\varphi}^2 l m_l sin(\varphi) + F + g m_l cos(\varphi) sin(\varphi)}{-m_l cos^2(\varphi) + m_c + m_l} \\ \dot{z}_b \\ -\frac{-\dot{z}_c d_c + z_b k_b + \dot{z}_b(d_b + d_c) + F}{m_b} \\ \dot{\varphi} \\ -\frac{-\dot{z}_c d_c \, cos(\varphi) + \dot{z}_b dc \, cos(\varphi) + \dot{\varphi}^2 l m_l cos(\varphi) sin(\varphi) + F cos(\varphi) + g sin(\varphi)(m_c + m_l)}{l(-m_l cos^2(\varphi) + m_c + m_l)} \end{bmatrix} \quad \text{with} \quad \vec{x} = \begin{bmatrix} z_c \\ \dot{z}_c \\ z_b \\ \dot{z}_b \\ \varphi \\ \dot{\varphi} \end{bmatrix} \tag{7}$$

$$y = z_c + l \cdot sin(\varphi) \tag{8}$$

This state space model can be linearized by means of the Taylor series expansion, yielding the linear state space model in (9) and (10). For the working points $z_{c0} = 0$,$z_{b0} = 0$ and $\varphi_0 = 0$ are chosen.

$$\dot{\vec{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -\frac{d_c}{m_c} & 0 & \frac{d_c}{m_c} & \frac{m_l g}{m_c} & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & \frac{d_c}{m_b} & -\frac{k}{m_c} & -\frac{d_b + d_c}{m_b} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & \frac{d_c}{l m_c} & 0 & -\frac{d_c}{l m_c} & -\frac{g(m_c + m_l)}{l m_c} & 0 \end{bmatrix} \vec{x} + \begin{bmatrix} 0 \\ \frac{1}{m_c} \\ 0 \\ -\frac{1}{m_b} \\ 0 \\ -\frac{1}{l m_c} \end{bmatrix} F \quad \text{with} \quad \vec{x} = \begin{bmatrix} z_c \\ \dot{z}_c \\ z_b \\ \dot{z}_b \\ \varphi \\ \dot{\varphi} \end{bmatrix} \tag{9}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 & l & 0 \end{bmatrix} \vec{x} \tag{10}$$

2.1 EVALUATION OF PHYSICAL CONSTANTS

Table 2.1: Given Parameters

Parameter	Description	Value
m_b	Mass of the base	$20\,\mathrm{kg}$
m_c	Mass of the cart	$2\,\mathrm{kg}$
m_l	Mass of the load	$1\,\mathrm{kg}$
l	Length of the rope	$0.25\,\mathrm{m}$
g	Gravitational acceleration	$9.81\,\mathrm{m\,s^{-2}}$
f_b	Natural frequency of the base	$4\,\mathrm{Hz}$
τ	Decay time	$25\,\mathrm{s}$
Δz	Deflection of the base	$10\,\mathrm{mm}$

In Table 2.1 the given parameters can be seen. Using those, and the fact that z occurs when $\ddot{z}_c = 10\,\mathrm{m\,s^{-2}}$, the stiffness k_b of the base can be calculated as shown in (11).

$$k_b = \frac{F_b}{\Delta z} = \frac{m_c \ddot{z}_c}{\Delta z} = 2000\,\mathrm{N\,m^{-1}} \tag{11}$$

In order to calculate the damping of the base the equation for the envelope of the vibration is used, shown in (12). It is assumed that if the amplitude is $0.01\,\mathrm{mm}$, it can be neglected. This value is reached after $\tau = 25\,\mathrm{s}$.

$$A(t) = A_0 e^{-\zeta\omega_0 t} \qquad 0.01\,\mathrm{mm} = 10\,\mathrm{mm}\, e^{-\zeta\omega_b \tau} \tag{12}$$

Using the damping according to lehr ζ, shown in (13), and the damping ratio δ, shown in (14), the damping of the base d_b can be calculated like shown in (15).

$$\zeta = \frac{-\ln\frac{0.01\,\mathrm{mm}}{10\,\mathrm{mm}}}{\omega_b \tau} = \frac{-\ln\frac{0.01\,\mathrm{mm}}{10\,\mathrm{mm}}}{25.13\,\mathrm{s^{-1}}\,25\,\mathrm{s}} = 0.011 \qquad \text{with} \qquad \omega_b = 2\pi_b = 25.13\,\mathrm{s^{-1}} \tag{13}$$

$$\zeta^2 = \frac{\delta^2}{\omega_b} \qquad \text{with} \qquad \delta = \frac{d}{2m} \qquad \text{and} \qquad m = m_b + m_c \tag{14}$$

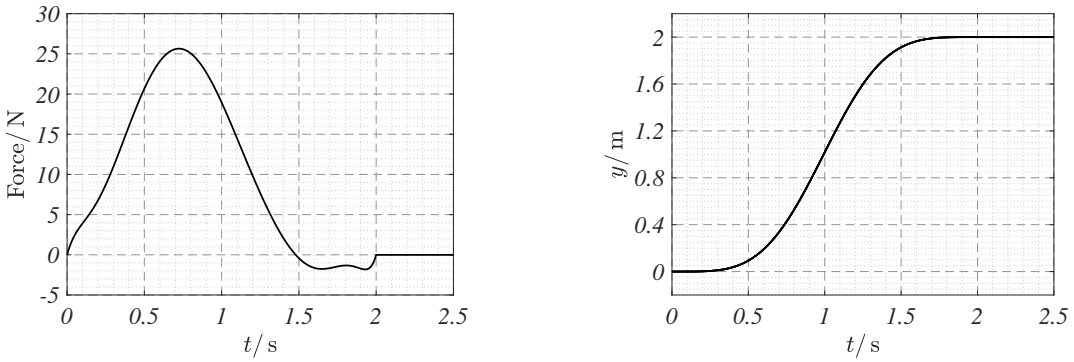
$$d = \sqrt{\zeta^2 4 (m_b + m_c)^2 \omega_b} = 12.16\,\mathrm{N\,s\,m^{-1}} \tag{15}$$

The cart is assumed to be mounted on a recirculating ball bearing guide carriage, which also introduces some damping. The values for this damping can vary drastically according to [1]. Thus the damping of the cart d_c is estimated to be $10\,\mathrm{N\,s\,m^{-1}}$.

3 PID

3.1 FEED FORWARD

Due to the complexity of the system, the feed forward control is not achieved with a flatness-based feed forward control but instead with trajectory planning using the linear state space model. Since the system has an order of $n = 6$ is the polynomial to calculate the reference trajectory of the order of $2n + 1 = 13$. The prototyping function is used to obtain the trajectory. The polynomial produced is always with a start value of 0 and an end value of 1. Therefore, it has to be scaled with the desired start and end value to get the reference trajectory y_{ref} for the load position and the reference input u_{ref}. In Figure 3.1, the trajectories for a start and end value of $y_{start} = 0\,\mathrm{m}$ and $y_{end} = 2\,\mathrm{m}$ with a transition time of $T = 2\,\mathrm{s}$, can be seen.



(a) u_{ref} (b) y_{ref}

Figure 3.1: Trajectory from 0 to 2 m in $T = 2\,\mathrm{s}$

3.2 IMPLEMENTATION OF THE PID CONTROLLER

In the figure 3.2 the PID implementation with a Luenberger observer in Simulink is illustrated.
For the PID values, the Matlab function *pidTuner()* is used. The following values are designated: $K_p = 12$, $K_i = 10$, $K_d = 0.8$.
The Luenberger observer is introduced in the green box. Since not all states are measured, its purpose is to estimate the full state vector $\hat{\vec{x}}$. The observer is designed using the linear state-space model of the system. The gain L is computed using the MATLAB function *acker()*. To ensure that the observer dynamics is faster than the system dynamics, the eigenvalues of matrix $\mathbf{A}$ are shifted to the left by 5; poles $=$ eig$(\mathbf{A}) - 5$.
As in (10) the output y is defined as the position of the load. The real world measurement is simulated in the violet box. The sensor to measure the cart position z_c is mounted on the base. Hence, it is needed to subtract the real base position z_b from the real z_c, to get the measured signal Δz, and later add the observed $\hat{z}_b$, since z_b is not measured. To step up the simulation, a noise of $\pm\ 1\,\mathrm{mm}$ is added to the cart position. Furthermore, an angle noise of $\pm\ \frac{720}{1024}\,^\circ$ and a quantization effect of $\frac{360}{1024}\,^\circ$ are introduced at the angle measurement.

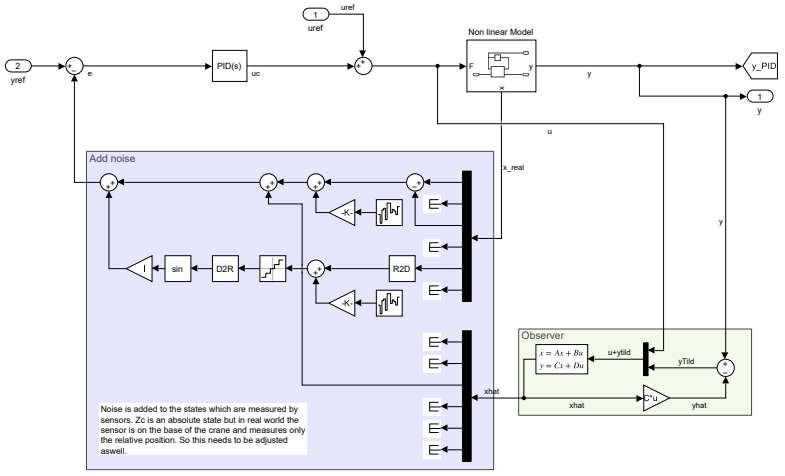
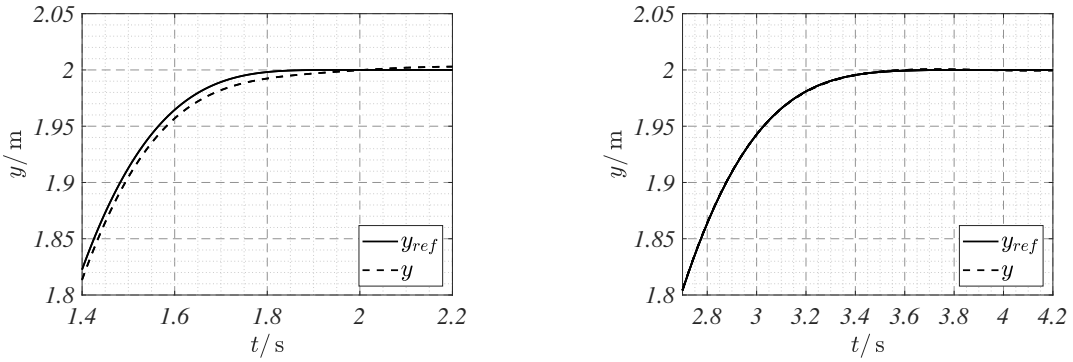


Figure 3.2: Implementation of the PID

Using the trajectories from Section 3.1 it can be seen in Figure 3.3b that the controlling of y is smoother using the transition time $T = 4\,\mathrm{s}$.
Due to a slower driving of the cart, the angle of the pendulum φ is staying low. The trajectory planning is realized with the linearized system, which results in a worse trajectory following the bigger φ gets. This effect is illustrated in figure 3.3 without noise and quantization for a better visualization.



(a) Comparison for $T = 2\,\mathrm{s}$

(b) Comparison for $T = 4\,\mathrm{s}$

Figure 3.3: Comparison with Output y and trajectory y_{ref} for both transition times

4 Linear Quadratic Regulator (LQR)

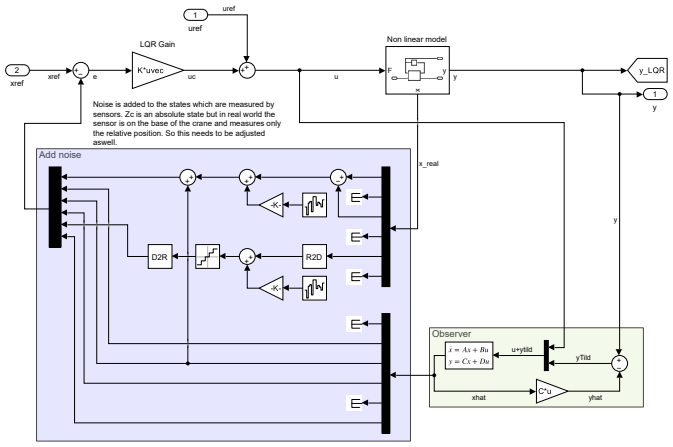


Figure 4.1: LQR with observer and reference trajectory as feed forward

In Figure 4.1 the block diagram of the Matlab Simulink simulation is depicted. The structure is similar to the PID in Figure 3.2 except the LQR gain replaces the PID and the feedback are the states instead of only the output signal.

The Linear Quadratic Regulator is a state feedback controller. For a normal state-feedback controller the desired poles are chosen, and based on the poles the state feedback gain is derived. But this can be non-intuitive, especially for more complex systems. Compared to a general state feedback controller, the LQR is a control strategy that minimizes a cost function (16), where the weights in $\mathbf{Q}$ and $\mathbf{R}$ directly influence each state. This gives more intuitive control and ease for complex systems. The optimization problem is solved by solving the Algebraic Riccati Equation (ARE). The result is a state feedback gain. For further mathematical details, see [2].

$$J = \int_0^\infty \left(\mathbf{x}^\top(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^\top(t)\mathbf{R}\mathbf{u}(t)\right) dt, \tag{16}$$

To obtain a state feedback gain, the two matrices $\mathbf{Q}$ and $\mathbf{R}$ have to be chosen. Important states get a higher entry value on the main diagonal of the $\mathbf{Q}$ matrix. So higher control errors will be penalized stronger. The elements of $\mathbf{R}$ describes the penalization of the inputs which are very expensive.

An approach to obtain $\mathbf{Q}$ and $\mathbf{R}$ is the *Bryson-Rule* (17) [3]. The idea is to choose the maximum tolerated error of each state, square it and put it in the denominator of the fraction. High sensitive or important states receive high entries in the $\mathbf{Q}$ matrix. The same applies for the $\mathbf{R}$ matrix. In the case of the gantry crane, there is only one input, so $\mathbf{R}$ is a scalar.

$$Q_{i,i} = \frac{1}{(\text{maximum acceptable value of } (error_{\text{states}}))^2}, \quad i \in \{1, 2, \ldots, n\} \tag{17}$$

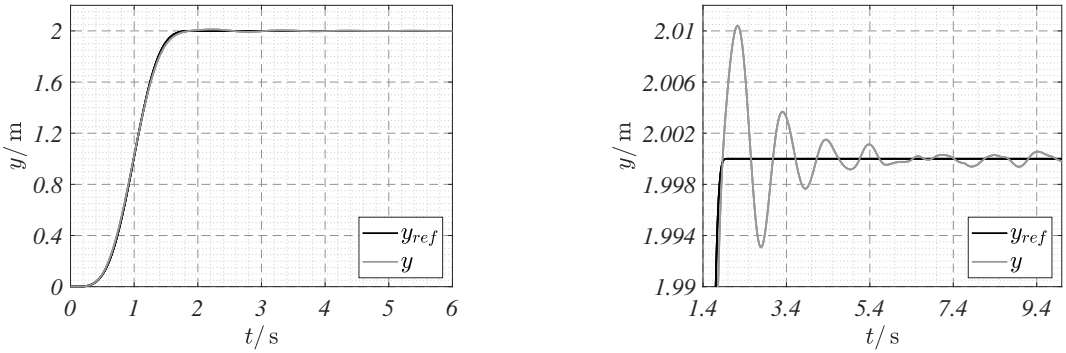
$$\mathbf{Q} = \begin{bmatrix} Q_{1,1} & 0 & \cdots & 0 \\ 0 & Q_{2,2} & \cdots & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & Q_{n,n} \end{bmatrix}$$

In Table 4.1 the tolerated errors for each state and the corresponding $Q_{i,i}$ values are depicted. The reasoning behind this is that the states which are in the calculation of the final output (z_c and φ) have a smaller acceptable error and should align with the given trajectory much faster. The value for $R = 0.01$ because it is not as relevant.

Table 4.1: Q entries for the LQR

states	z_b	$\dot{z}_b$	z_c	$\dot{z}_c$	φ	$\dot{\varphi}$
errors	0.01	0.05	0.1	0.2	0.01	0.5
$Q_{i,i}$	10^6	400	100	25	10^4	4

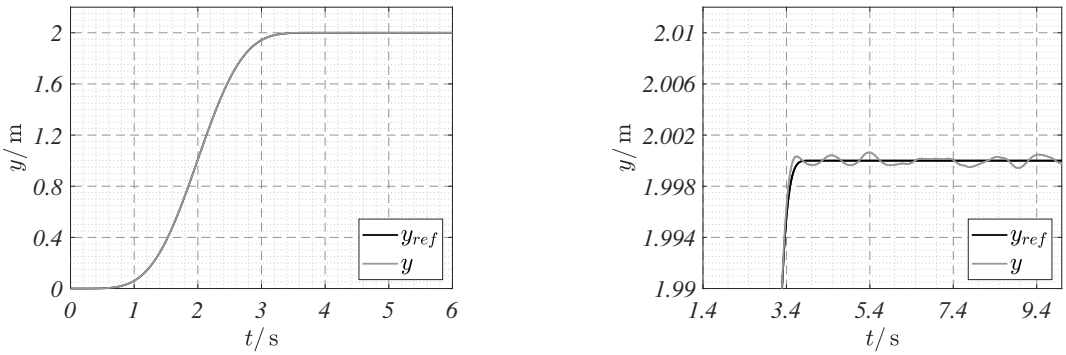
In Figure 4.2 and 4.3 are the results of two different transitions depicted. The first Figure 4.2 is a motion of the load from $0\,\mathrm{m}$ to $2\,\mathrm{m}$ within $2\,\mathrm{s}$. The detailed view Figure 4.2b oscillations with an overshoot of approximately $5\,\%$. The trajectory within $2\,\mathrm{s}$ is the limit. The faster the transition the higher the oscillations will be.



(a) whole transition (b) detailed view

Figure 4.2: Transition from 0 to 2 m in 2 s.

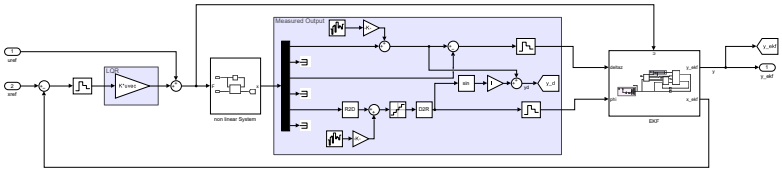
The result of a motion from $0\,\mathrm{m}$ to $2\,\mathrm{m}$ within $4\,\mathrm{s}$ is depicted in 4.3. In the detailed view 4.3b it can be seen that the oscillations are almost gone. The residuals are a result of the added noise in the feedback. But the position of the load deviates of not even $1\,\mathrm{mm}$.



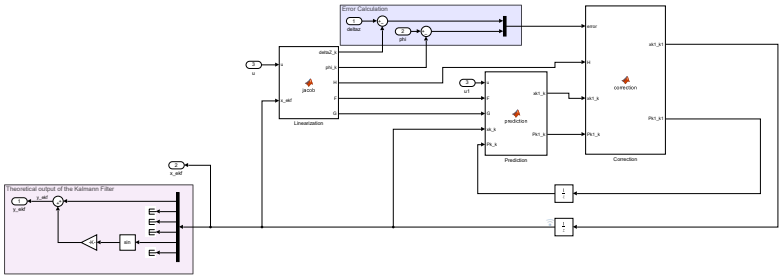
(a) whole transition (b) detailed view

Figure 4.3: Transition from 0 to 2 m in 4 s.

5 Extended Kalman Filter (EKF)



(a) Implementation of the Extended Kalman Filter with LQR



(b) Implementation of the EKF

Figure 5.1: Subsystem for the EKF

Figure 5.1a shows the subsystem where the EKF is implemented. In order to take advantage of the estimated states, an LQR combined with the trajectory planning is used. The non linear system is used to calculate the true states and also the true output. The states that are measured, $\Delta z = z_c - z_b$ and the angle of the load φ, are calculated and noise is added. Those states are then fed into another subsystem in which the EKF is implemented.

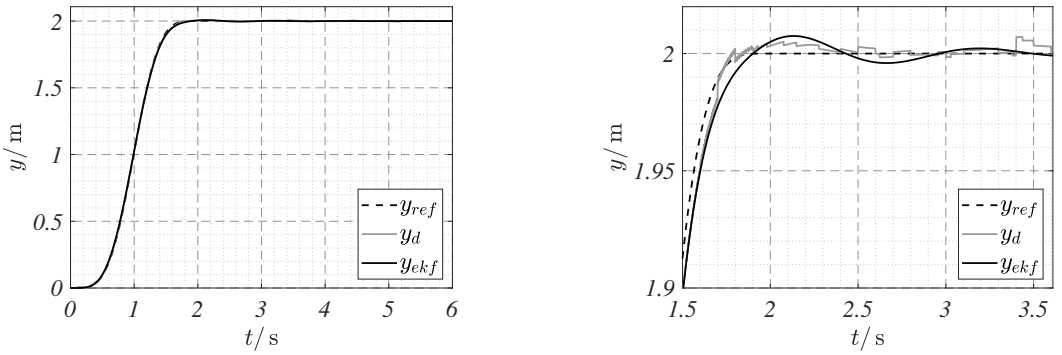
In Figure 5.1b the implementation of the EKF itself can be seen. In the Matlab function *jacob*, is the system being linearized around the current states, which is why the estimated states of the EKF are fed back into it, and discretized. To do this the Jacobian matrices, derived from the non linear system, are used. They were calculated beforehand in Matlab. The Jacobian of the matrix $\mathbf{C}$ has to reflect the measurements and therefore is form the size [2x6]. It can be seen in (18). The estimation of the measured states is also calculated in there and is further used to calculate the error.

$$\mathbf{C_{jacob}} = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{18}$$

The Matlab function *predict*, is used to predict the new states and covariance based on their values from the previous steps and the input u from the LQR. In the Matlab function *correction*, the states and the covariance are corrected using the error from the estimated states to measured one.

$$\mathbf{Q_{EKF}} = \begin{bmatrix} 10^{-8} & 0 & 0 & 0 & 0 & 0 \\ 0 & 10^{-12} & 0 & 0 & 0 & 0 \\ 0 & 0 & 10^{-12} & 0 & 0 & 0 \\ 0 & 0 & 0 & 10^{-12} & 0 & 0 \\ 0 & 0 & 0 & 0 & 10^{-6} & 0 \\ 0 & 0 & 0 & 0 & 0 & 10^{-12} \end{bmatrix} \quad \mathbf{R_{EKF}} = \begin{bmatrix} 1000 & 0 \\ 0 & 1000 \end{bmatrix} \tag{19}$$

For the LQR the same $\mathbf{Q}$ and $\mathbf{R}$ are used as in Chapter 4. For the EKF they can be seen in (19). In the process noise covariance matrix $\mathbf{Q_{EKF}}$ it can be seen that all states are expected to have small process noise, except for z_c and φ. This allows more flexibility for this values, which means they can deviate a bit more from the theoretical model. This has to do with the linearization of the model which is done in the EKF. The reson why z_c is effected can be seen in (7). It depends on the the $cos^2(\varphi)$. The same thing is true for φ itself, but even to a greater extend, since it depends on $sin(\varphi)$, $cos^2(\varphi)$ and more. The measurement noise covariance matrix $\mathbf{R_{EKF}}$ reflects the doubt in

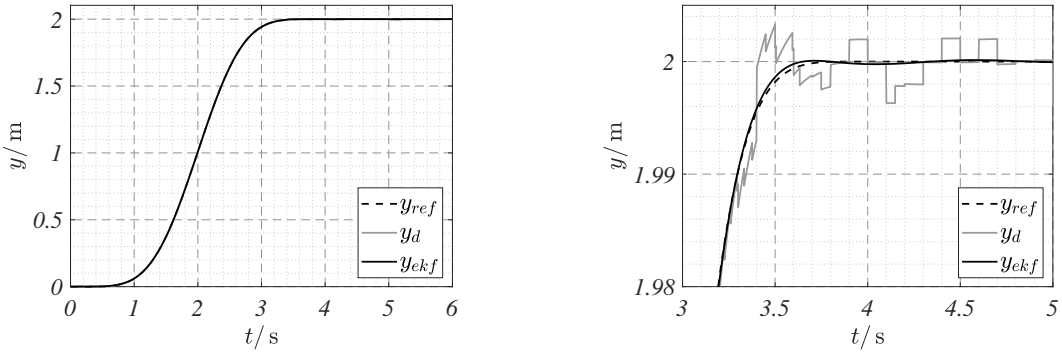


(a) whole transition (b) detailed view

Figure 5.2: Transition from 0 to 2 m in 2 s

the measurement. The values where chosen quit high and equal for both sensors since the experience a similar amount of noise.

In Figure 5.2 a transition from 0 to $2\,\mathrm{m}$ in $2\,\mathrm{s}$ can be seen. Here y_{ref} is the referencetrjectory, y_d is the output with disturbance and y_{EKF} is the output constructed from the estimated states of the EKF. The EKF follows the trajectory quit well, but has some overshoot of approximately $7.5\,\mathrm{mm}$, which is approximately $0.375\,\%$.



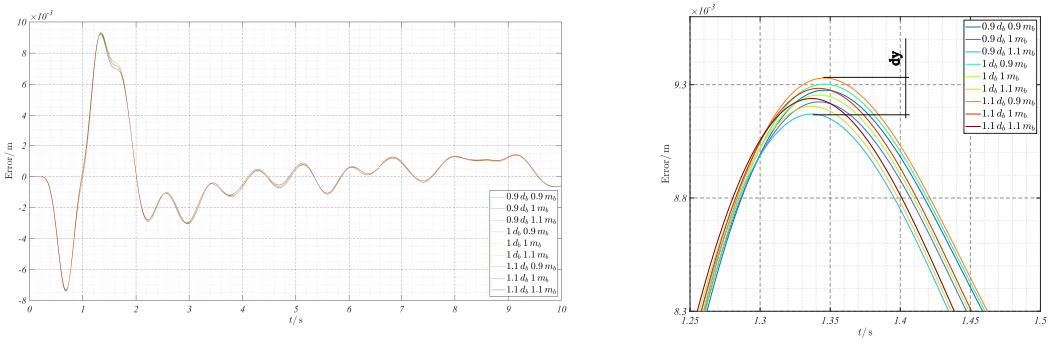
(a) whole transition (b) detailed view

Figure 5.3: Transition from 0 to 2 m in 4 s

In Figure 5.3 a transition from 0 to $2\,\mathrm{m}$ in $4\,\mathrm{s}$ can be seen. Here the overshoot is smaller than in the previous example with approximately $0.08\,\mathrm{mm}$. However, it can be seen that there is still some osculation of the output after the transition. Figure 5.3b shows that the EKF is removing the noise and smoothing the output.

6 Results

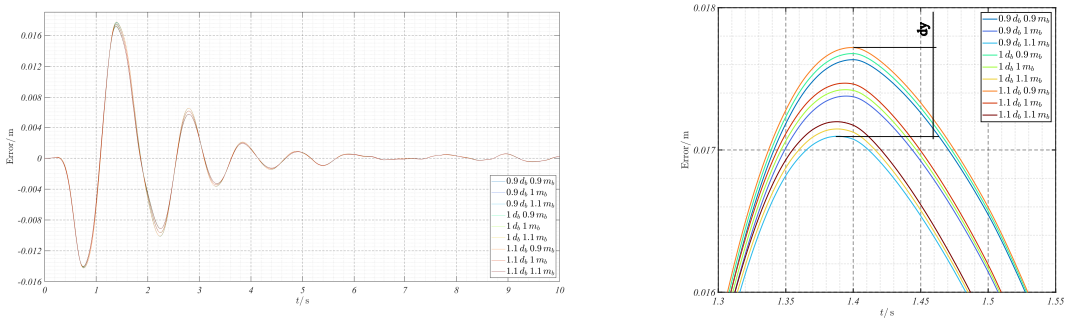
This section shows the comparison of the three control strategies and their reaction to uncertainties in the model parameters. For this, the mass and the damping of the base have been changed by $\pm 10\,\%$ in the non-linear model. The rest stays the same. The results are depicted in Figure 6.1, 6.2 and 6.3. In general, it can be seen that the overshoot of the PID is the smallest and produces the smallest difference to the reference trajectory at the overshoot. The plot highlights the maximum deviation d_y which gives information about what is the maximum difference between the parameter changes. Here EKF has by far the smallest which implies that it is not as sensitive to changes of the parameters and uncertainnesses of the model as the others.



(a) Error in the dynamic state

(b) Detailed view with $y_d = 1.5876 \cdot 10^{-4}$ m

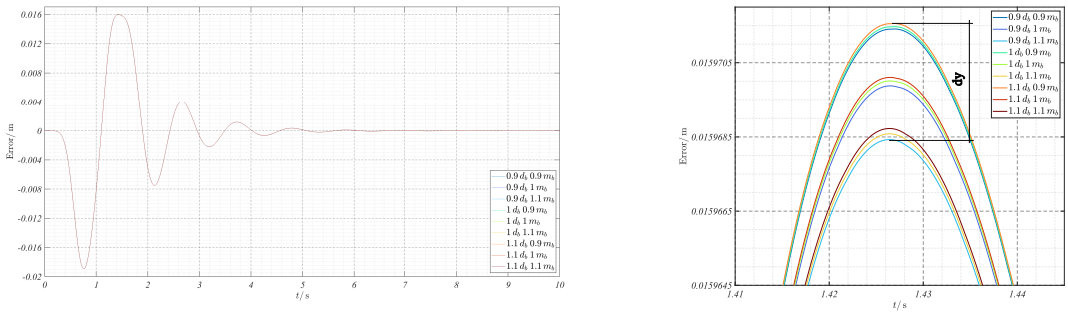
Figure 6.1: Errors resulting from the parameter variation for PID



(a) Error in the dynamic state

(b) Detailed view with $y_d = 6.2318 \cdot 10^{-4}$ m

Figure 6.2: Errors resulting from the parameter variation for LQR



(a) Error in the dynamic state

(b) Detailed view with $y_d = 3.1267 \cdot 10^{-6}$ m

Figure 6.3: Errors resulting from the parameter variation for EKF

7 Conclusion

This project was about to model, simulate and develop control strategies for the gantry crane 2.1. To model the system, the Lagrangian has been derived 2. The result of the non-linear system can be seen in (7) and (8). The linearized model is described in (9) and (10). The output of the system is the absolute position of the load, since this would be important for the real world as well. After modeling a PID has been tuned for the linear system and combined with a feed forward trajectory planner. The trajectory is also planned with the linear model because the complexity of the system did not allow to compute the flat output easily. This is a limitation, which will be discussed later. The next step was to implement a LQR instead of the PID. The tuning of the LQR is described in more detail in Section 4. As last step an Extended Kalman Filter has been added to the system to deal with the sensor noise. The results of each sensor can be seen in the individual chapters.

The different control strategies have been tested with two different trajectories. One is moderately fast with $4\,\mathrm{s}$ for a transition from 0 to $2\,\mathrm{m}$ and the other one has the same transition within $2\,\mathrm{s}$. For the slower transition all three follow the trajectory without any differences which are worth mentioning. Whereas, the faster transition shows major differences. Figure 7.1 shows the three strategies in comparison to the reference trajectory of the output. It can be seen the PID has the lowest overshoot but in the steady state region it oscillates the most. The EKF is in the middle between the PID and the LQR for the overshoot and in the steady state region the error is in the region of μm. With further tuning, the overshoot may be mitigated even more for the EKF and LQR.

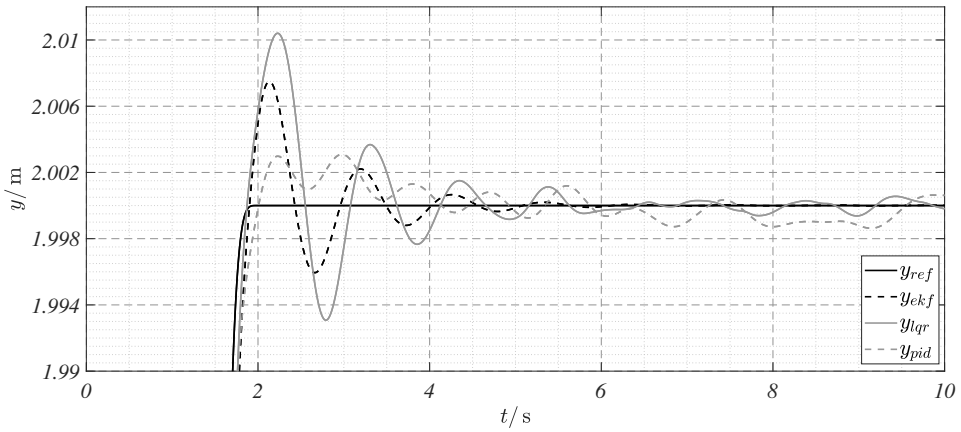


Figure 7.1: Comparison of EKF, LQR and PID for a transition from 0 to $2\,\mathrm{m}$ within $2\,\mathrm{s}$

In general the output signals are plotted without disturbances. The major advantage of the EKF is that it is able to predict and filter the noise which can not be seen in the plots. For this reason the oscillation of the EKF signal is also the smallest. It is also to be mentioned that the EKF is the least effected by variation of the parameters.

Furthermore, it is worth to mention that the transition could have been made faster with the flat output. Because for the current setting at the transition within $2\,\mathrm{s}$ the state of φ is about $30°$. The domain of small angles is not valid anymore. So the planning of the trajectory with the linear model reaches its boundaries here.

List of Figures

List of Tables

References

[1] I. Oleaga, J. Zulaika, F. Campa, and J. Hernando, "A method to measure the damping introduced by linear guides in large milling machines," *Mechanical Systems and Signal Processing*, vol. 171, p. 108908, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327022000954

[2] P. Kronthaler, "Advanced control engineering ii," Innsbruck, Austria, 2025, lecture notes for the Master Mechatronics program.

[3] MathWorks. (2025) lqr (Control System Toolbox). MathWorks. Accessed: 2025-06-10. [Online]. Available: https://de.mathworks.com/help/control/ref/lti.lqr.html