Homework 3 in EL2450 Hybrid and Embedded Control Systems

Alexander Ramm
931005
aleram@kth.se

Filip Stenbeck
930702
fstenb@kth.se

Gabriel Andersson Santiago
910706
gabras@kth.se

Anders Levin
personnummer
email

March 4, 2016

Task 1

$$u_r = \frac{2^{u_\omega + u_\psi}}{2} = u_w + \frac{u_{Psi}}{2} \tag{1}$$

$$u_l = u_\omega - \frac{u_\Psi}{2} \tag{2}$$

Task 2

By calculating the mean value of $\dot{x}$ from the data in *Forward.cvs* R could be estimated with equation (3)

$$\dot{x} = R * u_\omega * cos(\theta) \tag{3}$$

By calculating the mean value of $\dot{\theta}$ from the data in *Rotate.cvs* L could be estimated with equation (4)

$$\dot{\theta} = \frac{R}{L} u_\Psi \tag{4}$$

The estimated values are

R	L
111	111

Task 3

$\dot{\theta} = R/Lu_\psi$ won't be asymptotically stable due to that the output signal will oscillate with constant amplitude after a set time. It will not have Zeno behaviour due there is no finite time limit when the ouutput signal is stable.

Task 4

The system is asymptotically stable due to that the output signal will constantly converge to the desired value. Though by doing this will quicker and quicker reactions from the controller which will not be possible in real life. The system does not exhibit Zeno behaviour since the control signal will always have a value and it does not stabilize within a finite time.

Task 5

The system is stable but not asymptotically stable and therefore does also not exhibit Zeno behaviour. We see that the system is not asymptotically stable due to that the output signal does not converge to the desired value but will oscillate with a constant amplitude after infinite time.

Task 6

The discretized system can be seen in Equations (5),(6) and(7)

$$\frac{z-1}{T_s}x[k] = Ru_\omega[k]cos(\theta[k]) \tag{5}$$

$$\frac{z-1}{T_s}y[k] = Ru_\omega[k]sin(\theta[k]) \tag{6}$$

$$\frac{z-1}{T_s}\theta[k] = \frac{R}{L}u_\Psi[k] \tag{7}$$

Task 7

With euler forward method we have that

$$\theta[h] \approx \theta[k] + \dot{\theta}[k]\tau_s$$

We have $\dot{\theta}$ from equation (3) and control signal u_Ψ in the assignment. This give the following equation for the system.

$$\theta[k+h] \approx \frac{RK_\Psi\tau_s(\theta^R - \theta[k])}{L} + \theta[k]$$

$$\theta[k+h] \approx (1 - \frac{RK_\Psi\tau_s)}{L})\theta[k] + \frac{RK_\Psi\tau_s)}{L}\theta^R$$

This is stable i the absolute value of the eigenvalues are less than 1 for the Φ matrix. The eigenvalue is

$$\lambda = |1 - \frac{RK_\Psi\tau_s)}{L}| < 1$$

This gives the the upper/lower boundary of $0 < K_\Psi < \frac{2L}{R\tau_s}$.

Task 8

It is not possible to reach the goal angle due to we only have a proportional controller and therefore always have a small static error.

K = 0.7*L/R

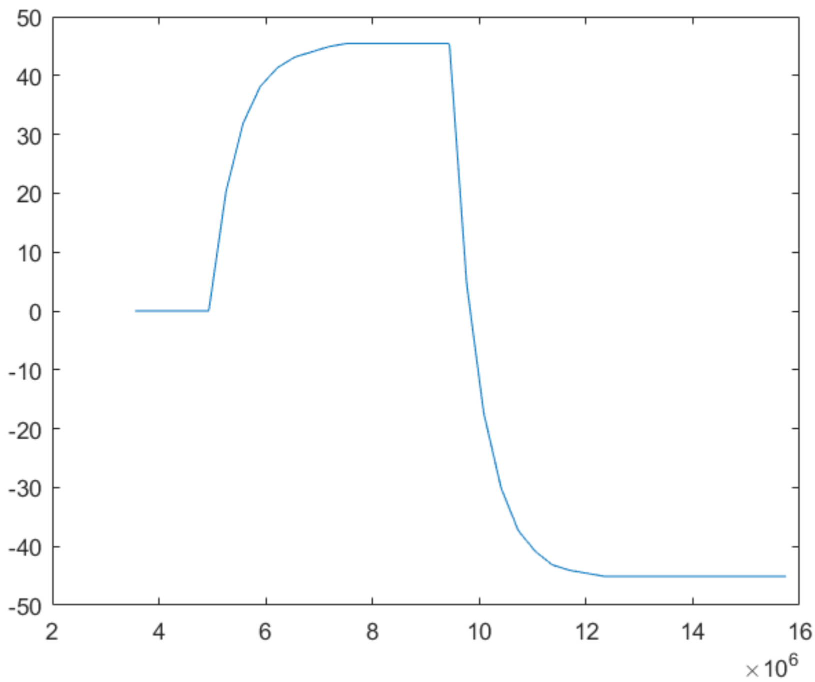


Figure 1: Performance of the controller

Task 9

With euler forward method we have that for the x variable

$$x[h] \approx x[k] + \dot{x}[k]\tau_s$$

We have $\dot{x}$ from equation (3) and control signal u_ω.

$$u_\omega[k] = K_\omega cos(\theta[k])(x_0 - x[k]) + K_\omega sin(\theta[k])(y_0 - y[k])$$

$$\dot{x[k]} = Ru_\omega[k]cos(\theta[k])$$

$$x[k+h] \approx (1 - K_\omega \tau_s R cos^2(\theta[k]))x[k] + K_1 + \Gamma y[k]$$

Where K_1 a constant of how $x[k+h]$ depends on x_0, y_0 and Γ is how it depends on the position i y direction $y[k]$. Assuming that $y[k]$ could be considered as an input signal we have stability if the absolute value of the eigenvalues in the stability equation is less than 1.

$$|\lambda| = |1 - K_\omega \tau_s R cos^2(\theta[k])| < 1$$

This gives the the upper/lower boundary of $0 < K_\omega < \frac{2}{R\tau_s}$ since the minumum upper boundary is found when $cos^2(\theta[k]) = 1$.

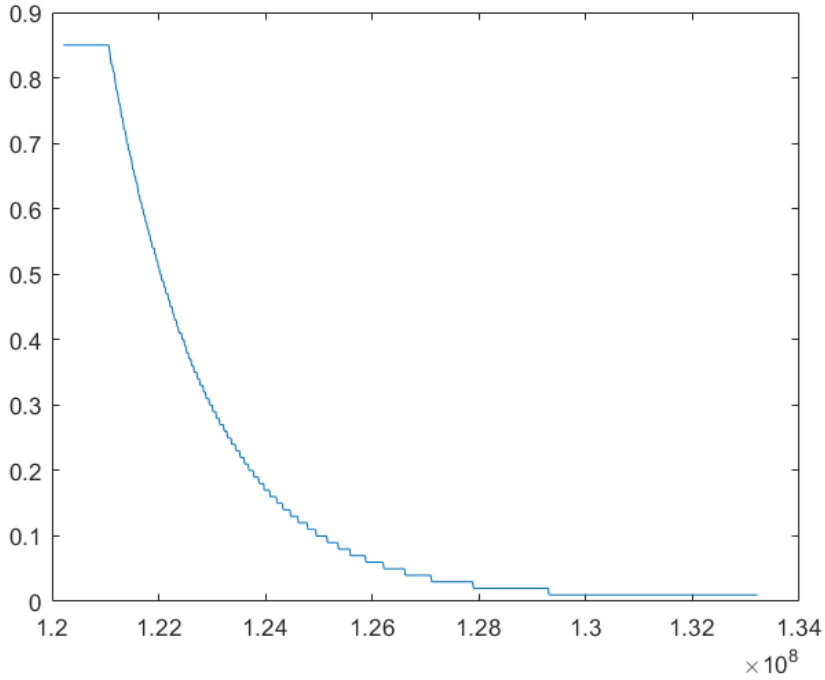
If the controller is analysed in the y direction the following eigenvalues are obtained.

$$|\lambda| = |1 - K_\omega \tau_s R sin^2(\theta[k])| < 1$$

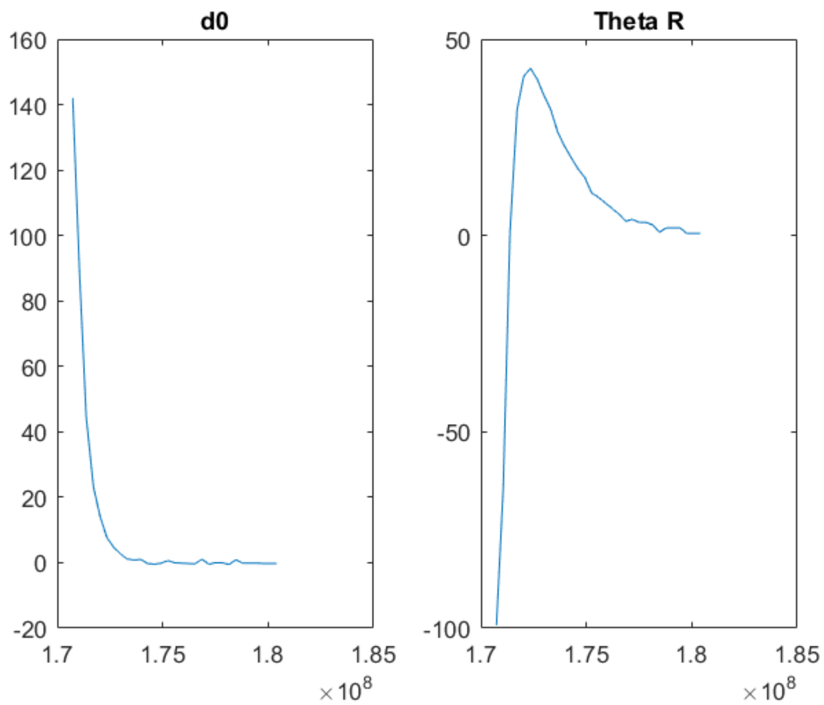
And since the maximum of $sin^2(\theta[k]) = cos^2(\theta[k]) = 1$ gives the same upper and lower bounds.

Task 10

K = 5 In the general case no due to the direction of the robot is not lined up with the point it wants to go to.



Task 11



Task 12

This section is very similar to task 9 whilst we have a slightly different controller thus this will not change the poles of the controller. Again the position approximated with euler forward.

$$x[h] \approx x[k] + \dot{x}[k]\tau_s$$

We have $\dot{x}$ from equation (3) and control signal u_ω.

$$u_\omega[k] = K_\omega cos(\theta[k])(x_g - x[k]) + K_\omega sin(\theta[k])(y_g - y[k])$$

$$\dot{x[k]} = Ru_\omega[k]cos(\theta[k])$$

$$x[k+h] \approx (1 - K_\omega \tau_s R cos^2(\theta[k]))x[k] + K_1 + \Gamma y[k]$$

Where K_1 a constant of how $x[k+h]$ depends on x_g, y_g and Γ is how it depends on the position i y direction $y[k]$. Assuming that $y[k]$ could be considered as an input signal we have stability if the absolute value of the eigenvalues in the stability equation is less than 1.

$$|\lambda| = |1 - K_\omega \tau_s R cos^2(\theta[k])| < 1$$

This gives the the upper/lower boundary of $0 < K_\omega < \frac{2}{R\tau_s}$ since the minumum upper boundary is found when $cos^2(\theta[k]) = 1$. This is just like the controller in task 9 but we control to a different position.

If the controller is analysed in the y direction the following eigenvalues are obtained.

$$|\lambda| = |1 - K_\omega \tau_s R sin^2(\theta[k])| < 1$$

And since the maximum of $sin^2(\theta[k]) = cos^2(\theta[k]) = 1$ gives the same upper and lower bounds.

Task 13

K = 30

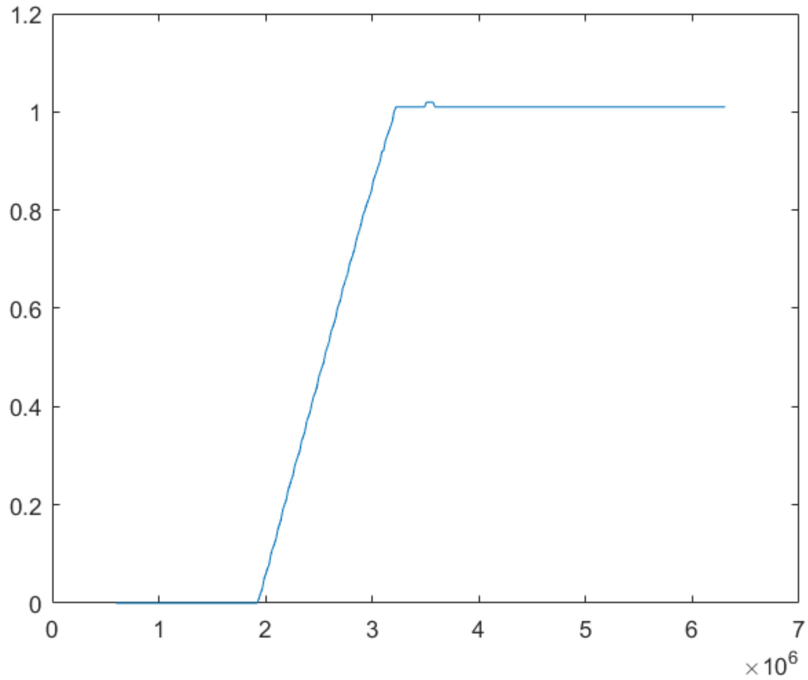


Figure 2: Controller performance

Task 14

Solution to the task

Task 15

Solution to the task

Task 16

Solution to the task

Task 17

Solution to the task

Task 18

$Q = \{Rotate, Line, Wait\}$
$X = \mathbb{R}^3$
$Init = (Rotation, (x0, y0, \theta 0)$
$f(Rotation, (x, y, \theta)) = (Ru_\omega * cos(\theta), Ru_\omega * sin(\theta), \frac{R}{L} u_\Psi)$
$f(Line, (x, y, \theta)) = (Ru_\omega * cos(\theta), Ru_\omega * sin(\theta), \frac{R}{L} u_\Psi)$

$f(Wait, (x, y, \theta)) = (0, 0, 0)$

$D(Rotation) = \{(x, y, \theta)\epsilon\mathbb{R}, -180 < \theta <= 180\}$
$D(Line) = \{(x, y, \theta)\epsilon\mathbb{R}, -180 < \theta <= 180\}$
$D(Wait) = \{(x, y, \theta)\epsilon\mathbb{R}, -180 < \theta <= 180\}$

$E = \{(Rotation, line), (Line, Wait), (Line, Rotation), (Wait, Rotation)\}$

$G(Rotation, Line) = \{(x, y, \theta)\epsilon\mathbb{R} : |\theta^R - \theta| < a, |x0 - x| < b, |y0 - y| < c\}$
$G(Line, Rotation) = \{(x, y, \theta)\epsilon\mathbb{R} : |xg - x| < d, |yg - y| < e\}$

$R((Rotation, Line), ((x, y, \theta)\epsilon\mathbb{R})) = R((Line, Wait), ((x, y, \theta)\epsilon\mathbb{R})) =$
$= R((Line, Rotation), ((x, y, \theta)\epsilon\mathbb{R})) = R((Wait, Rotation), ((x, y, \theta)\epsilon\mathbb{R})) =$
$\{(x, y, \theta)\epsilon\mathbb{R}\}$

Task 19

Solution to the task

Task 20

Done

Task 21

It was pretty hard to reach the exact point manually. This was probably due to the short delay in communication and that you needed to hold down the button for a short time for the robot to react.

Task 22

The path of the robot compared to the simulation with the same controllers can be seen in Figure 3.

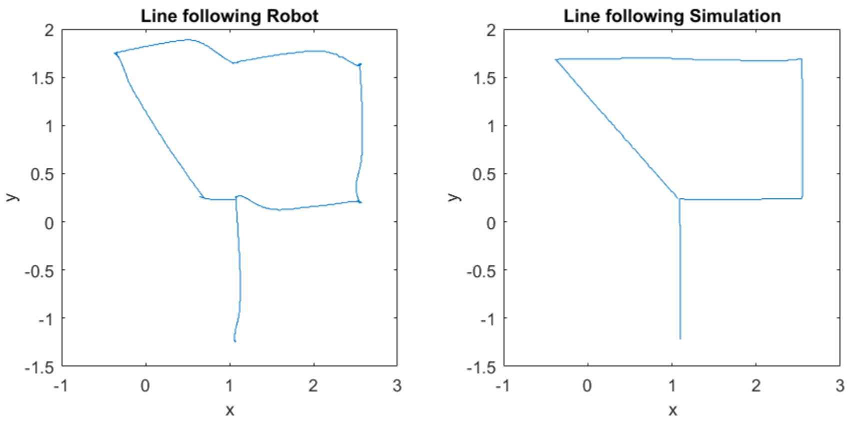


Figure 3: Comparison of line following between the real robot and the simulation

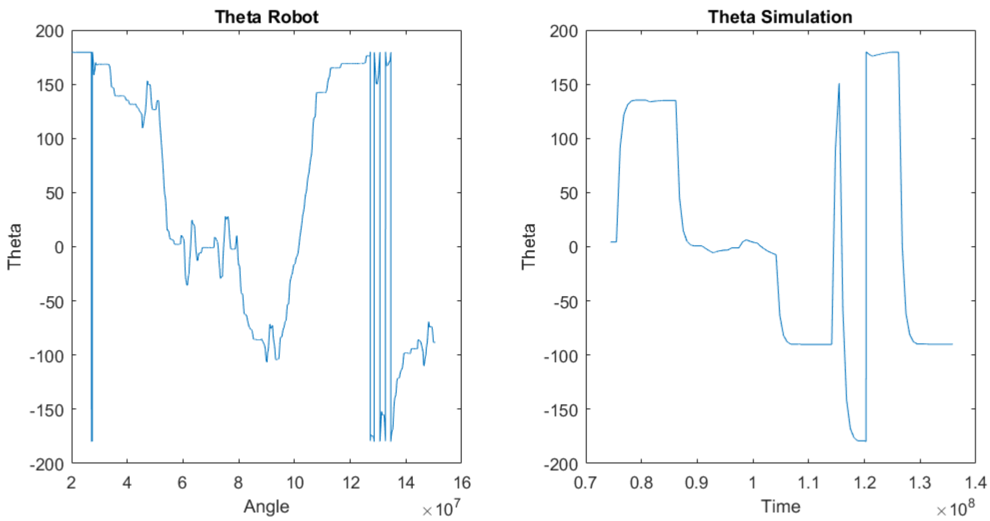


Figure 4: Comparison of θ between the real robot and the simulation

References

[1] Hassan K Khalil. *Nonlinear systems*. Prentice Hall, Upper Saddle river, 3. edition, 2002. ISBN 0-13-067389-7.

[2] Tobias Oetiker, Hubert Partl, Irene Hyna, and Elisabeth Schlegl. *The Not So Short Introduction to LaTeX 2ε*. Oetiker, OETIKER+PARTNER AG, Aarweg 15, 4600 Olten, Switzerland, 2008. http://www.ctan.org/info/lshort/.

[3] Shankar Sastry. *Nonlinear systems: analysis, stability, and control*, volume 10. Springer, New York, N.Y., 1999. ISBN 0-387-98513-1.