`RegT4` – Modern Control Engineering

Naoki Pross – `naoki.pross@ost.ch`

Spring Semester 2022

Abstract

This document is "open source", you can find the LaTeX sources at `https://github.com/NaoPross/RegT4`. All diagrams were made with TikZ. The content is based on the material of Prof. Dr. M. Kottman, from the course *Regelungstechnik 4* at the University of Applied Sciences Eastern Switzerland (OST). If you find typos or errors you can open an PR on Github or mail me at `naoki.pross@ost.ch` if I'm still around (until spring 2022) or `np@0hm.ch`.

Contributors

The following students have either directly or indirectly contributed to this work: Naoki Pross, Lukas Schmidt.
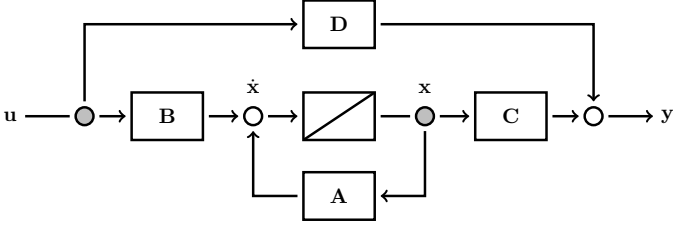
License

Contents

Figure 1: State space model of a LTI system.

1 State Space Models

1.1 Introduction

In modern control theory a *state* of a dynamic system is the smallest set of variables (called *state variables*) such that the knowledge of these variables at $t = t_0$, together with the knowledge of the input for $t \geq t_0$, completely determines the behavior of the system for any time $t \geq t_0$.

1.2 State Space Equations

The state space equations are, in their most general form, a set of first order differential equations and a set of outputs:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t)$$
$$\mathbf{y} = \mathbf{g}(\mathbf{x}, \mathbf{u}, t).$$

This can describe any system because higher order differential equations can (practically) always be rewritten as a set of first order differential equations. In the case where the system is *linear* they can be rewritten using linear algebra:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t),$$
$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t),$$

where $\mathbf{A} \in \mathbb{R}^{n \times n}, \mathbf{B} \in \mathbb{R}^{n \times m}, \mathbf{C} \in \mathbb{R}^{p \times n}, \mathbf{D} \in \mathbb{R}^{p \times m}$ (see figure 1). In general in a time varying system all four matrices are actually functions of time ($\mathbf{A} = \mathbf{A}(t), \ldots$). Whereas for *linear time invariant* (LTI) systems they are constants.

1.3 Linearization of non LTI systems

In general dynamics are not LTI systems, however it is possible to create linear approximation of complex dynamics around a *set point* $(\mathbf{x}_0, \mathbf{u}_0)$, if it is also a fixed point (that means $\mathbf{f}(\mathbf{x}_0, \mathbf{u}_0) = \mathbf{0}$ and $\mathbf{g}(\mathbf{x}_0, \mathbf{u}_0) = \mathbf{y}_0$), by computing the four Jacobian matrices:

$$\mathbf{A} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}_0, \mathbf{u}_0), \qquad \mathbf{B} = \frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\mathbf{x}_0, \mathbf{u}_0),$$
$$\mathbf{C} = \frac{\partial \mathbf{g}}{\partial \mathbf{x}}(\mathbf{x}_0, \mathbf{u}_0), \qquad \mathbf{D} = \frac{\partial \mathbf{g}}{\partial \mathbf{u}}(\mathbf{x}_0, \mathbf{u}_0).$$

Then the differences $\Delta \mathbf{x} = \mathbf{x} - \mathbf{x}_0$, $\Delta \mathbf{u} = \mathbf{u} - \mathbf{u}_0$ are used in the model:

$$\Delta \dot{\mathbf{x}}(t) = \mathbf{A}\Delta \mathbf{x} + \mathbf{B}\Delta \mathbf{u},$$
$$\Delta \mathbf{y}(t) = \mathbf{C}\Delta \mathbf{x} + \mathbf{D}\Delta \mathbf{u}.$$

1.4 Time-Domain Solutions

1.4.1 Homogeneous Case

To solve for the solution of the system at rest (no inputs)

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x},$$

we assume that the solution is a power series in t, or

$$\mathbf{x}(t) = \mathbf{b}_0 + \mathbf{b}_1 t + \mathbf{b}_2 t^2 + \cdots + \mathbf{b}_k t^k + \ldots.$$

Substituting into the differential equation yields a solution

$$\mathbf{x}(t) = \underbrace{\left(\mathbf{I} + \mathbf{A}t + \frac{1}{2!}\mathbf{A}^2 t^2 + \cdots + \frac{1}{k!}\mathbf{A}^k t^k + \ldots \right)}_{\text{Matrix exponential } e^{\mathbf{A}t}} \mathbf{x}(0),$$

which looks like the power series for the exponential function. Thus we define it to be the *matrix exponential* and write the solution as

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}(0).$$

The matrix exponential has the property that

$$e^{\mathbf{A}(t+s)} = e^{\mathbf{A}t}e^{\mathbf{A}s},$$

and in particular when $s = -t$ then the expression equals the identity matrix. Thus the inverse of $e^{\mathbf{A}t}$ is $e^{-\mathbf{A}t}$ and always exists. It is important to note that

$$e^{(\mathbf{A}+\mathbf{B})t} \neq e^{\mathbf{A}t}e^{\mathbf{B}t},$$

unless we are in the very special case when $\mathbf{A}\mathbf{B} = \mathbf{B}\mathbf{A}$.

1.4.2 Transition Matrix

The result of the exponential matrix is of such relevance that it has a special name. The expression

$$\mathbf{\Phi}(t) = e^{\mathbf{A}t}$$

is called the *transition matrix*, since it describes how the initial conditions change. The transition matrix has some important algebraic properties which are listed below:

$$\mathbf{\Phi}(0) = e^{\mathbf{A}0} = \mathbf{I},$$
$$\mathbf{\Phi}^{-1}(t) = \left(e^{\mathbf{A}t}\right)^{-1} = \mathbf{\Phi}(-t),$$
$$\mathbf{\Phi}(t)^n = \left(e^{\mathbf{A}t}\right)^n = \mathbf{\Phi}(nt),$$
$$\mathbf{\Phi}(t_1 + t_2) = e^{\mathbf{A}(t_1+t_2)} = \mathbf{\Phi}(t_1)\mathbf{\Phi}(t_2) = \mathbf{\Phi}(t_2)\mathbf{\Phi}(t_1),$$
$$\mathbf{\Phi}(t_2 - t_1)\mathbf{\Phi}(t_1 - t_0) = \mathbf{\Phi}(t_2 - t_0)$$
$$\dot{\mathbf{\Phi}}(t) = \mathbf{A}\mathbf{\Phi}(t).$$

A noteworthy result from the last property is that

$$\mathbf{A} = \dot{\mathbf{\Phi}}(t)\mathbf{\Phi}^{-1}(t) = \dot{\mathbf{\Phi}}(t)\mathbf{\Phi}(-t).$$

Finally, in the case when $\mathbf{A}$ is a diagonal matrix with entries $\lambda_1, \ldots, \lambda_n$ then

$$\mathbf{\Phi}(t) = e^{\mathbf{A}t} = \begin{bmatrix} e^{\lambda_1 t} & & 0 \\ & \ddots & \\ 0 & & e^{\lambda_n t} \end{bmatrix}.$$
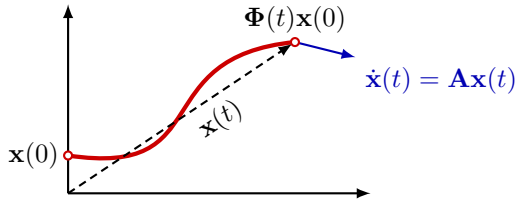
Figure 2: Interpretation of the $\mathbf{A}$ and $\mathbf{\Phi}$ matrices.

1.4.3 Nonhomogeneous Case

To expand the previous result, in the more general case when

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u},$$

a few steps that are the same as in the scalar case yield the solution

$$\mathbf{x}(t) = \mathbf{\Phi}(t)\mathbf{x}(0) + \int_0^t \mathbf{\Phi}(t - \tau)\mathbf{B}\mathbf{u}(\tau)d\tau.$$

1.5 Laplace s-Domain Solution

1.5.1 Transfer Matrix

By taking the Laplace transform the state space equations of an LTI system they become:

$$s\mathbf{X}(s) - \mathbf{x}(0) = \mathbf{A}\mathbf{X}(s) + \mathbf{B}\mathbf{U}(s),$$
$$\mathbf{Y}(s) = \mathbf{C}\mathbf{X}(s) + \mathbf{D}\mathbf{U}(s).$$

Similarly to the one dimensional case it is now possible to define the more general equivalent of a transfer function:

$$\mathbf{G}(s) = \mathbf{C}\left(s\mathbf{I} - \mathbf{A}\right)^{-1}\mathbf{B} + \mathbf{D},$$

which is the *transfer matrix*.

1.5.2 Impulse Response

The inverse Laplace transform of the transfer matrix is the *impulse response*. However, setting $\mathbf{u}(t) = \boldsymbol{\delta}(t)$ will also give the impulse response. By taking the result in section 1.4.3 and substituting into the output equation, we get that with a Dirac delta as input

$$\mathbf{g}(t) = \mathbf{C}\mathbf{\Phi}(t)\mathbf{B} + \mathbf{D}\boldsymbol{\delta}(t).$$

1.6 Controllability

A system is said to be *controllable* at time t_0 if it is possible my means of an unconstrained control vector to transfer the system from any initial state $\mathbf{x}(t_0)$ to any other state in a finite interval of time. To check if a system is completely controllable there is the $n \times nm$ *controllability matrix*

$$\mathbf{Q}_c = \begin{bmatrix} \mathbf{B} & \mathbf{A}\mathbf{B} & \cdots & \mathbf{A}^{n-1}\mathbf{B} \end{bmatrix}.$$

If rank $\mathbf{Q}_c < n$ (has less than n linearly independent rows), or in the case when $\mathbf{Q}_c$ is a square matrix $\det \mathbf{Q}_c = 0$, then the system is not totally state controllable.

It is also possible to determine the controllability from the transfer function (or matrix) of a system. It can be shown that if cancellation occurs (German: *Polkürzung*), then the system cannot be controlled in the direction of the canceled mode.

1.7 Observability

In the same (too) abstract spirit as in the controllability: a system is said to be *observable* at time t_0 if, with the system in state $\mathbf{x}(t_0)$, it is possible to determine this state from the observation of the output over a finite time interval. The observability can be determined through the $np \times n$ *observability matrix*

$$\mathbf{Q}_o = \begin{bmatrix} \mathbf{C} \\ \mathbf{C}\mathbf{A} \\ \vdots \\ \mathbf{C}\mathbf{A}^{n-1} \end{bmatrix} = \begin{bmatrix} \mathbf{C}^* & \mathbf{A}^*\mathbf{C}^* & \cdots & (\mathbf{A}^*)^{n-1}\mathbf{C}^* \end{bmatrix},$$

where the asterisk is the conjugate transpose (for real matrices $\mathbf{A}^* = \mathbf{A}^\mathsf{T}$). The system is observable only if rank $\mathbf{Q} \geq n$ (there are at least n linearly independent rows).

In terms of the transfer function: if cancellation occurs in the transfer function then the canceled mode cannot be observed for the output. In other words, no cancellation is a necessary and sufficient condition for total observability.

1.8 Stability

A state space model system is stable is the $\mathbf{A}$ matrix is positive definite, which means that the real part of all eigenvalues $\operatorname{Re}\{\lambda_i\} > 0$. Another way to determine if $\mathbf{A}$ is positive definite is to test whether the quadratic expression $\mathbf{x}^*\mathbf{A}\mathbf{x} > 0$ for all $\mathbf{x} \neq \mathbf{0}$.

1.9 Equivalent Representations

In this section we assume without loss of generality that the system has a transfer function

$$\frac{Y(s)}{U(s)} = d + \frac{b_{n-1}s^{n-1} + b_{n-2}s^{n-2} + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0}.$$

1.9.1 Controllable Canonical Form

This form (German: *Regelungsnormalform*) is important in discussing the pole-placement approach. It is given by

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_0 & -a_1 & -a_2 & \dots & -a_{n-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u.$$

And the output is

$$y = \begin{bmatrix} b_0 & \dots & b_{n-1} \end{bmatrix} \mathbf{x} + du.$$

1.9.2 Observable Canonical Form

This mode is similar to previous form, in fact, the $\mathbf{A}$ matrix is the transposed version of the controllable form:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \dots & -a_n \\ 1 & 0 & 0 & \dots & -a_{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & 0 & \dots & -a_2 \\ 0 & 0 & 0 & \dots & -a_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-2} \\ b_{n-1} \end{bmatrix} u.$$

1.9.3 Diagonal Form (Decoupling)

An important form is when the $\mathbf{A}$ matrix is diagonalized, i.e.

$$
\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} \lambda_1 & & & & \\ & \lambda_2 & & & \\ & & \ddots & & \\ & & & \lambda_{n-1} & \\ & & & & \lambda_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \tilde{\mathbf{B}}u.
$$

In this representation the state variables are the *eigenstates* of the system, and they are decoupled from each other. Since the diagonal contains the eigenvalues, as noted in the stability section 1.8, they describe the poles of the transfer function $G(s)$, and therefore the stability of the system. In other words

$$
\frac{Y(s)}{U(s)} = \frac{b_1 c_1}{s - \lambda_1} + \frac{b_2 c_2}{s - \lambda_2} + \cdots + \frac{b_n c_n}{s - \lambda_{n-1}}.
$$

1.10 Similarity Transform

In the previous section there were many representation that are practical to use, but in fact there is nothing special about them. In fact, the state space can be transformed into an infinite number of other equivalent representations.

Given an arbitrary regular matrix $\mathbf{P}$, the state space in $\mathbf{x}$ can be brought into a new set of state variables $\mathbf{x}' := \mathbf{P}\mathbf{x}$ by appropriately transforming the matrices as follows:

$$
\mathbf{A}' := \mathbf{P}\mathbf{A}\mathbf{P}^{-1}, \qquad \mathbf{B}' := \mathbf{P}\mathbf{B},
$$
$$
\mathbf{C}' := \mathbf{C}\mathbf{P}^{-1}, \qquad \mathbf{D}' := \mathbf{D}.
$$

1.11 Subsystem Aggregation

1.11.1 Series

When after a subsystem 1 a subsystem 2 is connected in series the model of the entire path is

$$
\begin{bmatrix} \dot{\mathbf{x}}_1 \\ \dot{\mathbf{x}}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{B}_2\mathbf{C}_1 & \mathbf{A}_2 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2\mathbf{D}_1 \end{bmatrix} \mathbf{u},
$$
$$
\mathbf{y} = \begin{bmatrix} \mathbf{D}_2\mathbf{C}_1 & \mathbf{C}_2 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} + \mathbf{D}_2\mathbf{D}_1\mathbf{u}.
$$

1.11.2 Parallel

When a subsystem 1 and a subsystem 2 are connected in parallel, the model of the entire system is

$$
\begin{bmatrix} \dot{\mathbf{x}}_1 \\ \dot{\mathbf{x}}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix} \mathbf{u},
$$
$$
\mathbf{y} = \begin{bmatrix} \mathbf{C}_1 & \mathbf{C}_2 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} + (\mathbf{D}_2 + \mathbf{D}_1)\mathbf{u}.
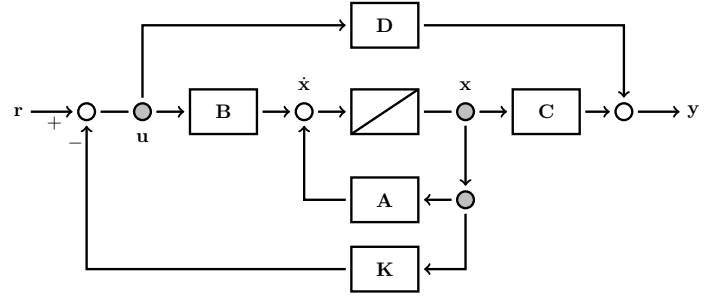$$

Figure 3: Pole placement through the gain matrix $\mathbf{K}$.

1.11.3 Feedback

When to a subsystem 1 a subsystem 2 is connected in the negative feedback path, the matrices of the entire system are

$$
\mathbf{A}' := \begin{bmatrix} \mathbf{A}_1 - \mathbf{B}_1\mathbf{K}_1\mathbf{D}_2\mathbf{C}_1 & -\mathbf{B}_1\mathbf{K}_1\mathbf{C}_2 \\ \mathbf{B}_2\mathbf{K}_2\mathbf{C}_1 & \mathbf{A}_2 - \mathbf{B}_2\mathbf{K}_2\mathbf{D}_1\mathbf{C}_2 \end{bmatrix},
$$
$$
\mathbf{B}' := \begin{bmatrix} \mathbf{B}_1\mathbf{K}_1 \\ \mathbf{B}_2\mathbf{K}_2\mathbf{D}_1 \end{bmatrix}, \qquad \mathbf{C}' := \begin{bmatrix} \mathbf{K}_2\mathbf{C}_1 & -\mathbf{K}\mathbf{D}_1\mathbf{C}_2 \end{bmatrix}
$$

and $\mathbf{D}' := \mathbf{K}_2\mathbf{D}_1$, where $\mathbf{K}_1 = (\mathbf{I} - \mathbf{D}_2\mathbf{D}_1)^{-1}$ and $\mathbf{K}_2 = (\mathbf{I} - \mathbf{D}_1\mathbf{D}_2)^{-1}$. In the case of a positive feedback the minus signs become pluses.

1.12 Create a State Space Model from a Transfer Function

[TODO: Ogata and skript p. 169, also from a diagram on p. 171.]

2 Control in State Space

2.1 Pole Placement

2.1.1 Gain Matrix

Consider the state space model

$$
\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t),
$$
$$
\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t),
$$

and assume that we can somehow always know the state $\mathbf{x}$. Then we add a feedback loop from the state to the input with the *gain matrix* $\mathbf{K}$ (see figure 3). This introduces a new term for the desired value $\mathbf{r}$ and changes the state space equation to

$$
\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{B}\mathbf{K})\,\mathbf{x} + \mathbf{B}\mathbf{r}.
$$

The addition of $\mathbf{K}$ allow to change the eigenvalues of the system matrix and thus move the poles. Therefore, the poles of the new matrix $\tilde{\mathbf{A}} := \mathbf{A} - \mathbf{B}\mathbf{K}$ are called the regulator poles. To allow an arbitrary placement of poles the system needs to be completely controllable. Without any additional construction, the gain matrix is basically a proportional controller (German: *P-Regler*).

2.1.2 Solve for the Gain Matrix

Now, if we wish to bring the poles of the closed loop system to the locations $\mu_1, \mu_2, \ldots, \mu_n$, in the most general case we can use the fact that

$$\det\left(s\mathbf{I} - \tilde{\mathbf{A}}\right) = \det\left(s\mathbf{I} - \mathbf{A} + \mathbf{BK}\right)$$
$$= (s - \mu_1)(s - \mu_2)\cdots(s - \mu_n)$$
$$= s^n + \alpha_1 s_{n-1} + \cdots + \alpha_{n-1}s + \alpha_n,$$

and solve for the entries of $\mathbf{K}$ by comparing coefficients.

If the state space representation is in the diagonal form (decoupled, eigenmodes), then finding $\mathbf{K}$ is as simple as solving for the entries of $\mathbf{K}$:

$$\mathbf{A} - \mathbf{BK} = \underbrace{\begin{bmatrix} \mu_1 & & \\ & \ddots & \\ & & \mu_n \end{bmatrix}}_{\mathbf{M}}.$$

Sometimes it is possible to do $\mathbf{K} = \mathbf{B}^{-1}(\mathbf{A} - \mathbf{K})$, but more often the system is overdetermined and we are free to choose some entries of $\mathbf{K}$ (which is even easier).

If the state space representation is in the controllable canonical form, and

$$\mathbf{K} = \begin{bmatrix} k_1 & k_2 & \ldots & k_n \end{bmatrix}$$

since $\tilde{\mathbf{A}} = \mathbf{A} - \mathbf{BK} =$

$$\begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \\ -a_n - k_1 & -a_{n-1} - k_2 & -a_{n-2} - k_3 & \ldots & -a_1 - k_n \end{bmatrix},$$

which means that

$$G(s) = \frac{b_{n-1}s^{n-1} + \cdots + b_1 s + b_0}{s^n + (a_{n-1} + k_n)s^{n-1} + \cdots + (a_0 + k_1)},$$

the poles are found by comparing the coefficients of the denominator:

$$s^n + (a_{n-1} + k_n)s^{n-1} + \cdots + (a_0 + k_1)$$
$$= (s - \mu_1)(s - \mu_2)\cdots(s - \mu_n).$$

2.2 Linear Quadratic Regulator (LQR)

A problem of the pole placement approach is that it is not always obvious which poles should be moved and where. The *linear quadratic regulator* (LQR) also called *quadratic optimal regulator* solves this problem.

In the LQR design we define a performance index to minimize:

$$J = \int_0^\infty \left(\mathbf{x}^*\mathbf{Q}\mathbf{x} + \mathbf{u}^*\mathbf{R}\mathbf{u}\right) dt,$$

where $\mathbf{Q}$ and $\mathbf{R}$ are positive definite (or semidefinite) Hermitian or real symmetric matrices. The $\mathbf{Q}$ matrix determines the importance of the relative error, whereas the $\mathbf{R}$ matrix

determines the cost (in energy) of modifying the state. Since in the pole placement $\mathbf{u} = -\mathbf{K}\mathbf{x}$ the problem becomes

$$\arg\min_{\mathbf{K}} \int_0^\infty \mathbf{x}^*\left(\mathbf{Q} + \mathbf{K}^*\mathbf{R}\mathbf{K}\right)\mathbf{x}\, dt.$$

Then, the expression can be further developed by using the fact that

$$\mathbf{x}(t) = e^{(\mathbf{A}-\mathbf{BK})t}\mathbf{x}(0),$$

which when substituted into the previous expression gives

$$\mathbf{x}(0)^* \underbrace{\int_0^\infty \left[e^{(\mathbf{A}-\mathbf{BK})t}\right]^* \left(\mathbf{Q} + \mathbf{K}^*\mathbf{R}\mathbf{K}\right) e^{(\mathbf{A}-\mathbf{BK})t}\, dt}_{\mathbf{S}(\mathbf{K})} \mathbf{x}(0).$$

The minimization problem is completed by computing

$$\text{minimize } J \iff 0 = \frac{\partial J}{\partial K_{ij}} = \mathbf{x}^*(0)\frac{\partial \mathbf{S}}{\partial K_{ij}}\mathbf{x}(0).$$

It is possible to show that the resulting optimal gain matrix is

$$\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^*\mathbf{S}.$$

The LQR optimization problem can also be described in terms of the output

$$J = \int_0^\infty \left(\mathbf{y}^*\mathbf{V}\mathbf{y} + \mathbf{u}^*\mathbf{R}\mathbf{u}\right) dt,$$

and in a state space model $\mathbf{A}, \mathbf{B}, \mathbf{C}$ with $\mathbf{D} = \mathbf{0}$ can be rewritten using $\mathbf{x}$ as follows: $\mathbf{y}^*\mathbf{V}\mathbf{y} = \mathbf{x}^*\mathbf{C}^*\mathbf{V}\mathbf{C}\mathbf{x}$.

2.3 PI-Control

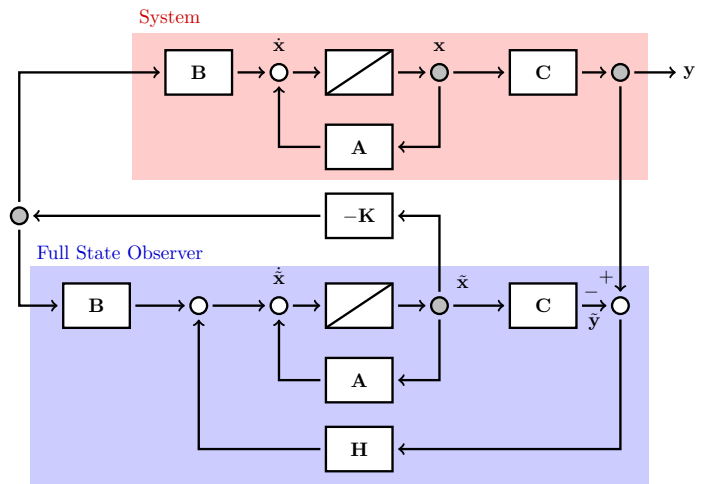[TODO: Skript p. 200.]

2.4 Observer

Figure 4: Full state observer to estimate $\mathbf{x}$ for the pole placement matrix $\mathbf{K}$.
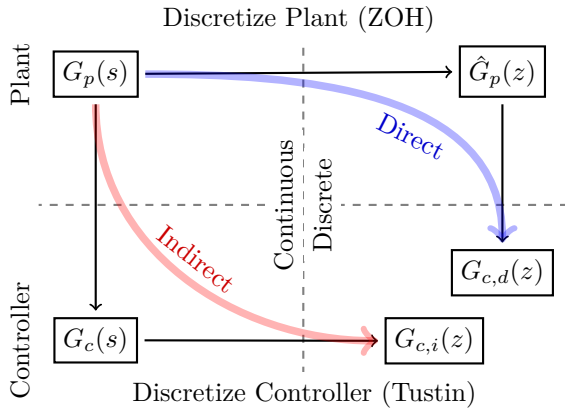
[TODO: Skript p. 203.]

Figure 5: Direct vs indirect design. In general $G_{c,i}(z) \neq G_{c,d}(z)$.

3 Discrete-Time Modelling

3.1 Sampling

In discrete time the system works with sampled signals. The sampling time T must be chosen such that the sampling theorem (Nyquist Shannon) is satisfied, i.e.

$$\omega_s = \frac{2\pi}{T} \geq 2\omega_r,$$

where ω_r is the is the highest frequency component in the reference signal $r(t)$.

In practice sampling means using an analog to digital converter (ADC), but mathematically the sampling is done through a multiplication with a Dirac comb

$$\sum_k \delta(t - kT).$$

The converse operation, which is done with a digital to analog converter (DAC), is represented in many ways depending on the type of reconstruction. The simplest reconstruction is the zero order hold (ZOH) reconstruction, which keeps the signal fixed at the value of the last digital sample until the new one comes.

3.1.1 Choosing T Empirically

[TODO: Stuff from skript p. 230, parameters from Takahashi et al.]

3.2 Indirect Design

In the indirect design we create a controller in the continuous time domain and then discretize it (see figure 5). This is okay as long as the sampling theorem is satisfied.

Thus, given a plant with transfer function $G_p(s)$ we create a continuous time controller $G_c(s)$. Then to discretize it we can choose how to approximate the integrator element from one of the following 3 options:

$$
\begin{aligned}
s = \frac{z-1}{T} && z = 1 + sT && \text{(Euler)}, \\
s = \frac{z-1}{Tz} && z = \frac{1}{1-sT} && \text{(Forward)}, \\
s = \frac{2}{T}\frac{z-1}{z+1} && z = \frac{2+sT}{2-sT} && \text{(Tustin)}.
\end{aligned}
$$

The last one (Tustin) is the most popular. To obtain the discretized controller transfer function $\hat{G}_c(z)$ we substitute s with the chosen approximation (for example Tustin):

$$\hat{G}_c(z) = G_c\left(s = \frac{2}{T}\frac{z-1}{z+1}\right).$$

3.3 ZOH Discretization (Direct Design)

3.3.1 Discretization of a Transfer Function

In the direct design the plant's continuous time model $G_p(s)$ is discretized. If we assume that the DAC is a ZOH, when the input is a discrete Dirac impulse in continuous time it becomes a rectangle of length T:

$$u(t) = \varepsilon(t) - \varepsilon(t-T) \rightsquigarrow U(s) = \frac{1 - e^{-sT}}{s}.$$

Then we compute the output of the system to this input in continuous time $U(s)G_p(s)$, inverse Laplace transform it, and then sample it in the time domain. By taking the z transform of the result we obtain a discretized transfer function

$$\hat{G}_p(z) = \left(1 + z^{-1}\right) \mathcal{Z}\left\{\left(\mathcal{L}^{-1}\left\{\frac{G(s)}{s}\right\}\right)\Big|_{t=kT}\right\}.$$

ZOH discretizations for a few common type of plants are listed in table 1.

Plant $G_p(s)$	ZOH $\hat{G}_p(z)$	Plant $G_p(s)$	ZOH $\hat{G}_p(z)$
K	$\dfrac{Kz}{z-1}$	$\dfrac{1}{s}$	$\dfrac{T}{z-1}$
$\dfrac{1}{s^2}$	$\dfrac{T}{2}\dfrac{z+1}{(z-1)^2}$	$\dfrac{K}{s\tau+1}$	$\dfrac{1-e^{-T/\tau}}{z-e^{T/\tau}}$

Table 1: ZOH discretization of a few plant types.

3.3.2 Discretization of a State Space Model

[TODO: Stuff from skript p. 250.]

3.3.3 Root Locus Method

[TODO: Stuff from skript p. 254.]

3.3.4 Pole Placement Design

Another way to design a controller is to simply state that the closed $G_f(z)$ has to have the poles at some determined

locations. Then since in a closed feedback loop with a plant $G_p(z)$ and a controller $G_c(z)$:

$$G_f(z) = \frac{G_c(z)G_p(z)}{1 + G_p(z)G_c(z)},$$

by solving for $G_c(z)$:

$$G_c(z) = \frac{1}{G_p(z)}\frac{G_f(z)}{1 - G_f(z)}.$$

3.4 Discrete State Space

The discrete state space model equations are

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k),$$
$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k),$$

and similarly to the Laplace domain solution there is a *transfer matrix* in the z-domain:

$$\mathbf{G}(z) = \mathbf{C}(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}.$$

Like in continuous time the eigenvalues of the $\mathbf{A}$ matrix are the poles of the transfer function $G(z)$, and thus determine the stability of the system. The observability and controllability can be checked with the same matrices as in continuous time: $\mathbf{Q}_o$, $\mathbf{Q}_c$ respectively. Because the system is a *difference* equation (instead of *differential* equation) future states can be computed recursively:

$$\mathbf{x}(x) = \mathbf{A}\mathbf{x}(n-1) + \mathbf{B}\mathbf{u}(n-1)$$
$$= \mathbf{A}^n\mathbf{x}(0) + \sum_{k}^{n-1}\mathbf{A}^k\mathbf{B}\mathbf{u}(n-1-k).$$

3.5 Deadbeat Controller Design

[TODO: Stuff from skript p. 256.]

A Linear Algebra

A.1 Determinant

The determinant is a measure of how the unit square ($n = 2$), cube ($n = 3$) or hypercube ($n > 3$) is scaled by the linear transformation. The determinant of a 2×2 matrix is computed using the "fish rule":

$$\det\begin{bmatrix} a & b \\ c & d \end{bmatrix} = ad - bc.$$

By hand, for a 3×3 matrix, Sarrus' rule is probably the fastest method:

$$\det\underbrace{\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}}_{\mathbf{A}} \rightsquigarrow \begin{bmatrix} a & b & c & a & b \\ d & e & f & d \\ g & h & i & g & h \end{bmatrix},$$

the red lines are added, while the blue ones are subtracted:

$$\det\mathbf{A} = aei + bfg + cdh - ceg - afh - bdi.$$

A.2 Inverse Matrix

The inverse of a 2×2 matrix is

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix},$$

I remember it as "transposed on the wrong diagonal and add minuses (and divide by the determinant)". For bigger matrices, there are the method of Gaussian elimination:

$$\left[\mathbf{A}\,|\,\mathbf{I}\right] \xrightarrow{\text{Gauss}} \left[\mathbf{I}\,|\,\mathbf{A}^{-1}\right],$$

or Cramers' rule $\mathbf{A}^{-1} = \operatorname{adj}\mathbf{A}/\det\mathbf{A}$.

A.3 Eigenvalues and Eigenvectors

The eigenvalue problem is to find the vectors that are unaffected by a matrix A (up to a scalar factor):

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v} \iff (\mathbf{A} - \lambda\mathbf{I})\mathbf{v} = \mathbf{0}.$$

There are n numbers λ, the *eigenvalues*, and equally many *eigenvectors* $\mathbf{v}$. This problem is solved by first finding the λ's and then later the eigenvectors.

In the right expression, we need a matrix $\mathbf{A} - \lambda\mathbf{I}$ to move any vector to the origin, this is accomplished by forcing

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0,$$

and solving for the λ's. Then to find the eigenvectors we put back each eigenvalue λ_i into the original equation to obtain a system of equations

$$(\mathbf{A} - \lambda_i\mathbf{I})\mathbf{v}_i = \mathbf{0}.$$

This system of equations is solved for the components of $\mathbf{v}_i$, and it is overdetermined. This is because there are infinitely many eigenvectors each eigenspace of $\mathbf{A}$.

For a 2×2 matrix the eigenvalue problem is a parabola, so there is a shortcut:

$$m = \frac{1}{2}\operatorname{tr}\mathbf{A}, \qquad\qquad p = \det\mathbf{A},$$

then $\lambda_{1,2} = m \pm \sqrt{m^2 - p}$.

A.4 Diagonalization

If a $n \times n$ matrix $\mathbf{A}$ has n distinct eigenvectors (linearly independent), it is *diagonalizable*. To diagonalize $\mathbf{A}$, first we compute the eigenvalues $\lambda_1, \ldots, \lambda_n$ and the eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$, then a matrix is formed by using the eigenvectors as columns:

$$\mathbf{T} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \ldots & \mathbf{v}_n \end{bmatrix}.$$

This matrix is then used for a change of bases into the eigenbasis of $\mathbf{A}$, where it is a diagonal matrix:

$$\mathbf{\Lambda} = \mathbf{T}\mathbf{A}\mathbf{T}^{-1} = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}.$$

A.5 Positive Definite

A Hermitian complex (or real symmetric) matrix $\mathbf{A}$ is said to be *positive definite* if the quadratic form $\mathbf{x}^* \mathbf{A} \mathbf{x} > 0$ for all $\mathbf{x} \neq \mathbf{0}$. If $\mathbf{x}^* \mathbf{A} \mathbf{x} \leq 0$ then it is *semi-definite*. If the quadratic form is smaller than 0 it is *negative definite* or (semi-definite). Matrix definiteness can be controlled by using the eigenvalues λ_i because:

- if all $\lambda_i > 0$, then the matrix is positive definite,

- if all $\lambda_i \geq 0$ the matrix is positive semi-definite.

And the converse for negative definiteness. Further, definiteness can be checked using the determinants of the submatrices $\mathbf{A}_{[ij,kl]}$ as follows: given a matrix

$$
\mathbf{A} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix},
$$

If $\det \mathbf{A}_{[11,11]} > 0$ (red), $\det \mathbf{A}_{[11,22]} > 0$ (orange), $\det \mathbf{A}_{[11,33]} > 0$ (yellow), etc. the matrix is positive definite.

The quadratic form $\mathbf{x}^* \mathbf{Q} \mathbf{x}$ for $\mathbf{x} \in \mathbb{C}^{2\times 1}$ a, b, q as complex numbers is

$$
\mathbf{x}^* \underbrace{\begin{bmatrix} a & q \\ q & b \end{bmatrix}}_{\mathbf{Q}} \mathbf{x} = a|x_1|^2 + q(x_1^* x_2 + x_1 x_2^*) + b|x_2|^2.
$$

If working only with real numbers:

$$
\mathbf{x}^\mathsf{T} \mathbf{Q} \mathbf{x} = a x_1^2 + 2q x_1 x_2 + b x_2^2 = \sum_{i,j}^{n^2} Q_{ij}\, x_i x_j.
$$

A.6 Jordan Normal Form

[TODO: From Axler.]

A.7 Caley-Hamilton Theorem

[TODO: From Ogata.]

A.8 Useful Matrices and Products

Column Row Product For two complex numbers a and b:

$$
\mathbf{x} = \begin{bmatrix} a \\ b \end{bmatrix} \quad \leadsto \quad \mathbf{x}\mathbf{x}^* = \begin{bmatrix} |a|^2 & ab^* \\ a^*b & |b|^2 \end{bmatrix}.
$$

Rotation Matrices In 2 dimensions:

$$
\mathbf{R}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \quad \mathbf{R}(-\theta) = \mathbf{R}(\theta)^\mathsf{T}.
$$

Rotation matrices belong to the special orthogonal group, therefore

$$
\mathbf{R}\mathbf{R}^\mathsf{T} = \mathbf{I} \iff \mathbf{R}^{-1} = \mathbf{R}^\mathsf{T} \text{ and } \det \mathbf{R} = 1.
$$

B Vector Analysis

B.1 Gradient Vector

The *gradient* of a function $f(\mathbf{x}), \mathbf{x} \in \mathbb{R}^m$ is a column vector containing the partial derivatives in each direction.

$$
\boldsymbol{\nabla} f(\mathbf{x}) = \sum_{i=1}^{m} \partial_{x_i} f(\mathbf{x}) \mathbf{e}_i = \begin{pmatrix} \partial_{x_1} f(\mathbf{x}) \\ \vdots \\ \partial_{x_m} f(\mathbf{x}) \end{pmatrix}
$$

The gradient vector always points towards *the direction of steepest ascent*, and thus is always perpendicular to contour lines.

B.2 Jacobian Matrix

The *Jacobian* $\mathbf{J}_f$ (sometimes written as $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}$) of a function $\mathbf{f} : \mathbb{R}^m \to \mathbb{R}^n$ is a matrix $\in \mathbb{R}^{m \times n}$ whose entry at the i-th row and j-th column is given by $(\mathbf{J}_f)_{i,j} = \partial_{x_j} f_i$, so

$$
\mathbf{J}_f = \begin{pmatrix} \partial_{x_1} f_1 & \cdots & \partial_{x_m} f_1 \\ \vdots & \ddots & \vdots \\ \partial_{x_1} f_n & \cdots & \partial_{x_m} f_n \end{pmatrix} = \begin{pmatrix} (\boldsymbol{\nabla} f_1)^t \\ \vdots \\ (\boldsymbol{\nabla} f_m)^t \end{pmatrix}
$$

C Laplace Transform

The Laplace transform of a function f is

$$
F(s) = \mathcal{L} f(t) = \int_0^\infty f(T)\, e^{-st}\, dt,
$$

and in the other direction the inverse Laplace transform is

$$
f(t) = \mathcal{L}^{-1} F(s) = \frac{1}{2\pi i} \lim_{T \to \infty} \oint_{\gamma - iT}^{\gamma + iT} F(s)\, e^{st}\, ds.
$$

The Laplace transform converts integration and differentiation problems into algebraic problems. Some important properties and transform pairs are listed in table 2.

D z-Transform

D.1 Definition

The z-transform of a sequence (f_n) is defined to be

$$
F(z) = \sum_{k=0}^\infty f_k z^{-k},
$$

where z is a complex number. The inverse transform is

$$
f_n = \frac{1}{2\pi i} \oint_C F(z) z^{n-1}\, dz,
$$

where C is a counterclockwise path encircling the origin and entirely in the region of convergence (ROC).

Property	Time Domain	s-Domain		Function	Laplace Tr.
Linearity	$\sum_i a_i f_i(t)$	$\sum_i a_i F_i(s)$		$\delta(t)$	1
Time shift $(\tau > 0)$	$f(t - \tau)$	$e^{-s\tau} F(s)$		$\varepsilon(t)$	$\dfrac{1}{s}$
Damping	$e^{at} f(t)$	$F(s - a)$		t^n	$\dfrac{n!}{s^{n+1}}$
Stretching $(a > 0)$	$f(at)$	$\dfrac{1}{a} F\left(\dfrac{s}{a}\right)$		$\cos(\omega t)$	$\dfrac{s}{s^2 + \omega^2}$
Differentiation	$f^{(n)}(t)$	$s^n F(s) - s^{n-1} f(0^+) - \ldots - f^{(n-1)}(0^+)$		$\sin(\omega t)$	$\dfrac{\omega}{s^2 + \omega^2}$
Integration	$\displaystyle\int_0^t f(t)\,dt$	$\dfrac{1}{s} F(s)$			

Table 2: Useful Laplace transform rules and pairs. All functions are assumed to be causal (multiplied with the unit step $\varepsilon(t)$).

D.2 Relation to the Laplace Transform

The z-transform is related to the Laplace transform, when a sequence is generated by sampling a function. In other words with a sampling time T we can construct a continuous time function

$$f(t) = \sum_k f_k \delta(t - kT)$$

of which is then possible to take the Laplace transform resulting in

$$\mathcal{L}\left\{\sum_k f_n \delta(t - kT)\right\} = \sum_{k=0}^{\infty} f_k e^{-kTs} = \sum_{k=0}^{\infty} f_k z^{-k},$$

where in the last step the substitution $z = e^{sT}$ was made. Therefore the multiplication by z^{-1} can interpreted as adding a delay of T.

E Idiot's section
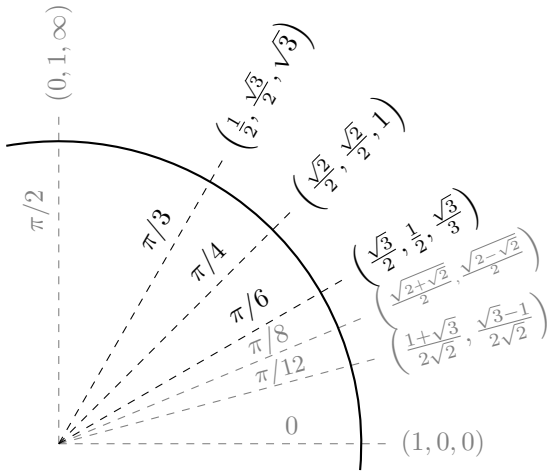
A few useful trigonometric identities:

$$\cos^2(x) + \sin^2(x) = 1 \quad \cosh^2(x) - \sinh^2(x) = 1$$

$$
\begin{aligned}
\cos(\alpha + 2\pi) &= \cos(\alpha) & \sin(\alpha + 2\pi) &= \sin(\alpha) \\
\cos(-\alpha) &= \cos(\alpha) & \sin(-\alpha) &= -\sin(\alpha) \\
\cos(\pi - \alpha) &= -\cos(\alpha) & \sin(\pi - \alpha) &= \sin(\alpha) \\
\cos(\tfrac{\pi}{2} - \alpha) &= \sin(\alpha) & \sin(\tfrac{\pi}{2} - \alpha) &= \cos(\alpha)
\end{aligned}
$$

$$
\begin{aligned}
\cos(\alpha + \beta) &= \cos\alpha \cos\beta - \sin\alpha \sin\beta \\
\sin(\alpha + \beta) &= \sin\alpha \cos\beta - \cos\alpha \sin\beta
\end{aligned}
$$

$$
\begin{aligned}
\cos(2\alpha) &= \cos^2\alpha - \sin^2\alpha \\
&= 1 - 2\sin^2\alpha \\
&= 2\cos^2\alpha - 1 \\
\sin(2\alpha) &= 2\sin\alpha\cos\alpha \\
\tan(2\alpha) &= (2\tan\alpha)(1 + \tan^2\alpha)^{-1}
\end{aligned}
$$

The geometric series has a closed form formula for $|q| < 1$:

$$\sum_{k=0}^{\infty} q^k = \frac{1}{1 - q}.$$

Rules for differentiation, f, u, v are differentiable functions of x:

$$
\begin{aligned}
(af)' &= af' & (u(v))' &= u'(v)v' \\
(uv)' &= u'v + uv' & \left(\frac{u}{v}\right)' &= \frac{u'v - uv'}{v^2} \\
\left(\sum u_i\right)' &= \sum u_i' & (\ln u)' &= \frac{u'}{u} \\
(f^{-1})' &= \frac{1}{f'(f^{-1}(x))}
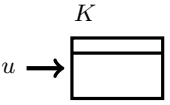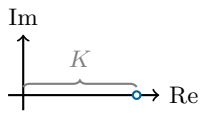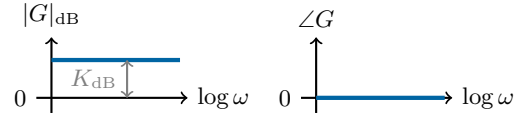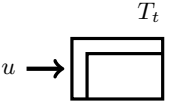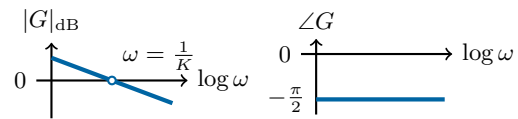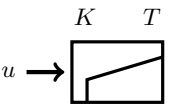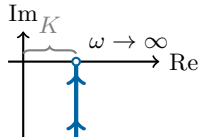\end{aligned}
$$
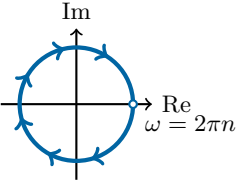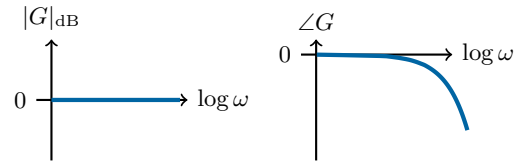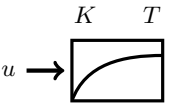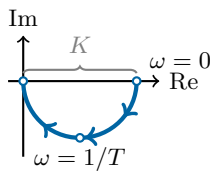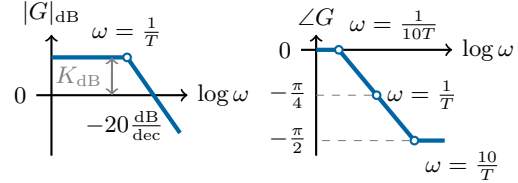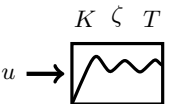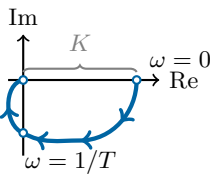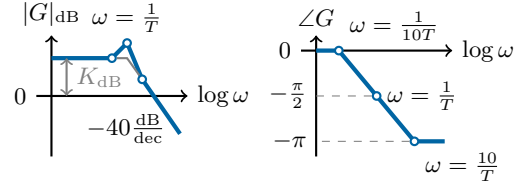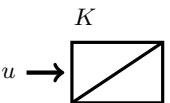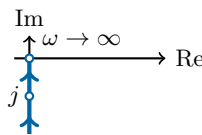
Some useful derivatives and integrals:
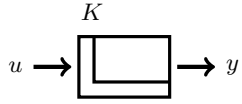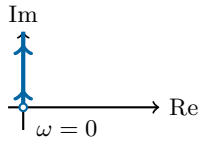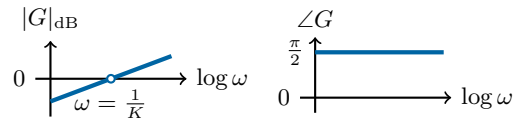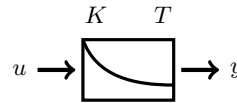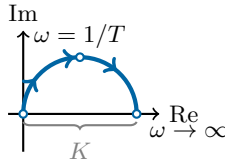
Property	Time Domain	z-Domain	Sequence	z-Tr.
Linearity	$\sum_i a_i f_k(k)$	$\sum_i a_i F_i(z)$	$\delta(k)$	1
Forward time shift	$f(k+n)$	$z^n F(z) - z^n f(0) - \cdots - z f(n-1)$	$\varepsilon(k)$	$\dfrac{z}{z-1}$
Backwards time shift	$f(k-n)$	$z^{-n} F(z)$	k	$\dfrac{z}{(z-1)^2}$
Convolution	$f_1 * f_2 = \sum_{i=0}^{k} f_1(k) f_2(k-i)$	$F_1(z) F_2(z)$	a^k	$\dfrac{z}{z-a}$

Table 3: Useful rules and transform pairs of the z-transform. All sequences can be converted into time-domain functions by using a Dirac comb. Some are easier for example $k \rightsquigarrow kT$.

f	f'	f	f'
x^n	$n x^{n-1}$	a^x	$a^x \ln a$
$\sqrt[n]{x}$	$1/\left(x^n \sqrt[n]{x^{n-1}}\right)$	$\ln x$	$1/x$
$\sin x$	$\cos x$	$\cos x$	$-\sin x$
$\tan x$	$1/\cos^2 x$	$1/\tan x$	$-1/\sin^2 x$
$\arcsin x$	$1/\sqrt{1-x^2}$	$\arccos x$	$-1/\sqrt{1-x^2}$
$\arctan x$	$1/\left(1+x^2\right)$		
$\sinh x$	$\cosh x$	$\tanh x$	$1/\cosh^2 x$
$\operatorname{arcsinh} x$	$1/\sqrt{1+x^2}$	$\operatorname{arccosh} x$	$1/\sqrt{x^2-1}$

F   Block Diagram Elements

Type	Symbol	Differential Equation	Frequency Response	Nyquist Diagram	Bode Plot
P		$y = Ku$	K		
T		$y(t) = \varepsilon(t)u(t - T_t)$	$e^{-j\omega T_t}$		
PT$_1$		$T\dot{y} + y = Ku$	$\dfrac{K}{j\omega T + 1}$		
PT$_2$		$T^2\ddot{y} + 2\zeta T\dot{y} + y = Ku$	$\dfrac{K}{T^2(j\omega)^2 + 2\zeta T(j\omega) + 1}$		
I		$\dot{y} = Ku$	$\dfrac{K}{j\omega}$		
PI		$y = K\left(u + \displaystyle\int_0^t \dfrac{u\, d\tau}{T}\right)$	$K\left(1 + \dfrac{1}{j\omega T}\right)$		

Type	Symbol	Differential Equation	Frequency Response	Nyquist Diagram	Bode Plot
D	$u \rightarrow \boxed{K} \rightarrow y$ Not realizable!	$y = K\dot{u}$	$j\omega K$	(Im/Re plot, $\omega = 0$)	$\|G\|_{\mathrm{dB}}$, $\omega = \frac{1}{K}$; $\angle G = \frac{\pi}{2}$
DT_1	$u \rightarrow \boxed{K\ T} \rightarrow y$	$T\dot{y} + y = K\dot{u}$	$\dfrac{j\omega K}{1 + j\omega T}$	(Im/Re plot, $\omega = 1/T$, $\omega \to \infty$, K)	

F.1 PT_2–Element

$$\omega = 1/T, \quad \omega_n = 1/T_n = \omega\sqrt{1-\zeta^2}$$
$$T_a = (\pi - \arccos\zeta)/\omega_n$$
$$T_m = T_n/2 = \pi/\omega_n$$
$$y_m = y_\infty e^{-\pi\zeta/\sqrt{1-\zeta^2}}$$
$$K\left[1 - e^{-\zeta t/T}\left(\cos(\omega_n t) + \zeta/\sqrt{1-\zeta^2}\sin(\omega_n t)\right)\right]$$

G Lead-Lag Element

$$K\frac{1+j\omega T_1}{1+j\omega T_2} = \begin{cases} \text{Lead} & T_1 > T_2 \\ \text{Lag} & T_1 < T_2 \end{cases}$$

H `MATLAB` Reference

Command	Description

`G = tf(N,D);`
 `G = tf(N,D,"InputDelay",T);`
 `G = tf(N,D,TS);`
	Create a continous time a transfer function with denominator `N` and denominator `D` (both arrays). Add a delay of e^{-sT}. Create a discrete time transfer function with sampling time `TS`.

`SYS = ss(A,B,C,D);`
 `SYS = ss(A,B,C,D,"InputDelay",T);`
 `SYS = ss(A,B,C,D,TS);`
	Create a continuous time state space model from the matrices `A B C` and `D`. Add a delay of e^{-sT}. Create a discrete time model with sampling time `TS`.

`CO = ctrb(A,B);` `CO = ctrb(SYS);`	Compute the controllability matrix of a system from the matrices `A` and `B` or from the system object.
`OB = obsv(A,C);` `OB = obsv(SYS);`	Compute the observability matrix of a system from the matrices `A` and `C` or from the system object.
`SYS = tf2ss(G);` `G = ss2tf(SYS);`	Conversion between state space and transfer function.
`K = dcgain(SYS);` `K = dcgain(G);`	Compute the DC gain of a system.
`K = place(A,B,P);`	Compute a state feedback matrix such that the eigenvalues of $\mathbf{A} - \mathbf{BK}$ are those specified in the vector `P`.

`[K,S,CLP] = lqr(SYS,Q,R,N);`
 `[K,S,CLP] = lqr(A,B,C,D,Q,R,N);`
 `[K,S,CLP] = dlqr(A,B,Q,R,N)`
	Calculates the optimal gain matrix `K` for the continuous or discrete state-space model `SYS`.

`[N,D] = pade(T,N);` `SYSX = pade(SYS,N);`	Returns a Nth order Padé approximation of the continous time delay e^{-sT}. Returns delay free approximation of a system `SYS` with a delay.