Hochschule Ravensburg-Weingarten
University of Applied Sciences

Embedded Control

Sliding-Mode Control for Nonlinear RC and RLC Circuits

Author:	*Student Number:*
Christian Dalea	34808
Simon Scheerer	34749

February 5, 2023

Contents

1 Introduction to sliding mode control

Sliding mode control (SMC) is a powerful, robust, and computing-wise easy-to-implement control method. Equivalent to the proportional integral derivative controller (PID controller), once the control parameters are defined, the controller can be easily implemented on a microcontroller. Moreover, due to its robustness, it can handle disturbances or inaccuracies in either the system model or other parameters to a certain degree. It is commonly used to control nonlinear systems. However, it can be used for linear control tasks as well [2].

Compared to other nonlinear control methods, such as model predictive control (MPC), the advantage of sliding mode control is its less complexity at run time. The sliding mode control parameters have to be defined only once to yield a constant controller structure, whereas MPC has an optimization algorithm that changes the control parameters during the runtime. Consequently, MPC needs more computational resources, which means more expensive controllers. Therefore sliding mode control is commonly used, in several applications like the aerospace industry [9], electric drives [4], and automotive systems [5].

1.1 Basic idea

The mathematical concept of SMC is based on using a sliding manifold, a mathematical function that represents the desired system behavior. To force the system to the desired behavior, SMC applies a control signal to the system such that the system will slide along the so-called sliding surface or more generally, sliding manifold. The sliding manifold is, in the case of a three-dimensional system, a two-dimensional surface, and in the case of a two-dimensional system a one-dimensional line. Generally, the order n of the sliding manifold can be calculated by the systems order m reduced by one ($n = m-1$). The control input is calculated by the usage of a switching function. Depending on the position of the system with respect to the sliding surface, the ultimate trajectory will not exist entirely with one control structure. Therefore, it will slide along the boundaries of the control structures [7].

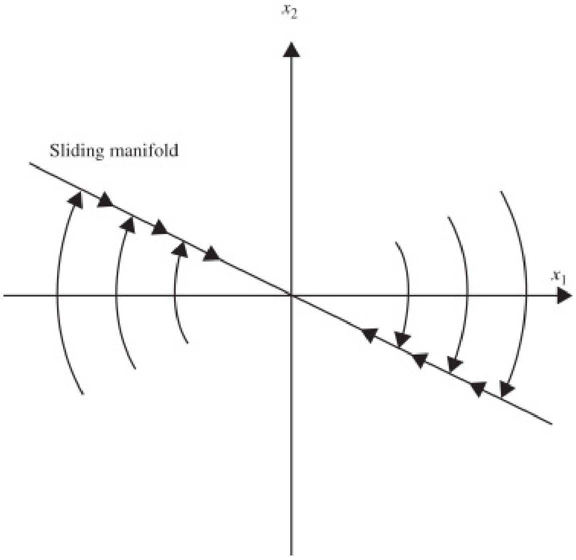


Figure 1: Sliding manifold of a 2^{nd} order system. [6]

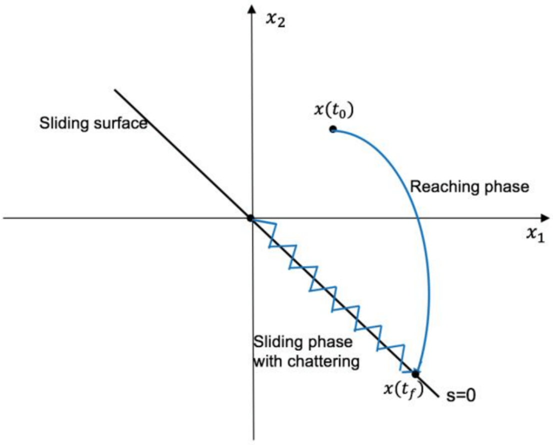


Figure 2: Sliding phase with chattering [8]

The pictures above show two different phase plots of a second order system that visualize the basic concept of SMC. In a second order system the sliding manifold is a line. Based on the starting condition $x(t_0)$, the trajectories will start on different starting positions within the phase plot (Figure 1). The first stage in SMC is the reaching phase, where the trajectory moves toward the sliding surface $s(\boldsymbol{x}) = 0$. As soon as it has reached this surface, it will slide along the sliding surfaces until it reaches its stable position in origin. Since the trajectory does not always hit the sliding surface perfectly, the trajectory chatters along the sliding surface (Figure 2).

1.2 General design rules for SMC

The subspace that fulfills the condition $s(\boldsymbol{x}) = 0$ is called the sliding surface. When the system reaches this area, the system is asymptotically stable. To reach this subspace the following conditions are considered. If $s(\boldsymbol{x}) > 0$ than its derivative has to be $\dot{s}(\boldsymbol{x}) < 0$. On the other hand if $s(\boldsymbol{x}) < 0$ than the derivative has to be $\dot{s}(\boldsymbol{x}) > 0$. These two conditions will ensure that the system will be attracted to move toward the desired system behavior $s(\boldsymbol{x}) = 0$ (see Figure 3). Derived from that, consequently, the following condition holds: $s(\boldsymbol{x})\dot{s}(\boldsymbol{x}) < 0$.[1]
Hence this condition is necessary, it does not ensure that the system will reach these states in a finite time. The reachability in a finite time is already proven for example in [7], therefore no further prove will be done again.

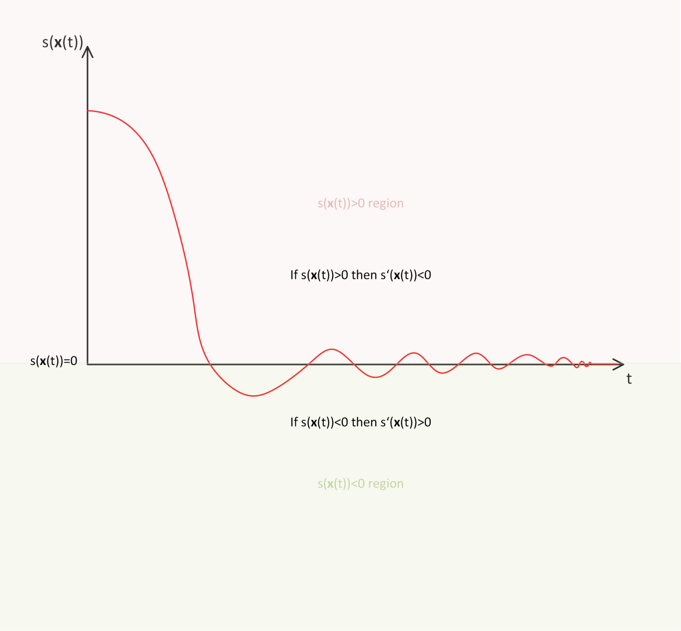

s(**x**(t))
s(**x**(t))>0 region
If s(**x**(t))>0 then s‘(**x**(t))<0
s(**x**(t))=0
t
If s(**x**(t))<0 then s‘(**x**(t))>0
s(**x**(t))<0 region

Figure 3: Sliding surface over time

The formulas to design the sliding mode control are introduced next. In this report the linear sliding surface

$$s(\boldsymbol{x}) = \boldsymbol{r}^T\boldsymbol{x} = r_1x_1 + r_2x_2 + ... + r_nx_n \tag{1.2.1}$$

with $r_n = 1$ is used.

The derivative of $s(\boldsymbol{x})$ is given in the following equation

$$\begin{aligned}\dot{s}(\boldsymbol{x}) = \boldsymbol{r}^T\dot{\boldsymbol{x}} &= grad^T s(\boldsymbol{x})\dot{\boldsymbol{x}},\\ \dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}u,\\ \dot{s}(\boldsymbol{x}) &= grad^T s(\boldsymbol{x})(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}u). \qquad (1.2.2)\end{aligned}$$

Based on the above mentioned design rules for SMC, the following condition holds

$$\dot{s}(\boldsymbol{x}) = \begin{cases} > 0 & \text{if } s(\boldsymbol{x}) < 0,\\ < 0 & \text{if } s(\boldsymbol{x}) > 0. \end{cases} \qquad (1.2.3)$$

From equation (1.2.3), the inequality

$$s(\boldsymbol{x})\dot{s}(\boldsymbol{x}) < 0 \qquad (1.2.4)$$

is obtained.

When does sliding occur?

Based on condition (1.2.3), it is evident in which direction the system should move. To investigate if this condition is fulfilled, the gradient will be used. By applying the scalar product $\nabla s(\boldsymbol{x}) \cdot f$, it can be calculated in which direction the vectors f_1 and f_2 are pointing. If the scalar product is positive, both vectors point in the same direction. Otherwise, the vectors point in the opposite direction. Finally, condition (1.2.3) can be proven, and sliding occurs, if the following condition will be respected [3]

$$\frac{\partial s(\boldsymbol{x})}{\partial \boldsymbol{x}} f_1 = L_{f_1} s(\boldsymbol{x}) < 0 \text{ and } \frac{\partial s(\boldsymbol{x})}{\partial \boldsymbol{x}} f_2 = L_{f_2} s(\boldsymbol{x}) > 0.$$

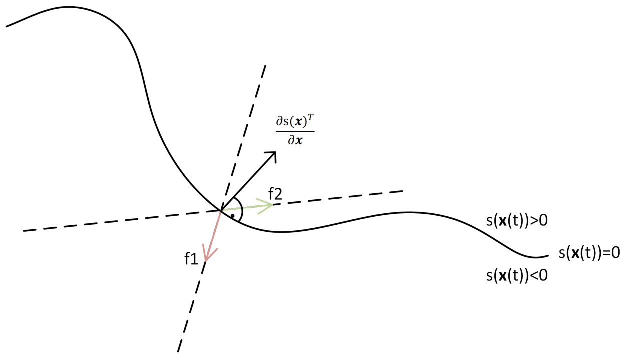

$\frac{\partial s(\boldsymbol{x})^T}{\partial \boldsymbol{x}}$
f2
f1
s(**x**(t))>0
s(**x**(t))=0
s(**x**(t))<0

Figure 4: Check when sliding occurs. f_1 is the system behavior if $s(\boldsymbol{x}) > 0$, f_2 is the system behavior if $s(\boldsymbol{x}) < 0$ and $s(\boldsymbol{x}) = 0$ is the sliding manifold.[3]

Controllable canonical form

The controller design for linear and nonlinear systems will be described in the next chapters. An essential condition for both cases (linear and nonlinear) is that the system has to be noted in controllable canonical form

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{pmatrix} = \begin{bmatrix} 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \\ -a_0 & -a_1 & \cdots & -a_{n-2} & -a_{n-1} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} u = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}u,$$

for linear system, and

$$\begin{pmatrix} \dot{z}_1 \\ \vdots \\ \dot{z}_{n-1} \\ \dot{z}_n \end{pmatrix} = \begin{bmatrix} z_2 \\ \vdots \\ z_n \\ \boldsymbol{L}_{\boldsymbol{a}}^n c(\boldsymbol{x}) + \boldsymbol{L}_{\boldsymbol{b}} \boldsymbol{L}_{\boldsymbol{a}}^{n-1} c(\boldsymbol{x}) u \end{bmatrix} = \begin{pmatrix} z_2 \\ \vdots \\ z_n \\ \alpha(\boldsymbol{z}) \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \beta(\boldsymbol{z}) \end{pmatrix} u = \tilde{\boldsymbol{a}}(\boldsymbol{z}) + \tilde{\boldsymbol{b}}(\boldsymbol{z}) u$$

for nonlinear system.

If the system is not already in this form, the state space model has to be transformed into the desired "controllable canonical form“. Therefore, it could be necessary to introduce new variables, such as in a cascaded RC circuit (see chapter 2.1.1).

Furthermore, in a nonlinear system, the new variables will be calculated with the help of the Lie derivative (see chapter 2.2.1). To better understand how this can be done, in chapter 2, several calculation examples for a second-order system will be shown.

1.3 Control methods

Several control methods with different properties will satisfy the general design rules (chapter 1.2) for SMC. In this report, three different methods were investigated. All these three methods use the approach to describe $\dot{s}(\boldsymbol{x})$ by a particular function.

1.3.1 Reaching law with constant rate

For this method, the following ansatz will be used [6]:

$$\dot{s}(\boldsymbol{x}) = -\varepsilon sign(s(\boldsymbol{x})) \text{ with control parameter } \varepsilon > 0 \tag{1.3.1}$$

$$\dot{s}(\boldsymbol{x}) = \begin{cases} -\varepsilon sign(s_+(\boldsymbol{x})) = -\varepsilon < 0 & \text{if } s(\boldsymbol{x}) > 0 \\ -\varepsilon sign(s_-(\boldsymbol{x})) = +\varepsilon > 0 & \text{if } s(\boldsymbol{x}) < 0 \end{cases} \tag{1.3.2}$$

With proof 1.3.2, it can be shown formula 1.3.1 fulfills the conditions 1.2.3 and 1.2.4.

1.3.2 Exponential reaching law by Gao and Hung

For this method, the following ansatz will be used [6]:

$$\dot{s}(\boldsymbol{x}) = -\varepsilon sign(s(\boldsymbol{x})) - ks(\boldsymbol{x}) \text{ with control parameters } \varepsilon > 0 \text{ and } k > 0 \tag{1.3.3}$$

$$\dot{s}(\boldsymbol{x}) = \begin{cases} -\varepsilon sign(s_+(\boldsymbol{x})) - ks_+(\boldsymbol{x}) = -\varepsilon - ks_+(\boldsymbol{x}) < 0 & \text{if } s(\boldsymbol{x}) > 0 \\ -\varepsilon sign(s_-(\boldsymbol{x})) - ks_-(\boldsymbol{x}) = +\varepsilon - ks_-(\boldsymbol{x}) > 0 & \text{if } s(\boldsymbol{x}) < 0 \end{cases} \tag{1.3.4}$$

With proof 1.3.4, it can be shown formula 1.3.3 fulfills the conditions 1.2.3 and 1.2.4.

1.3.3 Reaching law with power rate

For this method, the following ansatz will be used [6]:

$$\dot{s}(\boldsymbol{x}) = -k|s(\boldsymbol{x})|^{\alpha} sign(s(\boldsymbol{x})) \text{ with control parameters } k > 0 \text{ and } 0 < \alpha < 1 \tag{1.3.5}$$

$$\dot{s}(\boldsymbol{x}) = \begin{cases} -k|s_+(\boldsymbol{x})|^{\alpha} sign(s_+(\boldsymbol{x})) = -k|s_+(\boldsymbol{x})|^{\alpha} < 0 & \text{if } s(\boldsymbol{x}) > 0 \\ -k|s_-(\boldsymbol{x})|^{\alpha} sign(s_-(\boldsymbol{x})) = +k|s_-(\boldsymbol{x})|^{\alpha} > 0 & \text{if } s(\boldsymbol{x}) < 0 \end{cases} \tag{1.3.6}$$

With proof 1.3.6 it can be shown formula 1.3.5 fullfills the conditions 1.2.3 and 1.2.4.

1.4 Linear controller calculation

Derived from the definition of the sliding surface (equation (1.2.1)) and its derivative equation (1.2.2). It is possible to calculate the corresponding controller $u(t)$ for each control method individually. The following calculations consider a linear system which can be described as $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}u$.

1.4.1 Linear controller for reaching law with constant rate

General equation

$$u(\boldsymbol{x}) = -\frac{\boldsymbol{r}^T\boldsymbol{A}\boldsymbol{x} + \varepsilon sign(s(\boldsymbol{x}))}{\boldsymbol{r}^T\boldsymbol{b}} \tag{1.4.1}$$

Special case 2^{nd} order

The matrix $\boldsymbol{A}$, vector $\boldsymbol{b}$, vector $\boldsymbol{x}$ and the vector $\boldsymbol{r}^T$ will be defined as

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ a_0 & a_1 \end{bmatrix}, \tag{1.4.2}$$

$$\boldsymbol{b} = \begin{pmatrix} 0 \\ b_1 \end{pmatrix}, \tag{1.4.3}$$

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \tag{1.4.4}$$

$$\boldsymbol{r}^T = (r_1, 1). \tag{1.4.5}$$

With these definitions the equation (1.4.1) can be simplified as

$$\boxed{u(\boldsymbol{x}) = \frac{-x_1a_0 - x_2(r_1 + a_1) - \varepsilon sign(r_1x_1 + x_2)}{b_1}.}$$

1.4.2 Linear controller for reaching law with exponential rate

General equation

$$u(\boldsymbol{x}) = -\frac{\boldsymbol{r}^T\boldsymbol{A}\boldsymbol{x} + \varepsilon sign(s(\boldsymbol{x})) + ks(\boldsymbol{x})}{\boldsymbol{r}^T\boldsymbol{b}} \tag{1.4.6}$$

Special case 2^{nd} order

With the definitions 1.4.2 to 1.4.5 the equation (1.4.6) can be simplified as

$$\boxed{u(\boldsymbol{x}) = \frac{-x_1a_0 - x_2(r_1 + a_1) - \varepsilon sign(r_1x_1 + x_2) - k(r_1x_1 + x_2)}{b_1}.}$$

1.4.3 Linear controller for reaching law with power rate

General equation

$$u(\boldsymbol{x}) = -\frac{\boldsymbol{r}^T\boldsymbol{A}\boldsymbol{x} + k|s(\boldsymbol{x})|^\alpha sign(s(\boldsymbol{x}))}{\boldsymbol{r}^T\boldsymbol{b}} \tag{1.4.7}$$

Special case 2^{nd} order

With the definitions 1.4.2 to 1.4.5 the equation (1.4.7) can be simplified as

$$\boxed{u(\boldsymbol{x}) = \frac{-x_1a_0 - x_2(r_1 + a_1) - k|r_1x_1 + x_2|^\alpha sign(r_1x_1 + x_2)}{b_1}.}$$

1.5 Nonlinear controller calculation

In the case of a nonlinear system, the starting point to calculate the controller is slightly different because the system can not be represented in the same way as the linear system. A nonlinear system can be represented by $\dot{\boldsymbol{z}} = \tilde{a}(\boldsymbol{z}) + \tilde{b}(\boldsymbol{z})u$. Therefore applies analogously

$$\dot{s}(\boldsymbol{z}) = grad^T s(\boldsymbol{z})\dot{\boldsymbol{z}},$$

$$\dot{s}(\boldsymbol{z}) = grad^T s(\boldsymbol{z})(\tilde{a}(\boldsymbol{z}) + \tilde{b}(\boldsymbol{z})u). \tag{1.5.1}$$

By reshaping equation (1.5.1) to u based on the corresponding control method from section 1.3, it is possible to calculate the controller.

1.5.1 Nonlinear controller for reaching law with constant rate

General equation

$$u(\boldsymbol{z}) = -\frac{\boldsymbol{r}^T\tilde{\boldsymbol{a}}(\boldsymbol{z}) + \varepsilon sign(s(\boldsymbol{z}))}{\boldsymbol{r}^T\tilde{\boldsymbol{b}}(\boldsymbol{z})} \tag{1.5.2}$$

Special case 2^{nd} order

The vector $\tilde{\boldsymbol{a}}(\boldsymbol{z})$, vector $\tilde{\boldsymbol{b}}(\boldsymbol{z})$ and vector $\boldsymbol{z}$ will be defined as

$$\tilde{\boldsymbol{a}}(\boldsymbol{z}) = \begin{pmatrix} z_2 \\ \alpha(\boldsymbol{z}) \end{pmatrix}, \tag{1.5.3}$$

$$\tilde{\boldsymbol{b}}(\boldsymbol{z}) = \begin{pmatrix} 0 \\ \beta(\boldsymbol{z}) \end{pmatrix}, \tag{1.5.4}$$

$$\boldsymbol{z} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}. \tag{1.5.5}$$

With the definitions 1.5.3 to 1.5.5 and definition 1.4.5 the equation (1.5.2) can be simplified as

$$\boxed{u(\boldsymbol{z}) = \frac{-r_1 z_2 - \alpha(\boldsymbol{z}) - \varepsilon sign(r_1 z_1 + z_2)}{\beta(\boldsymbol{z})}.}$$

1.5.2 Nonlinear controller for reaching law with exponential rate

General equation

$$u(\boldsymbol{z}) = -\frac{\boldsymbol{r}^T \tilde{\boldsymbol{a}}(\boldsymbol{z}) + \varepsilon sign(s(\boldsymbol{z})) + k\boldsymbol{r}^T \boldsymbol{z}}{\boldsymbol{r}^T \tilde{\boldsymbol{b}}(\boldsymbol{z})} \tag{1.5.6}$$

Special case 2^{nd} order

With the definitions 1.5.3 to 1.5.5 and definition 1.4.5 the equation (1.5.6) can be simplified as

$$\boxed{u(\boldsymbol{z}) = \frac{-r_1 z_2 - \alpha(\boldsymbol{z}) - \varepsilon sign(r_1 z_1 + z_2) - k(r_1 z_1 + z_2)}{\beta(\boldsymbol{z})}.}$$

1.5.3 Nonlinear controller for reaching law with power rate

General equation

$$u(\boldsymbol{z}) = -\frac{\boldsymbol{r}^T \tilde{\boldsymbol{a}}(\boldsymbol{z}) + \varepsilon sign(s(\boldsymbol{z})) + k\boldsymbol{r}^T \boldsymbol{z}}{\boldsymbol{r}^T \tilde{\boldsymbol{b}}(\boldsymbol{z})} \tag{1.5.7}$$

Special case 2^{nd} order

With the definitions 1.5.3 to 1.5.5 and definition 1.4.5 the equation (1.5.7) can be simplified as:

$$\boxed{u(\boldsymbol{z}) = \frac{-r_1 z_2 - \alpha(\boldsymbol{z}) - k|r_1 z_1 + z_2|^{\alpha} sign(r_1 z_1 + z_2)}{\beta(\boldsymbol{z})}.}$$

2 State space models

For the project, we analyzed two circuits. These are a cascaded RC circuit (Figure 5) and a simple RLC circuit (Figure 6), and both circuits are second-order low-pass filters.

2.1 Linear circuits

2.1.1 Cascaded RC circuit

The following picture shows a cascaded RC circuit.

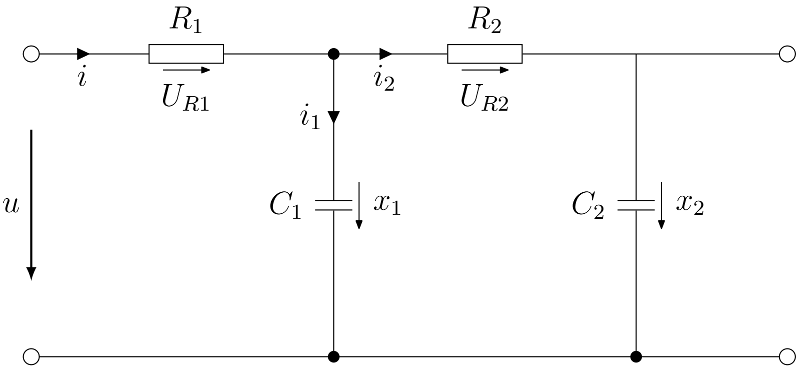


Figure 5: Cascaded RC circuit

The following equations are given from the circuit by Kirchhoff's law

$$i = i_1 + i_2 = \frac{U_{R1}}{R_1},$$

$$x_1 = \frac{1}{C_1} \int i_1 dt, \tag{2.1.1}$$

$$x_2 = \frac{1}{C_2} \int i_2 dt, \tag{2.1.2}$$

$$u = U_{R1} + x_1. \tag{2.1.3}$$

By reshaping equation (2.1.1) and (2.1.2) the following equations for current $i_1 = C_1\dot{x}_1$ and $i_2 = C_2\dot{x}_2$ will sustain.

The equation (2.1.1) can also be expressed as

$$x_1 = U_{R2} + x_2 = R_2 i_2 + x_2 = R_2 C_2 \dot{x}_2 + x_2. \tag{2.1.4}$$

In addition, the equation (2.1.3) can be expanded as

$$u = R_1 i + x_1 = R_1(i_1 + i_2) + x_1 = R_1 C_1 \dot{x}_1 + R_1 C_2 \dot{x}_2 + x_1. \tag{2.1.5}$$

Now, the equations (2.1.4) and (2.1.5) as

$$\begin{aligned} R_1 C_1 \dot{x}_1 + R_1 C_2 \dot{x}_2 &= -x_1 + u, && (2.1.6)\\ R_2 C_2 \dot{x}_2 &= x_1 - x_2. && (2.1.7) \end{aligned}$$

In this case, the equations (2.1.6) and (2.1.7) can be shown in the matrix representation

$$\begin{bmatrix} R_1C_1 & R_1C_2 \\ 0 & R_2C_2 \end{bmatrix} \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{bmatrix} -1 & 0 \\ 1 & -1 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u. \tag{2.1.8}$$

The equation (2.1.8) can be solved for $\dot{x}_1$ and $\dot{x}_2$

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{bmatrix} -\frac{1}{R_1C_1} - \frac{1}{R_2C_1} & \frac{1}{R_2C_1} \\ \frac{1}{R_2C_2} & -\frac{1}{R_2C_2} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} \frac{1}{R_1C_1} \\ 0 \end{pmatrix} u. \tag{2.1.9}$$

To simplify the equation (2.1.9), the parameters will be chosen as: $R_1 = R_2 = R$, and $C_1 = C_2 = C$

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \frac{1}{RC} \begin{bmatrix} -2 & 1 \\ 1 & -1 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \frac{1}{RC} \begin{pmatrix} 1 \\ 0 \end{pmatrix} u. \tag{2.1.10}$$

From equation (2.1.10) the following equations are obtained for $\dot{x}_1$ and $\dot{x}_2$

$$\begin{aligned} \dot{x}_1 &= \frac{1}{RC}(-2x_1 + x_2 + u), \\ \dot{x}_2 &= \frac{1}{RC}(x_1 - x_2). \end{aligned}$$

A substitution must be done for the equation (2.1.10) to be represented in the controllable canonical form, as mentioned in chapter 1.2.

For the substitution, the following variables will be defined

$$\begin{aligned} z_1 &= x_2, \\ \dot{z}_1 &= \dot{x}_2 = z_2, \\ \dot{z}_2 &= \ddot{x}_2. \end{aligned}$$

The variables x_1 and x_2 can be replaced with the following equations

$$\begin{aligned} &z_2 = x_1 - z_1 \rightarrow x_1 = z_1 + z_2, \\ &x_2 = z_1. \end{aligned}$$

For the substitution $\dot{z}_1$ is already known, but $\dot{z}_2$ is not known yet. Therefore, $\ddot{x}_2$ has to be calculated

$$\begin{aligned} \ddot{x}_2 &= \frac{1}{RC}(\dot{x}_1 - \dot{x}_2), \\ \ddot{x}_2 &= \frac{1}{RC}(-3x_1 + 2x_2 + u). \end{aligned} \tag{2.1.11}$$

In formula 2.1.11 x_1 and x_2 can be replaced and than is $\dot{z}_2$ also known

$$\dot{z}_2 = \frac{1}{RC}(-z_1 - 3z_2 + u).$$

Finally, it is possible to represent the system in the controllable canonical form

$$\begin{pmatrix} \dot{z}_1 \\ \dot{z}_2 \end{pmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{1}{RC} & \frac{-3}{RC} \end{bmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} + \frac{1}{RC} \begin{pmatrix} 0 \\ 1 \end{pmatrix} u.$$

2.1.2 RLC circuit

The following picture shows an RLC circuit.

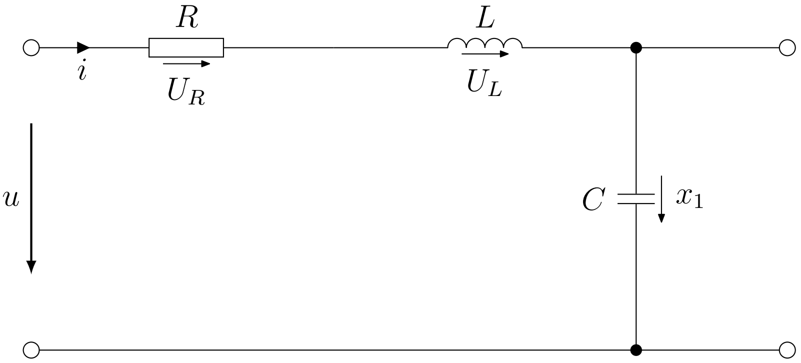


Figure 6: RLC circuit

The following equations are given from the circuit by Kirchhoff's law:

$$U_R = Ri(t),$$

$$U_L = L\frac{di(t)}{dt},$$

$$u(t) = U_R(t) + U_L(t) + x_1(t) = Ri(t) + L\frac{di(t)}{dt} + \frac{1}{C}\int i(t)dt. \tag{2.1.12}$$

Now, the equation for the current $i = C\dot{x}$ can be insert in equation 2.1.12

$$u(t) = RC\dot{x}_1 + L\frac{C\dot{x}_1}{dt} + x_1 = RC\dot{x}_1 + LC\ddot{x}_1 + x_1.$$

Next, the new variable $x_2(t)$ will be introcuded with $x_2(t) = \dot{x}_1(t)$. From this follows that $\dot{x}_2(t) = \ddot{x}_1(t)$ and with this ansatz the following equation is obtained

$$u(t) = RCx_2 + LC\dot{x}_2 + x_1. \tag{2.1.13}$$

Since $\dot{x}_1$ is already known it is possible to calculate $\dot{x}_2$ with equation (2.1.13) and the outcome therefore is

$$\dot{x}_2 = -\frac{1}{LC}x_1(t) - \frac{R}{L}x_2(t) + \frac{1}{LC}u(t).$$

With $\dot{x}_1$ and $\dot{x}_2$ the state space model can be described by

$$\begin{pmatrix}\dot{x}_1\\ \dot{x}_2\end{pmatrix} = \begin{bmatrix}0 & 1\\ -\frac{1}{LC} & -\frac{R}{L}\end{bmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix} + \begin{pmatrix}0\\ \frac{1}{LC}\end{pmatrix}u.$$

2.2 Nonlinear circuits

In this section, the nonlinear circuits will be handled. The nonlinear part of the circuit is represented as a nonlinear resistor. The voltage over the resistor will be described as a function $f(i)$.

2.2.1 Cascaded RC circuit

The following picture 7 shows a cascaded RC circuit with a nonlinear resistor R_2.

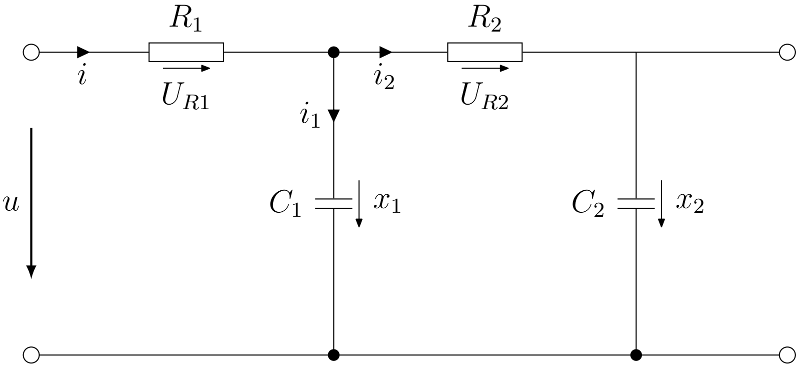


Figure 7: Cascaded RC circuit with a nonlinear R_2.

The following equations are given from the circuit by Kirchhoff's law

$$i = i_1 + i_2 = \frac{U_{R1}}{R_1} = C_1\dot{x}_1 + C_2\dot{x}_2, \tag{2.2.1}$$

$$U_{R2} = f(i_2) = f(C_2\dot{x}_2), \tag{2.2.2}$$

$$x_1 = \frac{1}{C_1}\int i_1 dt = U_{R2} + x_2 = f(i_2) + x_2.$$

By reshaping equation (2.2.2) the following equation for $\dot{x}_2$ will sustain

$$\dot{x}_2 = \frac{1}{C_2} f^{-1}(U_{R2}). \tag{2.2.3}$$

U_{R2} can be rewritten as following expression: $U_{R2} = x_1 - x_2$. The equation (2.2.3) can be expanded as

$$\dot{x}_2 = \frac{1}{C_2} f^{-1}(x_1 - x_2).$$

In the next step, formula (2.2.1) can be converted to

$$\dot{x}_1 = \frac{U_{R1}}{R_1C_1} - \frac{C_2}{C_1}\dot{x}_2 = \frac{u - x_1}{R_1C_1} - \frac{1}{C_1}f^{-1}(x_1 - x_2).$$

Now, there are the following expressions for $\dot{x}_1$ and $\dot{x}_2$ as

$$\begin{aligned}\dot{x}_1 = & -\frac{x_1}{R_1C_1} - \frac{1}{C_1}f^{-1}(x_1 - x_2) + \frac{1}{R_1C_1}u,\\ \dot{x}_2 = & \frac{1}{C_2}f^{-1}(x_1 - x_2).\end{aligned}$$

However, these expressions cannot be used to represent in the controllable canonical form. Therefore, the Lie derivative is a useful tool to derive the desired controllable canonical form for a nonlinear system

$$L_{\boldsymbol{f}}h(\boldsymbol{x}) = \frac{\partial h(\boldsymbol{x})}{\partial(\boldsymbol{x})}\boldsymbol{f}(\boldsymbol{x}) = grad^T h(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x}).$$

For example, the Lie derivative will apply to the output variable c

$$L_{\boldsymbol{a}}c(\boldsymbol{x}) = \frac{\partial c(\boldsymbol{x})}{\partial \boldsymbol{x}}\boldsymbol{a}(\boldsymbol{x}).$$

If the temporal derivative of y will be calculated, the following equation accrues

$$\dot{y} = \frac{dc(\boldsymbol{x})}{dt} = \frac{\partial c(\boldsymbol{x})}{\partial x_1}\dot{x}_1 + ... + \frac{\partial c(\boldsymbol{x})}{\partial x_n}\dot{x}_n = \frac{c(\boldsymbol{x})}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}}.$$

It is now possible to calculate $y, \dot{y}$ and $\ddot{y}$ with the following equations

$$y = c(\boldsymbol{x}),$$

$$\dot{y} = \frac{\partial c(\boldsymbol{x})}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}} = L_{\boldsymbol{a}}c(\boldsymbol{x}),$$

$$\ddot{y} = \frac{\partial L_{\boldsymbol{a}}c(\boldsymbol{x})}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}} = L_{\boldsymbol{a}}^2c(\boldsymbol{x}).$$

For the following calculations the convention $f^{-1}(x) = g(x)$ holds and it yields to the following equations

$$y = x_2,$$

$$\dot{y} = \frac{\partial y}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}} = \frac{\partial x_2}{\partial x_1}\dot{x}_1 + \frac{\partial x_2}{\partial x_2}\dot{x}_2 = \dot{x}_2 = \frac{1}{C_2}g(x_1 - x_2),$$

$$\ddot{y} = \frac{\partial \dot{y}}{\partial \dot{\boldsymbol{x}}}\dot{\boldsymbol{x}} = \frac{\partial \frac{1}{C_2}g(x_1 - x_2)}{\partial x_1}\dot{x}_1 + \frac{\partial \frac{1}{C_2}g(x_1 - x_2)}{\partial x_2}\dot{x}_2$$

$$= \frac{1}{C_2}g'(x_1 - x_2)\dot{x}_1 - \frac{1}{C_2}g'(x_1 - x_2)\dot{x}_2.$$

Next, the new state space coordinates $z_1 = y$, $z_2 = \dot{y}$ will be introduced

$$z_1 = x_2,$$

$$z_2 = \frac{1}{C_2}g(x_1 - x_2). \tag{2.2.4}$$

With equation 2.2.4, x_1 can be calculated:

$$x_1 = g^{-1}(C_2 z_2) + z_1 \tag{2.2.5}$$

In this regard, the following equations occur for $\dot{z}_1$ and $\dot{z}_2$

$$\dot{z}_1 = z_2,$$

$$\dot{z}_2 = \frac{1}{C_2}g'(x_1 - x_2)\dot{x}_1 - \frac{1}{C_2}g'(x_1 - x_2)\dot{x}_2$$

$$= \frac{1}{C_2}g'(g^{-1}(C_2 z_2) + z_1 - z_1)\dot{x}_1 - \frac{1}{C_2}g'(g^{-1}(C_2 z_2) + z_1 - z_1)z_2$$

$$= \frac{1}{C_2}g'(g^{-1}(C_2 z_2) + z_1 - z_1)(-\frac{x_1}{R_1 C_1} - \frac{1}{C_1}g(x_1 - x_2) + \frac{1}{R_1 C_1}u)$$
$$- \frac{1}{C_2}g'(g^{-1}(C_2 z_2) + z_1 - z_1)z_2$$

where $\dot{z}_2$ is further reformulated as

$$\dot{z}_2 = \frac{1}{C_2} g'(g^{-1}(C_2 z_2))(-\frac{(g^{-1}(C_2 z_2) + z_1)}{R_1 C_1} - \frac{1}{C_1} g(g^{-1}(C_2 z_2)) + \frac{1}{R_1 C_1} u) - \frac{1}{C_2} g'(g^{-1}(C_2 z_2)) z_2$$

$$= g'(g^{-1}(C_2 z_2))(\frac{-g^{-1}(C_2 z_2) - z_1}{R_1 C_1 C_2} - \frac{1}{C_1} z_2 - \frac{1}{C_2} z_2) + \frac{1}{R_1 C_1 C_2} g'(g^{-1}(C_2 z_2)) u.$$

The following function and their inverse and derivative will be used

$$g(z) = tanh(z),$$

$$g^{-1}(z) = atanh(z),$$

$$g'(z) = sech^2(z).$$

Now, the controllable canonical form for the nonlinear system can be represented by

$$\begin{pmatrix} \dot{z}_1 \\ \dot{z}_2 \end{pmatrix} = \begin{pmatrix} z_2 \\ \frac{1}{C_2} g'(g^{-1}(C_2 z_2))(-\frac{g^{-1}(C_2 z_2) - z_1}{R_1 C_1} - \frac{C_2}{C_1} z_2 - z_2) \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{R_1 C_1 C_2} g'(g^{-1}(C_2 z_2)) \end{pmatrix} u.$$

2.2.2 RLC circuit

The following picture 8 shows an RLC circuit with a nonlinear R.

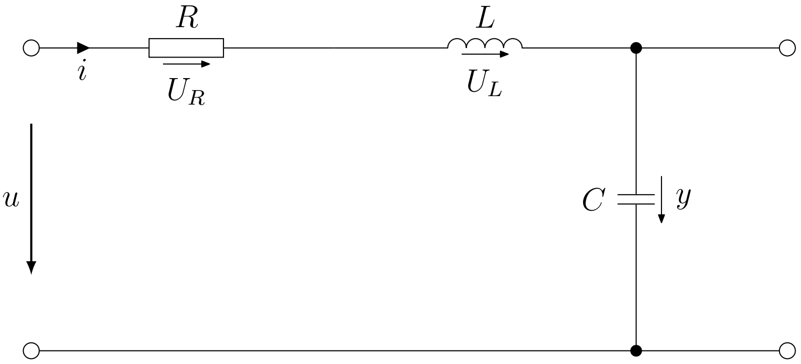


Figure 8: RLC circuit with a nonlinear R.

The following equations are given from the circuit by Kirchhoff's law:

$$U_R = f(i),$$

$$U_L = L\frac{di(t)}{dt},$$

$$y(t) = \frac{1}{C}\int i(t)dt, \tag{2.2.6}$$

$$u(t) = U_R(t) + U_L(t) + y(t) = f(i) + L\frac{di(t)}{dt} + \frac{1}{C}\int i(t)dt. \tag{2.2.7}$$

By reshaping formula 2.2.6 equation for the current will sustain

$$i(t) = C\dot{y}. \tag{2.2.8}$$

Therefore, the equation 2.2.8 can be insert in equation 2.2.7 and it yields to

$$u(t) = f(C\dot{y}) + L\frac{C\dot{y}}{dt} + y = f(C\dot{y}) + LC\ddot{y} + y. \tag{2.2.9}$$

Now, the variables x_1 and x_2 will be introduced

$$x_1 = y,$$

$$\dot{x}_1 = \dot{y} = x_2,$$

$$\dot{x}_2 = \ddot{y}.$$

These variables can be inserted in equation 2.2.9 and this leads to

$$u = f(Cx_2) + LC\dot{x}_2 + x_1. \tag{2.2.10}$$

If equation 2.2.10 is solved to $\dot{x}_2$, the needed variables $\dot{x}_1$ and $\dot{x}_2$ are known

$$\dot{x}_2 = -\frac{1}{LC}f(Cx_2) - \frac{1}{LC}x_1 + \frac{1}{LC}u,$$

and subsequently, the following state space model will sustain

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} x_2 \\ -\frac{1}{LC}x_1 - \frac{1}{LC}f(Cx_2) \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{LC} \end{pmatrix} u.$$

3 Controller design

In this chapter, the parameters which are mandatory for the controller calculation which are described in chapter 1.4 and chapter 1.5 will be extracted, such that finally, the controller $u(\boldsymbol{x})$ can be implemented.

3.1 Linear circuits

For the linear controller design, the matrix

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ a_0 & a_1 \end{bmatrix}$$

and the vector

$$\boldsymbol{b} = \begin{pmatrix} 0 \\ b_1 \end{pmatrix}$$

are needed. From the canonical form of each circuit the values a_0, a_1 and b_1 will be given.

Linear cascaded RC circuit

$$\begin{pmatrix} \dot{z}_1 \\ \dot{z}_2 \end{pmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{1}{RC} & \frac{-3}{RC} \end{bmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} + \frac{1}{RC} \begin{pmatrix} 0 \\ 1 \end{pmatrix} u$$

$$a_0 = \frac{1}{RC}$$

$$a_1 = \frac{-3}{RC}$$

$$b_1 = \frac{1}{RC}$$

Linear RLC circuit

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{1}{LC} & -\frac{R}{L} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \frac{1}{LC} \begin{pmatrix} 0 \\ 1 \end{pmatrix} u$$

$$a_0 = -\frac{1}{LC}$$

$$a_1 = -\frac{R}{L}$$

$$b_1 = \frac{1}{LC}$$

3.2 Nonlinear circuits

For the nonlinear controller design, the vector

$$\tilde{\boldsymbol{a}}(\boldsymbol{z}) = \begin{pmatrix} z_2 \\ \alpha(\boldsymbol{z}) \end{pmatrix},$$

and the vector

$$\tilde{\boldsymbol{b}}(\boldsymbol{z}) = \begin{pmatrix} 0 \\ \beta(\boldsymbol{z}) \end{pmatrix}.$$

are needed. And the variables $\alpha(\boldsymbol{z})$ and $\beta(\boldsymbol{z})$ have to be defined.

Nonlinear cascaded RC circuit

$$\begin{pmatrix} \dot{z}_1 \\ \dot{z}_2 \end{pmatrix} = \begin{pmatrix} z_2 \\ \frac{1}{C_2} g'(g^{-1}(C_2 z_2))(-\frac{g^{-1}(C_2 z_2) - z_1}{R_1 C_1} - \frac{C_2}{C_1} z_2 - z_2) \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{R_1 C_1 C_2} g'(g^{-1}(C_2 z_2)) \end{pmatrix} u,$$

$$\alpha(\boldsymbol{z}) = \frac{1}{C_2} g'(g^{-1}(C_2 z_2))(-\frac{g^{-1}(C_2 z_2) - z_1}{R_1 C_1} - \frac{C_2}{C_1} z_2 - z_2),$$

$$\beta(\boldsymbol{z}) = \frac{1}{R_1 C_1 C_2} g'(g^{-1}(C_2 z_2)).$$

Nonlinear RLC circuit

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} x_2 \\ -\frac{1}{LC}x_1 - \frac{1}{LC}f(Cx_2) \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{LC} \end{pmatrix} u$$

$$\alpha(\boldsymbol{x}) = -\frac{1}{LC}x_1 - \frac{1}{LC}f(Cx_2),$$

$$\beta(\boldsymbol{x}) = \frac{1}{LC}.$$

4 Simulation

In the following chapters, the results of the simulation will be visualized.

4.1 Linear circuits

4.1.1 Cascaded RC circuit

Linear controller for reaching law with constant rate

The following pictures will illustrate the impact of the different control parameters, r_1 (Figure 9) and ε (Figure 10). A linear cascaded RC circuit with $R = 1\Omega$ and $C = 1F$ is simulated. When r_1 gets bigger, the system gets faster, and the undershoot gets higher and vice versa. By increasing ε, the system will also get faster.

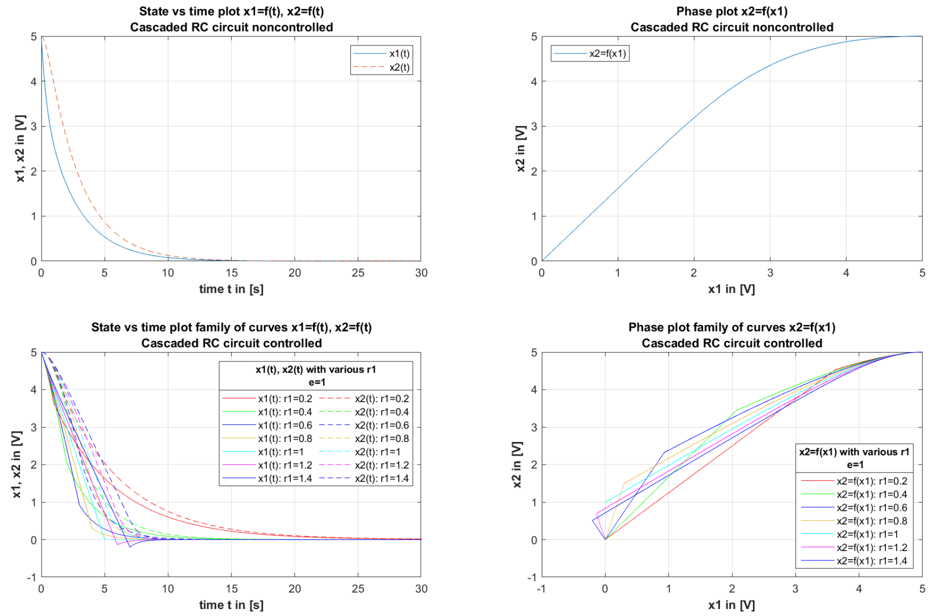


Figure 9: The two upper diagrams show the noncontrolled circuit. The subjacent diagrams show the controlled linear circuit with various values of r_1.

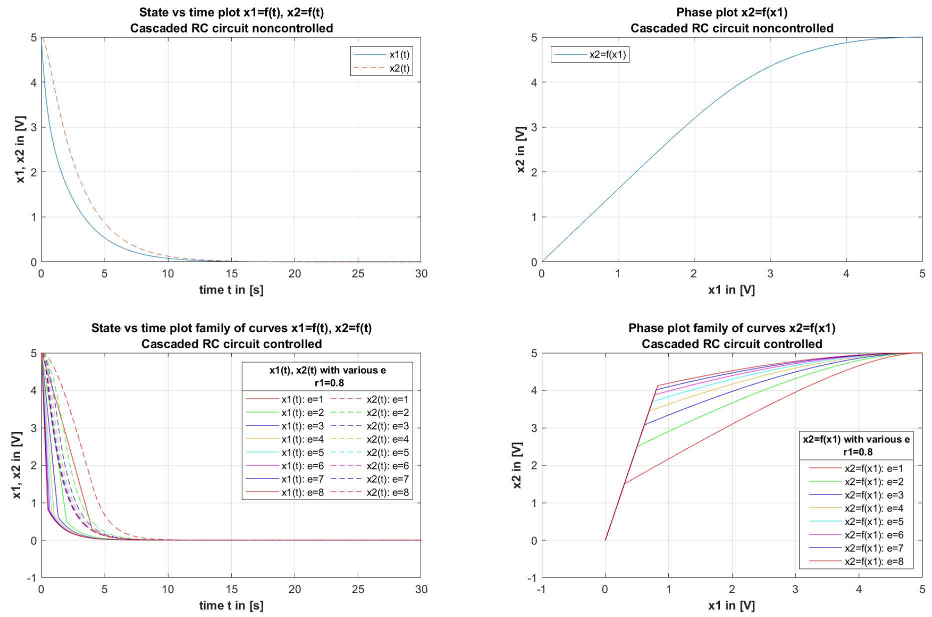


Figure 10: The two upper diagrams show the noncontrolled circuit. The subjacent diagrams show the controlled linear circuit with various values of ε

Linear controller for reaching law with exponential rate

In the following picture, only the impact of the additional control parameter k is simulated. Because, that is the only different parameter from the previous simulated controller. With the increase of the additional control parameter k the system gets even faster than the upper controller design.

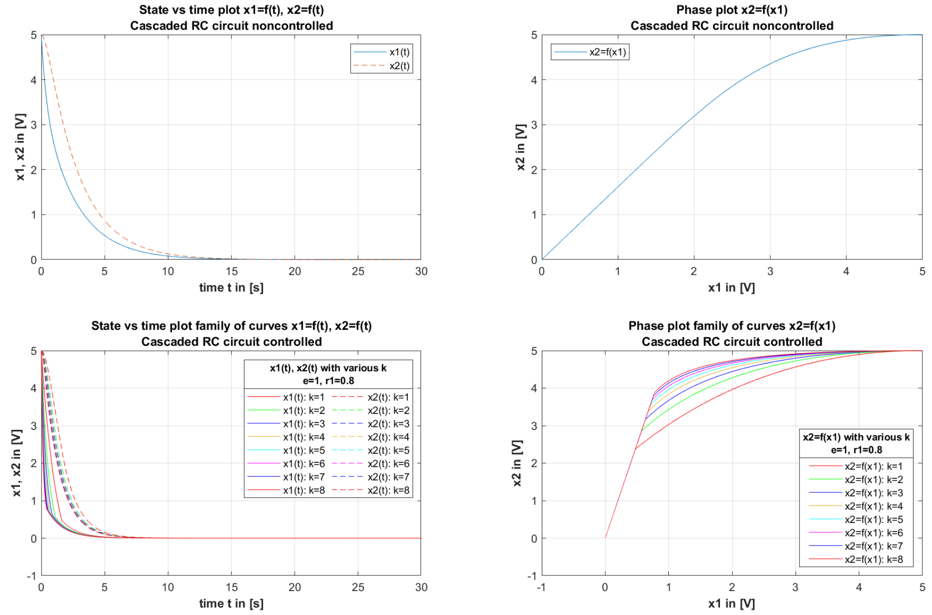


Figure 11: The two upper diagrams show the noncontrolled circuit. The subjacent diagrams show the controlled linear circuit with various values of k

Linear controller for reaching law with power rate

Finally, the following pictures illustrate the impact of the different control parameters, r_1 (Figure 12), α (Figure 13), and k (Fiugre 14), of the third controller of this project. Equally, as in the previous simulated controllers, if r_1 gets bigger, the system gets faster. However, also the undershoot gets higher and vice versa. When α gets bigger, the system gets faster as well. Lastly, by increasing k, the system will also fast. The general difference in the simulation results of the third controller is its smoother trajectory.

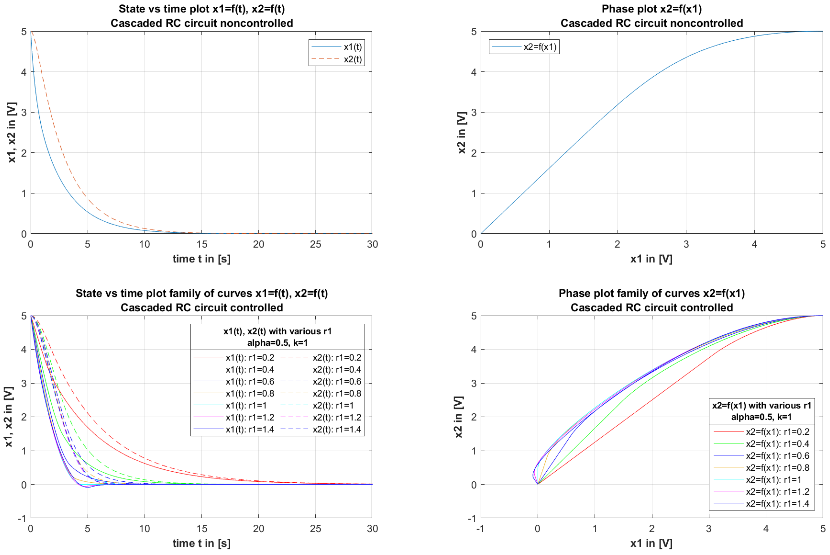


Figure 12: The two upper diagrams show the noncontrolled circuit. The subjacent diagrams show the controlled linear circuit with various values of $r1$

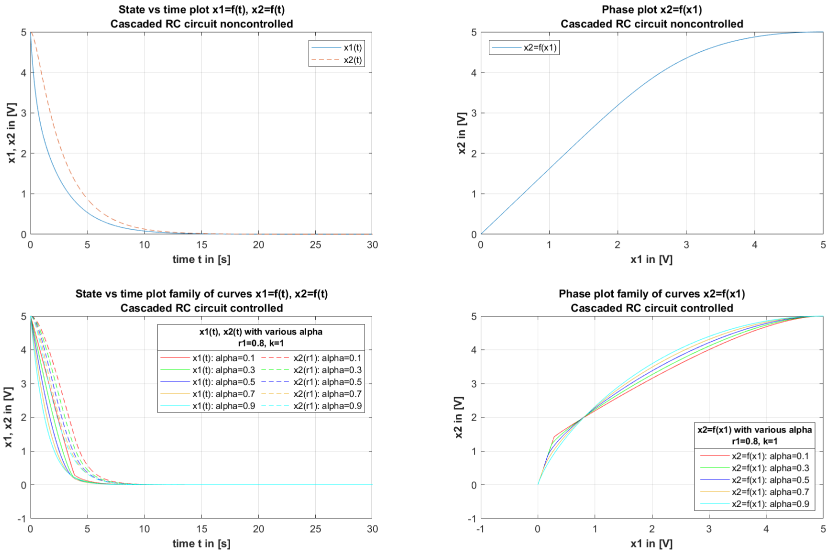


Figure 13: The two upper diagrams show the noncontrolled circuit. The subjacent diagrams show the controlled linear circuit with various values of α

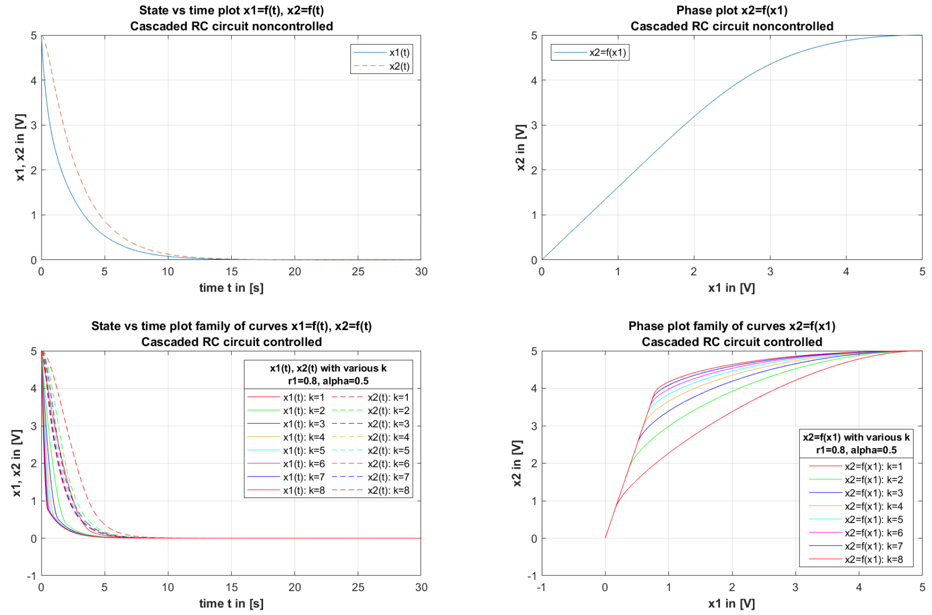


Figure 14: The two upper diagrams show the noncontrolled circuit. The subjacent diagrams show the controlled linear circuit with various values of k

4.1.2 RLC circuit

Linear controller for reaching law with constant rate

The following pictures will illustrate the impact of the different control parameters, r_1 (Figure 15) and ε (Figure 16). A linear RLC circuit with $R = 1\Omega$, $C = 1F$ and $L = 1H$ is simulated. When r_1 gets bigger, the system gets faster, and the undershoot gets higher and vice versa. By increasing of ε the system will also get faster, but without overshoots.

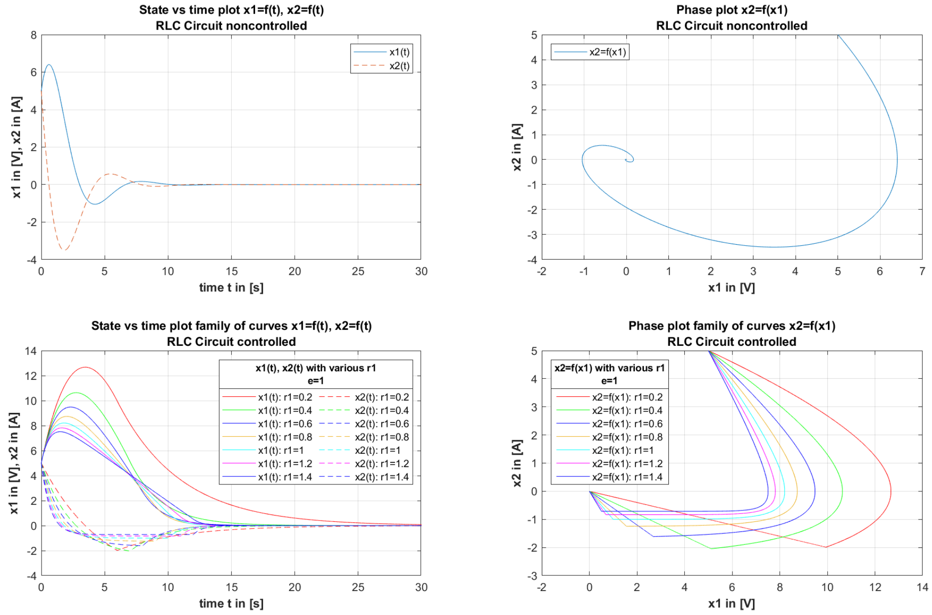


Figure 15: The two upper diagrams show the noncontrolled circuit. The subjacent diagrams show the controlled linear circuit with various values of r_1

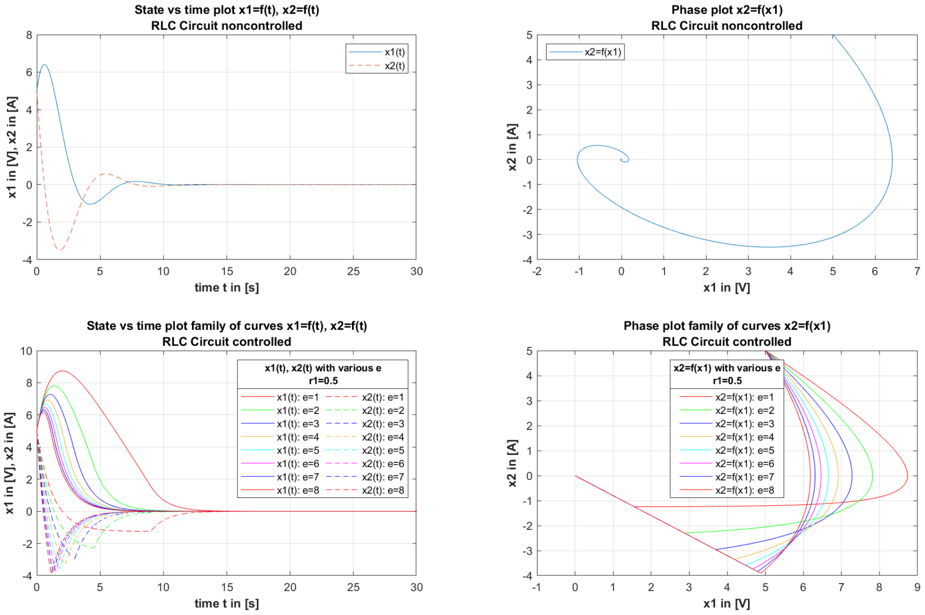


Figure 16: The two upper diagrams show the noncontrolled circuit. The subjacent diagrams show the controlled linear circuit with various values of ε

Linear controller for reaching law with exponential rate

In the following picture, only the impact of the additional control parameter k is simulated. Because, that is the only different parameter from the previous simulated controller. With the increase of the additional control parameter k the system gets even faster than the upper controller design.

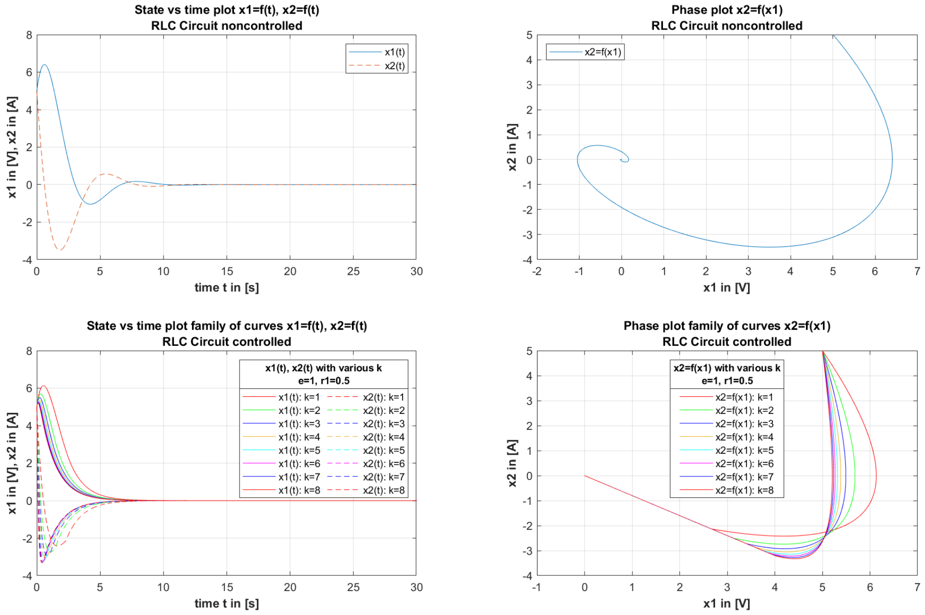


Figure 17: The two upper diagrams show the noncontrolled circuit. The subjacent diagrams show the controlled linear circuit with various values of k

Linear controller for reaching law with power rate

Finally, the following pictures illustrate the impact of the different control parameters, r_1 (Figure 18), α (Figure 19), and k (Fiugre 20), of the third controller of this project. Equally, as in the previous simulated controllers, if r_1 gets bigger, the system gets faster. However, also the undershoot gets higher and vice versa. When α gets bigger, the system gets faster as well. Lastly, by increasing k, the system will also fast. The general difference in the simulation results of the third controller is its smoother trajectory.

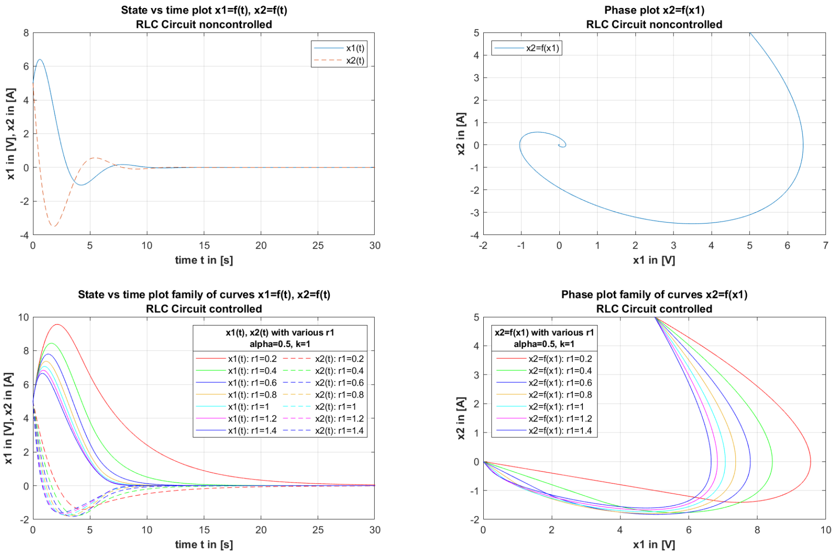


Figure 18: The two upper diagrams show the noncontrolled circuit. The subjacent diagrams show the controlled linear circuit with various values of $r1$

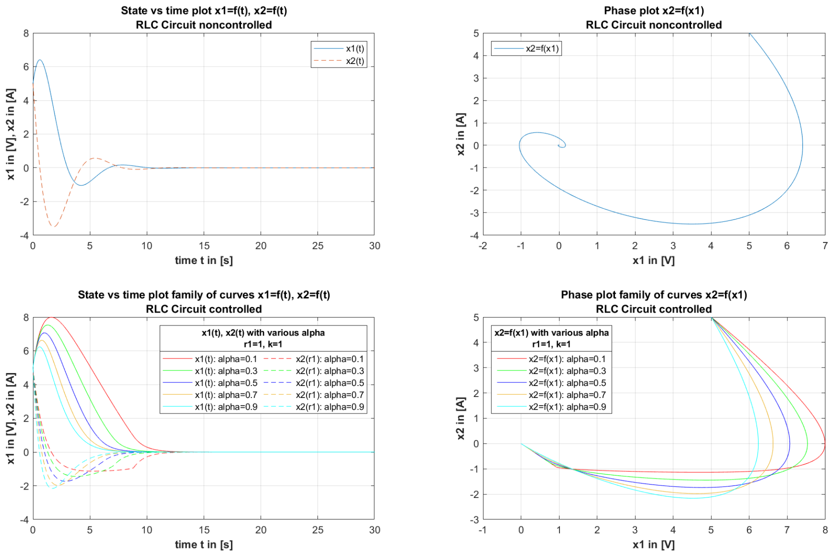


Figure 19: The two upper diagrams show the noncontrolled circuit. The subjacent diagrams show the controlled linear circuit with various values of α

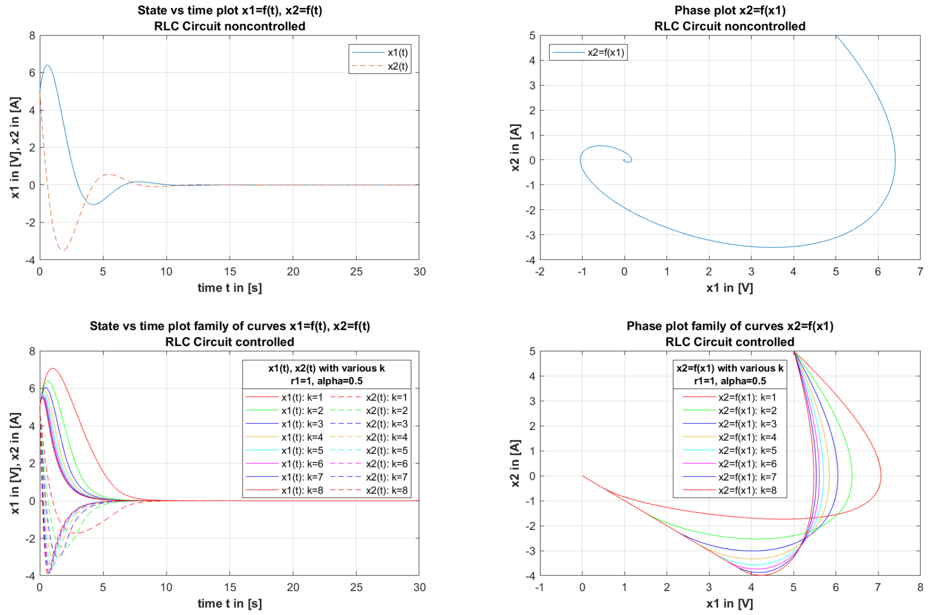


Figure 20: The two upper diagrams show the noncontrolled circuit. The subjacent diagrams show the controlled linear circuit with various values of k

4.2 Nonlinear circuits

4.2.1 Cascaded RC circuit

Nonlinear controller for reaching law with constant rate

The following pictures will illustrate the impact of the different control parameters, r_1 (Figure 21) and ε (Figure 22). A nonlinear cascaded RC circuit is simulated with $U_{R2} = f(i_2)$ and $C = 1F$. When r_1 gets bigger, the system gets faster, and the undershoot increases. In addition, if r_1 is selected too high, the undershoot will increase, and the return values will be in the complex number range. This results from the function which will be used because their domain is limited from $-1 < x < 1$. If the function's argument is outside of this domain, the returned value will be a complex number, and the source code is running into a numerical issue because it cannot handle complex numbers. Therefore, a min/max function was implemented. This function will check if the values taken for the next iteration are in the legal domain. If a value is not in the legal domain, it will shift the value so that the following calculation will not be complex. On the one hand, this ensured that the calculation was not complex anymore. On the other hand, it can be seen that the simulation results still show that if this function applies, the curves will take an unexpected shape, see the yellow curve in Figure 21. The same effect occurs if ε is increased. However, the undershoot will not rise to the same level as then r_1 will be increased.

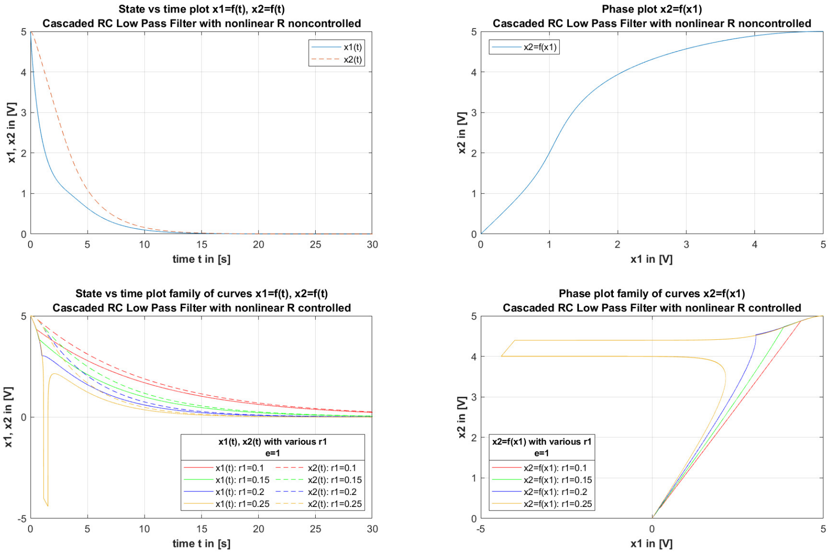


Figure 21: The two upper diagrams show the noncontrolled circuit. The subjacent diagrams show the controlled nonlinear circuit with various values of r_1. The yellow curve shows the impact of the min/max function.

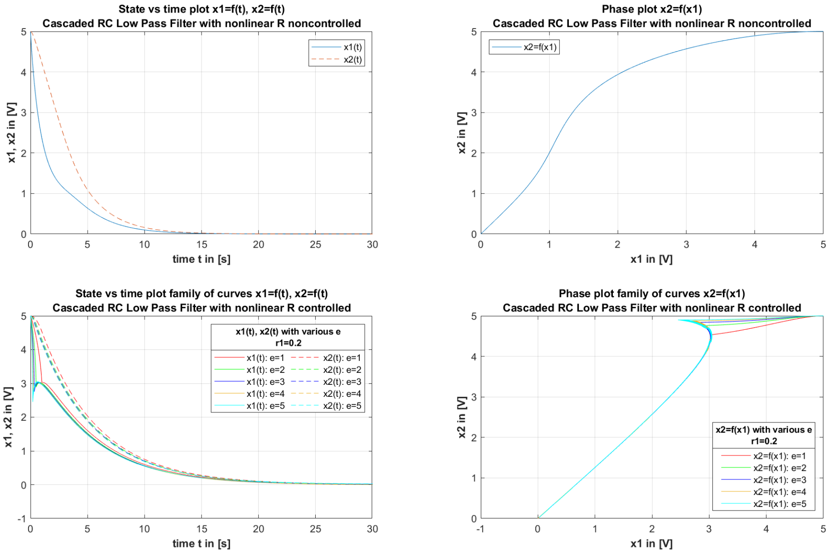
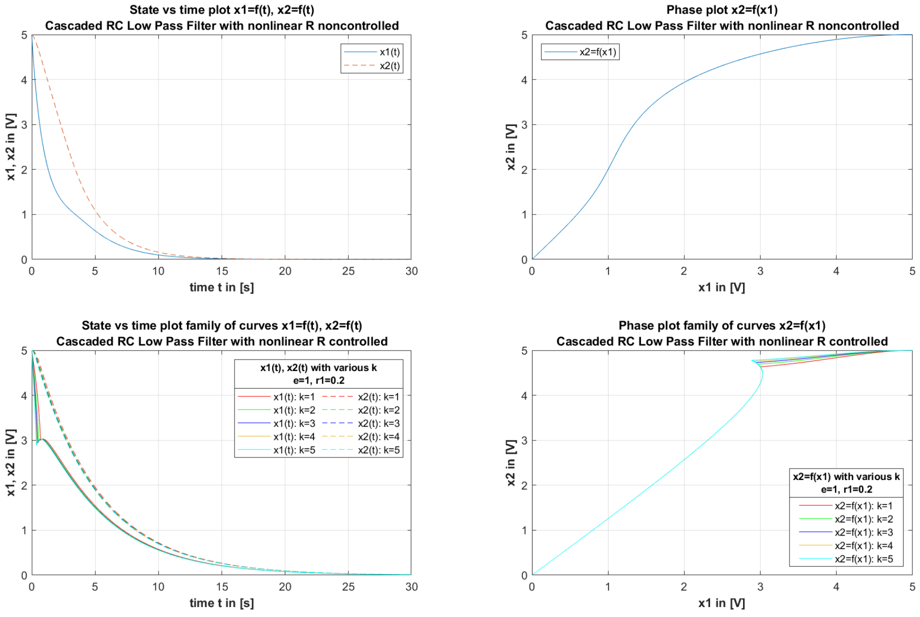


Figure 22: The two upper diagrams show the noncontrolled circuit. The subjacent diagrams show the controlled nonlinear circuit with various values of ε.

Nonlinear controller for reaching law with exponential rate

In the following picture, only the impact of the additional control parameter k is simulated because that is the only different parameter from the previous simulated controller. With the increase of the additional control parameter k, the same issue as the controller before. Therefore, k cannot be chosen arbitrarily large.



State vs time plot x1=f(t), x2=f(t)
Cascaded RC Low Pass Filter with nonlinear R noncontrolled

Phase plot x2=f(x1)
Cascaded RC Low Pass Filter with nonlinear R noncontrolled

State vs time plot family of curves x1=f(t), x2=f(t)
Cascaded RC Low Pass Filter with nonlinear R controlled

Phase plot family of curves x2=f(x1)
Cascaded RC Low Pass Filter with nonlinear R controlled

Figure 23: The two upper diagrams show the noncontrolled circuit. The subjacent diagrams show the controlled nonlinear circuit with various values of k.

Nonlinear controller for reaching law with power rate

In the following pictures, the graphs have the same behavior. If one control parameter is increased, the system gets an undershoot, and the system output will be in the complex number range.

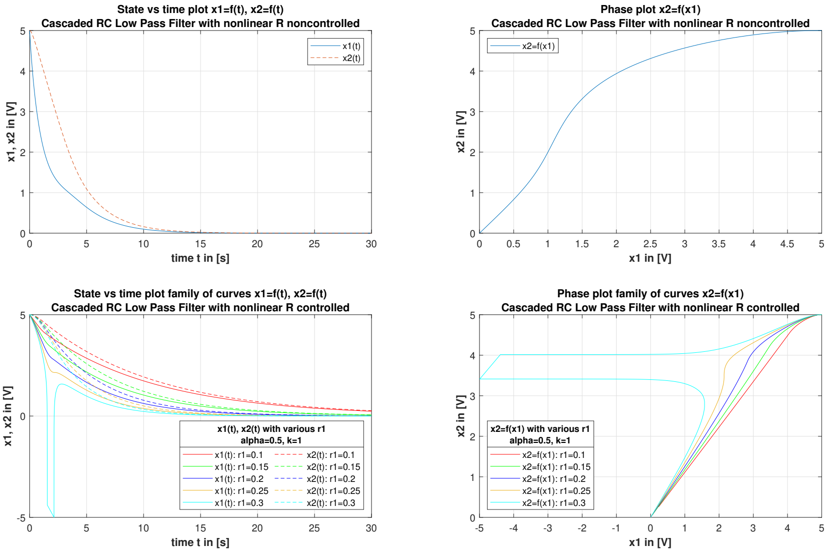


Figure 24: The two upper diagrams show the noncontrolled circuit. The subjacent diagrams show the controlled nonlinear circuit with various values of $r1$. The turquoise curve shows the impact of the min/max function.

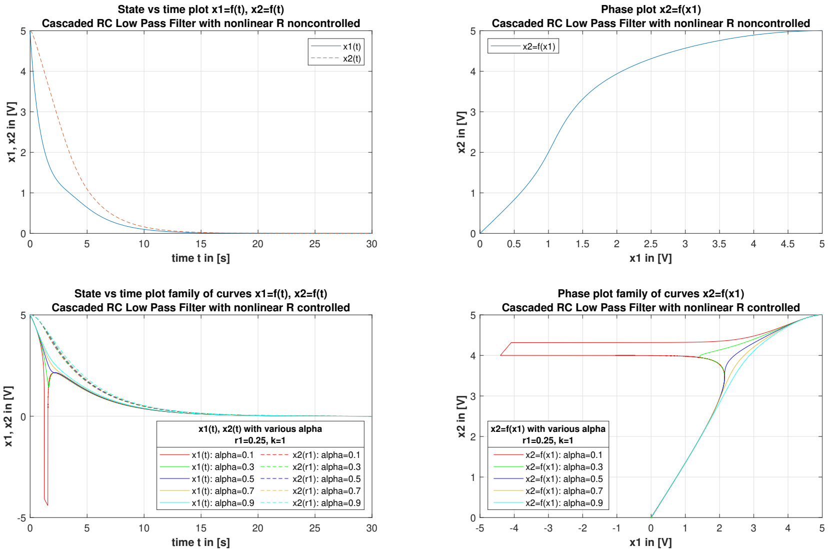


Figure 25: The two upper diagrams show the noncontrolled circuit. The subjacent diagrams show the controlled nonlinear circuit with various values of α. The magenta curve shows the impact of the min/max function.

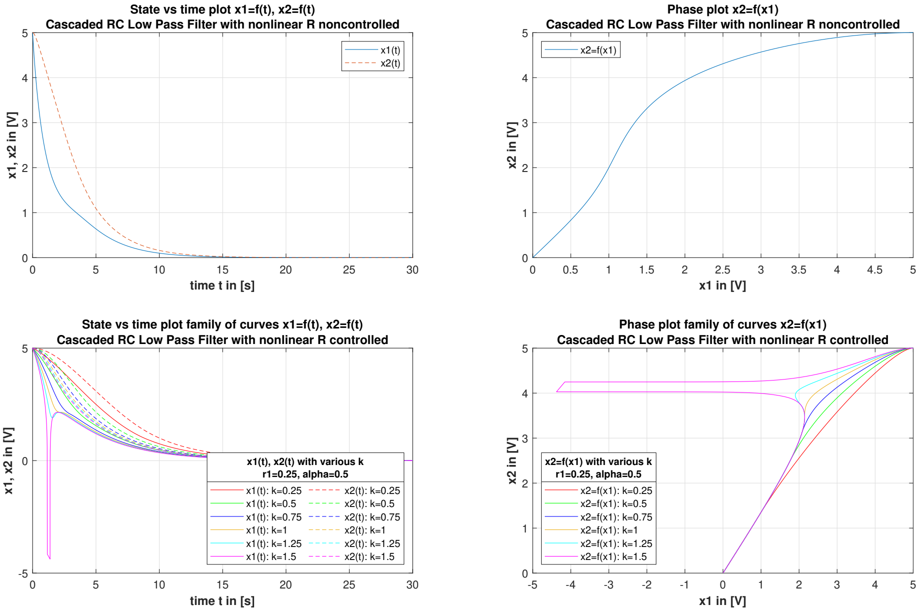


Figure 26: The two upper diagrams show the noncontrolled circuit. The subjacent diagrams show the controlled nonlinear circuit with various values of k. The yellow curve shows the impact of the min/max function.

4.2.2 RLC Circuit

Nonlinear controller for reaching law with constant rate

The following pictures will illustrate the impact of the different control parameters, r_1 (Figure 27) and ε (various 28). A nonlinear RLC circuit is simulated with $U_R = f(i)$, $C = 1F$ and $L = 1H$. If the RLC circuit is compared to the cascaded RC circuit, the noncontrolled circuits' behavior is different. At the cascaded RC circuit, the voltage will decrease to $0V$, the RLC circuit oscillates, and the system will be destroyed.
The controlled circuits also have another behavior compared to the cascaded circuit. When r_1 gets bigger, the undershoot and overshoot will decrease (Figure 27). There is similar behavior for an increasing ε, but with increasing ε the system will be faster, but the undershoots are increasing (Figure 28).

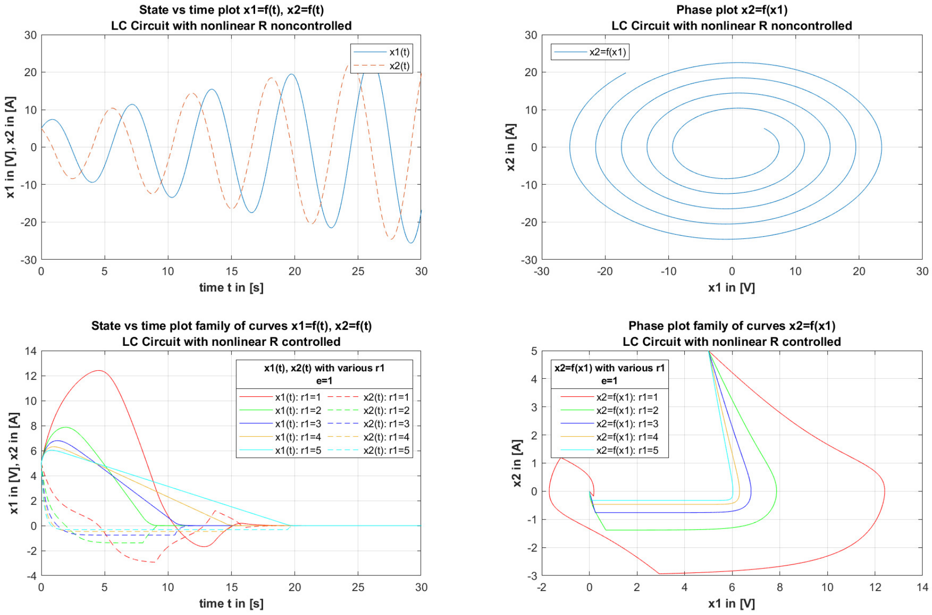


Figure 27: The two upper diagrams show the noncontrolled circuit. The subjacent diagrams show the controlled nonlinear circuit with various values of r_1.

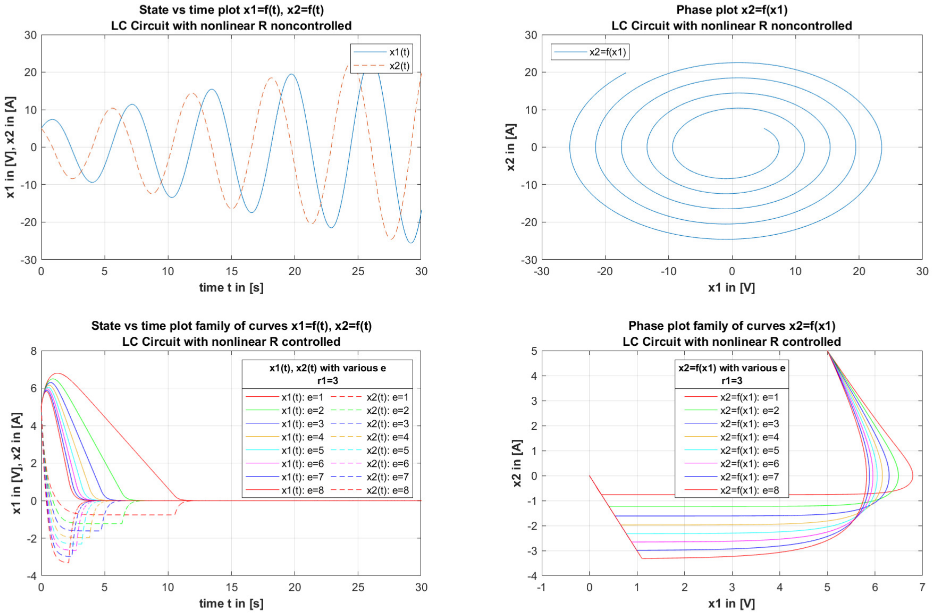


Figure 28: The two upper diagrams show the noncontrolled circuit. The subjacent diagrams show the controlled nonlinear circuit with variou values of ε.

Nonlinear controller for reaching law with exponential rate

In the following picture, only the impact of the additional control parameter k is simulated because that is the only different parameter from the previous simulated controller. With the increase of the additional control parameter k, the system will be faster but the undershoots are increasing.

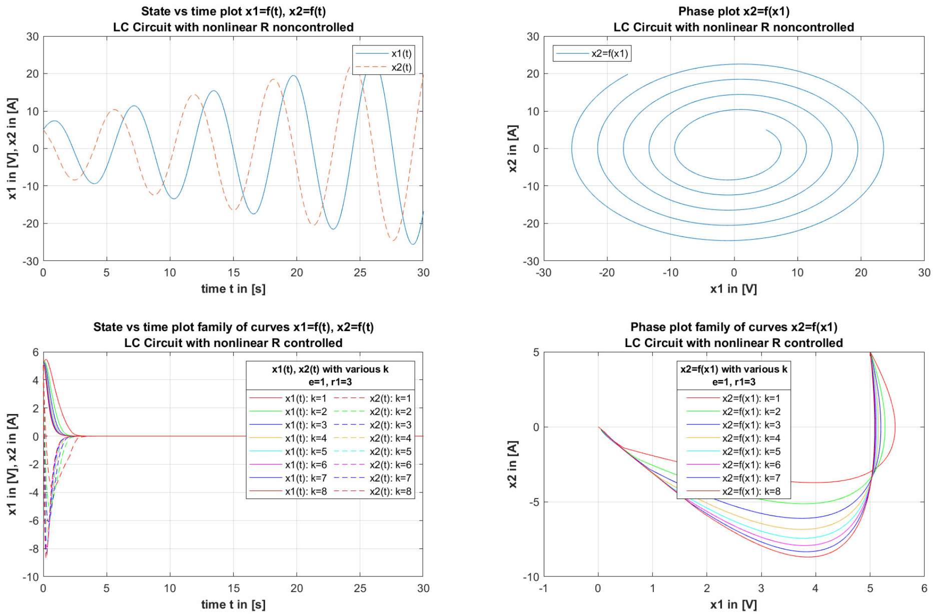


Figure 29: The two upper diagrams show the noncontrolled circuit. The subjacent diagrams show the controlled nonlinear circuit with various values of k.

Nonlinear controller for reaching law with power rate

In the following pictures, the graphs have the same behavior. If one control parameter is increased, the system gets some undershoot.

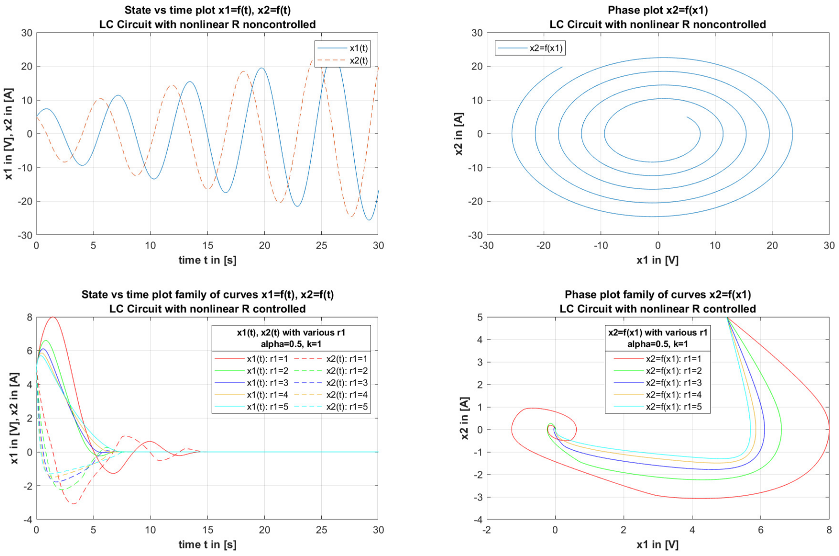


Figure 30: The two upper diagrams show the noncontrolled circuit. The subjacent diagrams show the controlled nonlinear circuit with various values of $r1$.

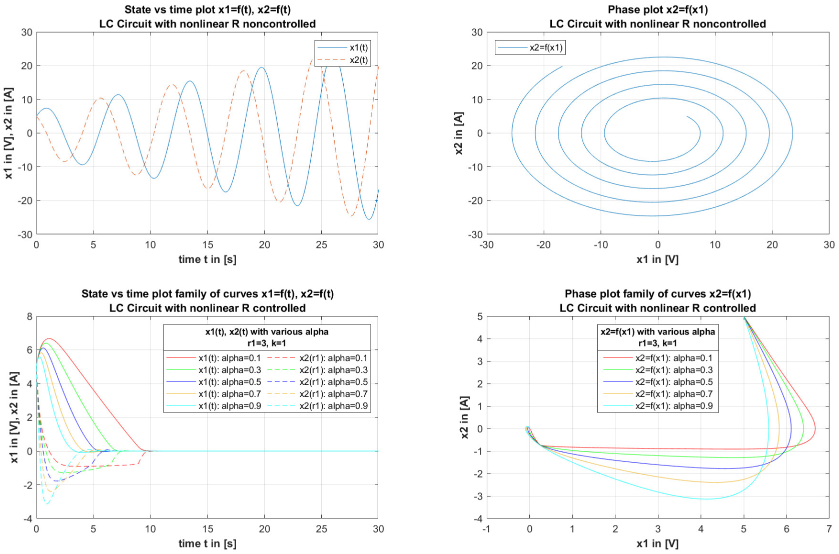
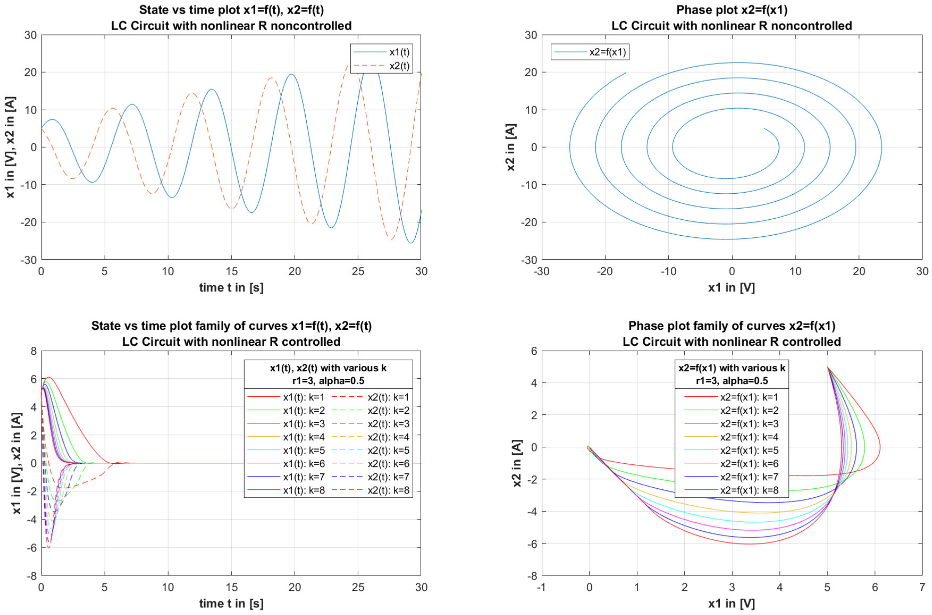


Figure 31: The two upper diagrams show the noncontrolled circuit. The subjacent diagrams show the controlled nonlinear circuit with various values of α.



State vs time plot x1=f(t), x2=f(t)
LC Circuit with nonlinear R noncontrolled

Phase plot x2=f(x1)
LC Circuit with nonlinear R noncontrolled

State vs time plot family of curves x1=f(t), x2=f(t)
LC Circuit with nonlinear R controlled

Phase plot family of curves x2=f(x1)
LC Circuit with nonlinear R controlled

Figure 32: The two upper diagrams show the noncontrolled circuit. The subjacent diagrams show the controlled nonlinear circuit with various values of k.

5 Controller design in C

The following chapters will illustrate the implementation in C of each linear and nonlinear controller. The associated C function with its inputs and return values will be visualized.

5.1 Linear circuit

The following subchapters will introduce the three linear controllers *Regler*1, *Regler*2, and *Regler*3.

Reaching law with constant

```
Regler1(x1(i),x2(i),a0,a1,r1,e,soll,b1)
    /*Returns the value u for the specified controller */
        u = ((-x1a0 - x2(r1+a1) - esign((r1x1)+x2 - refr1))/b1);
    return u
```

Exponential reaching law

```
Regler2(x1(i),x2(i),a0,a1,r1,e,k,soll,b1)
    /*Returns the value u for the specified controller */
        u = ((-x1a0 - x2(r1+a1) - esign((r1x1)+x2 - ref*r1)
            - k((r1x1)+x2 - refr1))/b1);
    return u
```

Reaching law with power rate

```
Regler3(x1, x2, a0, a1, r1, alpha, k, ref,b1)
    /*Returns the value u for the specified controller */
        u = ((-x1a0 - x2(r1+a1) - kabs(r1x1+x2-refr1)^alphasign
            (r1x1+x2-refr1))/b1);
    return u
```

5.2 Nonlinear circuit

The following subchapters will introduce the three nonlinear controllers *Regler*1, *Regler*2, and *Regler*3.

Reaching law with constant

```
Regler1(x1(i),x2(i),a0,a1,r1,e,soll ,beta)
    /*Returns the value u for the specified controller */
        u = (-((r1a0+a1+esign(r1x1+x2-r1ref))/beta));
    return u
```

Exponential reaching law

```
Regler2(x1(i),x2(i),a0,a1,r1,e,k,soll ,b)
    /*Returns the value u for the specified controller */
        u = (-((r1a0+a1+esign(r1x1+x2-r1ref)
            +k(r1x1+x2-r1ref))/b));
    return u
```

Reaching law with power rate

```
Regler3(x1, x2, a0, a1, r1, alpha, k, ref,b)
    /*Returns the value u for the specified controller */
        u = (-((r1a0+a1+kpow(abs_calc,alpha)
            sign(r1x1+x2-refr1))/b));
    return u
```

References

[1] Jürgen Adamy. *Nichtlineare System und Regelungen.* URL: https://link.springer.com/book/10.1007/978-3-642-45013-6.

[2] Alexander Barth. *Indirect Adaptive Higher-Order Sliding-Mode Control Using the Certainty-Equivalence Principle.* URL: https://www.db-thueringen.de/servlets/MCRFileNodeServlet/dbt_derivate_00052269/ilm1-2019000739.pdf.

[3] Magnus Egerstedt. *Sliding Mode Control.* URL: https://www.youtube.com/watch?v=BZbCdjbPTs4.

[4] IEEE. *Sliding mode control design principles and applications to electric drives.* URL: https://ieeexplore.ieee.org/document/184818.

[5] Ali J.Koshkouei and Keith J.Burnham. *Sliding Mode Controllers for Active Suspensions.* URL: https://www.sciencedirect.com/science/article/pii/S1474667016394794.

[6] Jinkun Liu. *Sliding Mode Control Using MATLAB.* URL: https://www.elsevier.com/books/sliding-mode-control-using-matlab/liu/978-0-12-802575-8?country=DE&format=print&utm_source=google_ads&utm_medium=paid_search&utm_campaign=germanyshopping&gclid=EAIaIQobChMI943z05Hy_AIVWofVCh0Ugg-1EAQYASABEgIUIvD_BwE&gclsrc=aw.ds.

[7] *Sliding mode control.* URL: https://en.wikipedia.org/wiki/Sliding_mode_control.

[8] *Sliding phase with chattering.* URL: https://www.researchgate.net/figure/Sliding-Mode-Control-depicting-the-sliding-surface-chattering-and-two-phases-namely_fig4_351245869.

[9] Jinkun Liu Xinhua Wang. *Sliding Mode Control for Aircraft.* URL: https://link.springer.com/chapter/10.1007/978-3-642-20907-9_12.