

# Signale & Systeme

## Analyse eines Systems

**Matrikelnr.:** 9085107 & 2147057

**Dozent:** apl. Prof. Dr. Lutz Gröll

**Kurs:** TINF22B1

**System:**  $PT_1T_t$ -Glied

15. Dezember 2023

# 1 Vorgehen in Regelungstechnik

Zur Ermöglichung einer strukturierten Vorgehensweise bedarf es einer Orientierung. Dieser Arbeit liegt die in der Vorlesung behandelte Vorgehensweise in der Regelungstechnik zugrunde, welche nachfolgend erläutert wird.

1. Detaillierungsgrad für das System festlegen
2. Physikalisches Modell einschließlich der Störsignale erstellen
3. Eingangs-, Ausgangs-, Zustands- und Störgrößen des Systems festlegen
4. Physikalische Einheiten festlegen und ggf. Normierung in Prozent
5. Analyse des Modells
  - (a) Ruhelagen, Anfangswerte, evtl. Linearisierung nichtlinearer Systeme
  - (b) Systemeigenschaften ermitteln
6. Entwurf
  - (a) Ziele, Arbeitspunkte und Trajektorien festlegen
  - (b) Entwurf: Struktur, Bereich, Verfahren und Kriterien
7. Simulation und Rückinterpretation

In dieser Vorlesung wird die Regelungstechnik außen vor gelassen, weswegen im Folgenden eine leicht abgewandelte Struktur angewendet wird. Bezüglich des Detaillierungsgrads sind alle in der Vorlesung behandelten Darstellungsformen darzulegen. Weiterführend werden die Punkte 3 und 4 des Vorgehens aus der Regelungstechnik zusammengefasst und tabellarisch dargestellt. Schwerpunktmäßig wird in dieser Arbeit Punkt 5 behandelt, in dem alle Darstellungsformen aus der Vorlesung dargelegt werden. Punkt 6 und 7 werden in dieser Arbeit nicht betrachtet.

## 2 Unser System

Wir betrachten einen Wassertank, der oben an einen Zufluss mit sehr hohem Druck angeschlossen ist. Am unteren Ende des Wassertanks befindet sich ein deutlich kleinerer Abfluss, der mit einer langen Wasserleitung verbunden ist. Erst am Ende der Wasserleitung befindet sich das Messgerät, welches den Volumenstrom misst.

Eine beispielhafte Skizze sieht folgendermaßen aus:

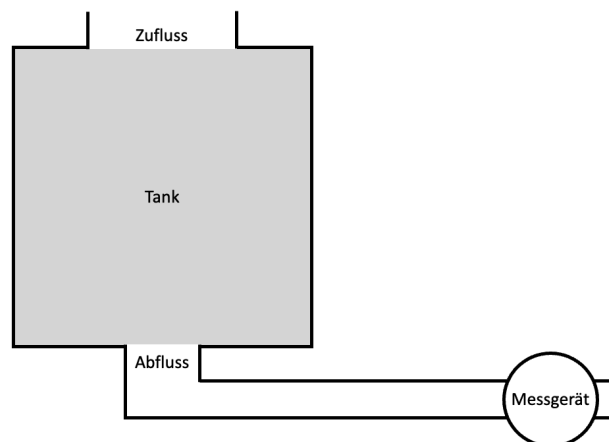


Abbildung 1: Skizze des Systems

Der Ablauf stellt sich wie folgt dar: Über den Zufluss fließt Wasser in den Tank. Da der Abfluss jedoch deutlich kleiner ist, fließt zu Beginn deutlich weniger Wasser heraus. Erst wenn der Tank voller wird, erhöht sich der Druck und es kommt zu einem immer größeren Volumenstrom. Schließlich ist der Tank so voll (Wasser und komprimierte Luft), dass der Druck des Zuflusses dafür sorgt, dass der Volumenstrom am Abfluss gleich dem am Zufluss ist. Da das Messgerät weit entfernt vom Tank liegt, misst dieses den ganzen Vorgang erst mit einer gewissen Verzögerung.

Je nachdem wie groß der Tank ist, verändert sich die Konstante  $T_1$ : Je kleiner der Tank, desto größer ist  $T_1$ . Aber auch das Verhältnis von Zuflussgröße zu Abflussgröße beeinflusst die Konstante  $T_1$ . Gleichzeitig bestimmt der Abstand zwischen dem Tank und dem Messgerät die Zeitkonstante  $T_t$ : Je größer der Abstand, desto größer ist  $T_t$ .

## 2.1 Größen

Im Folgenden werden Eingangs-, Zustands- und Ausgangsgrößen des Systems definiert:

Systemtheorie	Größe im System	Einheit
$u(t)$	Volumenstrom am Zufluss	$\frac{l}{min}$
$x(t)$	Volumenstrom am Abfluss	$\frac{l}{min}$
$y(t)$	Volumenstrom am Messgerät	$\frac{l}{min}$

## 2.2 Bezeichnung

Unser System hat ein proportionales Übertragungsverhalten, den Nennergrad 1 und eine Totzeit  $T_t$ . Daraus ergibt sich die Bezeichnung  $PT_1T_t$ .

## 2.3 Übertragungsfunktion

Formel 1 zeigt die Übertragungsfunktion, die in diesem Skript analysiert wird. Dabei gehen wir bei unserem System davon aus, dass  $T_1$  gleich 1 und  $T_t$  gleich 2 ist. Formel 3 zeigt eine etwas andere Darstellung der Formel, sodass die Eigenschaften des Systems leichter abzulesen sind.

$$\mathcal{G}(s) = \frac{Y(s)}{U(s)} \quad (1)$$

$$= \frac{1 \cdot e^{-2s}}{1 + s} \quad (2)$$

$$= \frac{1}{1 + s} \cdot e^{-2s} \quad (3)$$

## 2.4 Eingabe in Matlab

Für die Eingabe der Übertragungsfunktion in Matlab gibt es zwei Möglichkeiten.

In Computer-Algebra führt man  $s$  und  $G(s)$  als symbolischen Variablen ein und definiert anschließend die Übertragungsfunktion  $G(s)$ :

```
syms s, G(s)
G(s) = 1 / (1 + s) * exp(-2 * s)
```

Man erhält dasselbe Ergebnis wie bei der obigen, manuellen Definition von  $s$  und  $G(s)$ , wenn man den Transfer-Function-Befehl `tf()` in Matlab verwendet:

```
sys = tf([1], [1, 1], "IODElay", 2)
```

Hierbei werden in den beiden Vektoren die Koeffizienten vor  $s^0, s^1, s^2, \dots$  im Zähler und im Nenner angegeben. Bei komplizierteren Gleichungen hilft dabei der Befehl `expand(G(s))`, der einem die benötigten Koeffizienten liefert, was hier jedoch nicht nötig war. Darauf folgt der Befehl `IODelay`, mit dem die Totzeit  $t = 2$  angegeben wird.

### 3 Darstellungsformen des Systems

Im Folgenden werden die Darstellungen unseres Systems näher beleuchtet. Generell gilt: Mit jeder weiteren Darstellung verliert man bestimmte Informationen über das System, sodass die erste Darstellung die informationsreichste ist.

#### 3.1 Explizite Darstellung des Übertragungsoperators

Mit den Anfangswerten lässt sich  $x(t)$  und  $y(t)$  explizit berechnen:

$$\begin{aligned}x(t) &= e^{At} \cdot x(0) + \int_0^t e^{A(t-\tau)} Bu(\tau) d\tau \\y(t) &= Ce^{At} \cdot x(0) + C \cdot \int_0^t e^{A(t-\tau)} Bu(\tau) d\tau + Du(t)\end{aligned}$$

Hier ist bei der Eingabe in Matlab zu beachten, dass für  $e^{(\dots)}$  nicht `exp`, sondern `expm` verwendet wird, da man hier die Matrizenmultiplikation benötigt.

Durch Einsetzen der Heaviside-Funktion als Eingang und der jeweiligen Werte ergibt sich für unser System:

$$\begin{aligned}x(t) &= \int_0^t \left[ e^{-t+\tau} \right] d\tau & y(t) &= \int_0^t \left[ e^{-t+\tau} \right] d\tau \\x(t) &= \int_0^t \left[ e^{-t} \cdot e^{\tau} \right] d\tau & y(t) &= \int_0^t \left[ e^{-t} \cdot e^{\tau} \right] d\tau \\x(t) &= e^{-t} \cdot e^t - e^{-t} \cdot e^0 & y(t) &= e^{-t} \cdot e^t - e^{-t} \cdot e^0 \\x(t) &= 1 - e^{-t} & y(t) &= 1 - e^{-t}\end{aligned}$$

#### 3.2 Implizite Darstellung

##### 3.2.1 Zustandsraumdarstellung

Die allgemeine Form der Zustandsraumdarstellung lautet:

$$\begin{aligned}\dot{x} &= Ax + Bu \\y &= Cx + Du\end{aligned}$$

Mit dem Matlab-Befehl `ss(sys)` (ss für State Space) berechnet Matlab die Zustandsmatrizen für unser System. Somit ergibt sich die folgende Zustandsraumdarstellung:

$$\begin{aligned}\dot{x} &= \begin{bmatrix} -1 \end{bmatrix} x + \begin{bmatrix} 1 \end{bmatrix} u \\y &= \begin{bmatrix} 1 \end{bmatrix} x + \begin{bmatrix} 0 \end{bmatrix} u\end{aligned}$$

Sind die Zustandsmatrizen eines Systems bekannt, kommt man mit der folgenden Formel zur Übertragungsfunktion  $\mathcal{G}(s)$ , wobei  $I$  die Einheitsmatrix darstellt.

$$\mathcal{G}(s) = C \cdot (s \cdot I - A)^{-1} \cdot B + D$$

Sofern die Zustandsmatrizen gegeben sind, kann man das System alternativ mit der dazugehörigen Zustandsraumdarstellung übergeben, indem man `sys = ss(A, B, C, D)` in Matlab eingibt. Der Befehl `tf(sys)` liefert dann die Übertragungsfunktion.

In unserem System erhält man damit aber nicht die eigentliche Übertragungsfunktion, da die Totzeit  $T_t$  verloren geht.

Für die Funktionalbeziehung eines Totzeitglieds im Zeitbereich gilt:  $y(t) = u(t - T_t)$ . Somit ergibt sich für die Zustandsraumdarstellung eines zeitverzögerten Systems:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t - T_t) \\ y(t) &= Cx(t) + Du(t - T_t)\end{aligned}$$

In unserem System mit  $T_t = 2$  bedeutet das:

$$\begin{aligned}\dot{x}(t) &= \begin{bmatrix} -1 \end{bmatrix} x(t) + \begin{bmatrix} 1 \end{bmatrix} u(t - 2) \\ y(t) &= \begin{bmatrix} 1 \end{bmatrix} x(t) + \begin{bmatrix} 0 \end{bmatrix} u(t - 2)\end{aligned}$$

Anhand der Matrix  $D$ , die in unserem System eine Nullmatrix ist, lässt sich ablesen, dass es sich hier um ein nicht-sprungfähiges System handelt.

Außerdem ist ein System stabil, wenn die Eigenwerte der Matrix  $A$  einen negativen Realteil besitzen. Da die Eigenwerte bei unserem System  $\lambda = -1$  sind, ist das hier der Fall.

Ohne Matlab kann die Zustandsraumdarstellung in einfachen Fällen mithilfe des Substitutionstricks aus der Übertragungsfunktion errechnet werden. Für Systeme, bei denen auch  $\dot{u}, \ddot{u}, \dots$  in der Differentialgleichung vorkommt, ist das Ganze ein bisschen komplizierter, wird aber in Wikipedia beschrieben und kann entsprechend angewendet werden. Bei sprungfähigen Systemen sieht das noch etwas anders aus und es muss erst eine Polynomdivision gemacht werden, um  $D$  zu erhalten. Im Mehrgrößenfall ist es noch komplizierter.

## Anfangswerte

Für die Simulation sind Anfangswerte notwendig, da der Start der Funktion definiert sein muss. Hierbei werden die linksseitigen Grenzwerte verwendet, da sonst eventuelle Sprünge miteinbezogen würden. Es gilt:

$$\begin{aligned}y(0^-) &= Cx(0^-) + Du(0^-) \\ \dot{y}(0^-) &= CAx(0^-) + CBu(0^-) + D\dot{u}(0^-)\end{aligned}$$

Die Summanden, in denen  $\dot{u}, \ddot{u}, \dots$  vorkommt, können vernachlässigt werden. Schließlich sind die linksseitigen Anfangswerte aufgrund der Heavyside-Funktion gleich 0.

Darüber hinaus gibt es einen Zusammenhang mit der Ein-/Ausgangs-Differentialgleichung: Die Anfangswerte des Zustandsraumes  $x$  müssen mit denen der Ein-/Ausgangs-Differentialgleichung  $y$  korrespondieren. Zur Berechnung werden folgende Formeln verwendet:

$$\begin{aligned}\begin{pmatrix} y \\ \dot{y} \\ \ddot{y} \end{pmatrix} &= \begin{pmatrix} c^T \\ c^T A \\ c^T A^2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \\ \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} &= \begin{pmatrix} c^T \\ c^T A \\ c^T A^2 \end{pmatrix}^{-1} \begin{pmatrix} y \\ \dot{y} \\ \ddot{y} \end{pmatrix}\end{aligned}$$

Weiterhin gehen wir bei unserem System davon aus, dass der Tank zu Beginn leer ist, somit kein Abfluss vorliegt. Daraus ergibt sich  $x(0^-) = 0$  und mit voriger Formel gilt:

$$\begin{aligned} y &= 0 \\ x_1 &= 0 \end{aligned}$$

### 3.2.2 Integralgleichung

Im Gegensatz zur Differentialgleichung ergibt die Integralgleichung „mild solutions“, welche Sprünge abbilden können. Sie ergibt sich durch Integration der Differentialgleichung mit  $\int_0^t$ . Für die Integralgleichung gilt:

$$x(t) = x(0) + \int_0^t Ax(\tau) + Bu(\tau) d\tau \qquad y(t) = y(0) + \int_0^t Cx(\tau) + Du(\tau) d\tau$$

Für unser System folgt daraus:

$$x(t) = \int_0^t \begin{bmatrix} -1 \end{bmatrix} x(\tau) + \begin{bmatrix} 1 \end{bmatrix} u(\tau) d\tau \qquad x(t) = \int_0^t \begin{bmatrix} -1 \end{bmatrix} x(\tau) + \begin{bmatrix} 1 \end{bmatrix} u(\tau) d\tau$$

### 3.2.3 Differentialgleichung

Aus unserem physikalischen Modell lässt sich folgende Differentialgleichung ableiten:

$$\begin{aligned} \dot{x} &= u - x & \text{mit } x(0) &= 0 \\ y(t) &= x(t - 2) \end{aligned}$$

## 3.3 Ein-/Ausgangs Differentialgleichung

Für unser System ergibt sich unter Vernachlässigung des Totzeitglieds:

$$\begin{aligned} G_{PT_1}(s) &= \frac{Y(s)}{U(s)} = \frac{1}{1+s} \\ Y(s)(1+s) &= (1) \cdot U(s) \\ Y(s) + Y(s) \cdot s &= 1 \cdot U(s) \\ \dot{y} + y &= u \end{aligned}$$

In dieser Differentialgleichung taucht die Totzeit  $T_t$  nicht auf, da das Totzeitglied nicht mit einer gewöhnlichen Differentialgleichung beschrieben werden kann. Für die Funktionalbeziehung eines Totzeitglieds im Zeitbereich gilt:  $y(t) = u(t - T_t)$ .

Für unser System mit der Totzeit  $T_t = 2$  folgt somit:

$$\dot{y}(t) + y(t) = u(t - 2) \qquad \text{mit } y(0) = 0$$

## 3.4 Übertragungsfunktion

Die Übertragungsfunktion ergibt sich durch die Laplace-Transformation der Ein-/Ausgangs-Differentialgleichung, wird für unser System in Kapitel 2.3 definiert und lautet:

$$\mathcal{G}(s) = \frac{1}{1+s} \cdot e^{-2s}$$

### 3.5 Systemantworten

Im Folgenden werden die Antworten des Systems auf den Dirac-Impuls  $\delta(t)$  und den Einheitssprung  $\sigma(t)$  näher erläutert. Die Sprungantwort  $h(t)$  und die Impulsantwort  $g(t)$  sind jeweils ineinander überführbar: Durch Differenzieren von  $h(t)$  erhält man  $g(t)$  und durch Integrieren von  $g(t)$  erhält man  $h(t)$ .

#### 3.5.1 Gewichtsfunktion (Impulsantwort)

Die Impulsantwort bezeichnet die Antwort auf den technischen Dirac-Impuls. Die Gewichtsfunktion  $g(t)$  ist die dimensionslose Antwort eines Systems auf einen Dirac-Impuls.

In Matlab lässt sich die Gewichtsfunktion sowohl graphisch als auch analytisch ausgeben. `ilaplace()` liefert die inverse Laplace-Transformation der Übertragungsfunktion:

```
syms g(t)
g(t) = ilaplace(G(s))
```

Matlab Ausgabe:  $g(t) = \text{heaviside}(t - 2) * \exp(2 - t)$

Mathematische Notation:

$$g(t) = 1(t - 2) \cdot e^{(2-t)}$$

wobei  $1(t)$  für die Heavyside-Funktion steht.

Einen Plot der Impulsantwort erhält man mit `impulse(sys)`.



Abbildung 2: Impulsantwort

#### 3.5.2 Übergangsfunktion (Sprungantwort)

Die Sprungantwort bezeichnet die Antwort des Systems auf den technischen Einheitssprung. Die Übergangsfunktion  $h(t)$  ist somit die dimensionslose Antwort des Systems auf den Einheitssprung. Ein Beispiel ist die Heavyside-Funktion, die wie folgt definiert ist:

$$1(t) = \begin{cases} 0 & \text{für } t < 0 \\ \frac{1}{2} & \text{für } t = 0 \\ 1 & \text{für } t > 0 \end{cases}$$

In Matlab lässt sich das mit dem Befehl `step(sys)` simulieren, wodurch man die numerische Lösung erhält, wie in folgender Abbildung zu sehen ist.

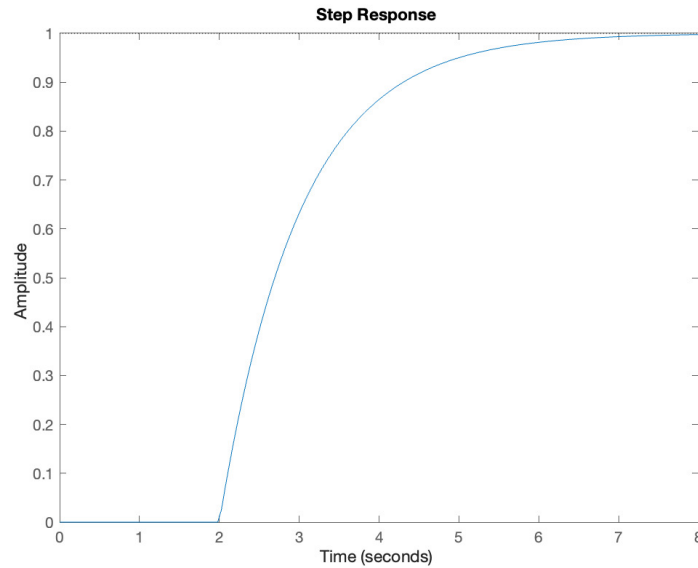


Abbildung 3: Sprungantwort

Die Übergangsfunktion  $h(t)$  lässt sich aber auch analytisch berechnen und man erhält die explizite Lösung.

```
H(s) = G(s)/s
syms h(t)
h(t) = ilaplace(H(s))
```

Matlab Ausgabe:  $h(t) = -\text{heaviside}(t - 2) * (\exp(2 - t) - 1)$

Mathematische Notation:

$$h(t) = -1(t - 2) \cdot (e^{(2-t)} - 1)$$

## 3.6 Frequenzgang

Zum Frequenzgang  $G(j\omega)$  gelangt man, indem man in  $\mathcal{G}(s)$  die Variable  $s$  durch  $j\omega$  ersetzt. Das  $\omega$  steht für die Frequenz in  $\frac{\text{rad}}{\text{s}}$  (Erinnerung:  $\omega = 2\pi f$ ). Das  $j$  steht für die imaginäre Einheit, da  $i$  in der Systemtheorie für den Strom steht und somit durch  $j$  ersetzt wird.

Das bedeutet, dass  $G(j\omega)$  für jede Frequenz  $\omega$  eine komplexe Zahl  $j$  mit Real- und Imaginärteil oder einen Betrag mit Phase darstellt. Systemtheoretisch entspricht der Frequenzgang  $G(j\omega)$  einem Schnitt der komplexen Funktion von  $\mathcal{G}(s)$  entlang der imaginären Achse:  $\mathcal{G}(s)|_{s=j\omega}$

### 3.6.1 Nyquist-Plot (Ortskurve)

Im Nyquist-Plot wird der über  $\omega$  parametrisierte Frequenzgang als Kurve in der komplexen Ebene mit Real- und Imaginärteil dargestellt.

Mit dem Matlab-Befehl `nyquist(sys)` wird ein Plot der Kurve des Frequenzgangs  $G(j\omega)$  ausgegeben. Matlab zeigt die Kurve des Frequenzgangs  $G(j\omega)$  immer für  $\omega$  zwischen  $-\infty$  und  $+\infty$  an:





Abbildung 4: Nyquist-Plot

Dabei ist es möglich, mit dem Cursor entlang der Kurve zu fahren, um sich den Real- und Imaginärteil zu dem jeweiligen Punkt anzeigen zu lassen:



Abbildung 5: Screenshot: Informationen im Nyquist-Plot

### 3.6.2 Bode-Diagramm

Im Bode-Diagramm sind der Betrag des Frequenzgangs über  $\omega$  in Dezibel ( $dB$ ) und die Phase des Frequenzgangs über  $\omega$  in Grad ( $^\circ$ ) dargestellt.

Streng genommen wird im oberen Diagramm der Abbildung 6 die Amplitudenverstärkung dargestellt, die ein Sinus-Signal an der Frequenz  $\omega$  stationär erfährt (wenn die Eigenvorgänge abgeklungen sind und sich das System eingeschwungen hat), während im unteren Diagramm die Phasenverschiebung zu sehen ist.

Der dazugehörige Matlab-Befehl ist `bode(sys)`, wobei dieser Befehl intern eine doppelt logarithmische Darstellung verwendet. Die Diagramm-Achse, auf der  $\omega$  aufgetragen ist, ist dabei logarithmisch skaliert, damit die Kurven schön aussehen.

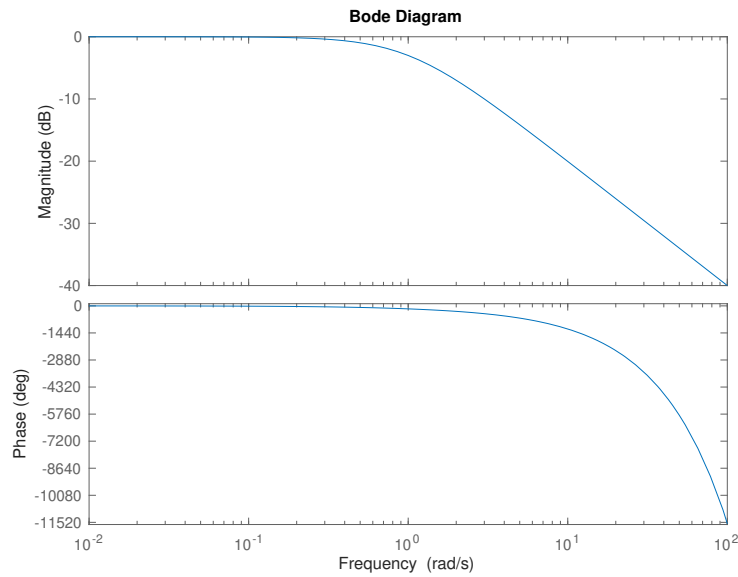


Abbildung 6: Bode-Plot

Anhand des Verlaufs zu Beginn des Bode-Plots lässt sich das Übertragungsverhalten feststellen. Beim P-Verhalten startet der Verlauf mit einer waagerechten Geraden, beim I-Verhalten mit einer Geraden mit negativer Steigung und beim D-Verhalten mit einer Geraden mit positiver Steigung. In unserem Bode-Plot ist eine waagerechte Linie zu erkennen, was mit dem P-Verhalten unseres Systems korrespondiert.

Außerdem lässt sich aus dem Beginn des Bode-Diagramms auch die Statik-Information entnehmen: Hierfür betrachtet man den Wert  $D$ , den der Bode-Plot an der Stelle  $f = 0$  hat. Aufgrund der logarithmischen Darstellung gibt es im Bode-Plot kein  $f = 0$ , jedoch ist es für uns ausreichend, den kleinsten Wert ( $f = 10^{-2}$ ) zu betrachten.

Nun nutzt man die Formel für die Dämpfung und stellt diese um:

$$20 \cdot \log_{10} \mathcal{K} = D$$

$$\mathcal{K} = 10^{\frac{D}{20}}$$

Da unser System bei  $f = 10^{-2}$  im Bode-Plot einen Wert von  $D \approx 0$  hat, ergibt sich:

$$\mathcal{K} = 10^{\frac{0}{20}} = 1$$

### 3.6.3 Simulink-Simulation

Mithilfe von Simulink lässt sich die Übertragungsfunktion simulieren. Die Schaltungen zeigen den Aufbau mit einem Sinus-Signal als Eingang. Dabei werden verschiedene Blöcke aus der Simulink-Bibliothek verwendet:

1. Sin Wave (Source-Ordner)
2. Transfer Function (Continuous-Ordner)
3. Transport Delay (Continuous-Ordner)
4. Scope (Sink-Ordner)
5. Add (Math-Operations-Ordner)
6. Integrator (Continuous-Ordner)

Eine Möglichkeit besteht darin, in den Block „Transfer Function“ die Übertragungsfunktion einzugeben:

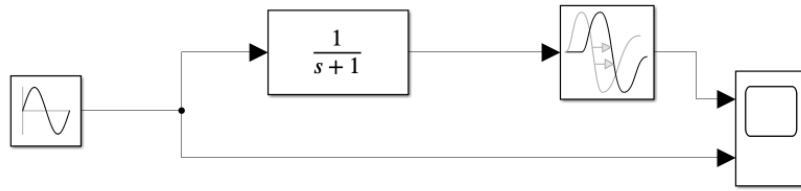


Abbildung 7: Darstellung des Systems anhand der Übertragungsfunktion

Die andere Möglichkeit ist, das System mithilfe von Integratoren darzustellen. Man stellt hierfür die Ein-/Ausgangs-Differentialgleichung nach der höchsten Ableitung um und integriert. Das bringt den Vorteil, dass man für die Integratoren Anfangswerte setzen kann:

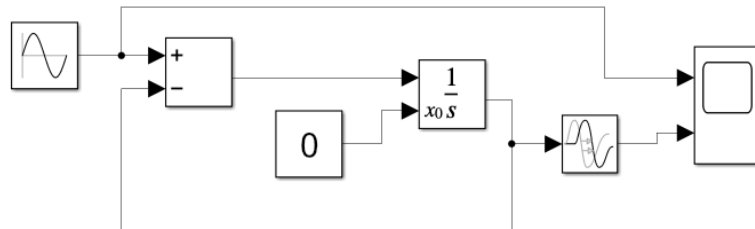


Abbildung 8: Darstellung des Systems anhand der Ein-/Ausgangs-Differentialgleichung

Die beiden abgebildeten Simulink-Schaltungen führen bei gleicher Eingangsfunktion auf die gleiche Ausgangsfunktion. Mit einem Klick auf den Oszillator wird diese Ausgangsfunktion sowie die am Eingang anliegende Sinus-Funktion in Form eines Graphen ausgegeben (siehe Abbildung 9 & 10). Dabei ist die blaue Linie die Sinus-Funktion am Eingang mit einer entsprechenden Frequenz  $\omega$  in  $\frac{rad}{s}$  und die rote Linie die Antwort  $y(t)$  unseres Systems.

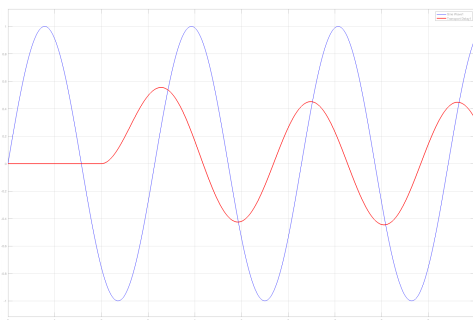


Abbildung 9: Systemantwort bei  $\omega = 2$

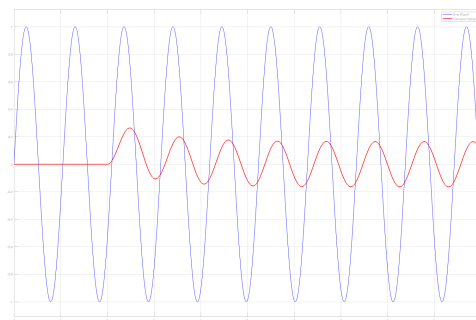


Abbildung 10: Systemantwort bei  $\omega = 6$

### 3.7 Vergleich Bode-Plot, Simulink und Handrechnung

Im Folgenden werden die Phasenverschiebung und Amplitudenverstärkung unseres Systems anhand des Bode-Diagramms, der Simulink-Schaltung und einer Handrechnung betrachtet und anschließend verglichen. Exemplarisch dient dabei die Frequenz  $\omega_0 = 6 \frac{rad}{s}$ .

Erhält das System ein Eingangssignal  $u(t) = A \cdot \sin(\omega_0 t + \varphi_0)$ , so gilt für den Ausgang:

$$y(t) = \text{Eigenvorgang} + y_{stat}(t)$$

Wobei für die Statik-Information  $y_{stat}$  gilt:

$$y_{stat}(t) = |\mathcal{G}(j\omega_0)| \cdot A \cdot \sin(\omega_0 t + \varphi_0 + \arg \mathcal{G}(j\omega_0))$$

Dabei entspricht  $|\mathcal{G}(j\omega_0)|$  der Amplitudenverstärkung und  $\arg \mathcal{G}(j\omega_0)$  der Phasenverschiebung.

### 3.7.1 Handrechnung

Berechnung des Frequenzgangs:

$$\mathcal{G}(s) = \frac{1}{s+1} \cdot e^{-2s}$$

$$\mathcal{G}(j\omega) = \frac{1}{j\omega+1} \cdot e^{-2j\omega}$$

Berechnung der Amplitudenverstärkung für  $\omega_0 = 6$ :

Für den Betrag eines Bruchs gilt:  $\text{BetragBruch} = \text{BetragZähler} / \text{BetragNenner}$

Für den Betrag von  $e^{-aj}$  gilt:  $e^{-aj} = 1$

$$\mathcal{G}(j6) = \frac{1}{6j+1} \cdot e^{-2j \cdot 6}$$

$$|\mathcal{G}(j6)| = \left| \frac{1}{6j+1} \cdot e^{-12j} \right|$$

$$|\mathcal{G}(j6)| = \frac{1}{\sqrt{37}} \cdot 1 \approx 0.1644$$

Berechnung der Phasenverschiebung für  $\omega_0 = 6$ :

Für das Argument  $\arg$  eines komplexen Bruches gilt:

$$\arg\left(\frac{a+bj}{c+dj}\right) = \text{WinkelZähler} - \text{WinkelNenner} = \arctan\frac{b}{a} - \arctan\frac{d}{c}$$

Für das Argument  $\arg$  einer Zeitverzögerung gilt:  $\arg e^{-aj} = -a$

$$\arg \mathcal{G}(j6) = \arctan\left(\frac{0}{1}\right) - \arctan\left(\frac{6}{1}\right) - 12$$

$$\arg \mathcal{G}(j6) \approx 0 - 1.4 - 12 = -13.4$$

$$\Rightarrow -13.4 \cdot \frac{180}{\pi} \approx -768.726^\circ$$

### 3.7.2 Bode-Plot

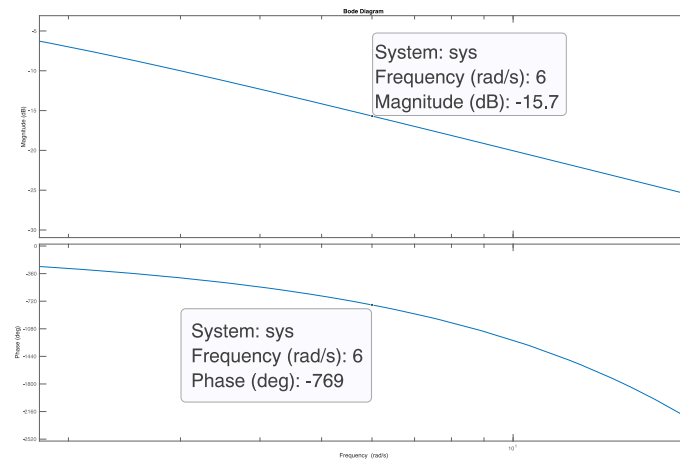


Abbildung 11: Bode-Plot mit Zusatzinformationen an der Stelle  $\omega_0 = 6$

Im Bodeplot lässt sich im oberen Bild für  $\omega_0 = 6$  eine Verstärkung von  $-15.7dB$  ablesen. Daraus lässt sich der Verstärkungsfaktor errechnen:

$$20 \cdot \log(\mathcal{K}) = -15.7$$

$$\mathcal{K} = 10^{-\frac{15.7}{20}}$$

$$\mathcal{K} \approx 0.1640$$

Im unteren Bild lässt sich für die Phasenverschiebung ein Winkel von  $-769^\circ$  ablesen.

### 3.7.3 Simulink

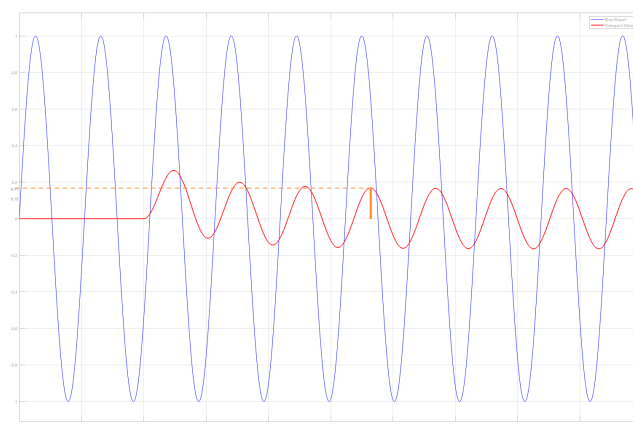


Abbildung 12: Messung der Amplitudenverstärkung der eingeschwungenen Ausgangsfunktion

Aus dem Simulink-Ergebnis lässt sich ablesen, dass die Amplitudenverstärkung bei eingeschwungener Ausgangsfunktion ungefähr 0.17 beträgt.

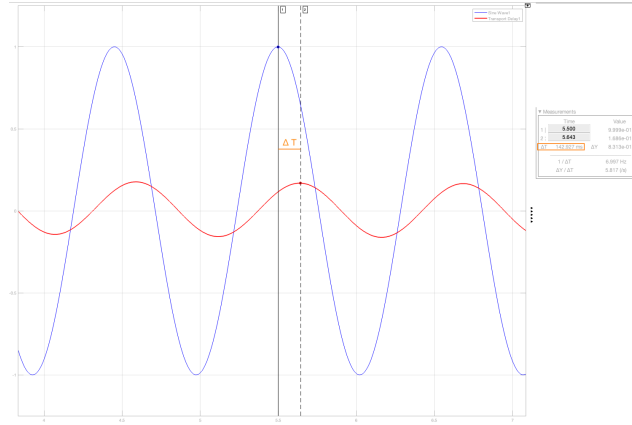


Abbildung 13: Messung der Zeitdifferenz zwischen der Sinus-Schwingung und der eingeschwungenen Ausgangsfunktion

Die Phasenverschiebung in Grad wird berechnet durch  $\varphi_0 = (\frac{\Delta t}{T} + T_t) \cdot 360^\circ$ , wobei  $T$  die Periodendauer ist, die mithilfe von  $T = \frac{2\pi}{\omega} = \frac{2\pi}{6} = \frac{\pi}{3}$  ausgerechnet werden kann.

Aus dem Simulink-Plot kann ein  $\Delta t = 0.142s$  abgelesen werden. Alle Werte werden in die Formel eingesetzt:

$$\begin{aligned}\varphi_0 &= \left(\frac{\Delta t}{T} + T_t\right) \cdot 360^\circ \\ \varphi_0 &= \left(\frac{0.142 \cdot 3}{\pi} + 2\right) \cdot 360^\circ \\ \varphi_0 &\approx 768.81^\circ\end{aligned}$$

Da die Ausgangsfunktion nach rechts verschoben ist, muss das Vorzeichen negativ sein, sodass das Ergebnis  $-768.81^\circ$  beträgt.

### 3.7.4 Vergleich

In der folgenden Tabelle sind die einzelnen Werte aus den verschiedenen Ermittlungen miteinander verglichen:

Ermittlung	Amplitudenverstärkung	Phasenverschiebung
Handrechnung	0.1644	$-768.726^\circ$
Bode-Plot	0.1640	$-769^\circ$
Simulink	0.17	$-768.81^\circ$

Es ist zu erkennen, dass die Werte mit minimalen Abweichungen (Rundungsfehler) übereinstimmen. Das bedeutet, dass unsere Rechnungen und Simulationen richtig sind.

## 3.8 Statische Kennlinie

Der Verlauf der statischen Kennlinie ist abhängig vom Übertragungsverhalten des Systems. Für P-Verhalten ist die statische Kennlinie eine Ursprungsgerade mit der Steigung  $\mathcal{K}$ , für I-Verhalten eine Gerade, die der x-Achse entspricht, und für D-Verhalten lässt sich formal keine statische Kennlinie angeben.

Da es sich bei unserem System um ein System mit proportionalen Übertragungsverhalten handelt, entspricht die statische Verstärkung  $\mathcal{K} = G(0)$  bzw.  $\mathcal{K} = \lim_{t \rightarrow \infty} h(t)$ . Diese lässt sich in Matlab auch mit dem Befehl `dcgain(sys)` berechnen.

Somit ergibt sich bei unserem System eine Ursprungsgerade der Steigung 1 als statische Kennlinie.

In weniger trivialen Fällen lässt sich die statische Kennlinie folgendermaßen aufnehmen: Man stellt einen konstanten Wert  $c$  für  $u(t)$  ein, wartet bis alle Eigenvorgänge abgeklungen sind und liest dann den zugehörigen  $y$ -Wert ab. Dies wiederholt man für verschiedene  $c$  und verbindet schließlich alle Punkte durch eine Linie, welche letztendlich die statische Kennlinie darstellt.

Dieses Verfahren erklärt, wieso bei instabilen System keine statische Kennlinie gezeichnet werden kann: Diese schwingen sich nie ein, die Eigenvorgänge klingen nie ab.

Noch ein anderer Weg, die statische Verstärkungsmatrix  $K$  auszurechnen, verwendet die Zustandsmatrizen (im mehrvariablen Fall ist  $K$  eine Matrix, die die jeweiligen statischen Verstärkung der einzelnen Eingangssignale angibt).

Dabei wird die Ableitung von  $x$  gleich 0 gesetzt, die Gleichung umgestellt, und in die Gleichung von  $y$  eingesetzt. Stellt man diese Gleichung erneut um, erhält man  $K$ :

$$\dot{x} = Ax + Bu$$

$$0 \stackrel{!}{=} Ax + Bu$$

$$x = -A^{-1}Bu$$

$$y = Cx + Du$$

$$y = C(-A^{-1}Bu) + Du$$

$$y := Ku$$

$$K = -CA^{-1}B + D$$

Mithilfe des Matlab-Befehls `fplot(@(x) 1 * x)` lässt sich unsere statische Kennlinie graphisch darstellen:

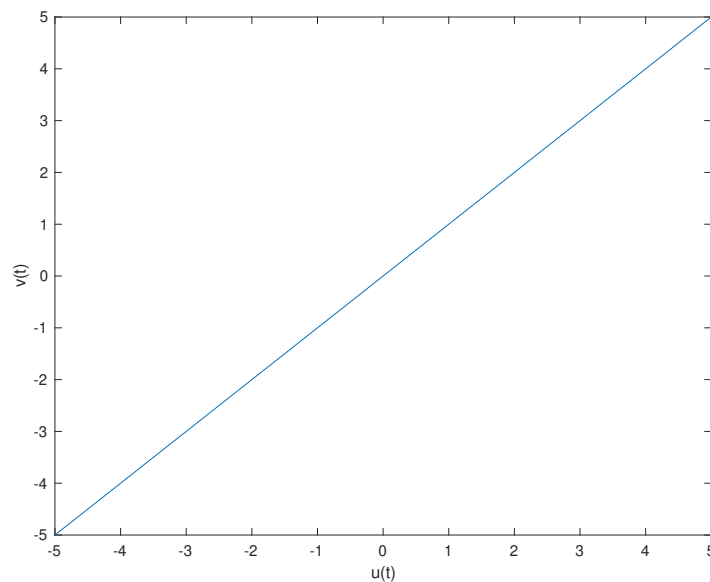


Abbildung 14: Statische Kennlinie

Bei Systemen mit einem Eingang und zwei Ausgängen verwendet man stattdessen ein 2-Balken-Diagramm, bei zwei Eingängen und einem Ausgang wird eine Eingangsgröße diskret und die andere kontinuierlich dargestellt, wodurch sich eine Funktionenschar ergibt.

Der Nachteil der statischen Kennlinie ist, dass sie im Gegensatz zum Pol-Nullstellen-Plot keine Dynamik-Informationen enthält.

### 3.9 Pol-Nullstellen-Plot

Abbildung 15 zeigt den Pol-Nullstellen-Plot zu unserem System. Der passende Matlab-Befehl hierfür lautet `pzplot(sys)`. Matlab markiert mit **o** die Nullstellen und mit **x** die Polstellen des Systems (doppelte Polstellen mit doppel-**x**).

In unserem Fall hat das System aber keine Nullstellen, da der Zählergrad 0 ist, weshalb in der Abbildung kein **o** zu sehen ist. Da der Nenner unseres Systems  $1 + s$  aber den Grad 1 hat, hat unser System eine reale Polstelle bei  $s = -1$ , was in der Abbildung mit den **x** gekennzeichnet ist.

Aufgrund dieses negativen Realteils handelt es sich um eine stabile Polstelle. Da somit alle Polstellen in der offenen linken Halbebene liegen, ist das System stabil.

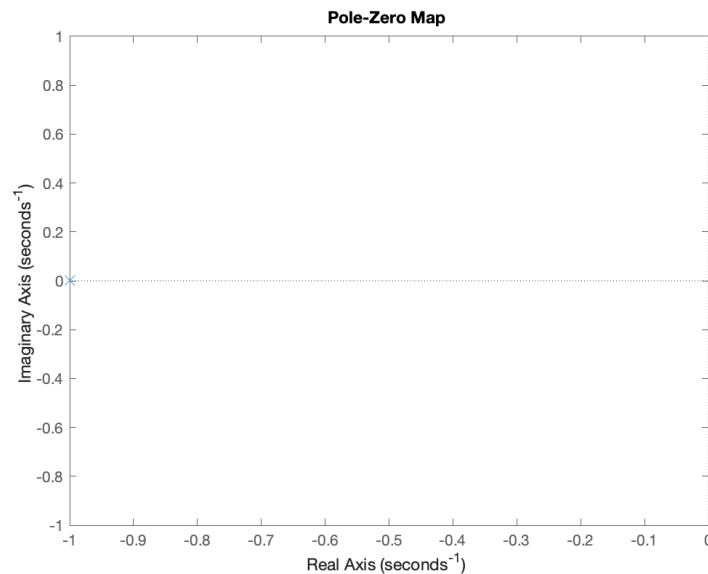


Abbildung 15: Pol-Nullstellen-Plot

Die Pole der Übertragungsfunktion korrespondieren mit den Eigenwerten der System-Matrix  $A$  unter der Voraussetzung, dass das System vollständig steuerbar und beobachtbar ist. Das heißt, dass keine Pol-Nullstellen-Kürzung auftreten darf. In unserem System ist das unkompliziert: Unsere System-Matrix  $A$  enthält nur das Element  $-1$  und passt somit zur Polstelle  $s = -1$ .

Der Nachteil des Pol-Nullstellen-Plots ist, dass er im Gegensatz zur statischen Kennlinie keine Information über die statische Verstärkung enthält.