Signale & Systeme
Analyse eines Systems

Matrikelnr.: 9085107 & 2147057
Dozent: Apl. Prof. Dr. Lutz Gröll
Kurs: TINF22B1
System: $\mathrm{PT}_1\mathrm{T}_t$-Glied

2. Dezember 2023

Jede Differentialgleichung bekommt Anfangswerte

Abbildungsbenennungen

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 unten Ende des Wassertanks befindet sich ein deutlich kleinerer Abfluss, an den eine lange Wasserleitung angeschlossen ist. Erst am Ende der Wasserleitung befindet sich ein Messgerät, welcher den Volumenstrom misst. Eine beispielhafte Skizze sieht folgendermaßen aus:

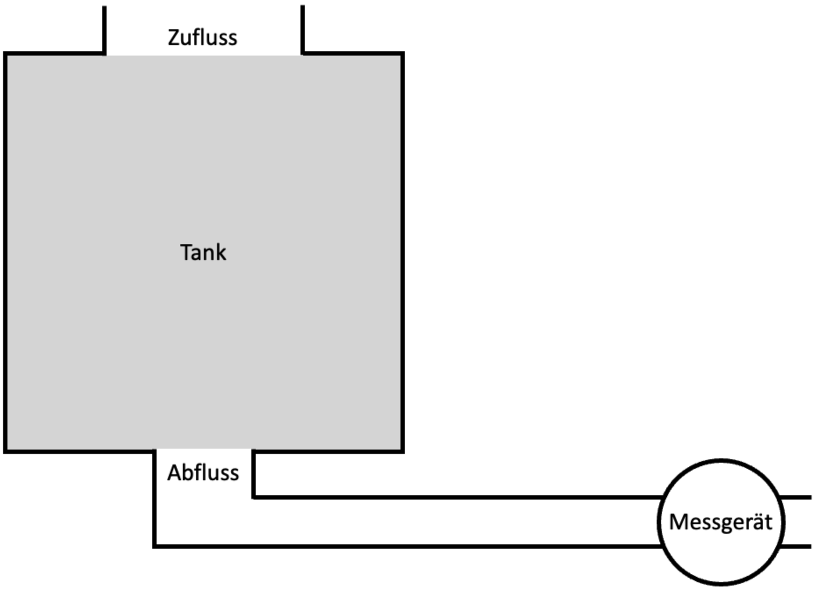

Abbildung 1: Skizze des Systems

Der Ablauf stellt sich wie folgt dar: Über den Zufluss fließt Wasser in den Tank. Da der Abfluss aber deutlich kleiner ist, fließt erstmal deutlich weniger Wasser heraus. Erst wenn der Tank voller wird, erhöht sich der Druck und es kommt zu einem immer größeren Volumenstrom bis der Tank schließlich so voll (Wasser und komprimierte Luft) ist, dass der Druck des Zuflusses dafür sorgt, dass der Volumenstrom am Abfluss gleich dem am Zufluss ist. Da das Messgerät aber 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 Abflussgrößte beinflusst die Konstante T_1. Gleichzeitig bestimmt der Abstand zwischen dem Tank und dem Messgerät die Zeitkonstante T_t: Je größer der Abstand, destor 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

Das System hat den Nennergrad 1, eine Totzeit T_t und ein proportionales Übertragungsverhalten. 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 gehen 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.

$$
\begin{aligned}
G(s) &= \frac{Y(s)}{U(s)} && (1)\\
&= \frac{1 \cdot e^{-2s}}{1+s} && (2)\\
&= \frac{1}{1+s} \cdot e^{-2s} && (3)
\end{aligned}
$$

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 kann anschließend die Übertragungsfunktion `G(s)` definieren:

```
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, ... 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.

In beiden Fällen wird in Matlab damit die Formel 3 dargestellt.

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:

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

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

Für unsere System ergibt sich durch Einsetzen der jeweiligen Werte und der Heaviside-Funktion als Eingang:

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

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 steht für State Space) berechnet Matlab automatisch die Werte für unser System. Somit ergibt sich für unser System 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 Matrizen A, B, C, D eines Systems bekannt, kommt man mit der folgenden Formel zur Übertragungsfunktion $G(s)$, wobei I die Einheitsmatrix darstellt.

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

Sofern die Matrizen gegeben sind, könnte man das System alternativ auch mit der dazugehörigen Zustandsraumdarstellung übergeben, indem `sys = ss(A, B, C, D)` in Matlab eingegeben wird. Der Befehlt `tf(sys)` liefert dann die Übertragungsfunktion über die Zustandsraumdarstellung.

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 also:

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

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

Ohne Matlab kann die Zustandsraumdarstellung in einfachen Fällen mithilfe des Substitutionstricks aus der Übertragungsfunktion erstellt werden. Für Systeme, bei denen auch $\dot{u}, \ddot{u}, \ldots$ 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 müssen die linken Grenzwerte verwendet werden, 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 $\dot{u}, \ddot{u}, \ldots$ vorkommt, können vernachlässigt werden. Schließlich sind die linksseiten 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:

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

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

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:

$$
\begin{pmatrix} y \\ \dot{y} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}
$$

$$
\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}
$$

3.2.2 Integralgleichung

Im Gegensatz zur Differentialgleichung ergibt die Integralgleichung „mild solutions“, welche Sprünge abbilden können. Als Integralgleichung gilt:

$$
x(t) = x(0) + \int_0^t Ax(\tau) + Bu(\tau)\, d\tau
$$

Für unser System ergibt sich somit:

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

3.2.3 Differentialgleichung

Aus unserem physikalischen Modell lässt sich folgende Differenzialgleichung ableiten:

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

3.3 Ein- /Ausgangs Differentialgleichung

Für unser System ergibt sich unter Vernachlässigung des Totzeitglieds:

$$
\begin{aligned}
G(s) &= \frac{1}{1+s} \cdot e^{-2s} \\
G_{PT_1}(s) = \frac{Y(s)}{U(s)} &= \frac{1}{1+s} \\
Y(s)(1+s) &= (1) \cdot U(s) \\
Y(s) + Y(s) \cdot s &= 1 \cdot U(s) \\
\dot{y}(t) + y(t) &= u(t)
\end{aligned}
$$

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:

$$\dot{y}(t) + y(t) = u(t - 2)$$

Vergessen werden darf auch hier nicht der Anfangswert. Es gilt:

$$y(0) = 0$$

3.4 Übertragungsfunktion

Formel 4 zeigt die Übertragungsfunktion die in diesem Skript analysiert wird. Dabei gehen wir bei unserem System gehen davon aus, dass T_1 gleich 1 und T_t gleich 2 ist.Formel 6 zeigt eine etwas andere Darstellung der Formel, sodass die Eigenschaften des Systems leichter abzulesen sind.

$$G(s) = \frac{Y(s)}{U(s)} \tag{4}$$

$$G(s) = \frac{1 \cdot e^{-2s}}{(1+s)} \tag{5}$$

$$G(s) = \frac{1}{1+s} \cdot e^{-2s} \tag{6}$$

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 abgetastete Antwort auf den technischen Dirac-Impuls. Die Gewichtsfunktion $g(t)$ ist die Antwort eines Sytems auf einen Dirac-Impuls $\delta(t)$.
In Matlab lässt sich die Gewichtsfunktion sowohl graphisch ausgeben als auch analytisch berechnen. Mithilfe des Befehls `ilaplace()`, wird die laplace-Transformierte Gewichtsfunktion $g(t)$ zurückliefert:

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

Matlab Ausgabe: `g(t) = 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.

Mit dem Matlab-Befehl `impulse(sys)` wird ein Plot der Impulsantwort erstellt, was in folgender Abbildung zu sehen ist.

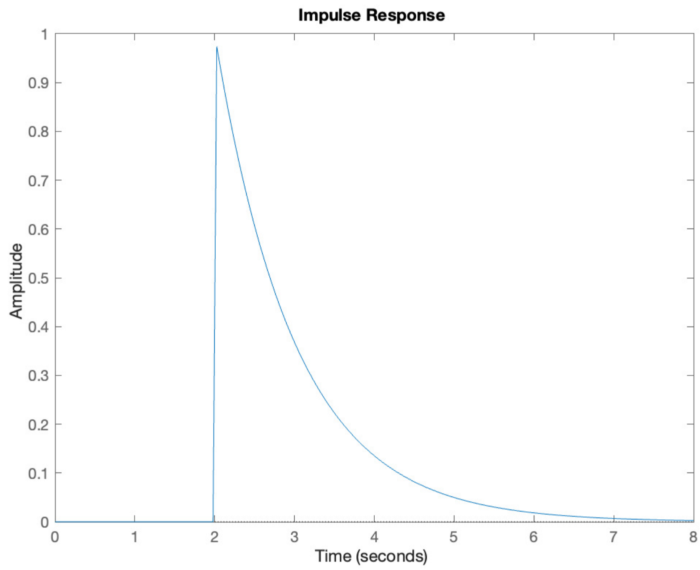

Abbildung 2: Impulsantwort

3.5.2 Übergangsfunktion (Sprungantwort)

Die Spungantwort bezeichnet die abgetastete Antwort des Systems auf den technischen Einheitssorung. Übergangsfunktion $h(t)$ ist die Antwort des Systems auf den Einheitssprung $\sigma(t)$. Ein Beipsiel ist die Heaviside-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.

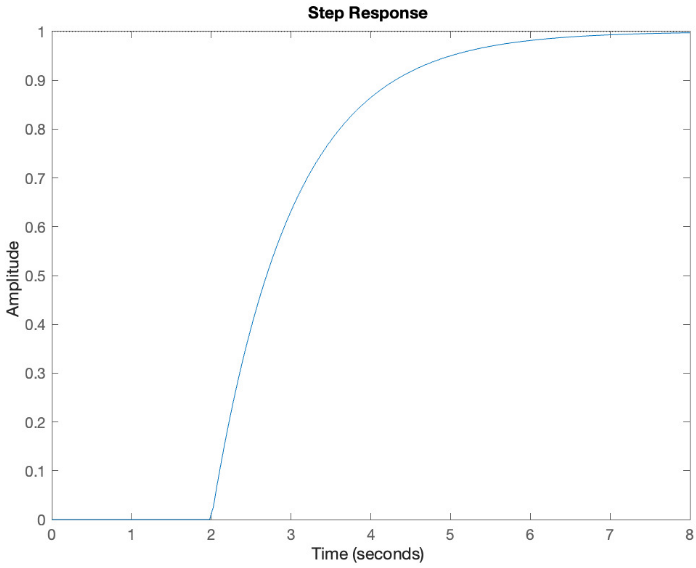

Abbildung 3: Sprungantwort

Die Übergangsfunktionfunktion $h(t)$ lässt sich aber auch analytisch berechnen und man erhält die explizite Lösung. Hierbei hilft einem der Befehl `ilaplace()`, der die Laplace-Transformierte zurückliefert:

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

Matlab Ausgabe: `h(t) = -heaviside(t - 2)*(exp(2 - t) - 1)`
Mathematische Notation:

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

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

3.6 Frequenzgang

Zum Frequenzgang $G(j\omega)$ gelangt man, indem man in $G(s)$ die Variable s durch $j\omega$ ersetzt. Das ω steht für die Frequenz in $\frac{rad}{s}$ (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 Frequzenz ω 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 $G(s)$ entlang der imaginären Achse: $G(s)|_{s=j*\omega}$

3.6.1 Nyquist-Plot (Ortskurve)

Im Nyquist-Plot wird der über ω 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 $G(\omega j)$ ausgegeben. Matlab zeigt die Kurve des Frequenzgangs $G(j\omega)$ immer für ω zwischen $-\infty$ und $+\infty$ an:

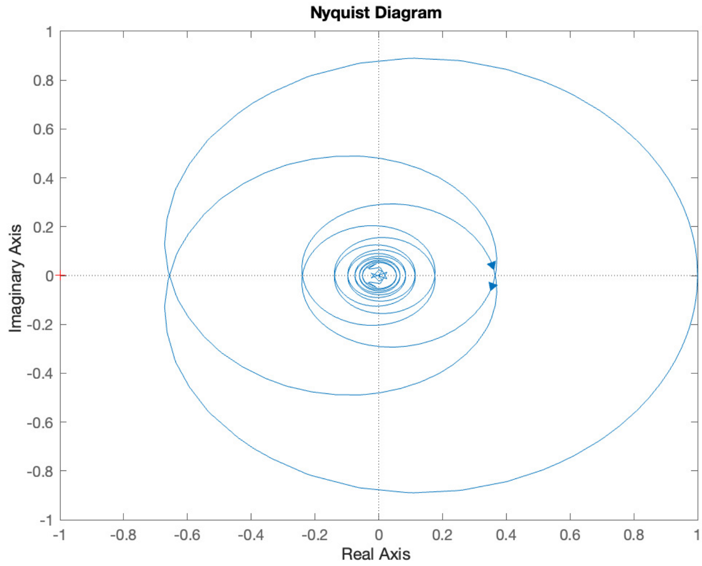

Abbildung 4: Nyquist-Plot

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:

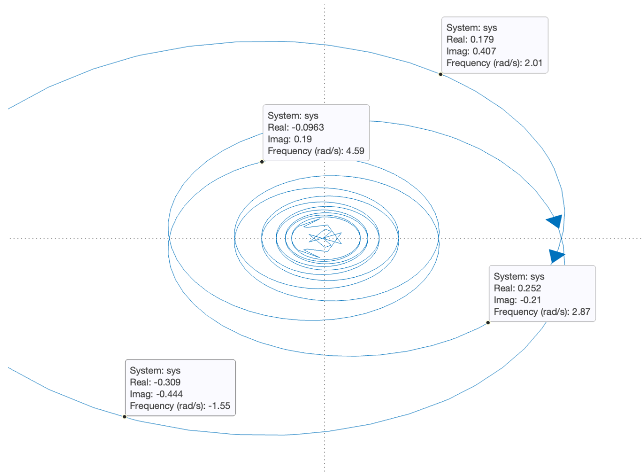

Abbildung 5: Screenshot: Informationen im Nyquist-Plot

3.6.2 Bode-Diagramm

Im Bode-Diagramm sind der Betrag des Frequenzgangs über ω in Dezibel(dB) und die Phase des Frequenzgangs über ω in Grad($^\circ$) dargestellt.
Streng genommen wird im oberen Diagramm der Abbildung 6 die Amplitudenverstärkung dargestellt, die ein Sinus-Signal an der Frequenz ω 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 ω aufgetragen ist, ist dabei logarithmisch skaliert, während der Betrag in Dezibel (dB) angegeben wird, damit die Kurven schön aussehen.

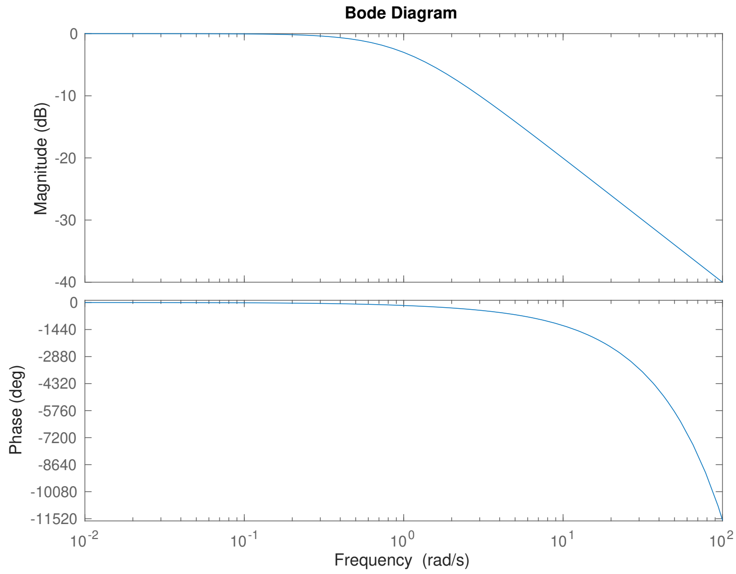

Abbildung 6: Bode-Plot

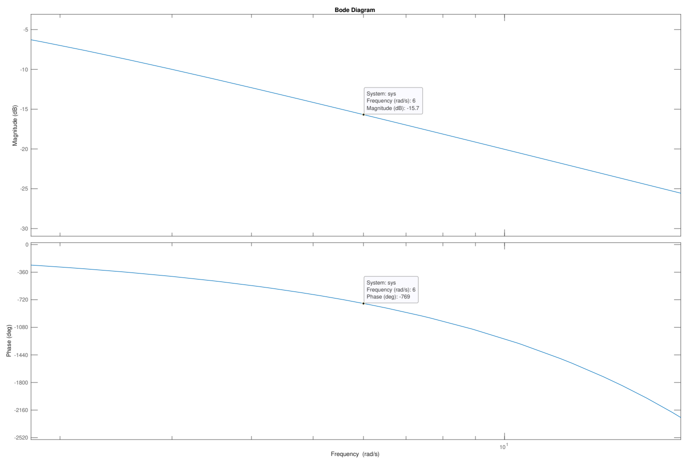

Abbildung 7: Bode-Plot

Beim Bode-Plot lässt sich das entsprechende Verhalten anhand des Verlaufs zu Beginn des Plots feststellen. Bei P-Verfahlten wäre der Verlauf eine waagerechte Gerade, beim I-Verhalten eine Gerade mit negativer Steitung, beim D-Verhalten eine Gerade mit positiver Steigung. Im Bode-Plot ist eine waagerechte Linie zu erkennen. Das impliziert ein P-Verhalten, was mit dem Verhalten unseres System korrespondiert.

3.6.3 Simulink-Simulation

Mithilfe von Simulink lässt sich die Übergangsfunktion simulieren. Die Schaltung zeigt den Aufbau mit einem Sinus-Signal als Eingang. Dabei werden verschiedene Blöcke aus der Simulink-Library verwendet.

1. Sin Wave (Source-Ordner)

2. Transfer Function (Continuous-Ordner)
3. Transport Delay (Continuous-Ordner)
4. Scope (Sink-Ordner)

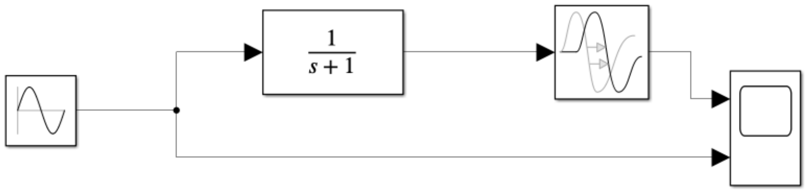

Abbildung 8: Theresa

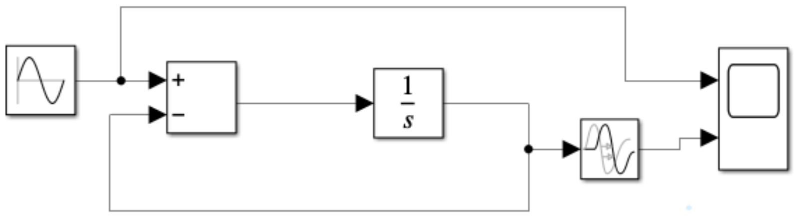

Abbildung 9: Theresa

Die beiden abgebildeten Simulink-Schlatungen sind äquivalent und haben somit bei gleicher Eingangsfunktionen die gleiche Ausgangfunktion. Mit einem Klick auf den Oszilator wird diese Ausgangsfunktion, sowie die am Eingang anliegende Sinus-Funktion in Form eins Graphen ausgegeben (Siehen Abbildung 10 und 11). Dabei ist die blaue Linie die Eingangs-Sinus-Funktion mit einer entsprechenden Frequenz ω in $\frac{rad}{s}$ und die Rote Linie der Ausgang $y(t)$ unseres Systems.

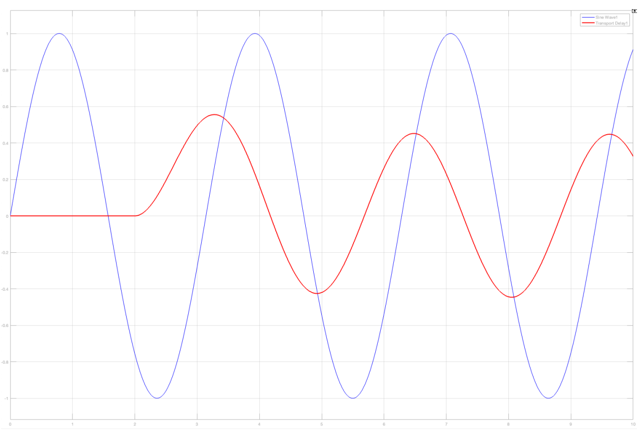

Abbildung 10: Ausgangsfunktion bei $\omega = 2$

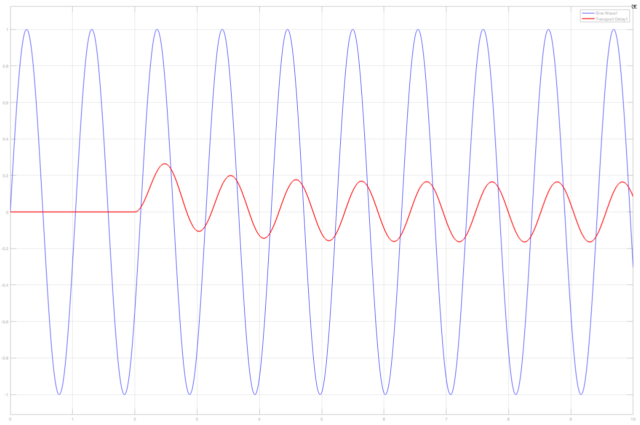

Abbildung 11: Ausgangsfunktion bei $\omega = 6$

3.6.4 Vergleich Bode-Plot, Simulink und Handrechnung

Berechnung des Stationäre Anteils des Ausgangssignals.

$$y_{stat} = |\mathcal{G}(j\omega_0)| \cdot A \cdot sin(\omega_0 t + \phi_0 + arg\ \mathcal{G}(j\omega))$$

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

Handrechnung

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

Berechnung der Amplitudenverstärkgung für $\omega = 6$:

$$\mathcal{G}(j6) = \frac{1}{6j+1} \cdot e^{-2j\cdot 6}$$
$$|\mathcal{G}(j6)| = \left|\frac{1}{6j+1} \cdot e^{-12j}\right|$$
$$|\mathcal{G}(j6)| = \frac{1}{\sqrt{37}} \cdot 1 \approx 0.1644$$

Berechnung der Phasenverschiebung für $\omega = 6$:

Es gilt: $e^{-aj} = -a$

$$arg\ \mathcal{G}(j2) = 0 - arctan\left(\frac{6}{1}\right) - 12$$
$$arg\ \mathcal{G}(j2) = -1.4 - 12 = -13.4$$
$$\Rightarrow -13.4 \cdot \frac{180}{3,14} = -768.726 \approx -769°$$

Vergleich

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

$$20log(\mathcal{K}) = -15.7$$
$$\mathcal{K} = 10^{-\frac{15.7}{20}}$$
$$\mathcal{K} = 0.1640$$

Der aus der Übertragungsfunktion errechnete Wert von 1.644 entspricht mit einer minimalen Abweichung dem Wert, der sich aus dem Bode-Diagramm ablesen lässt. Dieder Wert lässt sich ebenfalls graphisch aus dem Simulink-Plot ablesen:

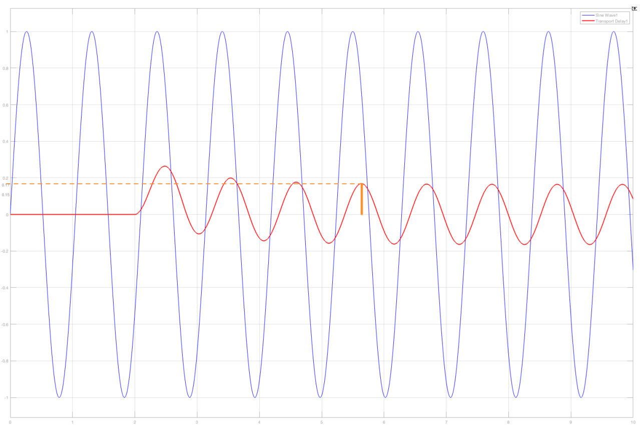

Abbildung 12: Messung der Aplitude der eingeschwungnen Ausganfunktion liegt ungefäht bei 0.17

Im Bodeplot lässt sich im unteren Bild für $\omega = 6$ ein Winkel von $-769°$ ablesen. Dieser Wert stimmt mit der analytischen Rechnung überein.

Die Phasenverschiebung in Grad wird berechnet durch $\phi = (\frac{\Delta t}{T} + T_t) \cdot 360$, wobei T die Periodendauer ist, die mit Hilfe der Formel $T = \frac{2\pi}{\omega} = \frac{2\pi}{6} = \frac{\pi}{3}$ ausgerechnet werden kann.

Aus dem Simulink-Plot kann ein $\Delta t = 0.142s$ abgelesen werden:

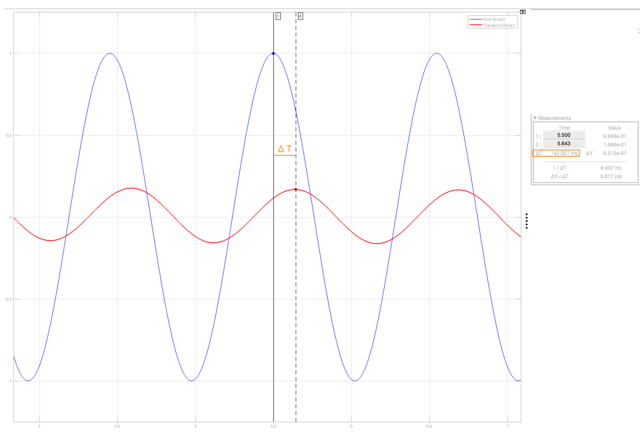

Abbildung 13: Messung der Zeitdifferenz zwischen der Sinusschwingung und der eingeschwungnen Ausganfunktion

Alle Werte werden in die Formel eingsetzt:

$$\phi = (\frac{\Delta t}{T} + T_t) \cdot 360$$
$$\phi = (\frac{0.142 \cdot 3}{\pi} + 2) \cdot 360$$
$$\phi = 768.81^\circ \approx 769^\circ$$

Folglich stimmt die abgelesene Phasenverschiebung aus Simulink mit der per Hand errechneten Lösung, sowie der Lösung aus dem Bode-Diagramm überein.

3.7 Statische Kennlinie

Da es sich bei unserem System um ein System mit proportionalen Übertragungsverhalten handelt, entspricht die statische Verstärkung $K = G(0)$ bzw. $K = \lim_{t\to\infty} h(t)$. Diese lässt sich in Matlab auch mit dem Befehl `dcgain(sys)` berechnen. Somit ergibt sich bei unserem System eine Verstärkung von 1, weshalb die statische Kennlinie nur eine Gerade mit der Steigung 1 ist.

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 Line, welche letztendlich die statische Kennlinie darstellt. Dieses Verfahren erklärt auch, 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 Versärkungsmatrix K auszurechnen (im mehrvariablen Fall ist K eine Matrix, die die jeweiligen statischen Verstärkung der einzelnen Eingangssignale angibt), verwendet die Matrizen aus dem Zustandsraum. Dabei wird die Ableitung von x gleich 0 gesetzt, die Gleichung umgestellt, und in die Gleichung von y eingesetzt. Stell man diese Gleichung erneut um, erhält man K:

$$\dot{x} = Ax + Bu$$
$$0 = Ax + Bu$$
$$x = -A^{-1}Bu$$

$$y = Cx + Du$$
$$y = (-A^{-1}Bu) + Du$$
$$K = -CA^{-1}Bu + D$$

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

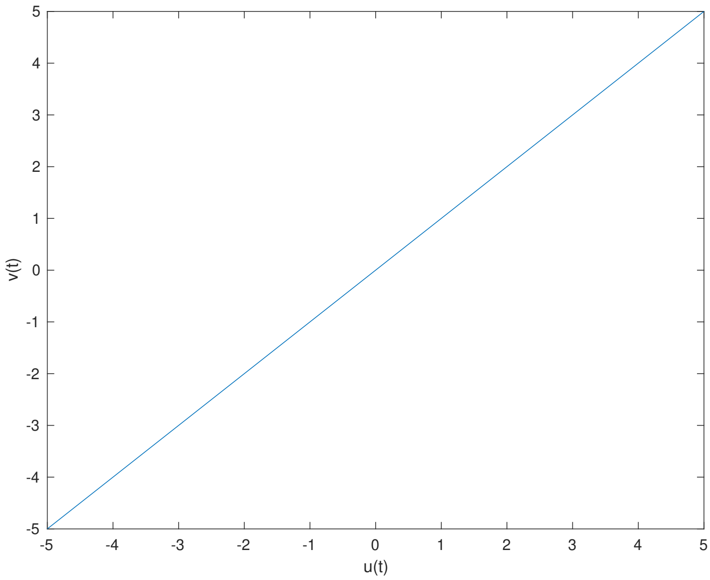

Abbildung 14: Statische Kennlinie

Bei Systemen mit einem Eingang und zwei Ausgängen verwendet man stattdessen ein 2-Balken-Diagramm, bei zwei Eingängen und einem Ausgang wird die 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.

3.8 Pol-Nulstellen-Plot

Abbildung 9 zeigt den Pol-Nulstellen-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 Nulstellen, da der Zählergrad 0 ist, weshalb in der Abbildung kein **o** zu sehen ist. Da der Nenner unseres Systems aber den Grad 1 mit $1+s$ ist, hat unserer System eine reale Polstelle bei $s=-1$, was in der Abbildung mit den **x** gekennzeichnet ist.
Aufgrund dieses negativen Realteils handelt es sich also um eine grenzstabile Polstelle. Würden alle Polstellen unseres Systems in der offenen linken Halbebene liegen, wäre das System stabil. Da die Polstelle genau auf der Imaginärachse liegt, nennt man das System grenzstabil.

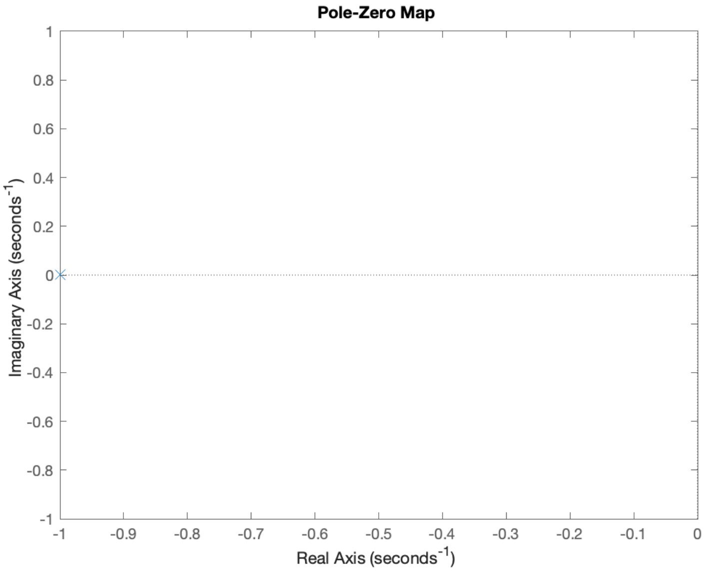

Abbildung 15: Pol-Nulstellen-Plot

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 ziemlich simple: 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.