State-space approach for linear control systems – lectures

Igor Reznichenko
Moscow State University

Chapter 3. Design of a Linear State Feedback Control Law

Lecture 5: State feedback control law - the introduction, eigenvalue placement

A linear time-invariant state equation

$$\begin{cases} \dot{\vec{x}}(t) = A \cdot \vec{x}(t) + B \cdot \vec{u}(t), \\ \vec{x}(t_0) = \overrightarrow{x_0}, \\ \vec{y}(t) = C \cdot \vec{x}(t) \end{cases} \tag{3.1}$$

with $A(n \times n), B(n \times r), \vec{u}(t)(r \times 1)$ represents the open-loop system or *plant* to be controlled. We apply the state feedback in the form

$$u(t) = -K \cdot \vec{x}(t) + \vec{r}(t) \tag{3.2}$$

Introducing (3.2) to (3.1) yields

$$\begin{cases} \dot{\vec{x}}(t) = (A - BK) \cdot \vec{x}(t) + B \cdot \vec{r}(t), \\ \vec{x}(t_0) = \overrightarrow{x_0}, \\ \vec{y}(t) = C \cdot \vec{x}(t) \end{cases} \tag{3.3}$$

where K is the *gain matrix* $(r \times n)$, $\vec{r}(t)$ is the new *external reference input* and $\vec{u}(t)$ we shall refer to as the *open-loop input*. The block diagram for this is

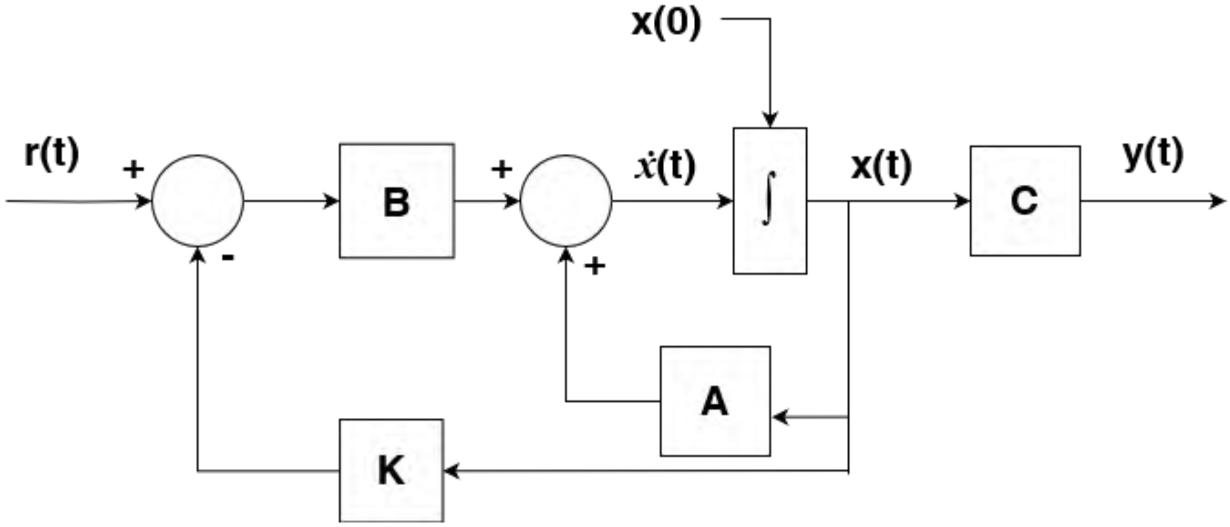

Figure 3.1 A block diagram for a system with a linear feedback control law [1].

In general yields

$$\begin{pmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_r(t) \end{pmatrix} = \begin{pmatrix} k_{11} & k_{12} & \cdots & k_{1n} \\ k_{21} & k_{22} & \cdots & k_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ k_{r1} & k_{r2} & \cdots & k_{rn} \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{pmatrix} + \begin{pmatrix} r_1(t) \\ r_2(t) \\ \vdots \\ r_n(t) \end{pmatrix} \tag{3.4}$$

but for the single-input, single-output case the feedback gain K is a $(1 \times n)$ row vector and the reference input $r(t)$ is a scalar signal. Then, the state feedback control law has the form

$$u(t) = -(K_1\ K_2\ \dots\ K_n)\begin{pmatrix} x_1(t) \\ x_2(t) \\ \cdots \\ x_n(t) \end{pmatrix} + r(t) = -K_1 x_1 - K_2 x_2 - \cdots - K_n x_n + r(t) \qquad (3.5)$$

This control law is mainly used to yield desired transient response characteristics and to counter disturbances in order to maintain an equilibrium state.

Theorem. A closed-loop system is asymptotically stable as $(A - BK)$ have strictly negative real-part eigenvalues.

Shaping the transient response.

Apart from stability we also care about the overshoot, rise time, peak time and the settling time of the step response. From the point of view of the state-space methods we would like to translate these values into closed-loop system eigenvalues, which is called *shaping the transient response*.

Control engineering often uses first- and second-order systems as approximations [2]. A first-order system without zeros is defined as $T_1(s) = a/(s + a)$ transfer function. With a unit step input $R(s) = 1/s$ the output of such system is

$$C(s) = R(s)T_1(s) = \frac{a}{s(s+a)} = \frac{1}{s} - \frac{1}{s+a}, \quad c(t) = 1 - e^{-at}, \quad \tau = \frac{1}{a}$$

where τ is the time constant of the process.

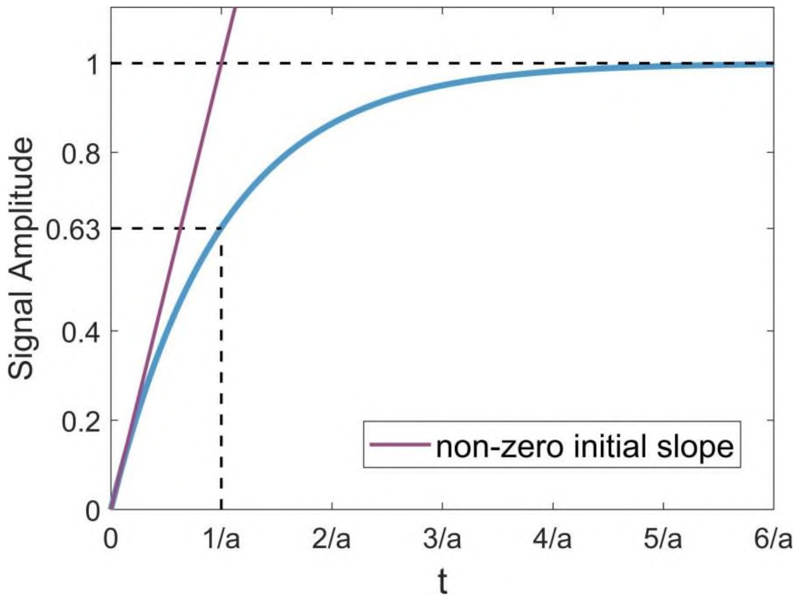

Figure 3.2 A step response of a first-order system.

Notice, that the initial signal slope at $t = 0$ is greater than zero $c'(0) = a$. This system has a single eigenvalue$\lambda = -1/\tau$. The time constant τ is defined as the time it takes the signal to reach 63% of its final value. The value $a = 1/\tau$ is sometimes called the exponential frequency. First-order system behavior is showed on Figure 2. First-order approximation I usually chosen if it is known (for example, by experiment) that the slope at $t = 0$ is nonzero.

A general second order differential equation

$$\ddot{y}(t) + a_1\dot{y}(t) + a_0 y(t) = b_0 u(t) \tag{3.6}$$

has the associated transfer function

$$H(s) = \frac{b_0}{s^2 + a_1 s + a_0}, \tag{3.7}$$

and the associated state space representation

$$\begin{cases} x_1(t) = y(t), \\ x_2(t) = \dot{y}(t) = \dot{x_1}(t), \\ \vec{x}(t = 0) = 0, \\ \dot{x_2}(t) = -a_1 x_2 - a_0 x_1 + b_0 u(t), \end{cases} \qquad \begin{pmatrix} \dot{x_1} \\ \dot{x_2} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -a_0 & -a_1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ b_0 \end{pmatrix} u(t). \tag{3.8}$$

As for the general case

$$\frac{d^n y}{dt^n} + a_{n-1}\frac{d^{n-1}y}{dt^{n-1}} + a_{n-2}\frac{d^{n-2}y}{dt^{n-2}} + \cdots + a_1\frac{dy}{dt} + a_0 y(t) = bu(t) \tag{3.9}$$

it translates to

$$A^{(n\times n)} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}; \; B = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ b_0 \end{pmatrix}; \tag{3.10}$$

The standard second-order transfer function form is

$$H(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}, \tag{3.11}$$

where ξ is called the *damping ratio*. Without it, if $\xi = 0$, the system just oscillates (see Figure 3a). The step response in the case of $0 < \xi < 1$ is seen in Figure 3.3b.

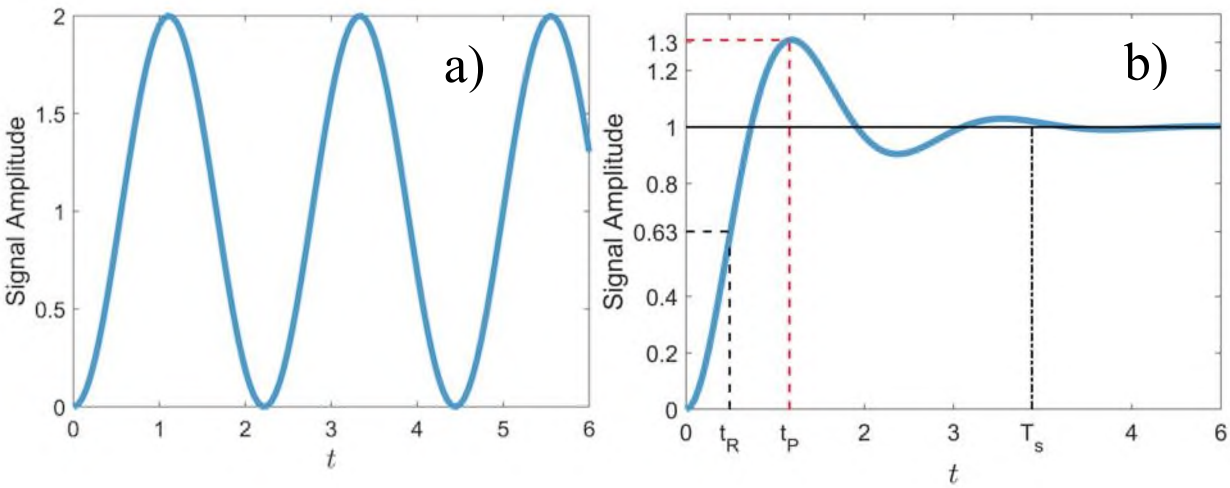

Figure 3.3 Step responses of second-order systems: a) ξ=0; b) 0<ξ<1;

For $\xi = 0$ the transfer function is $H(s) = \frac{{\omega_n}^2}{s^2+{\omega_n}^2}$, where ω_n is the *natural frequency*.

For 0 < ξ < 1 the transfer function will be

$$H(s) = \frac{{\omega_n}^2}{s^2 + 2\xi\omega_n s + {\omega_n}^2}.$$

The *damped natural frequency* is defined as

$$\omega_d = \omega_n\sqrt{1-\xi^2}. \#(3.11a)$$

For (3.11)

$$A = \begin{pmatrix} 0 & 1 \\ -{\omega_n}^2 & -2\xi\omega_n \end{pmatrix}; B = \begin{pmatrix} 0 \\ {\omega_n}^2 \end{pmatrix}; \#(3.11b)$$

the characteristic polynomial is

$$det(\lambda I - A) = \ det\begin{pmatrix} \lambda & -1 \\ {\omega_n}^2 & \lambda + 2\xi\omega_n \end{pmatrix} = 0$$

$$\lambda(\lambda + 2\xi\omega_n) + {\omega_n}^2 = 0, \qquad \lambda^2 + 2\xi\omega_n\lambda + {\omega_n}^2 = 0,$$

$$\frac{D}{4} = \xi^2{\omega_n}^2 - {\omega_n}^2 = {\omega_n}^2(\xi^2 - 1)\#(3.11c)$$

$$\lambda_{1,2} = -\xi\omega_n \pm \omega_n\sqrt{\xi^2 - 1} = -\xi\omega_n \pm i\cdot\omega_d, \#(3.12)$$

which is a complex conjugate pair. The unit step response for (3.11) is

$$\begin{cases} y(t) = 1 - \dfrac{e^{-\xi\omega_n t}}{\sqrt{1-\xi^2}} sin(\omega_d t + \theta) \\ \theta = arccos(\xi) \\ 0 < \xi < 1 \end{cases} \#(3.13)$$

where θ is the *damping angle*.

Homework: acquire $\Phi(t)$ with A and this expression for $y(t)$. Solve the state equation.

For the underdamped $0 < \xi < 1$ case these four step response characteristics can be related to ξ and ω_n. *Rise time* t_R (Fig. 3.3b) is the time the signal reaches from 10% to 90% of the steady-state value. For $0.3 < \xi < 0.8$ it approximately is [1]:

$$t_R \approx \frac{2.16\xi + 0.6}{\omega_n} \#(3.13)$$

Peak time t_P is given by exact expression

$$t_P = \frac{\pi}{\omega_n\sqrt{1-\xi^2}} = \frac{\pi}{\omega_d} \#(3.14)$$

As for the *percent overshoot*

$$PO = \frac{peak\ value - steady\text{—}state\ value}{steady\text{—}state\ value} \cdot 100\%$$

it can also be computed exactly:

$$PO = 100 \cdot exp\left(\frac{-\xi\pi}{\sqrt{1-\xi^2}}\right) \#(3.15)$$

Settling time is defined as the time t_S (Fig. 3.3b) for the signal to reach and **stay** within a $\pm 3\%$ band about the steady-state value. It is approximately by

$$t_S \approx \frac{4}{\xi\omega_n}. \#(3.16)$$

Formulas (3.13)-(3.16) are going to be used extensively in our design of the linear state feedback control law. From given values of PO and settling time one can obtain the desired damping ratio and natural frequency using the following formulas:

$$\xi = \frac{\left|ln\frac{PO}{100}\right|}{\sqrt{\pi^2 + ln^2\left(\frac{PO}{100}\right)}}, \qquad \#(3.17)$$

$$\omega_n = \frac{4}{\xi \cdot t_S}. \#(3.18)$$

Example 3.1. Step-response characteristics if a linear translational mechanical system from example 1.2.

Important: Before in example 1.2 we had $u(t) = f(t)$ as a *force*. Since in steady-state case the only force is $F = k \cdot \Delta x = k \cdot y(t) = k \cdot x_1$ we redefine the input as $u(t) = f(t) : k$ so that we get an input as the final displacement. That way since step function is $f(t) : k = u(t) = 1$ that way we get steady unit response as$y(t > t_s) = 1$.

Let $m = 2$ kg, μ $= 2\, N \cdot s/m,\ k = 20\, H/m$.

$$A = \begin{pmatrix} 0 & 1 \\ -k/m & -\mu/m \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -10 & -1 \end{pmatrix}, B = \begin{pmatrix} 0 \\ k/m \end{pmatrix}$$

since $\dot{x}_2 = -\frac{k}{m}x_1 - \frac{\mu}{m}x_2 + \frac{f(t)}{m}$ and $u(t) = f(t)/k$.

Comparing this to (3.11b):

$\Rightarrow \omega_n{}^2 = 10,\ \omega_n = \sqrt{10} \approx 3{,}16\frac{\text{rad}}{\text{s}},\ \xi = \frac{1}{2\sqrt{10}} \approx 0{,}158,\ \ \omega_d = \omega_n\sqrt{1-\xi^2} = 3{,}12$ rad/s.

The characteristic polynomial (3.11c) is $\lambda^2 + 2\xi\omega_n\lambda + \omega_n{}^2 = 0 \Rightarrow \lambda^2 + \lambda + 10 = 0$

From 3.12 $\lambda_{1,2} = -\frac{1}{2} \pm 3{,}12i$.

$$\theta = \arccos\xi = 80{,}9^{\circ} \approx 1{,}41 \text{ rad}$$

$$y(t) = 1 - 1{,}01e^{-0.5t}\, sin(3{,}12t + 80{,}9^{\circ}) = 1 - e^{-0.5t}\, sin(3{,}12t + 80{,}9^{\circ})$$

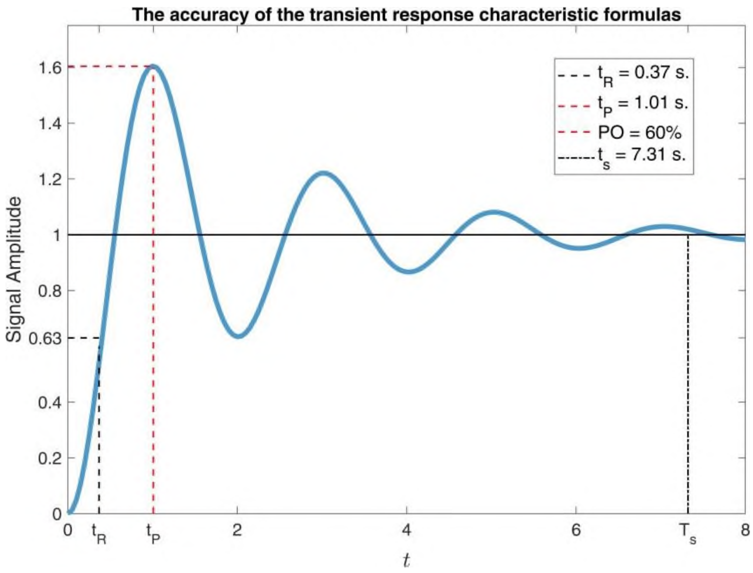

Figure 3.4 The step response and the actual step response characteristics.

Formulas (3.13) - (3.16) yield

$$t_R \approx 0{,}3 \text{ s.}, t_P \approx 1{,}01 \text{ s.}, PO = 0{,}59 \cdot 100\% = 59\%; \;\; t_S \approx 8 \text{ s.}$$

Figure 3.4 shows the step response. Note that since $\xi \approx 0{,}158 < 0{,}3$ the rise time approximation is a little off: t = 0,3 s. versus the actual t_R =0,37 s. Same for the settling time approximation t_S =8 s. versus the actual t_S =7,32 s.

The main goal of this chapter is to introduce the closed-loop eigenvalue placement via state feedback. It is a process matching values of the gain matrix K in (3.2) to the chosen optimal eigenvalues or desired eigenvalues. We are going to address this in the next lectures as we underline the importance of the second-order system's transient response characteristics formulas (3.13-3.16).

Lecture 6 - State feedback control law. Controllability. Closed-loop eigenvalue placement via state feedback. Controller canonical form.

Consider a linear time invariant differential equation:

$$\begin{cases}\dot{\vec{x}}(t) = A\vec{x}(t) + B\vec{u}(t)\\ \vec{x}(t_0) = \vec{x}_0\end{cases}, \#(3.19)$$

Definition. A state $\vec{x}^* \in R^n$ is **controllable to the origin** if for initial time to there exists a finite time $t_f > t_0$ and a piecewise continuous input signal $\vec{u}^*(t)$ defined on $[t_0, t_f]$ so that with the initial state $\vec{x}(t_0) = \vec{x}^*$the final state is

$$\vec{x}(t_f) = e^{A(t_f - t_0)}\vec{x}^* + \int_0^{t_f} e^{A(t_f-\tau)}B\vec{u}^*(\tau)\,d\tau = 0 \in R^n.$$

The state equation 3.19 is called **controllable** if energy state $\vec{x}$ possible for 3.19 is controllable to the origin.

This means that no matter what fixed state $\vec{x}^* = \vec{x}(t_0)$ we choose as a starting point for initial condition there exists a control signal $\vec{u}^*(t)$ that brings all the state variables to the zero eventually.

An easy way to determine whether a state equation is controllable is to use following method.

Theorem 3.1. The linear state equation 3.19 is controllable if and only if rank $P = rank\ (B \quad AB \quad A^2B \quad \dots \quad A^{n-1}B) = n.$

We shall call P the **controllability matrix**. Rank of a matrix is equal to the maximum number of linearly independent columns or rows in this matrix.

For example, $\vec{x}_1 = \begin{pmatrix}1\\1\end{pmatrix}$ and $\vec{x}_2 = \begin{pmatrix}-6\\-6\end{pmatrix}$ are **linearly dependent** vectors since $\vec{x}_2 = -6 \cdot \vec{x}_1$, $-6 \cdot \vec{x}_1 + \vec{x}_2 = \begin{pmatrix}0\\0\end{pmatrix}$. So there exist two numbers α_1 = -6 and α_2 = 1that $\alpha_1 \cdot \vec{x}_1 + \alpha_2 \cdot \vec{x}_2 = 0$ and both of these numbers are not zero.

On the other hand $\vec{x}_1 = \begin{pmatrix}0\\3\end{pmatrix}$ and $\vec{x}_2 = \begin{pmatrix}1\\4\end{pmatrix}$ are linearly independent since

$\alpha_1 \cdot \vec{x}_1 + \alpha_2 \cdot \vec{x}_2 = \begin{pmatrix}\alpha_2\\3\alpha_1 + 4\alpha_2\end{pmatrix} \neq \begin{pmatrix}0\\0\end{pmatrix}$ for

$\begin{cases}\alpha_2 = 0\\3\alpha_1 + 4\alpha_2 = 0\end{cases}$ has a solution only for $\alpha_1 = 0$ and $\alpha_2 = 0$ together.

Let's consider $\vec{x}_1 = \begin{pmatrix}1\\4\\7\end{pmatrix}$; $\vec{x}_2 = \begin{pmatrix}2\\5\\8\end{pmatrix}$; $\vec{x}_3 = \begin{pmatrix}3\\6\\9\end{pmatrix}$;

These three vectors are linearly dependent. Let's prove that.

$$\alpha_1 \cdot \vec{x}_1 + \alpha_2 \cdot \vec{x}_2 + \alpha_3 \cdot \vec{x}_3 = 0$$

$$\begin{cases}\alpha_1 + 2\alpha_2 + 3\alpha_3 = 0,\\ 4\alpha_1 + 5\alpha_2 + 6\alpha_3 = 0,\\ 7\alpha_1 + 8\alpha_2 + 9\alpha_3 = 0.\end{cases}$$

This is a linear equation $A\vec{x} = 0,\ A = \begin{pmatrix}1 & 2 & 3\\4 & 5 & 6\\7 & 8 & 9\end{pmatrix}$ and it has multiple solutions.

The easiest way to solve this is to get the matrix *A* to a reduced form using the process of Gauss-Jordan elimination. We can multiply any **row** by a non-zero number and add it to any other **row** (rows only!).

Step 1. Let's multiply the first row by (-4) and add it to row 2:

$$\begin{pmatrix}1&2&3\\4&5&6\\7&8&9\end{pmatrix}\begin{matrix}\text{(-4)}\\ \leftarrow +\\ \\ \end{matrix} \rightarrow \begin{pmatrix}1&2&3\\4-4&5-8&6-12\\7&8&9\end{pmatrix} \rightarrow \begin{pmatrix}1&2&3\\0&-3&-6\\7&8&9\end{pmatrix}.$$

Step 2. Multiply first row by (-7) and add it to row 3:

$$\begin{pmatrix}1&2&3\\0&-3&-6\\7&8&9\end{pmatrix}\begin{matrix}\text{(-7)}\\ \\ \leftarrow +\end{matrix} \rightarrow \begin{pmatrix}1&2&3\\0&-3&-6\\7-7&8-14&9-21\end{pmatrix} \rightarrow \begin{pmatrix}1&2&3\\0&-3&-6\\0&-6&-12\end{pmatrix}.$$

We can also divide or multiply a row by a non-zero number.

Step 3. Divide the second row by (-3) and third row by (-6): $\begin{pmatrix}1&2&3\\0&1&2\\0&1&2\end{pmatrix}$

Step 4. Row 2 times (-1) added to row 3:

$$\begin{pmatrix}1&2&3\\0&1&2\\0&1&2\end{pmatrix}\begin{matrix}\\ \text{(-1)}\\ \leftarrow +\end{matrix} \rightarrow \begin{pmatrix}1&2&3\\0&1&2\\0&0&0\end{pmatrix}.$$

Step 5. Row 2 times (-2) added t row 1:

$$\begin{pmatrix}1&2&3\\0&1&2\\0&0&0\end{pmatrix}\begin{matrix}\leftarrow +\\ \text{(-2)}\\ \\ \end{matrix} \rightarrow \begin{pmatrix}1&0&-1\\0&1&2\\0&0&0\end{pmatrix}$$

Now $\begin{pmatrix}1&0&-1\\0&1&2\\0&0&0\end{pmatrix}\begin{pmatrix}\alpha_1\\\alpha_2\\\alpha_3\end{pmatrix}=\begin{pmatrix}0\\0\\0\end{pmatrix}$.

Done.

In Matlab the same process is done by rret(A).

The general solution:

$$\begin{cases}\alpha_1-\alpha_3=0\\\alpha_2-2\alpha_3=0\end{cases}, \quad \begin{cases}\alpha_1=\alpha_3\\\alpha_2=-2\alpha_3\end{cases}.$$

For example, if α_1 = 1, α_3 = 1, and α_2= -2. Checking out:

$$\alpha_1\cdot\vec{x}_1+\alpha_2\cdot\vec{x}_2+\alpha_3\cdot\vec{x}_3=1\begin{pmatrix}1\\4\\7\end{pmatrix}-2\begin{pmatrix}2\\5\\8\end{pmatrix}+1\begin{pmatrix}3\\6\\9\end{pmatrix}=\begin{pmatrix}0\\0\\0\end{pmatrix};$$

$$1 \cdot 1 - 2 \cdot 2 + 1 \cdot 3 = 0$$
$$1 \cdot 4 - 2 \cdot 5 + 1 \cdot 6 = 0$$
$$1 \cdot 7 - 2 \cdot 8 + 1 \cdot 9 = 0$$

All true. So we have $(1, -2, 1) = (\alpha_1, \alpha_2, \alpha_3)$, for which $\alpha_1 \cdot \vec{x}_1 + \alpha_2 \cdot \vec{x}_2 + \alpha_3 \cdot \vec{x}_3 = 0$. You can check that any of the (2, -4, 2), (3, -6, 3), (-1, 2, -1), (-4, 8, -4) … are also complaint with it. Therefore, column vectors $\vec{x}_1 = \begin{pmatrix} 1 \\ 4 \\ 7 \end{pmatrix}$, $\vec{x}_2 = \begin{pmatrix} 2 \\ 5 \\ 8 \end{pmatrix}$ $and\ \vec{x}_3 = \begin{pmatrix} 3 \\ 6 \\ 9 \end{pmatrix}$ are linearly dependent.

Column vectors are constituents of matrices. Note that for linearly dependent column vectors in a square matrix the **determinant** of that matrix is always zero:

$$det \begin{pmatrix} 1 & -6 \\ 1 & -6 \end{pmatrix} = 1(-6) - 1 \cdot (-6) = 0,$$

$$det \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} = 1(5 \cdot 9 - 6 \cdot 8) - 2(4 \cdot 9 - 6 \cdot 7) + 3(4 \cdot 8 - 5 \cdot 7) = 0,$$

while for linearly independent column vectors the determinant is always non-zero:

$$det \begin{pmatrix} 0 & 1 \\ 3 & 4 \end{pmatrix} = -3 \neq 0,$$

$$det \begin{pmatrix} 1 & 7 & 3 \\ 0 & 2 & -4 \\ 0 & -2 & 1 \end{pmatrix} = 1\big(2 \cdot 1 - (-4) \cdot (-2)\big) - 7(0 \cdot 1 - (-4) \cdot 0) +$$

$$+3(0 \cdot (-2) - 2 \cdot 0) = -6 \neq 0.$$

NB! This means that for any square $(n \times n)$ matrix *P* the rank *P* = *n* if and only if $det\, P \neq 0$. Therefore if *P* is a controllability matrix from Theorem 3.1, corresponding to a linear state equation (3.19), this equation is controllable if and only if $det\, P \neq 0$.

Homework: Prove that *P* from theorem 3.1 is a square $(n \times n)$ matrix.

Example 3.3. A controller canonical form.

It can be shown that a transfer function

$$H(s) = \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0} \#(3.20)$$

has a state space realization of

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} u(t), \#(3.21)$$

$$y(t) = (b_0\ \ b_1\ \ b_2) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = b_0 x_1 + b_1 x_2 + b_2 x_3 \#(3.22)$$

The output depends on all three state variables. The controllability matrix is

$$P = (B \quad AB \quad A^2 B).$$

Calculate it for the following *A* and *B*:

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{pmatrix};\quad B = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

$$Then\ AB = \begin{pmatrix} 0 \\ 1 \\ -a_2 \end{pmatrix},\quad A^2 = \begin{pmatrix} 0 & 1 & 0 \\ -a_0 & 0 & 1 \\ a_0 a_2 & -a_0 + a_1 a_2 & -a_1 + {a_2}^2 \end{pmatrix},\quad A^2 B = \begin{pmatrix} 1 \\ -a_2 \\ -a_1 + {a_2}^2 \end{pmatrix}.$$

$$Thus\ P = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & -a_2 \\ 1 & -a_2 & -a_1 + {a_2}^2 \end{pmatrix};$$

$det\, P = -1 \neq 0$ **so the state equation (3.21) is always controllable and is valid for systems of higher order.**

A single input *u*(*t*), single output *y*(*t*), transfer function

$$H(s) = \frac{b_{n-1}s^{n-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0}$$

Corresponds to a **controller canonical form (CCF)**:

$$\begin{cases} \dot{\vec{x}}_{CCF} = A_{CCF} \cdot \vec{x} + B_{CCF} \cdot u(t) \\ y_{CCF}(t) = C_{CCF} \cdot \vec{x} \end{cases} \#(3.33)$$

which is always controllable with

$$A_{CCF} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-2} & -a_{n-1} \end{pmatrix}; \quad \#(3.34a)$$

$$B_{CCF} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 1 \end{pmatrix}; \quad C_{CCF} = (b_0 \quad b_1 \quad b_2 \quad \cdots \quad b_{n-2} \quad b_{n-1}).\#(3.34b)$$

To get (transform) a state space equation to a controller canonical form one uses a **coordinate transformation matrix:**

$$T_{CCF} = (B \quad AB \quad A^2B \quad \cdots \quad A^{n-1}B\,) \begin{pmatrix} a_1 & a_2 & a_3 & \cdots & a_{n-1} & 1 \\ a_2 & a_3 & a_4 & \cdots & 1 & 0 \\ a_3 & a_4 & a_5 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ a_{n-1} & 1 & 0 & \cdots & 0 & 0 \\ 1 & 0 & 0 & \cdots & 0 & 0 \end{pmatrix} = P \cdot {P_{CCF}}^{-1},$$

And the formulas from $\begin{cases} \dot{\vec{x}}(t) = A \cdot \vec{x}(t) + B \cdot \vec{u}(t) \\ y(t) = C\vec{x} \end{cases}$ to (3.33)

$$A_{CCF} = T_{CCF}{}^{-1} \cdot A \cdot T_{CCF},\ B_{CCF} = T_{CCF}{}^{-1} \cdot B,\ C_{CCF} = C \cdot T_{CCF},\ D_{CCF} = D, \#(3.35)$$

where $a_1, \ldots, a_{n-1}$ are coefficients of a characteristic polynomial

$$s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0 = 0, \qquad det(SI - A) = 0.$$

Even in Matlab there is no built-in function for this, so knowing to use formulas (3.33-3.35) is important.

Example 3.4. A three-dimensional state equation in a controller canonical form.

$$\begin{cases} \dot{\vec{x}}(t) = A\vec{x}(t) + B\vec{u}(t), \\ \vec{y}(t) = C\vec{x}(t) \end{cases} \qquad A = \begin{pmatrix} -2 & 0 & 8 \\ 4 & 1 & -3 \\ 7 & 12 & 5 \end{pmatrix}; \quad B = \begin{pmatrix} -1 \\ 2 \\ -3 \end{pmatrix}; \quad C = (4,\ 3,\ -3);$$

Let's transfer this state equation in a controller canonical form.

$$AB = \begin{pmatrix} -22 \\ 7 \\ 2 \end{pmatrix}; \quad A^2 = \begin{pmatrix} 60 & 96 & 24 \\ -25 & -35 & 14 \\ 69 & 72 & 45 \end{pmatrix}; \quad A^2B = \begin{pmatrix} 60 \\ -87 \\ -60 \end{pmatrix};$$

Since A is a (3×3) matrix, the controllability matrix is

$$P = (B\ AB\ A^2B) = \begin{pmatrix} -1 & -22 & 60 \\ 2 & 7 & -87 \\ -3 & 2 & -60 \end{pmatrix}; \quad \det P = -6636 \neq 0,$$

so *the state equation is controllable and it is possible to transfer it to a controller coordinate form.* A coordinate transformation $x(t) \to z(t)$ can be made by a non-singular *transformation matrix* T. It is called a *similarity transform:*

$$\vec{x}(t) = T\vec{z}(t), \quad \vec{z}(t) = T^{-1}\vec{x}(t).$$

Note, that it *does not change the characteristic polynomial*[1] *and the eigenvalues* of the new transformed matrix $\hat{A}$:

$$\hat{A} = T^{-1}AT, \quad |sI - \hat{A}| = |sI - A| \implies \hat{\lambda}_1 = \lambda_1,\ \hat{\lambda}_2 = \lambda_2,\ \dots\ \hat{\lambda}_n = \lambda_n. \tag{3.36}$$

Eigenvectors do change, but for our purposes it is not important. The characteristic polynomial of the matrix A is

$$\det(sI - A) = 0, \quad \det \begin{pmatrix} s+2 & 0 & 8 \\ 4 & s-1 & -3 \\ 7 & 12 & s-5 \end{pmatrix} = 0,$$

$$(s+2)((s-1)(s-5) + 3 \cdot 12) - 8(4 \cdot 12 + 7(s-1)) = 0,\ s^3 - 4s^2 - 27s - 246 = 0,$$

so $a_2 = -4$, $a_1 = -21$, $a_0 = -246$. Now, let's construct the controller canonical form transformation matrix T_{CCF} using (3.35):

$$T_{CCF} = PP_{CCF}^{-1} = P \begin{pmatrix} a_1 & a_2 & 1 \\ a_2 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} -1 & -22 & 60 \\ 2 & 7 & -87 \\ -3 & 2 & -60 \end{pmatrix} \begin{pmatrix} -27 & -4 & 1 \\ -4 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 175 & -18 & -1 \\ -169 & -1 & 2 \\ 13 & 14 & 3 \end{pmatrix};$$

Now we can construct

$$T_{CCF}^{-1} = \begin{pmatrix} 175 & -18 & -1 \\ -169 & -1 & 2 \\ 13 & 14 & 3 \end{pmatrix}; \ A_{CCF} = T_{CCF}^{-1}AT_{CCF} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 246 & 27 & 4 \end{pmatrix}; \ B_{CCF} = T_{CCF}^{-1}B = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix};$$

[1] $|sI - A| = \det(sI - A)$

$$C_{CCF} = CT_{CCF} = (154,\ -117,\ 11);\ D_{CCF} = D = 0;$$

Note that A_{CCF} and B_{CCF} can be calculated using (3.35) or written directly with (3.34) by a substitution of the characteristic polynomial values $a_0,\ a_1,\ a_2$. For C_{CCF}, however, the transformation matrix T_{CCF} is always required. See, how this is done in Matlab.

```
clc; clear; close all;

% Example 3.4. A three-dimensional state equation in a controller
% canonical form. Input the matrices.
A = [-2 0 8; 4 1 -3; 7 12 5  ]; B = [-1; 2; -3 ];
C = [4 3 -3 ]; D = 0;
% Controllability matrix:
P = [B A*B A^2*B];
% If determinant of P is not zero, the syste, is controllable.
det(P)
% % Coefficients of the characteristic polynomial of the matrix A:
% % s^3 -4s^2 -27s -246 = 0
char_poly = charpoly(A) % outputs [1    -4    -27   -246]
a1 = char_poly(3); a2 = char_poly(2);
% Controller canonical form transformation matrix:
T_CCF = P * [a1 a2 1; a2 1 0; 1 0 0 ];
% Transformed A, B, C, D matrices:
% If some of the values are less than 10^(-6), consider them zero.
A_CCF = T_CCF^(-1)*A*T_CCF; B_CCF = T_CCF^(-1)*B;
C_CCF = C*T_CCF; D_CCF = D;
```

Lecture 7: closed-loop eigenvalue placement using state feedback

Let's start with a closed-loop state equation with feedback control like we have seen on Figure 3.1:

$$\dot{\vec{x}}(t) = A\vec{x}(t) + B\vec{u}(t),$$

$$\vec{y}(t) = C\vec{x}(t),$$

where $\vec{u}(t) = -K\vec{x}(t) + Br(t)$, so

$$\dot{\vec{x}} = (A - BK)\vec{x} + Br(t). \tag{3.37}$$

Before, in lecture 5 we have established how one can place eigenvalues of proper choice based on certain requirements of the transient response characteristics like overshoot or settling time. We wrote explicit formulas for a second order approximation. This process amounts to *a set of distinct eigenvalues*: $\lambda_1, \lambda_2, \ldots, \lambda_{n-1}, \lambda_n$. The following theorem is the condition for bringing a closed-loop state equation, a pair (A, B), to these desired eigenvalues via a proper choice of the state feedback matrix K.

Theorem 3.2. For any symmetric[1] set of eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_{n-1}, \lambda_n$ there exists such a matrix K so that matrix $(A - BK)$ has the same eigenvalues if and only if the pair (A, B) is controllable.

Next, we show how to pick the right values for the feedback gain matrix K.

Case I. Pair (A, B) is already in the controller canonical form $(A_{CCF}, B_{CCF}$.

We shall discuss the three dimensional case for convenience.

$$A_{CCF} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{pmatrix}, \quad B_{CCF} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

It's characteristic polynomial is then

$$|sI - A_{CCF}| = s^3 + a_2 s^2 + a_1 s + a_0. \tag{3.38}$$

We denote the feedback gain vector in this case as K_{CCF} and

$$A_{CCF} - B_{CCF}K_{CCF} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 - \gamma_0 & -a_1 - \gamma_1 & -a_2 - \gamma_2 \end{pmatrix}.$$

It's characteristic polynomial is

$$|sI - A_{CCF} + B_{CCF}K_{CCF}| = s^3 + (a_2 + \gamma_2)s^2 + (a_1 + \gamma_1)s + a_0 + \gamma_0. \tag{3.39}$$

[1] to the real axis

So, (3.39) corresponds to a system pair (A_{CCF}, B_{CCF}) with an applied feedback controller in the form $\vec{u}(t) = -K\vec{x}(t) + r(t)$. In Lecture 5 we have seen how we can choose the desired eigenvalues $\lambda_1, \lambda_2, \lambda_3$. These determine the characteristic polynomial

$$(s-\lambda_1)(s-\lambda_2)(s-\lambda_3) = s^3 + \chi_2 s^2 + \chi_1 s + \chi_0, \tag{3.40}$$

where χ_0, χ_1, χ_2 are its coefficients. We want the coefficients of the polynomial of the controlled system to be equal to χ_0, χ_1, χ_2 so we get the desired transient response characteristics. Therefore

$$\begin{cases} a_0 + \gamma_0 = \chi_0, \\ a_1 + \gamma_1 = \chi_1, \\ a_2 + \gamma_2 = \chi_2 \end{cases} \implies \begin{cases} \gamma_0 = \chi_0 - a_0, \\ \gamma_1 = \chi_1 - a_1, \\ \gamma_2 = \chi_2 - a_2 \end{cases}$$

$$K_{CCF} = (\chi_0 - a_0,\ \chi_1 - a_1,\ \chi_2 - a_2), \tag{3.41}$$

where χ_0, χ_1, χ_2 are the coefficients of the characteristic polynomial of the *desired* controlled system and a_0, a_1, a_2 are the coefficients of the characteristic polynomial of the given *uncontrolled open-loop* system in a *controller canonical form.*

Example 3.5. Linear feedback control of a system in a controller canonical form.
Let

$$A_{CCF} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -12 & -9 & -4 \end{pmatrix}, \quad B_{CCF} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad C_{CCF} = (1\ 0\ 0).$$

Open-loop characteristic polynomial is by inspection $|sI - A_{CCF}| = s^3 + 4s^2 + 9s + 12|$, while the eigenvalues are $\lambda_1 = -2.34, \lambda_2 = -0.83 + 2.1i, \lambda_3 = -0.83 - 2.1i$. Let's acquire an overshoot of $PO = 5\%$ and a settling time of $t_s = 4$ seconds. Using

$$\xi' = \frac{|\ln(PO/100)|}{\sqrt{\pi^2 + \ln^2(PO/100)}} = 0.69,$$

which is within range of $0.3 < \xi' < 0.8$ so the approximation with (3.13) and (3.16) should be close. According to (3.16)

$$t_s \approx \frac{4}{\xi'\omega_n'} \implies \omega_n' \approx \frac{4}{\xi' t_s} = 1.45.$$

Using (3.12)

$$\lambda_{1,2}' = -\xi'\omega_n' \pm \omega_n'\sqrt{\xi^2 - 1} = -1 \pm 1.05i.$$

These are the two desired eigenvalues. However, the controlled system is three-dimensional (has three state variables) so we must *choose* the third eigenvalue. *A rule of thumb* for a second order-approximation is to choose the remaining eigenvalues to be real, negative and 10 times further away to the left from the origin than the desired eigenvalues. That way the third one does not affect the approximation too much. We choose $\lambda_3 = -10$. The three eigenvalues determine a characteristic polynomial

$$|sI - A'| = (s - \lambda_1')(s - \lambda_2')(s - \lambda_3') = s^3 + 12s^2 + 22s + 21.$$

Comparing it to the characteristic polynomial of the open-loop uncontrolled system $s^3+4s^2+9s+12$ we get

$$K_{CCF} = (\chi_0 - a_0,\ \chi_1 - a_1,\ \chi_2 - a_2) = (\gamma_0,\ \gamma_1,\ \gamma_2) = (21-12,\ 22-9,\ 12-4) = (9,\ 13,\ 8).$$

With such a controlled you can show that

$$A_{CCF} - B_{CCF}K_{CCF} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -21 & -22 & -12 \end{pmatrix}, \quad B_{CCF} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad C_{CCF} = (1\ 0\ 0).$$

Let there be zero initial conditions for each of the three state variables. You can observe the controlled and the open-loop step responses as well as the transient response of each of the three state variables in Figure 3.5. With $PO' = 4.94\%$ it is a big improvement over the original $PO = 17.3\%$,

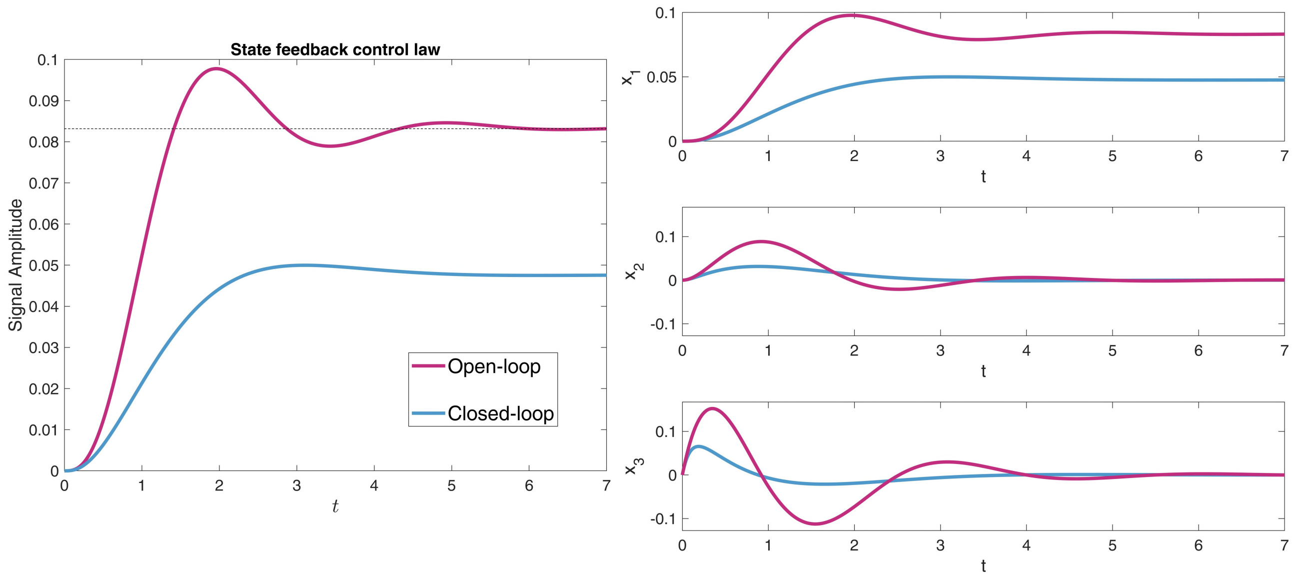

Fig. 3.5: The controlled and the open-loop step responses. Notice, how none of the two step responses reaches a steady-state value of 1. This is due to a lack of the steady-state error tracking, which we shall discuss later.

so we are in agreement with the design specifications. Notice, however, how none of the two step responses reaches a steady-state value of 1. This means that for a closed-loop system the steady-state output does not match the reference input signal $r(t)$. This is why later we shall modify the control law to the form

$$u(t) = -K\vec{x}(t) + Gr(t). \qquad \square \tag{3.42}$$

The code for this exercise is as follows.

```
clc; close all; clear
% Example 3.5 Linear feedback controll of a system in a controller
% canonical form. The open-loop uncontrolled system is already
% in a controller canonical form.
A_ccf = [ 0 1 0; 0 0 1; -12 -9 -4 ]; B_ccf = [ 0; 0; 1 ];
C_ccf = [1 0 0]; D_ccf = 0;
% A state-space representation of the open-loop uncontrolled system
system = ss(A_ccf,B_ccf,C_ccf,D_ccf);
% See the step response characteristics:
```

```
stepinfo(system)
% Constructing a vector [1 a_2 a_1 a_0 ] with the coefficients of
% the characteristic polynomial s^3 + a_2*s^2 + a_1*s + a_0:
A_poly = poly(A_ccf); % outputs [1  4   9   12]
a0 = A_poly(4); a1 = A_poly(3); a2 = A_poly(2);

% Set up initial conditions. Can be zero or any other.
X0 = zeros(3,1);
% Set up time.
t = 0:0.01:7;
% Set up the signal values. If all equal to 1, it is a
% unit step response.
U = ones(size(t) );

% The control starts. The desired values of overshoot
% and settling time.
PO = 0.05; ts = 4;
% Formula for the desired damping ratio and natural
% oscillation frequency.
xi = abs(  log(PO)/sqrt(pi^2 + (log(PO))^2 )  ); % = 0.69
omega_n = 4 / xi / ts;
% Eigenvalues associated with the desired damping
% ratio and natural oscillation frequency.
lambda1 = -xi*omega_n + omega_n*sqrt(xi^2 - 1 );
lambda2 = -xi*omega_n - omega_n*sqrt(xi^2 - 1 );
% The third chosen eigenvalue 10 times further to
% the negative part.
lambda3 = -10*xi*omega_n;
% Create a characteristic polynomial associated with
% these eigenvalues.
des_poly = poly( [lambda1 lambda2 lambda3 ] );
chi0 = des_poly(4); chi1 = des_poly(3); chi2 = des_poly(2);
% You can check that this polynomial has the eigenvalues
% as its roots.
Roots_poly_des = roots([ 1.0000    8.04    9.92    6.29 ]);
% The state-feedback vector in CCF:
K_CCF = [ chi0 - a0, chi1 - a1, chi2 - a2 ];

% Create a state-space representation of the
% closed-loop CONTROLLED system.
controlled_system = ss( A_ccf - B_ccf*K_CCF, B_ccf, C_ccf, D_ccf );
% See the step response characteristics:
stepinfo(controlled_system)


% % Collect unit step response data of the uncontrolled system:
[Y_opl, t_opl, X_opl] = lsim( system, U, t, X0 );
```

```
% % Collect unit step response data of the controlled system:
[Y_cont, t, X_cont] = lsim( controlled_system, U, t, X0 );

% Plot the step response of the output.
plot(t, Y_cont, 'Color', [ 0.3 0.6 0.8  ], 'LineWidth', 3)
hold on
plot(t_opl, Y_opl, 'Color', [ 0.76 0.18 0.49  ], 'LineWidth', 3)

% Plot the responses of the individual state variables.
figure
subplot(311), plot( t, X_cont(:,1) ); hold;
plot( t_opl, X_opl(:,1) );
subplot(312), plot( t,X_cont(:,2) ); hold;
plot( t_opl, X_opl(:,2) );
subplot(313), plot( t,X_cont(:,3) ); hold;
plot( t_opl, X_opl(:,3) );
```

Homework. A state feedback vector K may have negative values. Please show that for the same system as in example 3.5 for $PO = 4\%$ and $t_s = 6$ seconds the vector $K_{CCF} = (-5.71,\ 0.92,\ 4.04)$.

Case II. Pair (A, B) is not in the controller canonical form.

1) *The Bass-Gura formula.*

Before we did gain matrix K_{CCF} matching when the system was in a controller canonical form. In order to acquire these values for a general case one must consider the controller canonical form state variables transformation, which is [1]:

$$\vec{x}_{CCF} = T_{CCF}^{-1}\vec{x}, \qquad \vec{x} = T_{CCF}\vec{x}_{CCF}, \tag{3.43}$$

where

$$T_{CCF} = PP_{CCF}^{-1}, \quad P = (B\ AB\ A^2B\ \dots\ A^{n-1}B),$$
$$P_{CCF} = (B_{CCF}\ A_{CCF}B_{CCF}\ A_{CCF}^2B_{CCF}\ \dots\ A_{CCF}^{n-1}B_{CCF}). \tag{3.44}$$

Let pair (A, B) be the given state equation matrix pair with $n = 3$ state variables. We would like to transform it to a controller canonical form (A_{CCF}, B_{CCF}). Remember, that the *characteristic polynomial does not change with a coordinate transformation*, which includes the transformation to the controller canonical form.

$$|sI - A| = |sI - A_{CCF}| = s^3 + a_2s^2 + a_1s + a_0.$$

Let's say we have a desired (A', B') second-order model. It's characteristic polynomial is

$$|sI - A'| = s^3 + \chi_2s^2 + \chi_1s + \chi_0.$$

As we have seen in (3.41)

$$K_{CCF} = (\chi_0 - a_0,\ \chi_1 - a_1,\ \chi_2 - a_2).$$

Any coordinate transformation of a state-space equation constitutes

$$A_{new} = T^{-1}A_{old}T, \quad B_{new} = T^{-1}B_{old}.$$

A controller canonical form coordinate transformation matrix $T = T_{CCF} = PP_{CCF}^{-1}$ was built in (3.35). With a state feedback controller $u(t) = -K\vec{x}(t) + Br(t)$, we see that

$$\dot{\vec{x}}(t) = (A - BK)\vec{x} + Br(t)$$

and $A_{old} = A - BK$. The same system in a controller canonical form with state feedback $u_{CCF}(t) = -K_{CCF}\vec{x}_{CCF}(t) + B_{CCF}r(t)$ is

$$\dot{\vec{x}}_{CCF}(t) = (A_{CCF} - B_{CCF}K_{CCF})\vec{x}_{CCF}(t) + B_{CCF}r(t),$$

so $A_{new} = A_{CCF} - B_{CCF}K_{CCF}$ and, therefore

$$A_{new} = T_{CCF}^{-1}A_{old}T_{CCF} = T_{CCF}^{-1}(A - BK)T_{CCF} = A_{CCF} - B_{CCF}K_{CCF}. \tag{3.45}$$

At the same time

$$A_{CCF} - B_{CCF}K_{CCF} = T_{CCF}^{-1}AT_{CCF} - T_{CCF}^{-1}BK_{CCF}. \tag{3.46}$$

Equalizing (3.45) and (3.46) we see that

$$T_{CCF}^{-1}(A - BK)T_{CCF} = T_{CCF}^{-1}AT_{CCF} - T_{CCF}^{-1}BK_{CCF}$$

$$T_{CCF}^{-1}AT_{CCF} - T_{CCF}^{-1}B\underline{KT_{CCF}} = T_{CCF}^{-1}AT_{CCF} - T_{CCF}^{-1}B\underline{K_{CCF}}$$

from which $KT_{CCF} = K_{CCF}$ and, since $T = T_{CCF} = PP_{CCF}^{-1}$,

$$K = K_{CCF}T_{CCF}^{-1}. \tag{3.47}$$

This is the Bass-Gura formula. We get K_{CCF} the same way as in example 3.5 and then apply this formula to get the values of K for the system in a general form (A, B). Controllability is, of course, required for the pair (A, B).

Example 3.6 An implementation of the Bass-Gura Formula. Consider a single input system

$$A = \begin{pmatrix} 0 & 2 & 1 \\ 4 & 8 & 0 \\ -2 & 0 & 9 \end{pmatrix}; \qquad B = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix};$$

The controllability matrix can be calculated

$$P = \begin{pmatrix} 1 & 1 & 15 \\ 0 & 4 & 36 \\ 1 & 7 & 61 \end{pmatrix}, \quad \det P \neq 0, \text{ therefore, the system is controllable.}$$

The system's characteristic polynomial is

$$|sI - A| = s^3 + a_2s^2 + a_1s + a_0 = s^3 - 17s^2 + 66s + 56.$$

Let's say we would like the step response to have 5% overshoot PO and 6 seconds settling time t_s. It's easy to show that a second order approximation system with such stats has the eigenvalues $\lambda_{1,2} = -0.67 \pm 0.7i$. A third eigenvalue chosen as $\lambda_3 = -6.7$ amounts to

$$|sI - A'| = s^3 + \chi_2s^2 + \chi_1s + \chi_0 = s^3 - 8s^2 + 10s + 6.3.$$

Then

$$K_{CCF} = (6.3 - 56,\ 10 - 66,\ 8 + 17) = (-49.7,\ -56,\ 25).$$

A controllability matrix P_{CCF} corresponds to a pair (A_{CCF}, B_{CCF}). Since a state coordinate transformation $(A, B) \to (A_{CCF}, B_{CCF})$ does not change its characteristic polynomial and according to (3.35)

$$T_{CCF} = PP_{CCF}^{-1} = P\begin{pmatrix} a_1 & a_2 & 1 \\ a_2 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 15 \\ 0 & 4 & 36 \\ 1 & 7 & 61 \end{pmatrix}\begin{pmatrix} 66 & -17 & 1 \\ -17 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 64 & -16 & 1 \\ -32 & 4 & 0 \\ 8 & -10 & 1 \end{pmatrix}.$$

So, using the Bass-Gura formula:

$$K = K_{CCF}T_{CCF}^{-1} = (-49.7,\ -56,\ 25)\begin{pmatrix} 0.125 & 0.19 & -0.125 \\ 1 & 1.75 & -1 \\ 9 & -16 & -8 \end{pmatrix} = (163,\ 292.5,\ -138).$$

With that

$$A - BK = \begin{pmatrix} -163 & -291 & 139 \\ 4 & 8 & 0 \\ -165 & -293 & 147 \end{pmatrix}$$

has $\lambda_1 = -6.7$ and $\lambda_{1,2} = -0.67 \pm 0.7i$. as expected. Figure 3.6 shows the step responses of the open-loop and the closed-loop controlled systems with zero initial conditions. As can be seen the open-loop system is actually unstable without a controller. Each of the individual state variables $x_1,\ x_2,\ x_3$ tend to infinity with time. The controlled system on the other hand shows regular transient responses. While the Bass-Gura is rarely used outside of the learning process we believe it is still important to see how it can be implemented. In certain situations it may save computation time. Let's look at the way one can do this in Matlab.

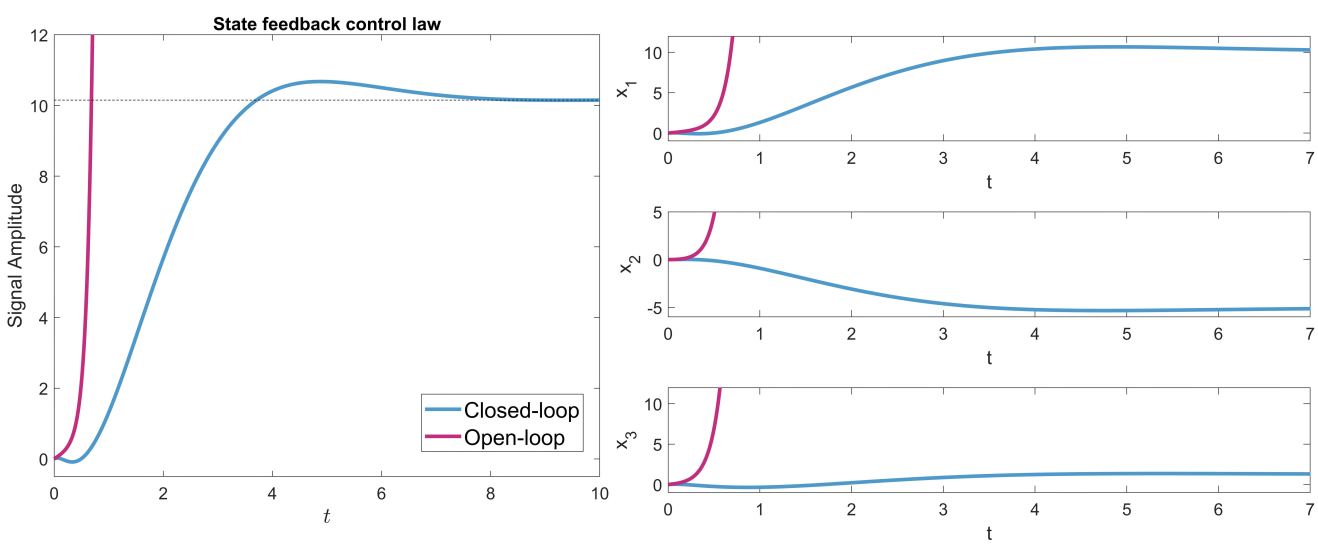

Fig. 3.6: The controlled and the open-loop step responses. The uncontrolled system is unstable. We are going to make the controlled step response reach a steady-state value of 1 in a latter example.

```
% Example 3.6. The Bass-Gura formula.
% Input the matrices:
A = [0 2 1; 4 8 0; -2 0 9]; B = [1; 0; 1]; C = [1 0 0]; D = 0;
% The controllability matrix:
P = [B A*B A^2*B];
% The system is controllable if the determinant is not zero.
determinant = det(P);

% Constructing a vector [1 a_2 a_1 a_0 ] with the coefficients of
% the characteristic polynomial s^3 + a_2*s^2 + a_1*s + a_0:
A_poly = poly(A); % outputs [1  -17   66   56]
a0 = A_poly(4); a1 = A_poly(3); a2 = A_poly(2);
% Input the desired eigenvalues manually or use the transient
% response characteristic formulas as in example 3.5.
lambda_1 = -6.7;
lambda_2 = -0.67 - 0.7*1i; % 1i is the imaginary unit in Matlab
lambda_3 = -0.67 + 0.7*1i;
% Defining the desired polynomial that possesses
% the chosen eigenvalues:
des_poly = poly( [lambda_1 lambda_2 lambda_3 ] );
chi0 = des_poly(4); chi1 = des_poly(3); chi2 = des_poly(2);
% The state-feedback vector in CCF:
K_CCF = [ chi0 - a0, chi1 - a1, chi2 - a2 ];

% Construct the inverse controllability matrix in the controller
% canonical form:
P_CCF_inv = [a1 a2 1; a2 1 0; 1 0 0];
% Transformation matrix:
T_CCF = P * P_CCF_inv;
% Use the Bass-Gura formula:
K = K_CCF * inv(T_CCF);
% See the new controlled A' matrix:
A_new = A - B*K;
% and check that it has the desired eigenvalues:
eig(A - B*K)

% A state-space representation of the open-loop uncontrolled system
system = ss( A, B, C, D );
% Set up initial conditions. Can be zero or any other.
X0 = zeros(3,1);
% Set up time. Since the open-loop system is unstable we
% make it shorter.
t_opl = 0:0.01:1;
% Set up the signal values. If all equal to 1, it is a
% unit step response.
U = ones(size(t_opl) );
```

```
% % Collect unit step response data of the uncontrolled system:
[Y_opl, t_opl, X_opl] = lsim( system, U, t_opl, X0 );

% A state-space representation of the closed-loop CONTROLLED system
controlled_system = ss( A_new, B, C, D );
% Set up initial conditions. Can be zero or any other.
X0 = zeros(3,1);
% Set up time.
t = 0:0.01:10;
% Set up the signal values. If all equal to 1, it is a unit
% step response.
U = ones(size(t) );
% % Collect unit step response data of the controlled system:
[Y_cont, t, X_cont] = lsim( controlled_system, U, t, X0 );

% Plot the step response of the output.
plot(t, Y_cont, 'Color', [ 0.3 0.6 0.8  ], 'LineWidth', 3)
hold on
plot(t_opl, Y_opl, 'Color', [ 0.76 0.18 0.49  ], 'LineWidth', 3)
% Set up the proper limits on the Y axis:
ylim( [ -0.5 12 ] )

% Plot the responses of the individual state variables.
figure
subplot(311), plot( t, X_cont(:,1) ); hold;
plot( t_opl, X_opl(:,1) ); ylim( [-1 12] ); xlim( [0 7] );
subplot(312), plot( t,X_cont(:,2) ); hold;
plot( t_opl, X_opl(:,2) ); ylim( [-6 5] ); xlim( [0 7] );
subplot(313), plot( t,X_cont(:,3) ); hold;
plot( t_opl, X_opl(:,3) ); ylim( [-1 12] ); xlim( [0 7] );
```

2) The Ackermann's formula is the shortest one to write and to program if one does not worry much about computation time. Consider a controllable system (A, B) in an arbitrary form and for a desired characteristic equation $|sI - A'| = s^3 + \chi_2 s^2 + \chi_1 s + \chi_0$. We use the case with three state variables for convenience. The Ackermann's formula is [1]:

$$K = (0\ 0\ 1)P^{-1}\alpha(A), \qquad \alpha(A) = A^3 + \chi_2 A^2 + \chi_1 A + \chi_0 I_{3\times 3}, \tag{3.48}$$

where P is the controllability matrix of the pair (A, B). This removes the necessity to pre-calculate K_{CCF}.

Example 3.7. An implementation of the Ackermann's formula.

Let's use the same system (A, B) and the same desired characteristic polynomial as in example 3.5.

$$A = \begin{pmatrix} 0 & 2 & 1 \\ 4 & 8 & 0 \\ -2 & 0 & 9 \end{pmatrix}; \quad B = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}; \quad |sI - A'| = s^3 + \chi_2 s^2 + \chi_1 s + \chi_0 = s^3 - 8s^2 + 10s + 6.3.$$

The controllability matrix and it's inverse are

$$P = \begin{pmatrix} 1 & 1 & 15 \\ 0 & 4 & 36 \\ 1 & 7 & 61 \end{pmatrix}, \quad \det P \neq 0, \quad P^{-1} = \begin{pmatrix} 0.25 & -1.38 & 0.75 \\ -1.13 & -1.44 & 1.13 \\ 0.13 & 0.19 & -0.13 \end{pmatrix},$$

$$\alpha(A) = A^3 + \chi_2 A^2 + \chi_1 A + \chi_0 I_{3\times 3} = 10^3 \begin{pmatrix} 0.1 & 0.29 & 0.17 \\ 0.58 & 1.3 & 0.1 \\ -0.34 & -0.1 & 1.42 \end{pmatrix}.$$

So,

$$K = (0\ 0\ 1)P^{-1}\alpha(A) = (163,\ 293,\ -138)$$

same as with the Bass-Gura formula in example 3.5. One can use Matlab for this.

```
% Example 3.7. Ackermann's formula
% Input the matrices:
A = [0 2 1; 4 8 0; -2 0 9]; B = [1; 0; 1];
% The controllability matrix:
P = [B A*B A^2*B];
% The system is controllable if the determinant is not zero.
determinant = det(P);

% Input the desired eigenvalues manually or use the transient
% response characteristic formulas as in example 3.5.
lambda_1 = -6.7;
lambda_2 = -0.67 - 0.7*1i; % 1i is the imaginary unit in Matlab
lambda_3 = -0.67 + 0.7*1i;
% Defining the desired polynomial that possesses
% the chosen eigenvalues:
des_poly = poly( [lambda_1 lambda_2 lambda_3 ] );
chi0 = des_poly(4); chi1 = des_poly(3); chi2 = des_poly(2);

% Pre-calculation:
alpha_A = A^3 + chi2*A^2 + chi1*A + chi0*eye(3);
% The Ackermann's formula.
K_acker = [0 0 1]*inv(P)*alpha_A

% Or one may use the built in functions
% For single-input single-output systems only - the
% Ackermann's formula:
des_eig = [ lambda_1 lambda_2 lambda_3 ];
K_acker = acker(A,B, des_eig)
% The function place can also handle the multiple input case.
K_place = place(A,B, des_eig)
```

Both `acker()` and `place()` functions have the same output. The Bass-Gura and the Ackermann's formulas are true only for a single-input, single-output case. For multiple-input, multiple output systems there exist similar formulas, however they go beyond the scope of this book. These methods are implemented in the `place()` function in Matlab.

Steady-state error tracking

Consider the feedback control law as [3]

$$u(t) = -K\vec{x}(t) + Gr(t). \qquad \square \tag{3.49}$$

in which the gain matrix is chosen to match a step reference input signal $r(t) = R,\ t \geq 0$. Then, the steady-state output satisfies $y_{ss} = R$ with

$$\dot{\vec{x}} = (A - BK)\vec{x} + BGr(t),$$

$$\vec{y}(t) = C\vec{x}(t),$$

In order for that, the gain matrix is

$$G = -(C(A - BK)^{-1}B)^{-1}. \tag{3.50}$$

If $R = 1,\ t \geq 0$ we have a unit step response with $y_{ss} = 1$. There may be a situation when it is required to match the step response steady state of the open-loop system. For this exists an explicit formula. If

$$u(t) = \begin{cases} 1 & t \geq 0, \\ 0 & t < 0 \end{cases}$$

is a unit step function, then

$$y_{ss} = -CA^{-1}B. \tag{3.51}$$

If $u(t) = M, t \geq 0$ and M is a real number $M \in R^1$

$$y_{ss} = -CA^{-1}BM. \tag{3.52}$$

Example 3.8. Steady state tracking. Using the same system and desired eigenvalues as in example 3.5 we bring the closed-loop controlled system to the same steady state value as the open-loop system's. There exist explicit formulas for that based on state equation matrices and the type of input signal. For a unit step response as an input signal an open-loop system yields

$$y_{ss} = -C_{CCF}A^{-1}_{CCF}B_{CCF}. \tag{3.53}$$

This formula is true if all of the elements of the input vector are the same, for example, like in the single-input single-output case. Figure 3.7 shows the change it makes compared to Figure 3.5.

The code is exactly the same as in example 3.5, however we add a few lines before the plotting part.

```
% Example 3.8. The steady state tracking.

% ... skipping the first half of the code in example 3.5 here

% A state-space representation of the closed-loop CONTROLLED system
```

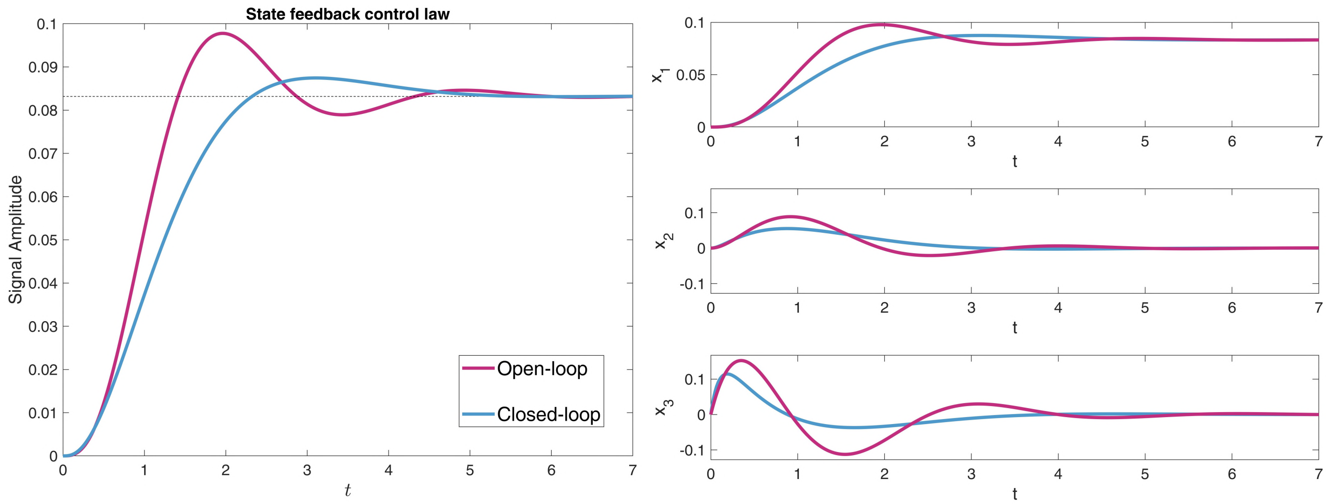

Fig. 3.7: The controlled and the open-loop step responses. Both, the responses reach the steady-state value of the open-loop system.

```
controlled_system = ss( A_new, B, C, D );
% Set up initial conditions. Can be zero or any other.
X0 = zeros(3,1);
% Set up time.
t = 0:0.01:10;
% Set up the signal values. If all equal to 1, it is a unit
% step response.
U = ones(size(t) );

% Create the gain matrix G for the controlled closed-loop
% system which enables the steady-state tracking:
G = - inv( C_ccf * inv( A_ccf - B_ccf*K_auto) * B_ccf  ); % 6.29
% Applying the formula for the steady state value
% of an open-loop system with a unit step response
% as the input signal:
y_ss_opl = - C_ccf *inv(A_ccf)*B_ccf; M = y_ss_opl;
% Multiply the input step signal
U = G*M*U;

% % Collect unit step response data of the controlled system:
[Y_cont, t, X_cont] = lsim( controlled_system, U, t, X0 );

% Plot the step response of the output.
% ...
```

Example 3.9. Feedback control of a DC motor with steady state tracking. We use the same system as in examples 1.5 and 2.3 as well as the desired eigenvalues from example 3.5. Figure 3.8 shows the step responses of the open-loop and the closed-loop controlled systems with zero initial conditions. As can be seen the open-loop system is actually unstable without a controller. Each

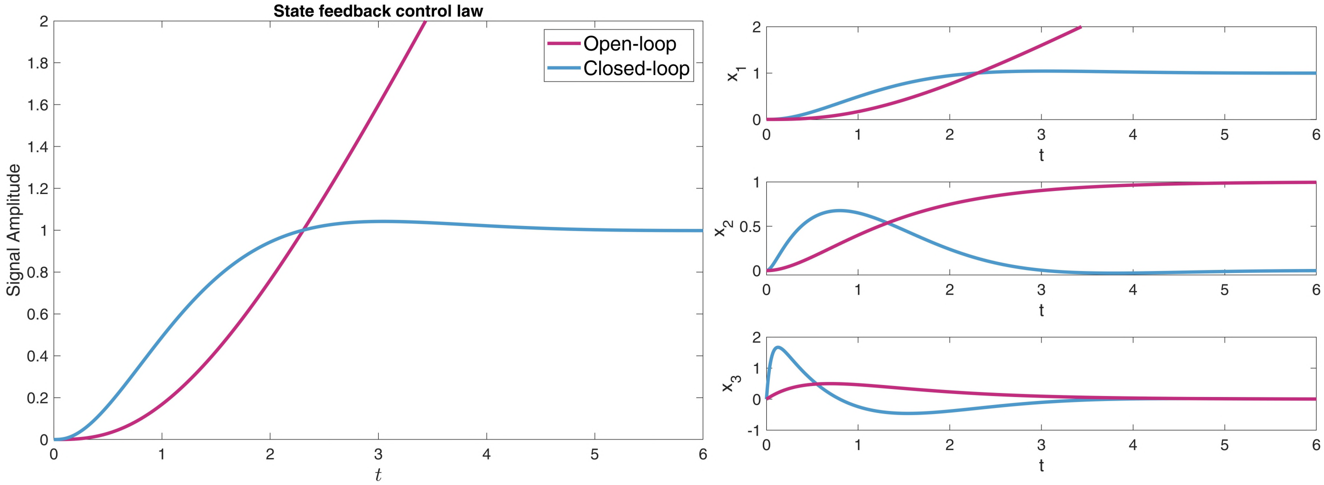

Fig. 3.8: The uncontrolled system is unstable. We have made the controlled step response reach a steady-state value of 1 as in (3.51).

of the individual state variables $x_1,\ x_2,\ x_3$ tend to infinity with time. The controlled system on the other hand shows regular transient responses. The code is almost the same as in example 3.7, however keep in mind that `M=1`.

References

[1] Williams II, R. L.; Lawrence, D. A. *Linear State-Space Control Systems*, 2nd ed.; John Wiley & Sons: 2007.

[2] Nise, N. S. *Control systems engineering*; John Wiley and Sons: 2019.

[3] Rowell D., Analysis and Design of Feedback Control Systems. Available online: http://web.mit.edu/2.14/www/Handouts/Handouts.html