Chapter 4

Controllability and Open-Loop control

4.1 Introduction

In this chapter, we will focus on computing a control signal $u(t)$ so that $\mathbf{x}(t)$ or $y(t)$[1] will follow a pre-defined pattern.
Note first that by saying that, we are not talking at all about feedback. Indeed, as useful as it is, feedback is not the only way to control a system.
Open-loop or feedforward control is another very important concept, which we will see in the present chapter.

An open-loop or feedforward controller uses the knowledge on the system model and a desired behavior/trajectory $\mathbf{x}_d(t)/y_d(t)$ to compute the desired actions $u_d(t)$.

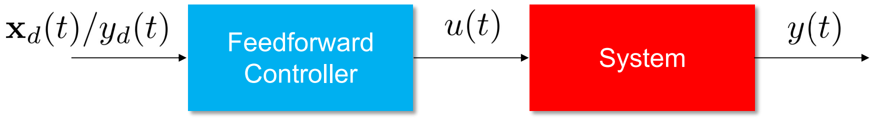

Figure 4.1: Feedforward controller

The main advantage of this technique is that no sensor is needed in the case of pure open-loop control. And no sensor can mean a cheaper product!

However, in some cases, the considered system is such that no matter what action is taken, the desired behavior *cannot* be realized. This concept is related to the notion of *controllability*, which helps to determine whether or not such cases will appear.

[1]In this chapter, we will mostly consider single-input single-output systems, hence the “non-bold” notation.

4.2 Controllability and the Kalman criterion

4.2.1 Controllability notions

The main idea for controllability can be explained as follows: starting from initial state $\mathbf{x}_0 = 0$, is it possible, using the control input $u(t)$, to bring/control the considered system to any desired state $\mathbf{x}_T$? If yes, the system is said to be *controllable*.[2]

A first simple way to consider the controllability concept consists in taking a quite visual perspective using block diagrams. Let us see that through the following example.

Example: A second-order system

Consider the dynamical system modeled by the following block diagram

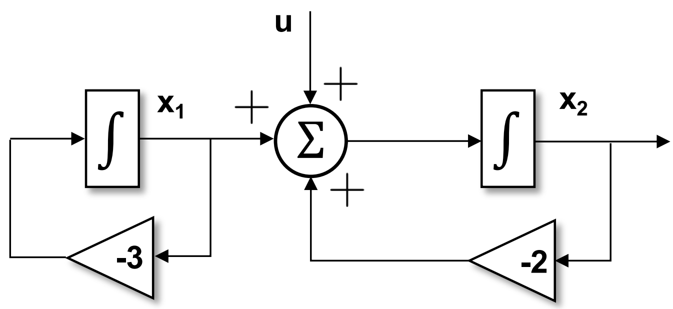

Figure 4.2: Block Diagram of a Second-Order System

One can obviously obtain the set of equations corresponding to this block diagram:

$$\begin{cases} \dot{x}_1 = -3x_1 \\ \dot{x}_2 = -2x_2 + x_1 + u \end{cases} \tag{4.1}$$

Coming back to the block diagram of figure 4.2, following the arrows representing the flow of information in the system, we can see that $u(t)$ will be able to modify or change $x_2(t)$ but does not have any influence on the component $x_1(t)$ of the state. Therefore the system is *not controllable*. □

Take now the new example below.

[2] More rigorous treatments actually call the above control-theoretic concept *Reachability*, while controllability instead refers to bringing any initial condition to 0. For linear continuous-time systems, both notions are equivalent, and some texts put both concepts into the controllability moniker, hence our somewhat loose treatment.

Example: Mass-Spring-Damper with $m = 1kg$

whose block diagram is shown in figure 4.3,

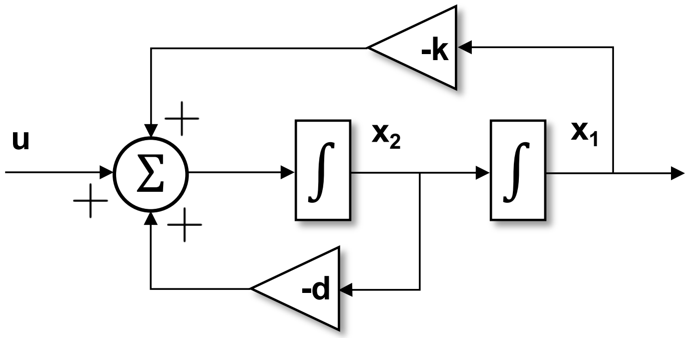

Figure 4.3: Block Diagram of a Mass-Spring-Damper System

we can now see that, in this new example, $u(t)$ influences the whole state (directly or not), thus indicating that the system is in fact controllable. □

We are now ready to state a more formal definition, in line with the few ideas we have seen so far.

Definition: controllability

The system represented by: $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$ is controllable if, for $\mathbf{x}(0) = 0$ and for any point $\mathbf{x}_T$ of the state-space, there exists a finite time $T > 0$ and a control input $u(t)$, $t \in [0, T]$, such that $\mathbf{x}(T) = \mathbf{x}_T$. □

In the above definition, note that $u(t)$ is considered on a *finite* interval of time.
Checking the flow/path of the block diagram of the system, like we have done here above, might be a bit tedious, especially if one think of systems with higher dimensions. Hence we would like to have a mathematical tool which would allow us to conclude on the controllability property just by using the knowledge we have on the model, i.e. embedded in the state-space representation or its matrices $\mathbf{A}$ and $\mathbf{B}$.
This is, roughly speaking, what the well-known Kalman criterion is about. But before digging right into it, we will need a few useful mathematical results, all related to the solutions of the first-order differential equations that are linear state-space representations.

4.2.2 Solutions of state-space representations for LTI systems

Firstly, we would like to answer the following question: what does the solution of the first-order differential equation $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t)$, $\mathbf{x}(0) = \mathbf{x}_0$ look

like?
To compute it, first we pre-multiply the above equation with the *matrix exponential*[3] $e^{-\mathbf{A}t}$:

$$e^{-\mathbf{A}t}\dot{\mathbf{x}}(t) - e^{-\mathbf{A}t}\mathbf{A}\mathbf{x}(t) = e^{-\mathbf{A}t}\mathbf{B}u(t) \tag{4.5}$$

Then, rewrite (4.5) as follows

$$\frac{d}{dt}\left(e^{-\mathbf{A}t}\mathbf{x}(t)\right) = e^{-\mathbf{A}t}\mathbf{B}u(t) \tag{4.6}$$

which is then integrated to give

$$\begin{aligned}
\left[e^{-\mathbf{A}t}\mathbf{x}(\tau)\right]_{\tau=0}^{\tau=t} &= \int_0^t e^{-\mathbf{A}t}\mathbf{B}u(\tau)d\tau \\
&\Downarrow \\
e^{-\mathbf{A}t}\mathbf{x}(t) - \mathbf{x}(0) &= \int_0^t e^{-\mathbf{A}t}\mathbf{B}u(\tau)d\tau \\
&\Downarrow \\
\mathbf{x}(t) &= e^{\mathbf{A}t}\mathbf{x}_0 + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}u(\tau)d\tau \qquad (4.7)
\end{aligned}$$

Impulse response

The impulse response, is, simply put, the response of the system in the case the control input is an impulse, i.e. we have

$$u(t) = \delta(t) \qquad \text{and} \qquad \mathbf{x}(0) = 0 \tag{4.8}$$

Then, the impulse response is given by

$$\mathbf{x}(t) = \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\delta(\tau)d\tau = e^{\mathbf{A}t}\mathbf{B} \tag{4.9}$$

Another useful property: the derivative response property

If $u(t)$ has the response $\mathbf{x}(t)$, then its derivative $\dot{u}(t)$ has the response of $\dot{\mathbf{x}}(t)$.

4.2.3 The Kalman criterion

What we will consider now can be summarized by the following problem statement: can we check whether the system $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}+\mathbf{B}u$ is controllable or not, just

[3]The matrix exponential is defined as

$$e^{\mathbf{A}t} = \sum_{k=0}^{\infty} \frac{1}{k!}\mathbf{A}^k t^k \tag{4.2}$$

Note that a matrix exponential is *not* the same as the usual exponential since

$$\left[e^{\mathbf{A}t}\right]_{ij} \neq e^{a_{ij}t} \tag{4.3}$$

Useful properties of the matrix exponential include

$$\frac{d}{dt}\left(e^{\mathbf{A}t}\right) = \mathbf{A}e^{\mathbf{A}t} = e^{\mathbf{A}t}\mathbf{A} \qquad \text{and} \qquad e^{\mathbf{A}t}e^{-\mathbf{A}t} = \mathbf{I} \tag{4.4}$$

by considering/looking at the matrices $\mathbf{A}$ and $\mathbf{B}$? Rudolf Kalman solved this problem by proposing a simple recipe to check controllability of a considered system.

More precisely, what we would like to determine are the conditions on matrices $\mathbf{A}$ and $\mathbf{B}$ such that, starting from $\mathbf{x}_0 = 0$, we can reach any target $\mathbf{x}_T$ just by using the control input $u(t)$? The problem might not be as simple as it could appear. Indeed, $\mathbf{x}(t)$ is of dimension n while $u(t)$ is only of dimension 1 (i.e. only one input in the present case).

The main idea for solving this problem consists in constructing the signal $u(t)$ on $[0, T]$ so that it is made of a sum of several basic elements whose combination allows the state to span the whole state-space.

To do so, let us define n basic signal elements. First, define the first basic control element as an impulse, whose response is the impulse response (see previous section), i.e. we write

$$\delta(t) \Rightarrow \mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{B} \tag{4.10}$$

Then, we would like to be able to modulate/change the amplitude of this basic control element with a constant α_0 which we can choose at our convenience. Hence we replace (4.10) with

$$u_0(t) := \alpha_0\delta(t) \Rightarrow \mathbf{x}(t) = \alpha_0 e^{\mathbf{A}t}\mathbf{B} \tag{4.11}$$

Next, we proceed similarly with the definition of a second control element u_1 made this time of the *derivative* of an impulse, again modulated by a constant α_1:

$$u_1(t) := \alpha_1\dot{\delta}(t) \Rightarrow \mathbf{x}(t) = \alpha_1\mathbf{A}e^{\mathbf{A}t}\mathbf{B} \tag{4.12}$$

and we proceed similarly for all the n control elements until the last one:

$$u_{n-1}(t) = \alpha_{n-1}\delta^{(n-1)}(t) \Rightarrow \mathbf{x}(t) = \alpha_{n-1}\mathbf{A}^{n-1}e^{\mathbf{A}t}\mathbf{B} \tag{4.13}$$

so that we now have our n basic control elements $u_0(t)$ to $u_{n-1}(t)$.

In order to simplify the discussion further, we let $t = 0^+$ (i.e. we want to steer the system from 0 to x_T in a very short amount of time). Then, if the control input signal is a sum of all the n above control elements, i.e. if

$$u(t) = \sum_{i=0}^{n-1} u_i(t) = \alpha_0\delta(t) + \alpha_1\dot{\delta}(t) + \alpha_2\ddot{\delta}(t) + ... + \alpha_{n-1}\delta^{(n-1)}(t) \tag{4.14}$$

then we have the following response, which, by linearity, is just the sum of the individual responses, so that we have

$$x(t) = \alpha_0\mathbf{B} + \alpha_1\mathbf{AB} + \alpha_2\mathbf{A}^2\mathbf{B} + ... + \alpha_{n-1}\mathbf{A}^{n-1}\mathbf{B} \tag{4.15}$$

Since the α_i coefficients are parameters *which we are free to choose*, then we can steer the system to any point **provided that the vectors B, AB, $\mathbf{A}^2$B,**

... , $\mathbf{A}^{n-1}\mathbf{B}$ are *linearly independent*.
Said differently, we have to check that the matrix composed of these vectors, called *controllability matrix* and expressed as

$$\mathbf{W}_c := \left[\mathbf{B}, \mathbf{AB}, \mathbf{A}^2\mathbf{B}, ..., \mathbf{A}^{n-1}\mathbf{B}\right] \tag{4.16}$$

is nonsingular.
A standard way to check whether a square matrix is nonsingular is to check that its determinant is different from 0. Hence the system is controllable provided $\det(\mathbf{W}_c) \neq 0$.

Main result

The system $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$ is controllable if, and only if the controllability matrix $\mathbf{W}_c$ as given by equation (4.16) is **invertible**! □

Example: Third-order system

Consider the following third-order system

$$\dot{\mathbf{x}} = \begin{bmatrix} -1 & 1 & 0 \\ 1 & -1 & 1 \\ 0 & 1 & -1 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} u \tag{4.17}$$

Is this system controllable?

To see this, let us compute the controllability matrix

$$\mathbf{W}_c = \left[\mathbf{B}, \mathbf{AB}, \mathbf{A}^2\mathbf{B}\right] = \begin{bmatrix} 0 & 1 & -2 \\ 1 & -1 & 3 \\ 0 & 1 & -2 \end{bmatrix} \Rightarrow \det(\mathbf{W}_c) = 0$$

whose determinant is equal to zero, thus showing that the system is *not* controllable.
There is also another way to find the result. Indeed, note that, after we have computed the 3 vectors $\mathbf{B}$, $\mathbf{AB}$ and $\mathbf{A}^2\mathbf{B}$, we have

$$\mathbf{A}^2\mathbf{B} = -4\mathbf{AB} - \mathbf{B} \tag{4.18}$$

meaning that one of the vectors is a linear function of the 2 others, i.e.. the column vectors of $\mathbf{W}_c$ are not linearly independent! Here again, the conclusion is clear: not controllable. □

4.2.4 Controllability and the Controllability Canonical Form (CCF)

In the first chapter, we have seen a special kind of linear state-space representation, which had the form

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 & ... & 0 \\ 0 & 0 & 1 & 0 & ... & 0 \\ 0 & 0 & 0 & 1 & ... & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & ... & 1 \\ -a_0 & -a_1 & -a_2 & -a_3 & ... & -a_{n-1} \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ b \end{bmatrix} u(t). \tag{4.19}$$

This form is called *Controllability Canonical Form*, or CCF, for short. A question which naturally arises in the context of what we have just seen is whether or not this representation is actually controllable. To see that it is indeed the case, let us compute the column vectors of $\mathbf{W}_c$, so that we have

$$\mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ b \end{bmatrix}, \quad \mathbf{AB} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ b \\ * \end{bmatrix}, \quad \mathbf{A}^2\mathbf{B} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ b \\ * \\ * \end{bmatrix}, ..., \quad \mathbf{A}^{n-1}\mathbf{B} = \begin{bmatrix} b \\ * \\ * \\ \vdots \\ * \\ * \end{bmatrix} \tag{4.20}$$

Putting them together to form matrix $\mathbf{W}_c$ makes the determinant of the latter equal to zero only if $b = 0$, which is not really supposed to happen for this particular form, otherwise the input has no impact at all.
Hence a Controllability Canonical Form is *always* controllable. Now its name makes a little bit more sense, does it not? :-)

4.2.5 Systems with several inputs

At first glance, the case where one has more than one input might seem more complicated, and it is indeed, in some sense. However, once again, the core idea is to obtain a set of independent column vectors that will allow us to span the whole state-space. To see this, first consider the next example.

Example: 3 state components, 2 inputs

Consider the following state-space representation with two inputs:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{4.21}$$

Here again, we would like to check whether this system is controllable.

First, note that, considering u_1 and u_2 separately, we can rewrite system (4.21) as

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix} = \begin{bmatrix}1 & 0 & 0\\ 0 & 0 & 1\\ 0 & 1 & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix} + \begin{bmatrix}1\\ 0\\ 0\end{bmatrix} u_1 + \begin{bmatrix}0\\ 0\\ 1\end{bmatrix} u_2 \tag{4.22}$$

where we define $\mathbf{b}_1 := [1, 0, 0]^T$ and $\mathbf{b}_2 := [0, 0, 1]^T$ (which also means that we have $\mathbf{B}$ from (4.21) expressed as $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2]$). Let us first analyze the controllability for each input separately. For the first case, where $u_1 \neq 0$ while $u_2 = 0$, we have the following controllability matrix

$$\mathbf{W}_{c1} = \left[\mathbf{b}_1, \mathbf{A}\mathbf{b}_1, \mathbf{A}^2\mathbf{b}_1\right] = \begin{bmatrix}1 & 1 & 1\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix} \Rightarrow \det(\mathbf{W}_{c1}) = 0 \tag{4.23}$$

thus showing that the system with only u_1 as input is not controllable. Proceeding with the same analysis for the other case (i.e. $u_1 = 0$ and $u_2 \neq 0$), we get the same conclusion, since

$$\mathbf{W}_{c2} = \left[\mathbf{b}_2, \mathbf{A}\mathbf{b}_2, \mathbf{A}^2\mathbf{b}_2\right] = \begin{bmatrix}0 & 0 & 0\\ 0 & 1 & 1\\ 1 & 1 & 2\end{bmatrix} \Rightarrow \det(\mathbf{W}_{c2}) = 0 \tag{4.24}$$

This would seem to indicate that system (4.21) would not be controllable, at least at first glance... However, there are actually combinations of vectors of both W_{c1} and W_{c2} that span the whole state-space: take, for example

$$[\mathbf{b}_1, \mathbf{b}_2, \mathbf{A}\mathbf{b}_2] = \begin{bmatrix}1 & 0 & 0\\ 0 & 0 & 1\\ 0 & 1 & 1\end{bmatrix} \Rightarrow \det([\mathbf{b}_1, \mathbf{b}_2, \mathbf{A}\mathbf{b}_2]) = -1 \tag{4.25}$$

(exercise: find another combination of 3 vectors of $\mathbf{W}_{c1}$ and $\mathbf{W}_{c2}$ which does the trick as well.) More generally, we have hence a total of $2 \times 3 = 6$ vectors from which we need to find 3 linearly independent ones for the system to be controllable. In other words, putting all these vectors together in a large matrix $\mathbf{W}_c$ of dimensions 3×6

$$\mathbf{W}_c = \left[\mathbf{b}_1, \mathbf{b}_2, \mathbf{A}\mathbf{b}_1, \mathbf{A}\mathbf{b}_2, \mathbf{A}^2\mathbf{b}_1, \mathbf{A}^2\mathbf{b}_2\right] = [\mathbf{B}, \mathbf{A}\mathbf{B}, \mathbf{A}^2\mathbf{B}] \tag{4.26}$$

if this matrix has its column rank equal to 3 or *full rank* (the number of linearly independent vectors is 3, the same number as the dimension of the system), then the system is controllable. □

The above discussion can be generalized to any dimension and any number of inputs, giving the following theorem.

Main result

The system with m inputs $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$ is controllable if, and only if the $n \times (mn)$ controllability matrix

$$\mathbf{W}_c = \begin{bmatrix}\mathbf{B} & \mathbf{A}\mathbf{B} & \mathbf{A}^2\mathbf{B} & \cdots & \mathbf{A}^{n-1}\mathbf{B}\end{bmatrix} \tag{4.27}$$

has full rank, ie if $rank(\mathbf{W}_c) = n$. □

Mathematically, a simple way to check whether $\mathbf{W}_c$ has full rank is to check that $\det(\mathbf{W}_c\mathbf{W}_c^T) \neq 0$. In Matlab, computing the rank of a matrix $\mathbf{W}$ can simply done by using the command `rank`.

4.3 Open-Loop control

In this section, we will be interested in computing a control signal $u(t)$ to steer the system from $\mathbf{x}_0$ to $\mathbf{x}_T$, where information on the state is *not* typically used to make online or real-time corrections on the control input. Hence, $u(t)$ will only depend on time.

4.3.1 Open-Loop Control and the Controllability Canonical Form

Let us start with our familiar CCF, which we will put in component form so that we have

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \dot{x}_3 = x_4 \\ \vdots \\ \dot{x}_{n-1} = x_n \\ \dot{x}_n = -a_0 x_1 - a_1 x_2 - ... - a_{n-1} x_n + bu \end{cases} \tag{4.28}$$

where the output equation reads

$$y = x_1. \tag{4.29}$$

What we would like is for the output variable $y(t)$, or equivalently $x_1(t)$, to follow a pre-defined/desired trajectory $y_d(t)/x_{1d}(t)$. What is the corresponding control input $u(t) = u_d(t)$?

To answer this question, let us first assume that our desired trajectory $y_d(t)$ is a polynomial in the time variable. Then, if the output $y(t)$ follows this desired trajectory, then

$$x_1(t) = x_{1d}(t) = y_d(t) \qquad \text{is a polynomial.} \tag{4.30}$$

But thanks to the special structure of the CCF, we can also say that

$$x_2(t) = \dot{x}_1(t) = \dot{x}_{1d}(t) = \dot{y}_d(t) \qquad \text{is a polynomial,} \tag{4.31}$$

and similarly for $x_3(t)$, so that

$$x_3(t) = \dot{x}_2(t) = \ddot{x}_1(t) = \ddot{x}_{1d}(t) = \ddot{y}_d(t) \qquad \text{is a polynomial} \tag{4.32}$$

and so on and so forth until the last state component $x_n(t)$

$$x_n(t) = \dot{x}_{n-1}(t) = x_1^{(n-1)}(t) = y_d^{(n-1)}(t) \qquad \text{also a polynomial,} \tag{4.33}$$

while its derivative $\dot{x}_n(t)$

$$\dot{x}_n(t) = x_1^{(n)}(t) = y_d^{(n)}(t) \qquad \text{again a polynomial.} \tag{4.34}$$

Then, using expressions (4.30) to (4.34) and taking the last line in our CCF expression (4.28), we can isolate our desired control signal to have

$$u(t) = u_d(t) = \frac{1}{b}\left[a_0 y_d(t) + a_1 \dot{y}_d(t) + ... + a_{n-1} y_d^{(n-1)}(t) + y_d^{(n)}(t)\right] \tag{4.35}$$

or, equivalently,

$$u_d(t) = \frac{1}{b}\left[a_0 x_{1d}(t) + a_1 \dot{x}_{1d}(t) + ... + a_{n-1} x_{1d}^{(n-1)}(t) + x_{1d}^{(n)}(t)\right] \tag{4.36}$$

and because a sum of polynomials is also a polynomial, then this means that expression (4.35) and (4.36) are also polynomial in time!

At this point, let us take a step back and discuss the consequences of these few derivations. If $y_d(t)$ is a polynomial trajectory going from one point to another (i.e. from $y_d(0)$ to $y_d(T)$ in a time T), then by taking the successive derivatives of $y_d(t) = x_{1d}(t)$ up to order n, we can compute an open-loop control signal $u_d(t)$ that will steer the system variable $y(t) = x_1(t)$ from $y_d(0)$ to $y_d(T)$ and similarly for the other state components since we will have $x_i(0) = y_d^{(i-1)}(0)$ and $x_i(T) = y_d^{(i-1)}(T)$. Hence we can conclude that, since we can decide on the desired trajectory $y_d(t)$, it is possible to compute $u_d(t)$ so that we steer the state of system (4.28) from any initial state x_0 to any desired target state x_T. Hence equation (4.35) is an expression of a open-loop/feedforward controller for system (4.28), that is an expression computing a control input signal $u_d(t)$ that will make the considered system output follow the desired trajectory $y_d(t)$.

Example: Open-Loop controller for the Mass-Spring-Damper system

Taking now our well-known mass-spring-damper system

$$\begin{bmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\end{bmatrix} = \begin{bmatrix}0 & 1\\ -k/m & -d/m\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix} + \begin{bmatrix}0\\ 1/m\end{bmatrix}u(t) \tag{4.37}$$

and assume, as we have seen, that we want its output $y(t) = x_1(t)$ to follow the desired polynomial trajectory $y_d(t)$.
Then, if $x_1(t)$ follows $y_d(t)$, then $x_1(t) = y_d(t)$. Similarly, we have then

$$x_2(t) = \dot{x}_1(t) = \dot{x}_{1d}(t) = \dot{y}_d(t) \tag{4.38}$$

which makes $x_2(t)$ a polynomial, and

$$\dot{x}_2(t) = \ddot{x}_1(t) = \ddot{y}_d(t) \tag{4.39}$$

also a polynomial. Isolating $u(t)$ in (4.37), we have the following feedforward controller:

$$u_d(t) = k y_d(t) + d\dot{y}_d(t) + m\ddot{y}_d(t) \tag{4.40}$$

□

Interestingly enough, note that even though we have used a polynomial desired trajectory in the above reasoning, we should stress that $y_d(t)$ does not have to be a polynomial. It just has to be differentiable a sufficient number of times. Then a feedforward controller is just the sum of these derivatives. However, because of their simplicity, polynomials are of special interest. Also for their computer efficiency.

4.3.2 Computing polynomial trajectories

Let us now look at how to define a desired trajectory for a second-order system ($n = 2$). Once we are there, generalization should not be a problem.
For a start, we want to steer our system

$$\text{from} \qquad \mathbf{x}(0) = \begin{bmatrix} x_{10} \\ x_{20} \end{bmatrix} =: \mathbf{x}_0 \qquad \text{to} \qquad \mathbf{x}(T) = \begin{bmatrix} x_1(T) \\ x_2(T) \end{bmatrix} = \begin{bmatrix} x_{1T} \\ x_{2T} \end{bmatrix} =: \mathbf{x}_T \tag{4.41}$$

this using a polynomial trajectory written as

$$y_d(t) = \alpha_0 + \alpha_1 t + \alpha_2 t^2 + \alpha_3 t^3 + \dots \tag{4.42}$$

Because we have 4 boundary conditions, i.e. 2 initial conditions and 2 final conditions as we have seen in (4.41), then we only need 4 coefficients in (4.42), from α_0 to α_3 (and all the other coefficients of higher order can therefore be set to zero).

What we would like to obtain is a way to compute the coefficients α_i of $y_d(t)$. Since we are working with a CCF, recall that we have $x_2(t) = \dot{x}_1(t)$, which gives

$$\begin{cases} x_1(t) = y_d(t) = \alpha_0 + \alpha_1 t + \alpha_2 t^2 + \alpha_3 t^3 \\ x_2(t) = \dot{x}_1(t) = \dot{y}_d(t) = \alpha_1 + 2\alpha_2 t + 3\alpha_3 t^2 \end{cases} \tag{4.43}$$

Interestingly, evaluating these polynomial expressions at $t = 0$ and $t = T$ will allow us to match them with the initial conditions ($\mathbf{x}_0$) and final conditions ($\mathbf{x}_T$), giving us our α_i coefficients.
For $t = 0$ and the initial conditions, the polynomial expressions (4.43) give

$$x_1(0) = y_d(0) = x_{10} = \alpha_0, \qquad x_2(0) = \dot{y}_d(0) = x_{20} = \dot{x}_1(0) = \alpha_1 \tag{4.44}$$

i.e. we directly have α_0 and α_1 from initial conditions x_{10} and x_{20}.
For $t = T$ and the final conditions, equation (4.43) reads

$$\begin{cases} x_1(T) = y_d(T) = \alpha_0 + \alpha_1 T + \alpha_2 T^2 + \alpha_3 T^3 = x_{1T} \\ x_2(T) = \dot{y}_d(T) = \alpha_1 + 2\alpha_2 T + 3\alpha_3 T^2 = x_{2T} \end{cases} \tag{4.45}$$

which is a bit more involved but not that much. Indeed, we know α_0 and α_1 since we have just calculated them. The only unknowns are α_2 and α_3 and we have precisely two equations, so that should not be a problem. First isolate the unknowns

$$\begin{cases} x_{1T} - (x_{10} + x_{20}T) = \alpha_2 T^2 + \alpha_3 T^3 \\ x_{2T} - x_{20} = 2\alpha_2 T + 3\alpha_3 T^2 \end{cases} \tag{4.46}$$

and rewrite in vectorial form to obtain

$$\begin{bmatrix} x_{1T} - (x_{10} + x_{20}T) \\ x_{2T} - x_{20} \end{bmatrix} = \begin{bmatrix} T^2 & T^3 \\ 2T & 3T^2 \end{bmatrix} \begin{bmatrix} \alpha_2 \\ \alpha_3 \end{bmatrix}. \tag{4.47}$$

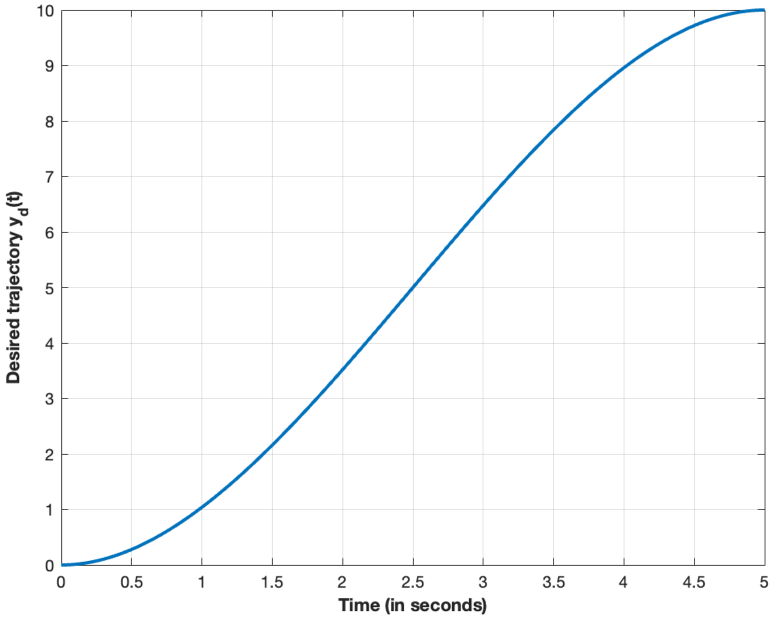

Figure 4.4: A polynomial desired trajectory

Inverting then the T-dependent matrix, we finally have the α_2, α_3 coefficients[4]

$$\begin{bmatrix} \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} T^2 & T^3 \\ 2T & 3T^2 \end{bmatrix}^{-1} \begin{bmatrix} x_{1T} - (x_{10} + x_{20}T) \\ x_{2T} - x_{20} \end{bmatrix} \tag{4.49}$$

An example of a desired trajectory is given in Figure 4.4. In this example, one would like to generate a trajectory going from $y_d(0) = 0$ to $y_d(T) = 10$ in $T = 5$ seconds, with $\dot{y}_d(0) = \dot{y}_d(T) = 0$. Using then expressions (4.44) and (4.49), we get $\alpha_0 = 0$, $\alpha_1 = 0$, $\alpha_2 = 1.2$ and $\alpha_3 = -0.16$.

Example: Back to the Mass-Spring-Damper system

We have previously seen that a feedforward controller for the mass-spring-damper system can be written as

$$u_d(t) = ky_d(t) + d\dot{y}_d(t) + m\ddot{y}_d(t) \tag{4.50}$$

[4]Sometimes, it is possible that the matrix in T in expression (4.47) is not very well conditioned, ie is it close to singularity when T is quite big. A simple way to improve on that consists in first multiplying the second line of (4.46) by T so that, instead of equation (4.47), one gets

$$\begin{bmatrix} x_{1T} - (x_{10} + x_{20}T) \\ (x_{2T} - x_{20})\,T \end{bmatrix} = \begin{bmatrix} T^2 & T^3 \\ 2T^2 & 3T^3 \end{bmatrix} \begin{bmatrix} \alpha_2 \\ \alpha_3 \end{bmatrix} \tag{4.48}$$

where the new matrix in T is much easier to invert.

and since the considered system is of order 2, then the polynomial desired trajectory can be written as

$$y_d(t) = \alpha_0 + \alpha_1 t + \alpha_2 t^2 + \alpha_3 t^3 \tag{4.51}$$

which also gives

$$\dot{y}_d(t) = \alpha_1 + 2\alpha_2 t + 3\alpha_3 t^2 \tag{4.52}$$

and the derivative of (4.52):

$$\ddot{y}_d(t) = 2\alpha_2 + 6\alpha_3 t \tag{4.53}$$

Putting (4.51), (4.52) and (4.53) into (4.50) gives

$$u_d(t) = k(\alpha_0+\alpha_1 t+\alpha_2 t^2+\alpha_3 t^3)+d(\alpha_1+2\alpha_2 t+3\alpha_3 t^2)+m(2\alpha_2+6\alpha_3 t) \tag{4.54}$$

The above feedforward controller is indeed a function of time only and a polynomial. □

4.3.3 General LTI systems

As we have seen above, doing open-loop control when the system is represented by a CCF is quite straightforward.
For a more general linear state-space representation, and associated to the fact that for any particular system, there exists many possible state-space representations, we will simply try to transform the current state-space representation back into a CCF, which will then allow us to apply the open-loop control technique we have seen in the previous section.

State-coordinate transformation

In the first chapter, we have seen that, for a particular system, there are several ways to define the state vector, leading to different state-space representations. What we would like to do here is to find an explicit relation between 2 particular state-space representations and their associated state-vectors.

Let us first start with our usual linear time-invariant state-space representation

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \\ y = \mathbf{C}\mathbf{x} \end{cases} \tag{4.55}$$

and consider another state-vector $\mathbf{z}(t)$ defined differently from $\mathbf{x}(t)$. What we know for now about $\mathbf{z}(t)$ is that it is a linear invertible combination of $\mathbf{x}(t)$, written as

$$\mathbf{z}(t) = \mathbf{T}\mathbf{x}(t), \qquad \text{with } \mathbf{T} \in \mathbb{R}^{n\times n} \tag{4.56}$$

which implies that (dropping the time dependencies for convenience)

$$\mathbf{x} = \mathbf{T}^{-1}\mathbf{z} \qquad \text{and} \qquad \dot{\mathbf{z}} = \mathbf{T}\dot{\mathbf{x}} \tag{4.57}$$

Then, using these few relations, equation (4.55) first transforms into

$$\begin{cases} \dot{\mathbf{z}} = \mathbf{T}(\mathbf{A}\mathbf{x} + \mathbf{B}u) \\ y = \mathbf{C}\mathbf{T}^{-1}\mathbf{z} \end{cases} \tag{4.58}$$

and then

$$\begin{cases} \dot{\mathbf{z}} = \mathbf{T}\mathbf{A}\mathbf{T}^{-1}\mathbf{z} + \mathbf{T}\mathbf{B}u \\ y = \mathbf{C}\mathbf{T}^{-1}\mathbf{z} \end{cases} \tag{4.59}$$

so that we have the following state-space representation in the **z**-space:

$$\begin{cases} \dot{\mathbf{z}} = \tilde{\mathbf{A}}\mathbf{z} + \tilde{\mathbf{B}}u \\ y = \tilde{\mathbf{C}}\mathbf{z} \end{cases} \tag{4.60}$$

where

$$\tilde{\mathbf{A}} = \mathbf{T}\mathbf{A}\mathbf{T}^{-1}, \qquad \tilde{\mathbf{B}} = \mathbf{T}\mathbf{B}, \qquad \tilde{\mathbf{C}} = \mathbf{C}\mathbf{T}^{-1} \tag{4.61}$$

Example: Second-order system

Consider the second-order system

$$\ddot{y} + 3\dot{y} = 2y + u, \tag{4.62}$$

whose corresponding CCF is

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ 2 & -3 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \tag{4.63}$$

and a coordinate transform matrix

$$\mathbf{T} = \begin{bmatrix} 3 & 1 \\ 1 & 0 \end{bmatrix} \tag{4.64}$$

The new state-space representation after coordinate transform is obtained by computing the matrices $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$, which gives

$$\tilde{\mathbf{A}} = \mathbf{T}\mathbf{A}\mathbf{T}^{-1} = \begin{bmatrix} 3 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 2 & -3 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & -3 \end{bmatrix} = \begin{bmatrix} 0 & 2 \\ 1 & -3 \end{bmatrix} \tag{4.65}$$

and

$$\tilde{\mathbf{B}} = \mathbf{T}\mathbf{B} = \begin{bmatrix} 3 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{4.66}$$

so that we have the new state-space representation

$$\dot{\mathbf{z}} = \begin{bmatrix} 0 & 2 \\ 1 & -3 \end{bmatrix} \mathbf{z} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u \tag{4.67}$$

□

Transforming a state-space representation into a CCF

Our aim is thus to find a coordinate transform $\mathbf{z} = \mathbf{Tx}$ such that, after the transform, the state-space representation in the z-space

$$\dot{\mathbf{z}} = \tilde{\mathbf{A}}\mathbf{z} + \tilde{\mathbf{B}}u \tag{4.68}$$

is a controllability canonical form.
First, note that if such a transform exists (let us remember that important "if" for later), then we have the two following important relations

$$\tilde{\mathbf{A}} = \mathbf{TAT}^{-1} \Rightarrow \tilde{\mathbf{A}}\mathbf{T} = \mathbf{TA}, \qquad \tilde{\mathbf{B}} = \mathbf{TB} \tag{4.69}$$

Hence the transformation $\mathbf{T}$ that we are looking for is really the solution of these two algebraic equations.
In order to solve it in a quite easy way, let us first decompose $\mathbf{T}$ into n stacked row vectors $\mathbf{T}_i$, i.e. we have

$$\mathbf{T} = \begin{bmatrix} \mathbf{T}_1 \\ \mathbf{T}_2 \\ \vdots \\ \mathbf{T}_n \end{bmatrix} \tag{4.70}$$

and rewrite the first equation of (4.69), $\tilde{\mathbf{A}}\mathbf{T} = \mathbf{TA}$, by taking into account the special structure of $\tilde{\mathbf{A}}$ and $\mathbf{T}$

$$\begin{bmatrix} 0 & 1 & ... & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & ... & 0 & 1 \\ -a_0 & -a_1 & ... & -a_{n-1} \end{bmatrix} \begin{bmatrix} \mathbf{T}_1 \\ \mathbf{T}_2 \\ \vdots \\ \mathbf{T}_n \end{bmatrix} = \begin{bmatrix} \mathbf{T}_1\mathbf{A} \\ \mathbf{T}_2\mathbf{A} \\ \vdots \\ \mathbf{T}_n\mathbf{A} \end{bmatrix} \tag{4.71}$$

which gives

$$\begin{bmatrix} \mathbf{T}_2 \\ \mathbf{T}_3 \\ \vdots \\ -a_0\mathbf{T}_1 - a_1\mathbf{T}_2 - ... - a_{n-1}\mathbf{T}_n \end{bmatrix} = \begin{bmatrix} \mathbf{T}_1\mathbf{A} \\ \mathbf{T}_2\mathbf{A} \\ \vdots \\ \mathbf{T}_n\mathbf{A} \end{bmatrix} \tag{4.72}$$

This last expression has the following important consequence: if it happens so that we know $\mathbf{T}_1$, then $\mathbf{T}_2$ is readily obtained because $\mathbf{T}_2 = \mathbf{T}_1\mathbf{A}$...
...if we know $\mathbf{T}_2$, then $\mathbf{T}_3$ is readily obtained because $\mathbf{T}_3 = \mathbf{T}_2\mathbf{A}$....
...and if we know $\mathbf{T}_{n-1}$, then $\mathbf{T}_n$ is readily obtained because $\mathbf{T}_n = \mathbf{T}_{n-1}\mathbf{A}$....

This simple reasoning means that if we have $\mathbf{T}_1$, then the rest of the rows of the coordinate transform $\mathbf{T}$ can be deduced directly from this first row $\mathbf{T}_1$ and the matrix $\mathbf{A}$!

Next question is of course how to obtain this first row vector $\mathbf{T}_1$. To answer that question, let us turn to the second expression of (4.69), $\tilde{\mathbf{B}} = \mathbf{TB}$. Use the structure of $\mathbf{T}$ and $\tilde{\mathbf{B}}$ to write

$$\tilde{\mathbf{B}} = \begin{bmatrix} \mathbf{T}_1\mathbf{B} \\ \mathbf{T}_2\mathbf{B} \\ \mathbf{T}_3\mathbf{B} \\ \vdots \\ \mathbf{T}_n\mathbf{B} \end{bmatrix} = \begin{bmatrix} \mathbf{T}_1\mathbf{B} \\ \mathbf{T}_1\mathbf{AB} \\ \mathbf{T}_1\mathbf{A}^2\mathbf{B} \\ \vdots \\ \mathbf{T}_1\mathbf{A}^{n-1}\mathbf{B} \end{bmatrix} \tag{4.73}$$

where note that, because $\mathbf{T}_1$ is a row vector and $\mathbf{B}$ is a column vector, then $\mathbf{T}_1\mathbf{B}$ is a scalar, and similarly for the terms $\mathbf{T}_1\mathbf{A}^i\mathbf{B}$.
Transposing expression (4.73) thus gives

$$\tilde{\mathbf{B}}^T = \begin{bmatrix} 0 & ... & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{T}_1\mathbf{B} & \mathbf{T}_1\mathbf{A}\mathbf{B} & \mathbf{T}_1\mathbf{A}^2\mathbf{B} & ... & \mathbf{T}_1\mathbf{A}^{n-1}\mathbf{B} \end{bmatrix} \tag{4.74}$$

which is nothing but

$$\tilde{\mathbf{B}}^T = \mathbf{T}_1\mathbf{W}_c \tag{4.75}$$

i.e. $\mathbf{T}_1$ multiplied by the controllability matrix!
Hence, provided $\mathbf{W}_c$ is nonsingular and therefore invertible, $\mathbf{T}_1$ is given by

$$\mathbf{T}_1 = \tilde{\mathbf{B}}^T\mathbf{W}_c^{-1} \tag{4.76}$$

At this point, several important remarks can be made. First, if the original system is *not* controllable, it means that $\mathbf{W}_c$ is singular and *not* invertible. Hence $\mathbf{T}_1$ *cannot* be obtained. Hence the "if such a transform exists" statement at the beginning of the discussion. This can be also explained by the fact that a non-controllable system should *not* be able to be put under a CCF, which we have seen to be always controllable.
Secondly, note that having set $\tilde{\mathbf{B}}$ as $\begin{bmatrix} 0 & ... & 0 & 1 \end{bmatrix}^T$ is not the only possibility, i.e. we could have decided that $\tilde{\mathbf{B}} = \begin{bmatrix} 0 & ... & 0 & 3 \end{bmatrix}^T$ for example, as long as $\tilde{\mathbf{B}}$ has the required CCF shape. In this case, however, we would have obtained a different state-space representation.

Example: Back to our previous example

Let us go back to our previous example, and just assume that it is written in the x-space as follows

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 2 \\ 1 & -3 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 1 \\ 0 \end{bmatrix}u \tag{4.77}$$

We want to find our way back to a CCF.
To do this, compute first the controllability matrix $\mathbf{W}_c$:

$$\mathbf{W}_c = \begin{bmatrix} \mathbf{B} & \mathbf{A}\mathbf{B} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.78}$$

whose determinant is obviously 1, so that we know that the system is controllable, and hence that a transform $\mathbf{T}$ to a CCF exists.
Then, defining $\tilde{\mathbf{B}}$ as

$$\tilde{\mathbf{B}}^T = \begin{bmatrix} 0 & 1 \end{bmatrix} \tag{4.79}$$

we can now obtain the first row vector of $\mathbf{T}$, $\mathbf{T}_1$:

$$\mathbf{T}_1 = \tilde{\mathbf{B}}^T\mathbf{W}_c^{-1} = \begin{bmatrix} 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \end{bmatrix} \tag{4.80}$$

and then $\mathbf{T}_2$:

$$\mathbf{T}_2 = \mathbf{T}_1\mathbf{A} = \begin{bmatrix} 0 & 1 \end{bmatrix}\begin{bmatrix} 0 & 2 \\ 1 & -3 \end{bmatrix} = \begin{bmatrix} 1 & -3 \end{bmatrix} \tag{4.81}$$

so that we finally have the transform

$$\mathbf{T} = \begin{bmatrix} 0 & 1 \\ 1 & -3 \end{bmatrix} \tag{4.82}$$

It is then easy to check that we obtain, after transformation, the following CCF

$$\dot{\mathbf{z}} = \begin{bmatrix} 0 & 1 \\ 2 & -3 \end{bmatrix} \mathbf{z} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \tag{4.83}$$

$\square$

Once both the coordinate transform $\mathbf{T}$ and the CCF in the z-space are obtained, we can apply the open-loop control technique we have learnt in this chapter.
This is summarized in the following "algorithm".

Algorithm - Open-Loop control for general LTI systems

Given the following state-space representation

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \\ y = \mathbf{C}\mathbf{x} \end{cases} \tag{4.84}$$

- After checking the controllability of the system, compute the transform $\mathbf{T}$ and obtain a CCF.
- We want the state $\mathbf{x}(t)$ to go from $\mathbf{x}_0$ to $\mathbf{x}_T$. Translate this in the z-space: we want the state $\mathbf{z}(t)$ to go from $\mathbf{z}_0 = \mathbf{T}\mathbf{x}_0$ to $\mathbf{z}_T = \mathbf{T}\mathbf{x}_T$.
- Construct polynomial trajectory $y_d(t)$ using $\mathbf{z}_0$ and $\mathbf{z}_T$ as boundary conditions (ie we want to have $y_d(0) = z_{1,0}$, $\dot{y}_d(0) = z_{2,0}$, ..., for the initial conditions and $y_d(T) = z_{1,T}$, $\dot{y}_d(T) = z_{2,T}$, ..., for the final conditions).
- Implement feedforward controller
$$u_d(t) = a_0 y_d(t) + a_1 \dot{y}_d(t) + ... + a_{n-1} y_d^{(n-1)}(t) + y_d^{(n)}(t)$$

$\square$

4.4 Flatness Theory

Flatness is a theory/tool in control systems which was proposed in the 1990s by Michel Fliess and his co-workers, which turns out to be very useful for, among other things, open-loop control. It can be applied to a wide variety of systems and is quite used in the industry.

It stems for a different way of looking at controllability.

4.4.1 The linear case

Previously, we have seen that one can only find a linear transform $\mathbf{T}$ (from a state-space representation to a controllability canonical form) if the considered system is controllable, and that this transform will change the state-space representation in the x-space to another state-space representation in the z-space.

What we have also seen is the special role played by the first row vector of $\mathbf{T}$, i.e. $\mathbf{T}_1$, which links the first component of the state $\mathbf{z}$ with state $\mathbf{x}$ or $z_1 = \mathbf{T}_1\mathbf{x}$.

Indeed, mathematically, we have

$$\mathbf{z} = \mathbf{T}\mathbf{x} \tag{4.85}$$

or

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} z_1 \\ \dot{z}_1 \\ \ddot{z}_1 \\ \vdots \\ z_1^{(n-1)} \end{bmatrix} = \begin{bmatrix} \mathbf{T}_1\mathbf{x} \\ \mathbf{T}_1\mathbf{A}\mathbf{x} \\ \mathbf{T}_1\mathbf{A}^2\mathbf{x} \\ \vdots \\ \mathbf{T}_1\mathbf{A}^{n-1}\mathbf{x} \end{bmatrix} \tag{4.86}$$

since recall that the state-space representation in the z-space is a CCF.
Hence the starting relation

$$z_1 = \mathbf{T}_1\mathbf{x} \tag{4.87}$$

seems quite central to find the whole transform $\mathbf{T}$. If we now try to express relation (4.87) in plain words, we will just say that **scalar variable $\mathbf{z}_1$ is a linear function of the state $\mathbf{x}$**.

Conversely, we also have that

$$\mathbf{x} = \mathbf{T}^{-1}\mathbf{z} = \mathbf{\Lambda}\mathbf{z}, \qquad \text{with} \qquad \mathbf{\Lambda} := \mathbf{T}^{-1} \tag{4.88}$$

which also means

$$\mathbf{x} = \mathbf{\Lambda} \begin{bmatrix} z_1 \\ \dot{z}_1 \\ \ddot{z}_1 \\ \vdots \\ z_1^{(n-1)} \end{bmatrix} \tag{4.89}$$

or, if we state things in plain words again, **the state $\mathbf{x}$ is a linear function of z_1 and its derivatives up to order $(n-1)$**.

Finally, and as we have seen when we did open-loop control for a CCF, we also have the relation

$$u = a_0 z_1 + a_1 \dot{z}_1 + ... + a_{n-1} z_1^{(n-1)} + z_1^{(n)} \tag{4.90}$$

or, in plain words, **the input u is a linear function of z_1 and its derivatives up to order n.**

Hence variable z_1 is quite central to the above discussion! In flatness theory, this variable is called flat variable or output and is sometimes noted F. We will use this notation in the rest of this section.

We are now ready to give a different definition of controllability using the concept of flat output:

Definition: If, for the system represented by $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$ ($dim(u) = 1$), there exists a linear function of the state, $F = \mathbf{T}_1\mathbf{x}$ (F is called flat output or variable), such that:

- the state $\mathbf{x}$ can be described as a linear function of F and its derivatives up to order $n - 1$,
- the input u can be described as a linear function of F and its derivatives up to order n,

then the system is controllable. □

Important remark

As we will see, it is not always easy to pick the right candidate for the flat output F. Indeed, for some choices of F, it is difficult or not possible to express $\mathbf{x}$ as a function of F and its derivatives.

In this case, one can only pick another candidate for the flat output. If the system is controllable however, it is guaranteed that there is one such variable. For linear systems, finding F can be systematically obtained by the method we saw in the previous section.

To see how to check that a system is controllable using the above definition and the flat output concept, let us consider (again) our previous example.

Example: our (now-infamous) second-order system

Recall that we have the following state-space representation

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 2 \\ 1 & -3 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u \tag{4.91}$$

To begin with, we choose a flat output *candidate* (i.e. we do not know just yet if that is the right candidate). Let us be as simple as possible and pick

$$F = x_1$$

Then we would like to express the *whole* state as a (linear) function of the flat output and its derivative. We already have x_1, so that we only have left to determine x_2.

To do so, rewrite system (4.91) in component form by replacing x_1 with F so that we have

$$\begin{cases} \dot{F} = 2x_2 + u \\ \dot{x}_2 = F - 3x_2 \end{cases} \tag{4.92}$$

Remember that we are supposed to express the state components as a (linear) function of F and its derivatives, and that only. Hence in the above expression, it looks a bit tricky, because x_2 would be a function of $\dot{F}$ and control input u in the first line, while x_2 would be a function of F and $\dot{x}_2$, which is not really working either...
Hence it looks like we do not have the right flat output *candidate*. OK, let us pick instead the new flat output candidate

$$F = x_2 \quad \text{which gives} \quad x_2 = F \tag{4.93}$$

Now we need to express x_1 in terms of its derivatives. Let us see: using the new flat output candidate, we have

$$\begin{cases} \dot{x}_1 = 2F + u \\ \dot{F} = x_1 - 3F \end{cases} \tag{4.94}$$

In the above set of equations, the second line turns out to be quite useful. Indeed, x_2 can be easily isolated, so that, combined with (4.93), we get

$$\begin{cases} x_1 = 3F + \dot{F} \\ x_2 = F \end{cases} \tag{4.95}$$

meaning that **the state is now expressed as a (linear) function of the flat variable F and its derivative (up to order 1)!**
We now have to consider control input u. To do so, isolate u in the first equation of (4.94) to get

$$u = -2F + \dot{x}_1 \tag{4.96}$$

which, thanks to (4.95), gives

$$u = -2F + 3\dot{F} + \ddot{F} \tag{4.97}$$

Hence **the control input u is also a (linear) function of the flat output F and its derivatives (up to order 2)!**
To conclude, these two statements, given by (4.95) and (4.97) allow us to conclude that the system (4.91) is controllable. □

Application to open-loop control:

Because of the way controllability is defined in flatness theory, the verification of whether a system is controllable leads directly to the definition of a feedforward controller. Indeed, letting the flat output $F(t)$ follow a desired trajectory, which we will note for convenience $F_d(t)$, we are then able to express the feedforward control input $u_d(t)$ as a (linear) function of $F_d(t)$ and its derivatives up to order n.
In our example above, we would have

$$u_d(t) = -2F_d(t) + 3\dot{F}_d(t) + \ddot{F}_d(t) \tag{4.98}$$

4.4.2 The nonlinear case

Roughly speaking, the above method can be directly extended to nonlinear systems by simply replacing the word "linear" by the word "nonlinear" in the definition of controllability.

Definition: If, for the system represented by $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u)$ ($\dim(u) = 1$), there exists a nonlinear function of the state, $F = \phi(\mathbf{x})$ (F is called flat output or variable), such that:

- the state $\mathbf{x}$ can be described as a nonlinear function of F and its derivatives up to order $n-1$, i.e. we have

$$\mathbf{x} = \boldsymbol{\psi}_x \left(F, \dot{F}, \ldots, F^{(n-1)} \right) \tag{4.99}$$

- the input u can be described as a nonlinear function of F and its derivatives up to order n, i.e. we have

$$u = \psi_u \left(F, \dot{F}, \ldots, F^{(n)} \right) \tag{4.100}$$

then the system is controllable. □

Let us now see how the above can be applied on a simple example.

Example: Robot "elbow"/pendulum

Recall a state-space representation of the controlled pendulum

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} x_2(t) \\ -\dfrac{g}{l}\sin x_1(t) + \dfrac{1}{ml^2}u(t) \end{bmatrix} \tag{4.101}$$

First, note that by choosing the flat output candidate $F = x_1$ (i.e. we have $\phi(\mathbf{x}) = x_1$), we get the simple expression

$$\begin{cases} x_1 = F \\ x_2 = \dot{F} \end{cases} \Rightarrow \boldsymbol{\psi}_x(F, \dot{F}) = \begin{bmatrix} F \\ \dot{F} \end{bmatrix} \tag{4.102}$$

i.e. the state is a function of F and its derivative.
Next, we look at the control input, and from the second line of (4.101), we get

$$u = mgl \sin F + ml^2 \ddot{F} = \psi_u(F, \dot{F}, \ddot{F}) \tag{4.103}$$

i.e. the control input is a function of F and its derivatives. From (4.102) and (4.103), we can conclude that the system is controllable. □

Algorithm: open-loop control for nonlinear systems

- check that the system is controllable using Flatness Theory.

- compute $F(0)$ and $F(T)$ from the function $\phi(\mathbf{x}(t))$, evaluated at times $t = 0$ and $t = T$. Also compute the derivatives of $F(t)$, also evaluated at times $t = 0$ and $t = T$,
- construct a polynomial trajectory $F_d(t)$ from $F_d(0)$ to $F_d(T)$, with derivatives at times $t = 0$ and $t = T$ corresponding to the boundary conditions on the derivatives of $F(t)$ computed in the previous step,
- Define the feedforward controller using the function ϕ_u, that is
$$u_d(t) = \psi_u\left(F_d(t), \dot{F}_d(t), ..., F_d^{(n)}(t)\right)$$

□

For systems with multiple inputs, the flat output is now a **vector F** whose dimension *is the same as the dimension of the control input vector* **u**. Indeed, if we have m inputs, this means

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix} \Rightarrow \mathbf{F} = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_m \end{bmatrix} \tag{4.104}$$

but the rest of the procedure for checking controllability is exactly the same. Let us see that through our examples in mobile robotics.

Example: Flatness of the two-wheeled robot

Recall the model of a two-wheeled robot we saw in the previous chapter:

$$\begin{cases} \dot{x}_1 = u_1 \cos\theta \\ \dot{x}_2 = u_1 \sin\theta \\ \dot{\theta} = u_2 \end{cases} \tag{4.105}$$

Since we have 2 control inputs (u_1 and u_2), then we will need a flat output vector **F** with 2 components (or two flat outputs). Let us choose the flat output candidates

$$F_1 = x_1 \quad \text{and} \quad F_2 = x_2 \tag{4.106}$$

i.e. the position of the robot in the plane.
In order to get the function $\boldsymbol{\psi}_x$, we need to have the 3rd component of the state as a function of the flat output and derivatives. To do this, replace the terms in x_1 and x_2 by F_1 and F_2 in model (4.105), and divide the second line of the model by the first line to get

$$\frac{\dot{F}_2}{\dot{F}_1} = \tan\theta \tag{4.107}$$

so that the multi-dimensional function/vector field $\mathbf{x} = \boldsymbol{\psi}_x(\mathbf{F}, \dot{\mathbf{F}})$ reads

$$\mathbf{x} = \boldsymbol{\psi}_x(\mathbf{F}, \dot{\mathbf{F}}) = \begin{bmatrix} F_1 \\ F_2 \\ \arctan\left(\dot{F}_2/\dot{F}_1\right) \end{bmatrix} \tag{4.108}$$

Hence the state is a nonlinear function of $\mathbf{F}$ and its derivative. Now for the control inputs. For u_1, add the squared first line to the squared second line of (4.105) to obtain (after replacement with the flat output components)

$$\dot{F}_1^2 + \dot{F}_2^2 = u_1^2 \tag{4.109}$$

which gives

$$u_1 = \pm\sqrt{\dot{F}_1^2 + \dot{F}_2^2}. \tag{4.110}$$

For u_2, taking the third line of (4.105) and using (4.108), we have

$$u_2 = \frac{d}{dt}\left(\arctan\left(\frac{\dot{F}_2}{\dot{F}_1}\right)\right) \tag{4.111}$$

which, after differentiating with respect to time and simplification, gives

$$u_2 = \frac{\ddot{F}_2\dot{F}_1 - \ddot{F}_1\dot{F}_2}{\dot{F}_1^2 + \dot{F}_2^2} \tag{4.112}$$

to finally obtain

$$\mathbf{u} = \boldsymbol{\psi}_u(\mathbf{F}, \dot{\mathbf{F}}, \ddot{\mathbf{F}}) = \begin{bmatrix} \pm\sqrt{\dot{F}_1^2 + \dot{F}_2^2} \\ \dfrac{\ddot{F}_2\dot{F}_1 - \ddot{F}_1\dot{F}_2}{\dot{F}_1^2 + \dot{F}_2^2} \end{bmatrix} \tag{4.113}$$

so that, combining (4.108) and (4.113), we can conclude that the system is controllable. □

Exercise: Car-like robot

Prove that the car-like robot is controllable.

4.5 Concluding remarks on open-loop control

Before closing this chapter, it might be good to reflect a little bit on the use of open-loop control in general. When is it useful to pre-compute a control input made to correspond to a desired trajectory is obviously a matter of context, and a correct answer, if not very informative, is that it depends.

First, it is worth mentioning that open-loop control should only be used by itself if the system to be controlled is stable in some sense (i.e. it forgets initial conditions). Otherwise, a feedforward controller will either need to be helped by a feedback stabilizer as it tries to follow the desired trajectory, or the state of the system will have to be caught as it reaches a vicinity of the target state, again by a feedback stabilizer.

Generally speaking, open-loop control can be very useful for plants evolving in controlled environments where the model of the system, as well as its parameters, are quite well-known and constant or varying in a predictable way (think of an industrial robot for example). However, open-loop control might be less useful when models are very changing and very uncertain, with highly unpredictable disturbances (think of a ship in a storm). In this case, feedback will be more prevalent.