Control Theory (RMS-1-612-s ; RIMA-1-612-s)

Exercises

Part 2 – State-Space Methods for Control System Design (updated: 5th June 2017)

Issued: 5th April 2021
Due: 24th May 2021

Phong Ba Dao, dr hab. inż., prof. AGH
Department of Robotics and Mechatronics
AGH University of Science and Technology

Keywords: state-space models, state-space coordinate transformation (similarity transform), inverted pendulum on cart, state feedback controller, state observer, pole placement, controllable canonical form, observable canonical form, modelling and simulation, Matlab/Simulink.

Remarks:

(1) Students are required to complete the following exercises/problems. Some tasks must be done without using MATLAB/Simulink, but the other tasks are required to solve with MATLAB/Simulink therefore you should read the questions and requirements carefully. Although some parts of the exercises were explained and guided in the instructions but they are just general solutions and not yet complete. Therefore, based on the instructions provided you should present and discuss in the exercise report your solutions and results obtained in detail.

(2) Before using Matlab commands (or functions) you should first check the syntax, descriptions, input arguments, output arguments, and illustrated examples by typing "help command".

(3) The exercise report must be typed on a word-processor and submitted as a single PDF file via **UPeL platform**.

1. State-space models and transformations

- **Exercise 1a** is given in "Lecture 7. State-space methods for control system design" on **slide 26**.
- **Exercise 1b** is given in "Lecture 7. State-space methods for control system design" on **slides 33-35**.

Notice that both exercises must be solved without using MATLAB/Simulink.

2. State-space control for inverted pendulum on cart

2.1. Introduction

Linearization of the nonlinear model of the inverted pendulum on a cart at the "upright" position (i.e. an equilibrium point) – performed in the previous exercise – resulted in a linearized model in terms of linear state-space equations with the state matrices (A, B, C, D)

$$A=\begin{bmatrix} 0 & 1 & 0 & 0 \\ 42.51 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -2.943 & 0 & 0 & 0 \end{bmatrix};\ B=\begin{bmatrix} 0 \\ -3.333 \\ 0 \\ 1 \end{bmatrix};\ C=\begin{bmatrix} C_1 \\ C_2 \end{bmatrix}=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix};\ D=\begin{bmatrix} D_1 \\ D_2 \end{bmatrix}=\begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad (1)$$

Two transfer functions were derived from the state-space model:

$$G_1(s) = \frac{\theta(s)}{U(s)} = \frac{-3.333s^2}{s^4 - 42.51s^2} \quad \text{and} \quad G_2(s) = \frac{X(s)}{U(s)} = \frac{s^2 - 32.7}{s^4 - 42.51s^2} \tag{2}$$

where

$G_1(s)$ is the transfer function from the input external force $u(t)$ to the output rod angle $\theta(t)$;

$G_2(s)$ is the transfer function from the input external force $u(t)$ to the output cart position $x(t)$.

From Equations (1) and (2) it is easy to know that the linearized model of the inverted pendulum on a cart has four (open-loop) poles, which are two poles locating at the origin ($p_1 = p_2 = 0$), one negative pole ($p_3 = -6.52$) and one positive pole ($p_4 = 6.52$), and therefore the linearized system is clearly unstable.

The impulse and step responses were respectively plotted in Figures 1 and 2 to investigate the dynamic characteristics of the linearized model of the inverted pendulum on a cart in case the system is not controlled. Based on the behaviour observed in Figures 1 and 2, some comments are given: Because of the fact that the linearization was performed only for small displacements around the "upright" position (i.e. $\bar{x} = 0$ and $\bar{\theta} = 0$) with an assumption that the external force acting on the cart is equal to zero (i.e. $\bar{u} = 0$). Hence, on running the simulation model, after a certain (short) time (e.g. t = 1 [s]) under the impact of input signals (e.g. impulse or unit step signals) the inverted pendulum has no longer stayed at the "upright" position (or in other words, it departed from this equilibrium) and therefore the linearized model at the "upright" equilibrium position is not correct anymore. For this reason, a controller is highly needed to keep the rod staying "upright". In other words, we are going to design a stable closed-loop control system for the unstable inverted pendulum on a cart.

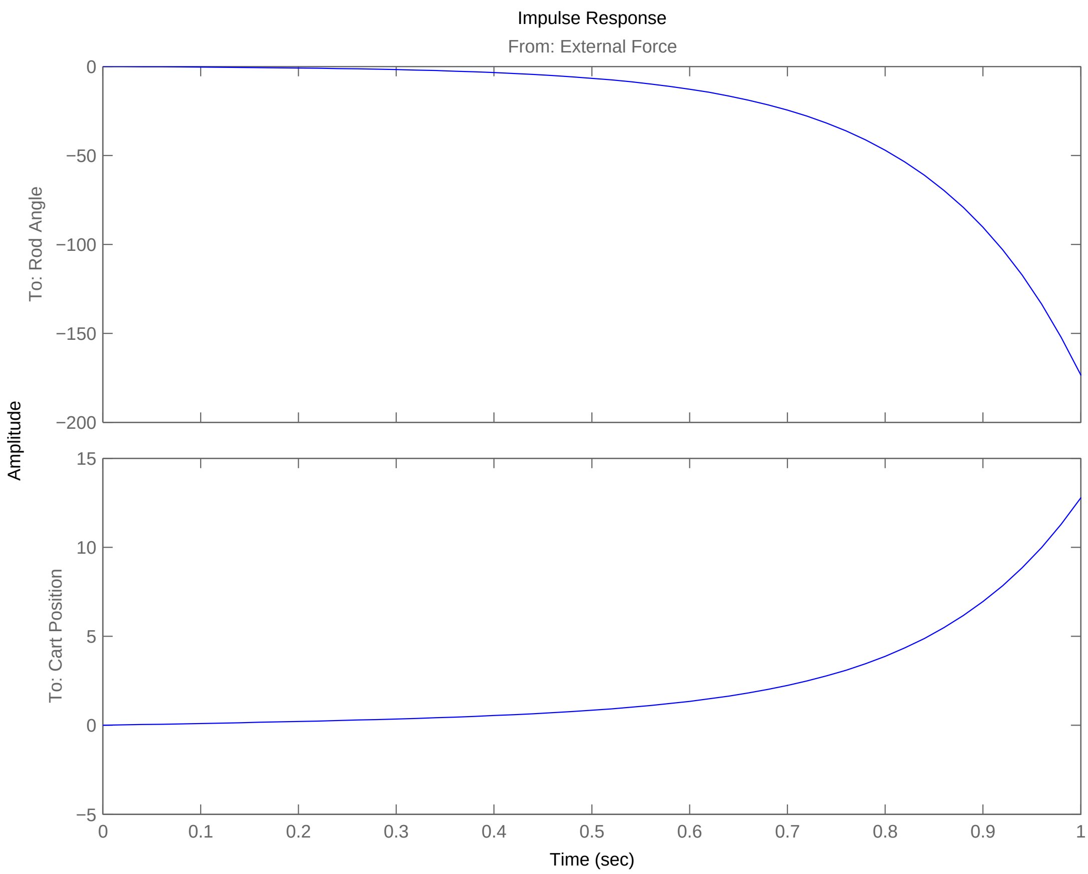

Figure 1. Impulse response of the linearized system in case without a controller.

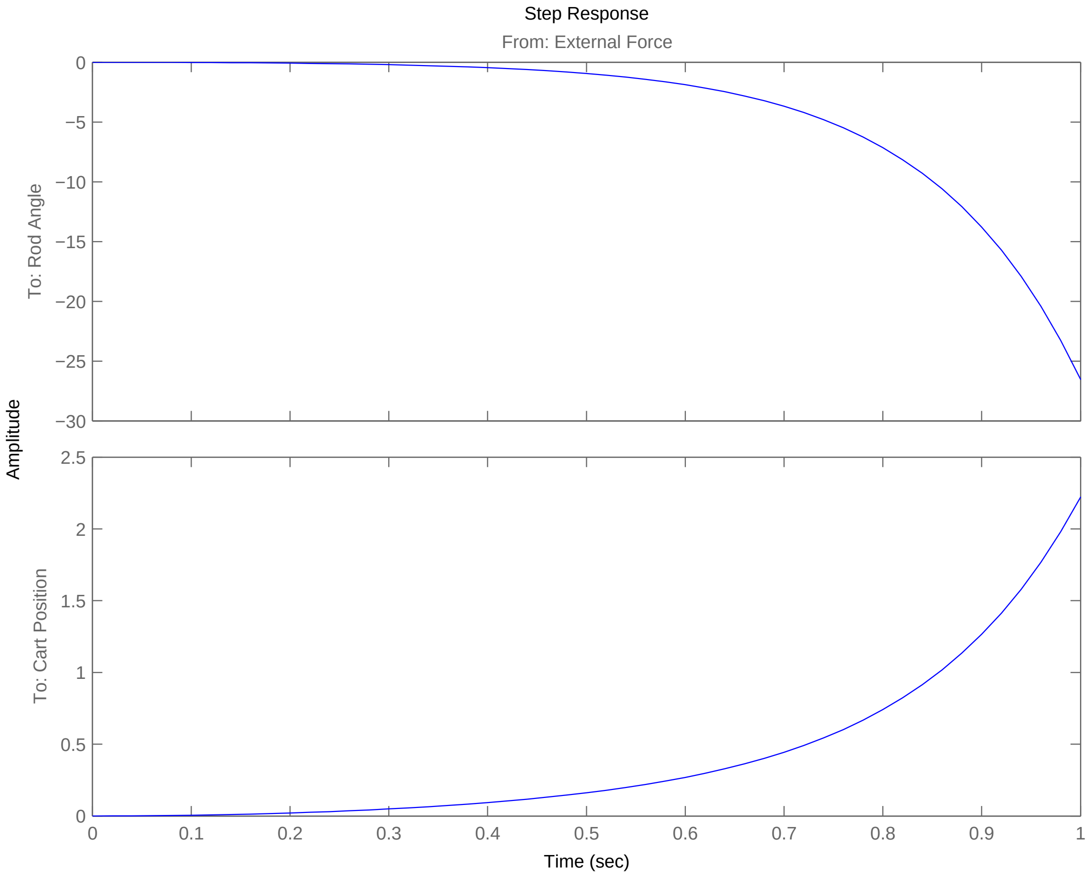

Figure 2. Step response of the linearized system in case without a controller.

2.2. Problems

Based on the linearized model of the inverted pendulum on a cart you are required to solve the following seven tasks:

1. Investigate the controllability and observability of the linearized model.
2. Assume that only the cart position $x(t)$, or in other words, only the state variable $x_3(t)$ can be measured, design a state feedback controller using pole placement technique – which is based on the *controllable canonical form* method – such that the linearized system will be assigned the four new stable closed-loop poles at [-1 -1 -5 -6.52].
3. **Use Matlab** to compute and simulate the closed-loop state feedback controller and check if it is assigned the four desired stable poles, i.e. [-1 -1 -5 -6.52] and then test performances of the controlled system by plotting the impulse and step responses. Give some comments about the dynamic characteristics of the system based on the simulation results.
4. Again, suppose that only the cart position $x(t)$, or in other words, only the state variable $x_3(t)$ can be measured, design a state observer using pole placement technique – which is based on the *observable canonical form* method – such that the state observer of the linearized system will be assigned four poles at [-10 -10 -50 -65.2], i.e. the observer poles are 10 times greater than the desired closed-loop poles.
5. **Use Matlab** to compute the closed-loop state observer and check if it is assigned the four desired poles, i.e. [-10 -10 -50 -65.2].
6. **Use Matlab** to compute and simulate the closed-loop control system (i.e. the observer-based state feedback control system) by combining the state feedback controller (obtained in Task

2) with the state observer (obtained in Task 4). Test performances of the observer-based state feedback control system by plotting impulse and step responses. Compare and discuss the obtained results with the relevant ones when only the state feedback controller is used.

7. Repeat Tasks 2–6 by applying the Ackermann's formula "acker" and the pole placement method using the function "place". It means that this task is done by **using Matlab**. Do these two functions "acker" and "place" work well?
   - If these two functions work then compare the results obtained with the relevant ones obtained by using the controllable and observable canonical form methods. Does "acker" and "place" give the same results (i.e. the resulting state feedback gain F and observer gain L) as the canonical form methods?
   - In case if they do not work then explain why.

Notice that Tasks 1, 2 and 4 must be solved without using MATLAB/Simulink.

2.3. Instructions

Recall from the previous exercise that the state variables were selected as follows:

$$\begin{cases} x_1(t) = \theta(t) \\ x_2(t) = \dot{\theta}(t) \\ x_3(t) = x(t) \\ x_4(t) = \dot{x}(t) \end{cases} \Rightarrow \begin{cases} \dot{x}_1(t) = \dot{\theta}(t) = x_2(t) \\ \dot{x}_2(t) = \ddot{\theta}(t) \\ \dot{x}_3(t) = \dot{x}(t) = x_4(t) \\ \dot{x}_4(t) = \ddot{x}(t) \end{cases} \tag{3}$$

and the system has one input $u(t)$ and two outputs, i.e. the output equations take the form

$$y(t) = \begin{bmatrix} \theta(t) \\ x(t) \end{bmatrix} = \begin{bmatrix} x_1(t) \\ x_3(t) \end{bmatrix} \Rightarrow y(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} \tag{4}$$

Notice that in Equations (3) and (4): $x_1(t),\ x_2(t),\ x_3(t),\ x_4(t)$ are state variables; and $x(t),\ \dot{x}(t)$ and $\ddot{x}(t)$ are cart position, cart velocity and cart acceleration, respectively. This implies that the linearized model of the inverted pendulum on a cart at the "upright" position is a *single-input multi-output (SIMO)* system.

In general, the output equation depends on the measurement taken, which depends on the sensors available. The cart position can be measured by placing one or two optical encoders on the wheels and the rod angle can be measured by placing an encoder at the rod pivot point. With regard to the cart velocity and the rod angle velocity, these parameters can be acquired by placing tachometers on a wheel and at the rod pivot point. The output equations for different cases can be found in the **Appendix** (on **page 16**) of the previous exercise (Part 1 – Modelling of Dynamic Systems and Linearization of Nonlinear Systems).

It is necessary to note that, in order to obtain the results displayed as desired, the following Matlab commands should be used to set (or configure) the resulting output format and accuracy (the number of digits after the coma).

```
format('long', 'g');  % display numbers as, e.g. F = [30.0440825688073   4.75985321100917
                                                      0.996941896024464   2.34617737003058]
format('short', 'g');  % display numbers as, e.g. F = [30.044   4.7599   0.99694   2.3462]
format('bank');  % display numbers as, e.g. F = [30.04   4.76   1.00   2.35]
```

Task 1:

In general, checking the controllability for a SIMO system is straightforward because it has only one control input. More specifically, the controllability of the given system can be checked by verifying if the rank of the controllability matrix M is equal to the order n of the system ($n = 4$ in this case). This is done by verifying whether $\det(M) \neq 0$, where $M = [B \quad AB \quad A^2B \quad A^3B]$. This implies that the hand calculations should be performed to obtain M and $\det(M)$.

By analogy, the observability of the given system can be checked by verifying if the rank of the observability matrix N is equal to the order n of the system. This is done by validating whether $\det(N) \neq 0$, where $N = [C \quad CA \quad CA^2 \quad CA^3]^T$. This implies that the hand calculations should be performed to obtain N and $\det(N)$.

However, because the analysed system has two outputs so that one should check whether it is observable from each single output (i.e. from either the output rod angle $\theta(t)$ or the output cart position $x(t)$), besides the normal case. Therefore, the observability of the given system should be checked for three cases:

- When only the rod angle $\theta(t)$ (i.e. the state variable $x_1(t)$) can be measured, thereby $C = C_1$;
- When only the cart position $x(t)$ (i.e. the state variable $x_3(t)$) can be measured, thereby $C = C_2$;
- In the normal case when both the rod angle $\theta(t)$ (i.e. the state variable $x_1(t)$) and the cart position $x(t)$ (i.e. the state variable $x_3(t)$) can be measured.

After that you can use Matlab to examine or verify the results calculated by hand.

```
N = obsv(A, C1);  % y = x1 = "rod angle". In this case, the system is unobservable
N = obsv(A, C2);  % y = x3 = "cart position". In this case, the system is observable
N = obsv(A, C);  % y = [x1 ; x3]. In this case, the system is observable
```

where $C_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$ and $C_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}$.

Task 2:

You need to perform the following five steps, which have been already presented in "Lecture 7. State-space methods for control system design" on **page 61**.

Step 1: Given the closed-loop poles that one wants to place/assign, determine the desired closed-loop characteristic polynomial (P_{cl})

You should obtain

$$P_{cl} = s^4 + 13.52s^3 + 56.64s^2 + 76.72s + 32.6 \tag{5}$$

Step 2: Find the similarity transform matrix (T) based on the method presented in the lecture slides and then transform the state matrices (A, B, C, D) of the original state-space model to the controllable canonical form (A_c, B_c, C_c, D_c) using the following transformation formulas

$$A_c = T^{-1}AT\,;\; B_c = T^{-1}B\,;\; C_c = CT\,;\; D_c = D \tag{6}$$

Notice that one can use **use Matlab** to calculate the inverse of the controllability matrix (i.e. M^{-1}) and T^{-1}. Then one should obtain

$$T = \begin{bmatrix} 0 & -3.333 & 0 & 0 \\ -3.333 & 0 & 0 & 0 \\ 0 & 1 & 0 & -32.7 \\ 1 & 0 & -32.7 & 0 \end{bmatrix} \tag{7}$$

$$A_c = \begin{bmatrix} 0 & 42.51 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{8}$$

$$B_c = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{9}$$

$$C_c = \begin{bmatrix} 0 & -3.333 & 0 & 0 \\ 0 & 1 & 0 & -32.7 \end{bmatrix} \tag{10}$$

$$D_c = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{11}$$

It can be noticed that A_c and B_c are in the controllable canonical form.

You should check if (A, B, C, D) and (A_c, B_c, C_c, D_c) have the same transfer functions and open-loop eigenvalues. Of course, they do because we have performed a similarity transform which allows us to transform a state-space model into another equivalent one but retains the same eigenvalues. You can use Matlab to perform this task.

In the general case, a state-space system in the controllable canonical form has the matrices A_c, B_c, C_c given by

$$A_c = \begin{bmatrix} a_1 & a_2 & a_3 & \dots & a_n \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix};\ B_c = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix};\ C_c = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix} \tag{12}$$

The controllable canonical form has the following properties:

- The open-loop characteristic polynomial can be directly determined from the first line of A_c, i.e.

  $$P = \det(sI - A_c) = s^n - a_1 s^{n-1} - a_2 s^{n-2} - \dots - a_n \tag{13}$$

- A system of this structure is always controllable as its controllability matrix always has full rank.
- The transfer functions of the system are immediately given by using the first line of A_c and C_c, i.e.

$$G_1(s) = \frac{c_{11}s^{n-1} + c_{12}s^{n-2} + \cdots + c_{1n}}{s^n - a_1 s^{n-1} - a_2 s^{n-2} - \dots - a_n}, \dots, G_n(s) = \frac{c_{n1}s^{n-1} + c_{n2}s^{n-2} + \cdots + c_{nn}}{s^n - a_1 s^{n-1} - a_2 s^{n-2} - \dots - a_n} \tag{14}$$

For examples: one can straightforwardly obtain the two transfer functions $G_1(s)$ and $G_2(s)$ given in Equation (2) by using Equation (14) with A_c given in Equation (8) and C_c given in Equation (10).

Step 3: Determine the open-loop characteristic polynomial (P_{op})

As discussed above that (A, B, C, D) and (A_c, B_c, C_c, D_c) have the same transfer functions and open-loop eigenvalues so that the open-loop characteristic polynomial (P_{op}) can be directly determined from the first line of A_c using Equation (13), instead of calculating $\det(sI - A)$.

You should obtain

$$P_{op} = s^4 - 42.51s^2 \tag{15}$$

Step 4: Define F_c

You should obtain

$$F_c = \begin{bmatrix} -13.52 & -99.15 & -76.72 & -32.6 \end{bmatrix} \tag{16}$$

Step 5: Compute the state feedback gain F for the system (A, B, C, D) using the formula

$$F = F_c T^{-1} \tag{17}$$

You should obtain

$$F = \begin{bmatrix} 30.044 & 4.759 & 0.997 & 2.346 \end{bmatrix} \tag{18}$$

In essence, the key idea of this method is that instead of directly calculating the state feedback gain F for the original state-space model (A, B, C, D), we first transform (A, B, C, D) to the controllable canonical form (A_c, B_c, C_c, D_c) and then calculate the state feedback gain F_c for this system. Finally, we obtain the state feedback gain F for the original system (A, B, C, D) by using Equation (17). To confirm this, you can use Matlab to check $eig(A + BF)$ and compare the obtained results with $eig(A_c + B_c F_c)$. Indeed, they should result in the same desired closed-loop poles or eigenvalues (i.e. [-1 -1 -5 -6.52]).

Task 3:

In this task you will first implement the whole five steps presented in Task 2 **using Matlab**. Then the closed-loop state feedback system can be formed by using the code

```
Asf = A+B*F; Bsf = B; Csf = C; Dsf = D;  % "sf" means "state feedback"
```

As discussed above, you can use either $eig(A + BF)$ or $eig(A_c + B_c F_c)$ to check if the closed-loop state feedback controller is assigned the four stable poles as desired.

You should obtain the impulse and step responses of the linearized model of the inverted pendulum on a cart – controlled by the state feedback controller designed – as shown in Figures 3 and 4, respectively. What can you say about the dynamic characteristics of the controlled system based on these simulation results?

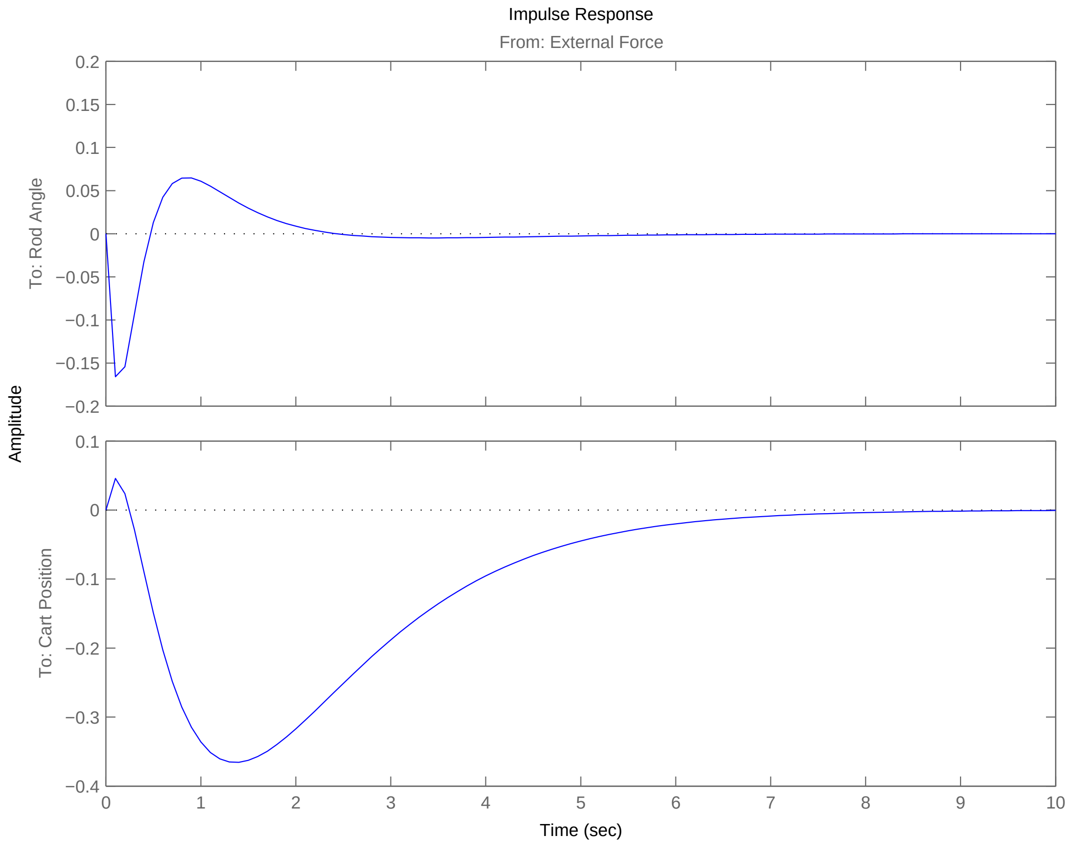

Figure 3. Impulse response of the linearized system controlled by the state feedback controller.

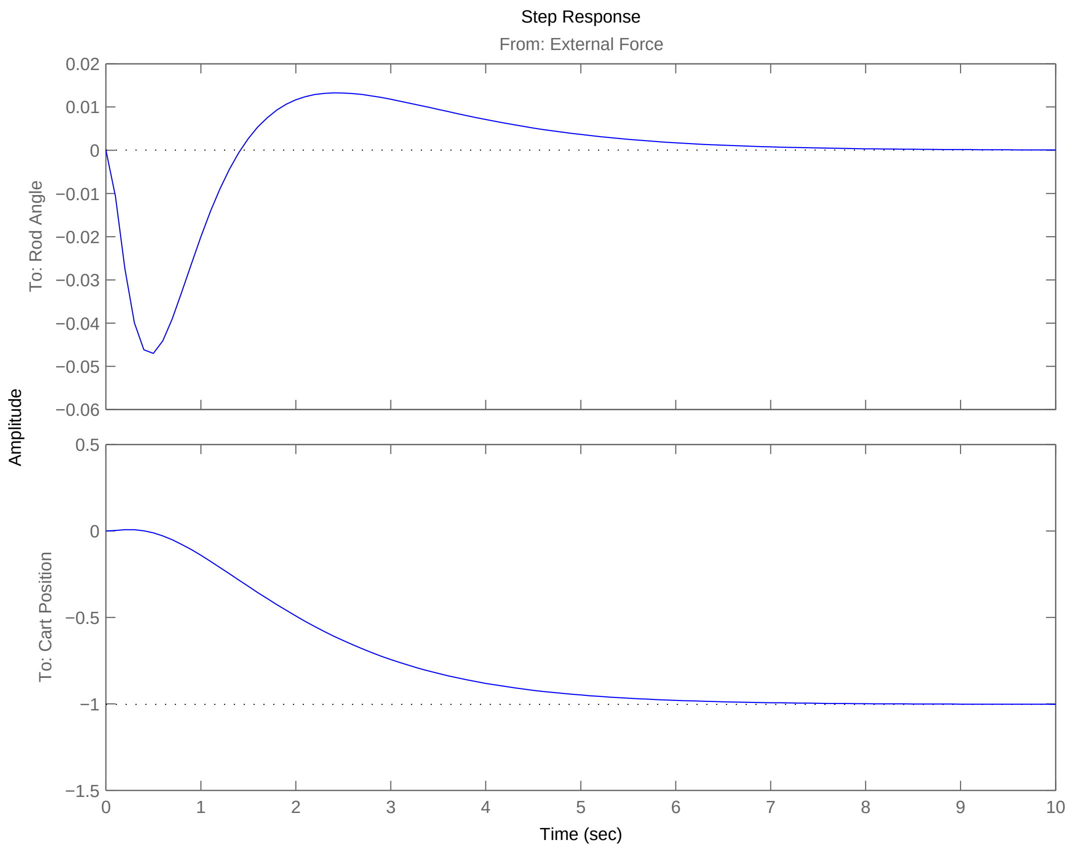

Figure 4. Step response of the linearized system controlled by the state feedback controller.

Task 4:

You need to perform the following five steps, which have been already presented in "Lecture 7. State-space methods for control system design" on **page 83**.

Step 1: Given the desired poles that one wants to place/assign for the observer, determine the desired observer polynomial (P_o)

You should obtain

$$P_o = s^4 + 135s^3 + 5664s^2 + 76720s + 326000 \tag{19}$$

Step 2: Find the similarity transform matrix (S) based on the method presented in the lecture slides and then transform the state matrices (A, B, C, D) of the original state-space model to the observable canonical form (A_o, B_o, C_o, D_o) using the following transformation formulas

$$A_o = S^{-1}AS\ ;\ B_o = S^{-1}B\ ;\ C_o = CS\ ;\ D_o = D \tag{20}$$

Since it is assumed that only the cart position $x(t)$, or in other words, only the state variable $x_3(t)$ can be measured, therefore you should use correctly the form of the observability matrix N when calculating the similarity transform matrix (S).

Notice that one can **use Matlab** to calculate the inverse of the observability matrix (i.e. N^{-1}) and S^{-1}. Then one should obtain

$$S = \begin{bmatrix} -14.444 & 0 & -0.339 & 0 \\ 0 & -14.444 & 0 & -0.339 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{21}$$

$$A_o = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 42.51 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{22}$$

$$B_o = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -32.7 \end{bmatrix} \tag{23}$$

$$C_o = \begin{bmatrix} -14.444 & 0 & -0.339 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{24}$$

$$D_o = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{25}$$

It is noticed that A_o is in the observable canonical form. However, one can notice in Equation (24) that only the second row of C_o is in the observable canonical form. This is because of the fact that we have only considered the output "cart position" $x(t)$ in the state observer design. In other words, only the second row of C was transformed to the observable canonical form.

In addition, it can be seen that $A_o = A_c^T$.

You should check if (A, B, C, D) and (A_o, B_o, C_o, D_o) have the same transfer functions and open-loop eigenvalues. Of course, they do because we have performed a similarity transform which allows us to transform a state-space model into another equivalent one but retains the same eigenvalues. You can use Matlab to perform this task.

Hint: The three commands $eig(A)$, $eig(A_c)$ and $eig(A_o)$ should give the same results, i.e. the open-loop poles or eigenvalues.

Step 3: Determine the open-loop characteristic polynomial (P_{op})

As discussed above that (A,B,C,D), (A_c,B_c,C_c,D_c) and (A_o,B_o,C_o,D_o) have the same transfer functions and open-loop eigenvalues so that this can be done in the same way as described in Step 3 of Task 2.

You should obtain

$$P_{op} = s^4 - 42.51s^2 \tag{26}$$

Step 4: Define L_o

You should obtain

$$L_o = \begin{bmatrix} -135 & -5706 & -76720 & -326000 \end{bmatrix}^T \tag{27}$$

Step 5: Compute the observer gain L for the system (A,B,C,D) using the formula

$$L = SL_o \tag{28}$$

You should obtain

$$L = \begin{bmatrix} 28022 & 193200 & -135 & -5706 \end{bmatrix}^T \tag{29}$$

In essence, the key idea of this method is that instead of directly calculating the observer gain L for the original state-space model (A,B,C,D), we first transform (A,B,C,D) to the observable canonical form (A_o,B_o,C_o,D_o) and then calculate the observer gain L_o for this system. Finally, we obtain the observer gain L for the original system (A,B,C,D) by using Equation (28). To confirm this, you can use Matlab to check $eig(A + LC(2,:))$ and compare the obtained results with $eig(A_o + L_oC_o(2,:))$. You can see that only the second row of C and C_o is considered in the computations. The reason was already explained above. Indeed, both should give the same desired observer poles or eigenvalues (i.e. [-10 -10 -50 -65.2]).

Task 5:

In this task you will first implement the whole five steps presented in Task 4 **using Matlab**. Then the closed-loop state observer can be formed by using the code

```
Aso = A+L*C(2,:); Bso = B; Cso = C; Dso = D;  % "so" means "state observer"
```

Remember that when computing the closed-loop state observer, only the matrix C_2 or $C(2,:)$ is used. You can use $eig(A + LC(2,:))$ to check if the closed-loop state observer is assigned the four desired poles.

Task 6:

The closed-loop control system (i.e. the observer-based state feedback controller) can be formed by combining the state feedback controller with the state observer using the codes below

```
Areg = [ (A+B*F)  B*F;  zeros(size(A))  (A+L*C(2,:)) ];  % only C2 is used
Breg = [ B;  zeros(size(B)) ];
Creg = [ C(2,:)  zeros(size(C(2,:))) ];  % only C2 is used
```

```
Dreg = 0;
```

Next, the impulse and step responses should be plotted for the output "cart position" $x(t)$ in two cases:

1. When only the state feedback controller is present;
2. When both the state feedback controller and the state observer are present (i.e. the observer-based state feedback control system is used).

You should obtain the responses as depicted in Figures 5 and 6.

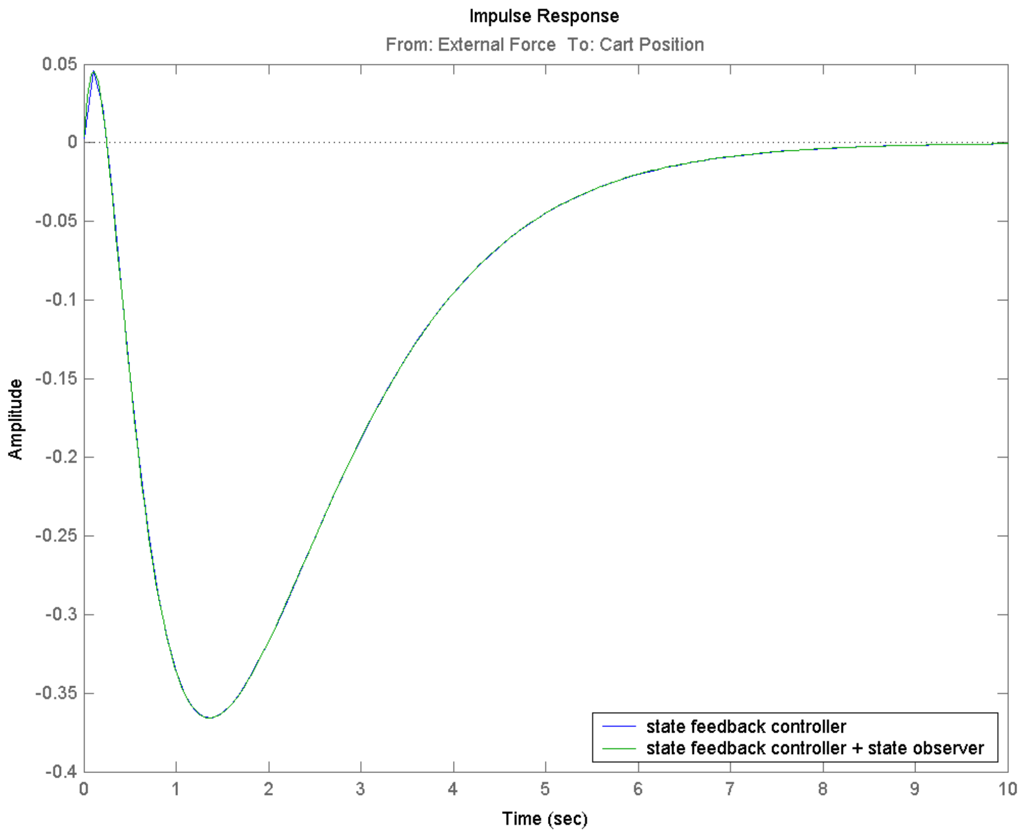

Figure 5. Impulse response of the output "cart position" $x(t)$ in two cases: (i) only the state feedback controller is present; (ii) the observer-based state feedback control system.

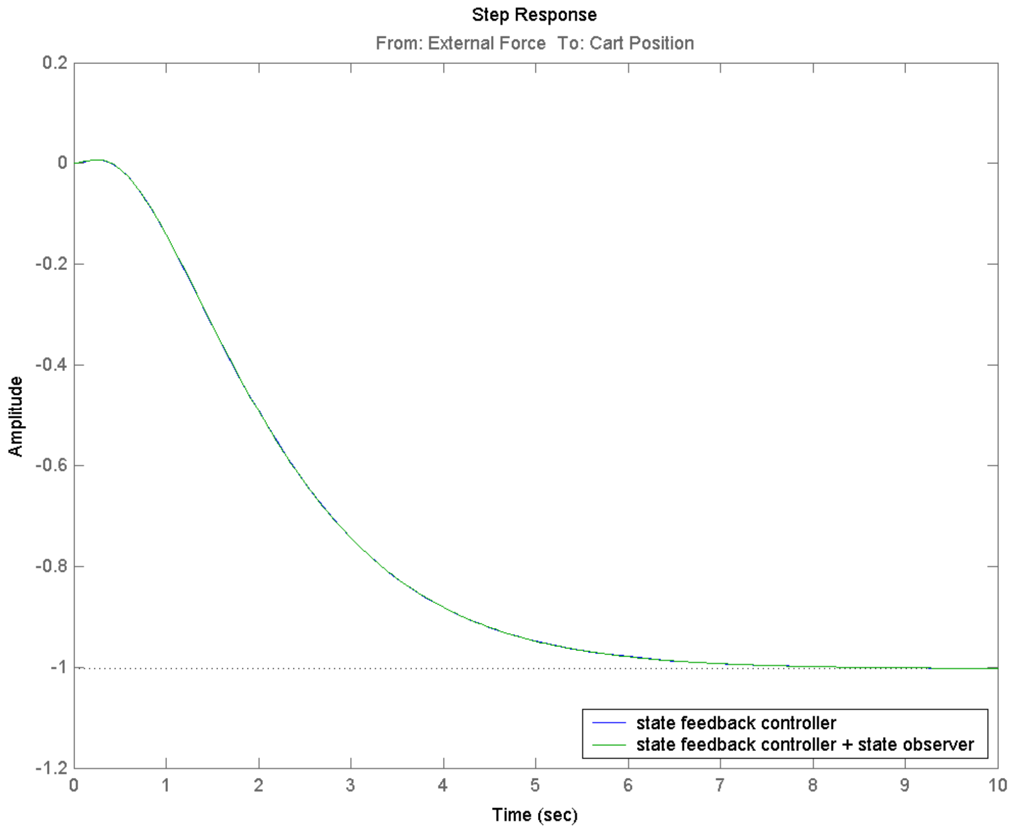

Figure 6. Step response of the output "cart position" $x(t)$ in two cases: (i) only the state feedback controller is present; (ii) the observer-based state feedback control system.

Give some comments on the simulation results obtained from the two design cases.

Recall from **"Lecture 7. State-space methods for control system design"** that the observer-based state feedback control system in Figure 7 with the state feedback gain F and the observer gain L results in **2n** closed loop poles, coinciding with the eigenvalues of the two matrices ($A+BF$ and $A+LC$). In this case, the approximated states $\hat{x}$ – obtained from the observer – are used to create the state feedback control law (i.e. $u = F\hat{x}$). This is different to the case when only the state feedback controller is used in which the state feedback control law uses directly the states of the system (i.e. $u = Fx$). Hence, when one wants to design a state feedback controller for a system, two assumptions should be met: (i) the system must be controllable and (ii) all state variables are available for feedback. In case if the second assumption is not met then a state observer might be considered.

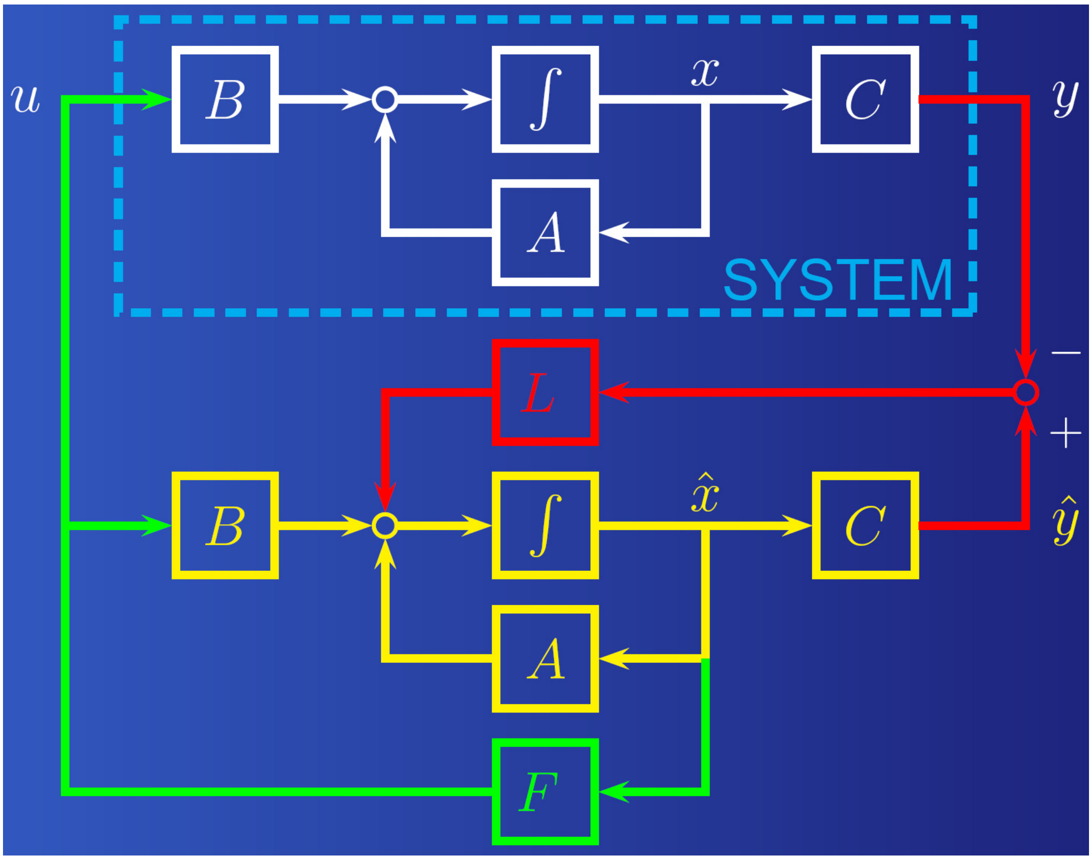

Figure 7. Structure of the observer-based state feedback control system.

Task 7:

Both the Ackermann's formula "acker" and the pole placement method using the function "place" have some limitations as discussed in the following. You are required to confirm those statements through solving Tasks 2–6 using these two functions with the support of Matlab. Performing this task will help you to understand this issue and select a proper pole placement method for a certain system (i.e. SISO, SIMO, MISO, or MIMO systems).

Limitations:

k = acker(A,b,p)

- "acker" uses Ackermann's formula [1] to calculate a gain vector k such that the state feedback $u=-kx$ places the closed-loop poles at the desired locations p. Here A is the state transmitter matrix and b is the input-to-state transmission vector.
- This function is only applicable to single-input single-output (SISO) systems and the pair (A,b) must be controllable. Note that this method is not numerically reliable and starts to break down rapidly for systems of order greater than 5 or for weakly controllable systems so that it should only be used for systems with a small number of states. The function "place" is a more general and numerically robust alternative to "acker".

K = place(A,B,p)

- "place" computes a gain matrix K such that the state feedback $u=-Kx$ places the closed-loop poles at the desired locations p, provided that (A,B) is controllable. This function works for multi-input multi-output (MIMO) systems.
- "place" uses the algorithm of [2] which, for multi-input systems, optimizes the choice of eigenvectors for a robust solution. It is recommended to use "place" rather than "acker" even

for single-input systems. However, if you want to place two or more poles at the same position, "place" will not work. In this case you can use "acker".

- In high-order systems, some choices of pole locations result in very large gains. The sensitivity problems attached with large gains suggest caution in the use of pole placement techniques. See [3] for results from numerical testing.

Where to place the desired closed-loop poles?

The role of dominant poles (or modes) leads to the following design procedure:

- First, choose a second-order system with desired dynamics;
- Next, place two desired poles at the two poles of this second-order system;
- Then, choose all other desired poles to be faster (to render them less dominant) but not too fast (to avoid too large control action);
- Finally, assign the desired poles or modes and then evaluate the system by dynamic simulation.

Typically this process has to be iterated to achieve a good design.

Caution:

Pole placement can be badly conditioned if you choose unrealistic pole locations. In particular, you should avoid [4]:

- Placing multiple poles at the same location.
- Moving poles that are weakly controllable and/or observable. This typically requires high gain, which in turn makes the entire closed-loop system very sensitive to perturbation (i.e. process disturbances and measurement noise).

Another approach is to use the so-called **Linear-Quadratic-Gaussian (LQG) control** design. LQG control is a modern state-space technique for designing optimal dynamic regulators and servo controllers with integral action (also known as set point trackers). This technique allows users to trade off regulation/tracker performance and control effort, and to take into account process disturbances and measurement noise.

To design LQG regulators and set point trackers, you perform the following steps [4]:

1. Construct the LQ-optimal gain.
2. Construct a Kalman filter (state estimator).
3. Form the LQG design by connecting the LQ-optimal gain and the Kalman filter.

For more information about using LQG design to create LQG regulators, see [4].

References

[1] Kailath, T., Linear Systems, Prentice-Hall, 1980, p. 201.

[2] Kautsky, J. and N.K. Nichols, "Robust Pole Assignment in Linear State Feedback," Int. J. Control, 41 (1985), pp. 1129-1155.

[3] Laub, A.J. and M. Wette, Algorithms and Software for Pole Assignment and Observers, UCRL-15646 Rev. 1, EE Dept., Univ. of Calif., Santa Barbara, CA, Sept. 1984.

[4] Functions for Compensator Design: Linear-Quadratic-Gaussian (LQG) Design.
Link: http://www.mathworks.com/help/control/getstart/functions-for-compensator-design.html