Assignment 8

TTK4215

Elias Olsen Almenningen

Version: 1.0 | Date: October 19, 2023

Contents

Problem 1

Recall the pendulum dynamics given in assignment 7:

$$ml^2\ddot{q} + \beta\dot{q} + mglsin(q) = \tau \tag{1}$$

where q is the angle in radians, measured clockwise from the downward hanging equilibrium, m is the mass, β is a damping coefficient, l is the length, g is the gravitational acceleration and τ is a control torque.

a)

In assignment 7, a controller was designed to locally stabilize the pendulum around the upright equilibrium $(q, \dot{q}) = (\pi, 0)$ *under the assumption that the parameters are known. Using simulations, try to get an estimate of the region of attraction of the closed-loop system, i.e., for what range of initial values of* q, $\dot{q}$ *does the pendulum state converge to the upright equilibrium?*

Since it is a closed-loop system, any initial conditions will converge to the upright equilibrium. This is confirmed by some test performed in simulink, with somewhat RAD initial conditions, see Figure 1.

Furthermore, since state feedback chooses the gains in K such that the closed-loop system is stable, all initial values must converge to the equilibrium. Therefore, the whole state space is the *region of attraction.*

b)

Now, in your simulations, impose the limited torque constraint $|\tau| \leq 0.5$ *Nm. Is the range of initial values from which the pendulum state now converges different than in a)?*

Adding a torque constraint is done using a saturation block in Simulink.

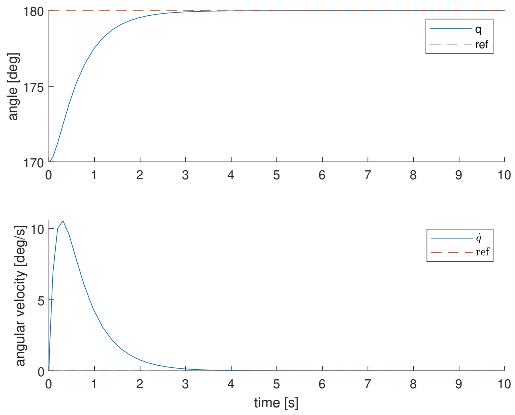
The Simulink test shows that the limit for q is $\approx 180° \pm 14.76°$, when $\dot{q} = 0$, see Figure 2. Therefore, the initial values for which the states converge to $(\pi,\ 0)$ are different from a).

c)

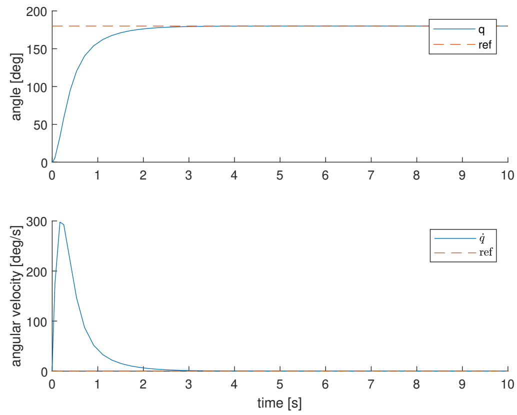
Define the total energy of the inverted pendulum as

$$E(q, \dot{q}) = \frac{1}{2}ml^2\dot{q}^2 + mgl(1 - cos(q)) \tag{2}$$

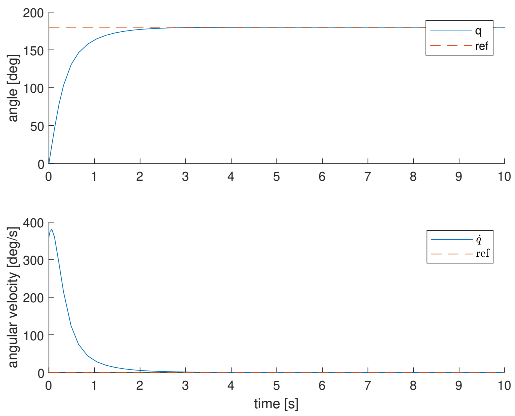
What is the desired total energy at the upright equilibrium? Denote this value Ed.



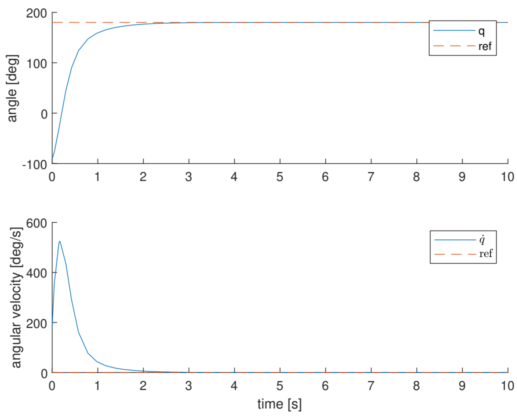
(a) $q = 170$, $\dot{q} = 0$



(b) $q =$, $\dot{q} = 0$



(c) $q = 0$, $\dot{q} = 360$



(d) $q = -90$, $\dot{q} = 180$

Figure 1: Finding region on attraction

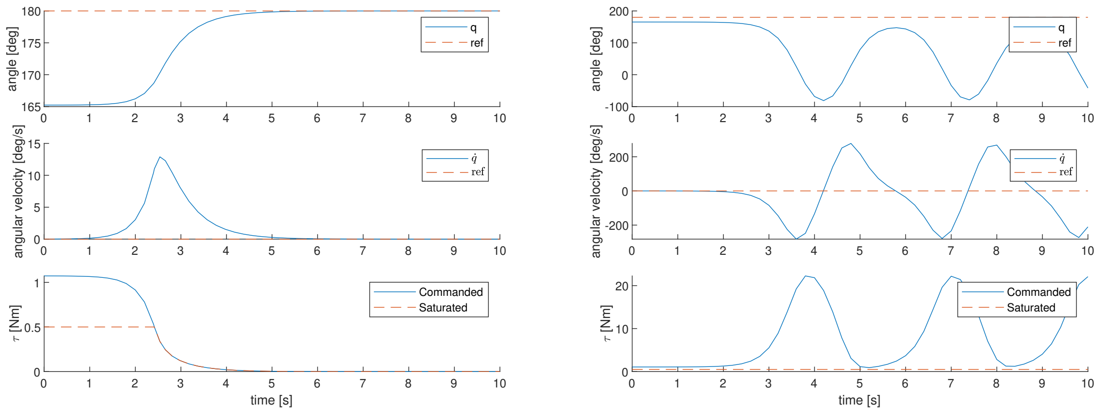
At the upright equilibrium $\dot{q} = 0$ and $q = \pi$, inserting this into Equation 2 gives

$$\begin{aligned} E(0, \pi) &= \frac{1}{2} ml^2 0^2 + mgl(1 - cos(\pi)) \\ E_d &= 2mgl \\ &= 0.2 \cdot 9.81 \cdot (1 - (-1)) \\ &= 3.924 \end{aligned} \tag{3}$$

d)

We will design a swing-up algorithm for the inverted pendulum using an energy-based approach. To this end, define the Lyapunov function candidate

$$V(\tilde{E}) = \frac{1}{2}\tilde{E}^2 \tag{4}$$



(a) $q = 165.24°$ **(b)** $q = 165.23°$

Figure 2: Finding region of attraction with saturation

where $\tilde{E} \triangleq E - E_d$. Choose a control law τ such that the function V is decreasing along the system trajectories when $\tilde{E} \neq 0$, $\dot{q} \neq 0$. i.e. such that $\dot{V} = \frac{\partial V}{\partial \tilde{E}} \dot{\tilde{E}} \leq 0$, and with a strict inequality when both $\tilde{E}$ and $\dot{q}$ do not equal zero. Try to argue why it is not necessarily a problem that $\dot{V}$ can be zero when $\dot{q}$ is zero.

Differentiating V gives

$$\begin{aligned}
\dot{V} &= \tilde{E}\dot{\tilde{E}} \\
&= \tilde{E}\dot{E} \\
\dot{E} &= ml^2\ddot{q} + mglsin(q)\dot{q} \\
\dot{V} &= \tilde{E}\left(ml^2\ddot{q}\dot{q} + mglsin(q)\dot{q}\right) \\
\ddot{q} &= \frac{1}{ml^2}\left(\tau - \beta\dot{q} - mglsin(q)\right) \\
\dot{V} &= \tilde{E}\left(\tau\dot{q} - \beta\dot{q}\right) \qquad (5)
\end{aligned}$$

Choosing $\tau = -k\tilde{E}\dot{q} + \beta\dot{q}$, where k is a constant of our choosing, gives

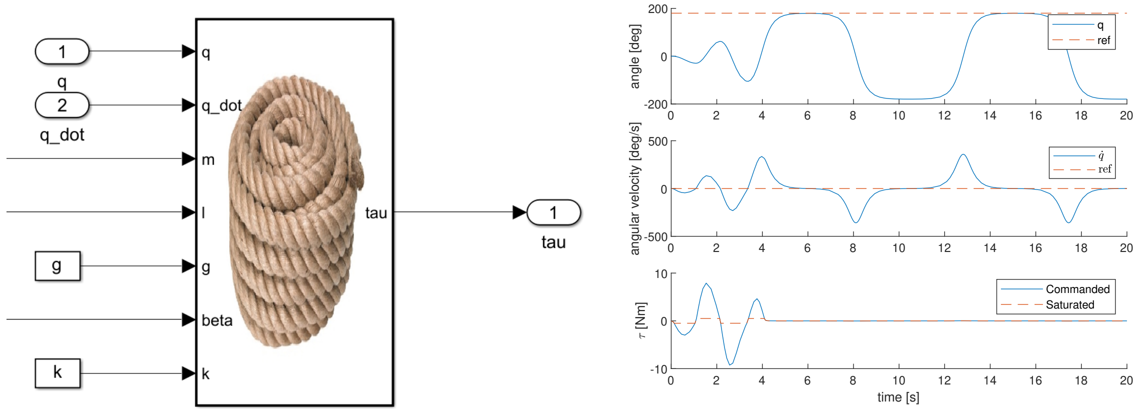
$$\dot{V} = -k\tilde{E}^2\dot{q}^2 < 0, \quad \text{when } \tilde{E},\, \dot{q} \neq 0 \qquad (6)$$

When $\dot{q}$ is zero, $\dot{V}$ is zero. $\tilde{E}$ is not necessarily zero, since $\tilde{E} = mgl - mglcos(q)$. This equation basically states that if the pendulum has some angle, there is some potential energy. This is not a problem, as this energy cannot increase, but only decrease.

e)

The implementation of the *swing-up* controller in Simulink can be seen in Figure 3

and the code in Figure 3a is



(a) The simulink function block **(b)** The swing up controller in action

Figure 3: Swing-up controller in Simulink

```
function tau = tau(q,q_dot,m,l,g,beta,k)
E =1/2 * m * l^2 * q_dot^2 + m * g * l * ( 1 - cos(q) );
E_d = 2 * m * g * l;
TildeE = E-E_d;
tau = -k*TildeE*q_dot + beta*q_dot;
```

The pendulum begins to oscillate back and forth between $[\pi - \pi]$, and after 4 seconds τ becomes ≈ 0, since there is enough energy in the system to keep then pendulum penduluming. If β had been larger, this would not have happened.

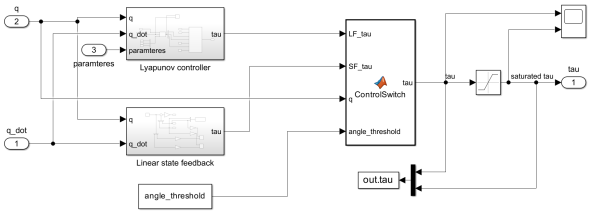
f)

The switching logic is implemented with an if loop in Simulink, with the layout and functional sequence shown in Figure 4

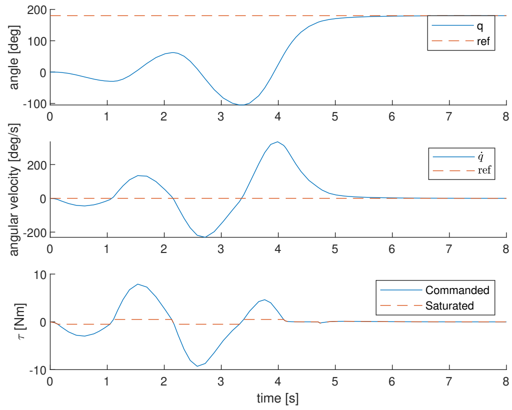
The switching logic code is

```
function tau = ControlSwitch(LF_tau, SF_tau,q,angle_threshold)
if q ≥ angle_threshold
    tau = SF_tau;
else
    tau = LF_tau;
end
```

The pendulum starts to swing back and forth, and when the threshold angle is reached, the LQR controller takes over.



(a) The simulink function block

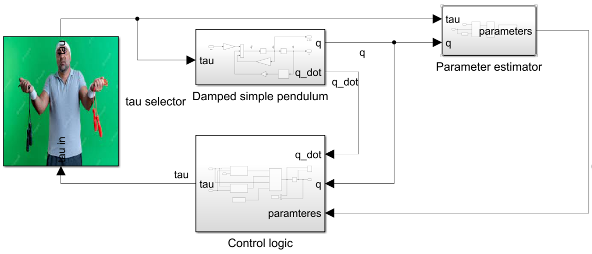


(b) The swing up controller and LQR in action with switching angle at 160°

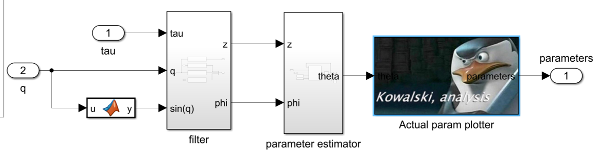
Figure 4: Swing-up and LQR in Simulink

g)

Combining the control algorithms with the parameter estimator from assignment 7 is done in Simulink as shown in Figure 5



(a) The complete simulink model



(b) The parameter estimator from Assingment 7

Figure 5: Swing-up and LQR in Simulink

The estimation can be done either by the regualr control signal generated by the control algorithms, or by the same signal used in Assignmnet 7, and then be switched over to the control signals. The last step is done in the *tau selector*. Both approaches work, but the parameters do not go to their correct values:

If only the control signal is used:

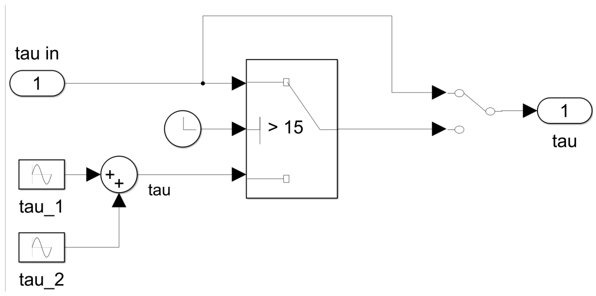
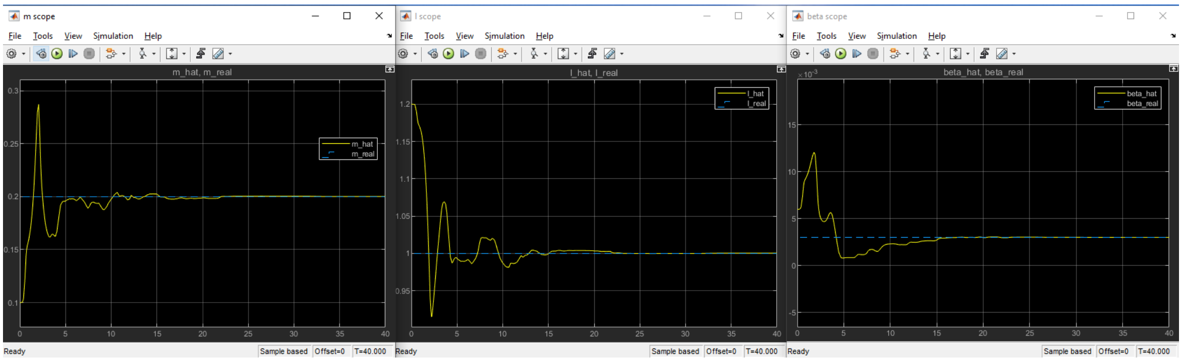
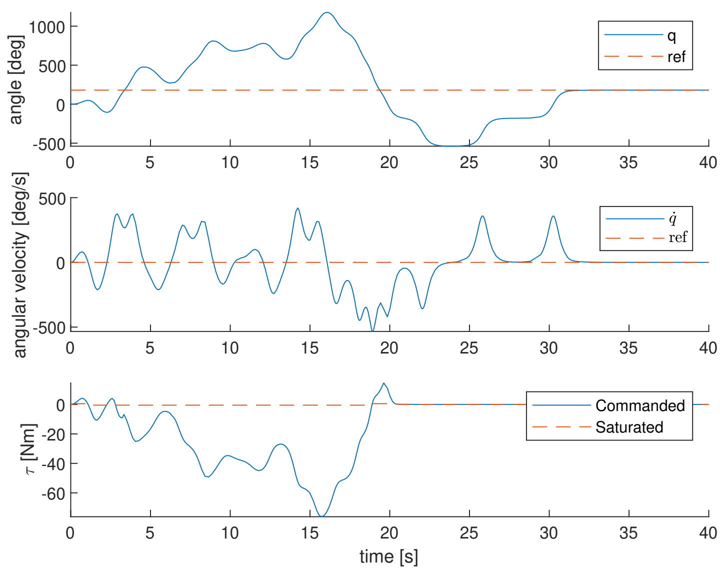


Figure 6: Tau selector

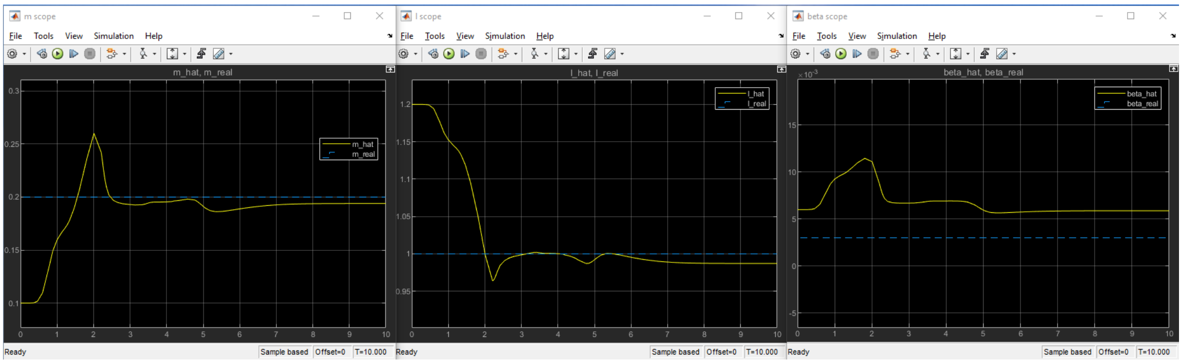


(a) Estimates reaching the correct values

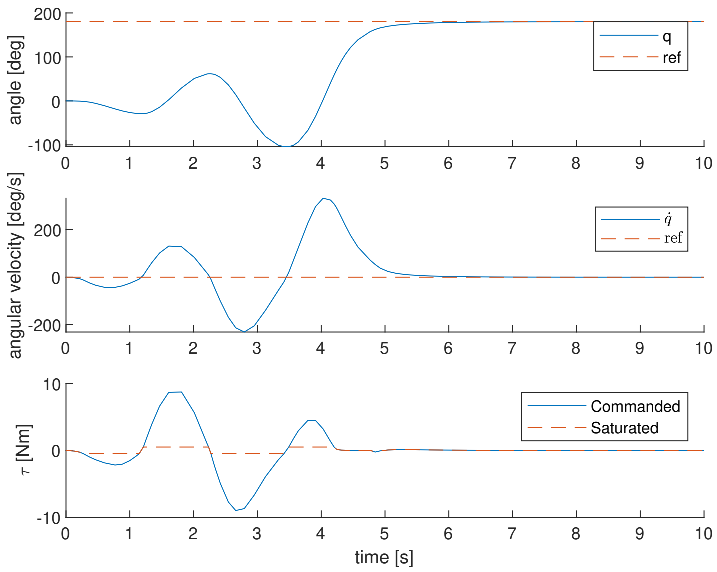


(b) Trajectory, but wierd :(

Figure 7: Convergence to correct parameters, and wierd trajectory



(a) Estimates reaching the correct values



(b) Trajectory, but nice :)

Figure 8: Convergence to wrong parameters, and nuce trajectory