Beschreibungsformen des PT_1-Glieds

Matrikelnummer: 7371122

1. **Physikalische Beschreibung / Umcodierung**
   Unser System hat P-Verhalten und deutet auf einen Integrator hin (Bsp. Widerstand-Kondensator Serienschaltung).

	Physikalisch	Normiert	Systemtheoretisch
Eingang	Spannungsquelle	1V	u
Ausgang	Spannung im Kondensator	1V	y
Zustand	Spannung im Kondensator	1V	x
Parameter	Widerstand, Kapazität des Kondensators	1Ω, $1\mu F$	T1,K

2. **Übertragungsfunktion**
   Die Übertragungsfunktion ist die Systembeschreibung im Laplace-Bereich und berücksichtigt die Anfangswerte nicht. Sie geht aus der Eingangs- / Ausgangs Differentialgleichung durch Laplace-Transformation hervor, wobei die Anfangswerte auf 0 gesetzt werden. Folglich hat sie eine geringere Aussagekraft als die E-/A-Differentialgleichung, bei der die Anfangswerte immer mit dabei stehen.
   Die Übertragungsfunktion sieht hier folgendermaßen aus:

$$G(s) = \frac{1}{s+1}$$

3. **Zustandsraumbeschreibung (implizite Systembeschreibung)**
   Die Zustandsraumbeschreibung wird durch zwei Gleichungen beschrieben. Die dynamische und die statische Gleichung. Diese folgen der Form:

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

$$\mathbf{y} = C\mathbf{x} + D\mathbf{u}$$

   wobei $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times p}, C \in \mathbb{R}^{q\times n}, D \in \mathbb{R}^{q\times p}$ mit p Eingängen, q Ausgängen und n Zustandsvariablen.
   Die Matrizen A, B, C, D lassen sich in Matlab mit folgendem Befehl generieren:

```
sys= tf([1], [1 1])
ss(sys)
```

Matlab liefert uns folgende Werte:

$$\dot{x} = \begin{pmatrix}-1\end{pmatrix} x + \begin{pmatrix}1\end{pmatrix} u$$
$$y = \begin{pmatrix}1\end{pmatrix} x + \begin{pmatrix}0\end{pmatrix} u$$

Bei unsererem System wählen wir den Anfangswert $x(0) = 0$.

- **Ähnlichkeitstransformation**
  Durch eine Ähnlichkeitstransformation können wir mit einer Hilfsmatrix T eine andere Zustandsraumbeschreibung erzeugen. Beispiel:

$$\text{mit } T = (2)$$

$$\dot{x} = \begin{pmatrix}-1\end{pmatrix} x + \begin{pmatrix}2\end{pmatrix} u$$
$$y = \begin{pmatrix}0.5\end{pmatrix} x + \begin{pmatrix}0\end{pmatrix} u$$

4. **Ein- Ausgangsdifferentialgleichung**
   Aus der Übertragungsfunktion können wir auf die Ein-/Ausgangsgleichung schließen. Durch Überkreuzmultiplikation von:

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

   gelangt man zu:

$$Y(s) \cdot (s+1) = 1 \cdot U(s)$$

   und unter der Betrachtung $s^k \mathrel{\widehat{=}} \frac{d^k}{dt^k}$ zu:

$$\dot{y} + y = u$$

   Das $\dot{y}$ in der Differentialgleichung weist darauf hin, dass wir einen frei wählbaren Anfangswert angeben müssen.
   (Bei gegebenem Beispiel: Anfangsspannung des Kondensators $y(0) = 0$)

5. **Explizite Darstellung des Übertragungsoperators**
   Die Allgemeine Form des Übertragungsoperators sieht wie folgt aus:

$$\mathbf{x}(t) = e^{At} \cdot \mathbf{x}_0 + \int_0^t e^{A(t-\tau)} \mathbf{b} u(\tau) d\tau$$

$$\mathbf{y}(t) = \mathbf{c}^{\mathbf{T}} \mathbf{x}(t) + du(t)$$

Für unser konkretes System kann man das mit Matlab berechnen:

```
syms t
expm(A*t)
```

Mit unserer Systemmatrix $A = \begin{pmatrix}-1\end{pmatrix}$, dem Eingangsvektor $\mathbf{b} = \begin{pmatrix}1\end{pmatrix}$, dem Ausgangsvektor $\mathbf{c} = \begin{pmatrix}1\end{pmatrix}$, dem Durchgangsfaktor $d = 0$ und dem Anfangswert $x_0 = 0$ liefert Matlab uns:

$$x(t) = \int_0^t e^{\tau - t} u(\tau) d\tau$$

$$y(t) = x(t)$$

In unserer Arbeit wählen wir die Step-Funktion für den Eingang, das heißt $u(\tau) = 1$ für $\tau \geq 0$. Im konkreten erhalten wir mit folgendem Befehl die Formel:

```
int(exp(tau - t), tau, 0, t)
```

$$y(t) = x(t) = 1 - e^{-t}$$

Diese ist geplottet identisch mit der Darstellung in Abbildung 1.

6. **Übergangsfunktion / Sprungantwort**

   (a) **Parametrische Darstellung**

   Die Übergangsfunktion $h(t)$ können wir durch die Laplace-Rücktransformation von der Übertragungsfunktion geteilt durch s erhalten

```
syms s
syms t
G(s) = 1/(s+1)
h(t) = ilaplace(G(s)/s)
```

   Das liefert uns $h(t) = 1 - e^{-t}$

   Alternativ kann $h(t)$ auch durch Integration der Gewichtsfunktion $g(t)$ berechnet werden:

```
g(t) = ilaplace(G(s))
h(t) = int(g(t), 0, t)
```

   (b) **Nichtparametrische Darstellung**

   - *Plot mit MATLAB*

     Mit dem wissen der parametrischen Darstellung, können wir diese in Matlab plotten

```
t=[0:0.1:10]
plot(t, 1-exp(-t))
```

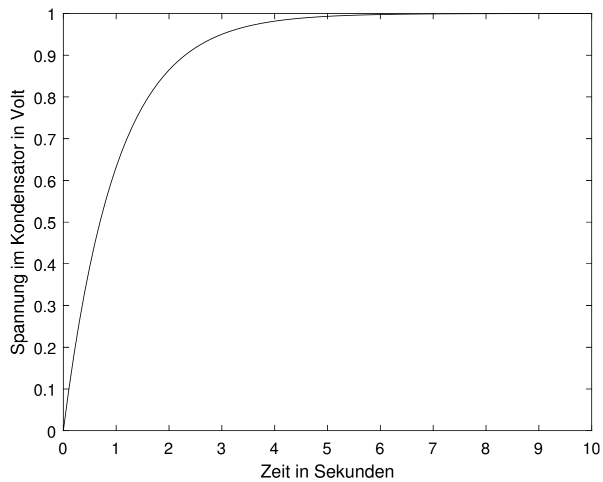

Abbildung 1: Sprungantwort: Plot mit MatLab

- *Plot mit Step-Funktion*
  Denselben Plot bekommt man auch über die Matlab `Step` Funktion

```
sys = tf([1], [1 1])
step(sys)
```

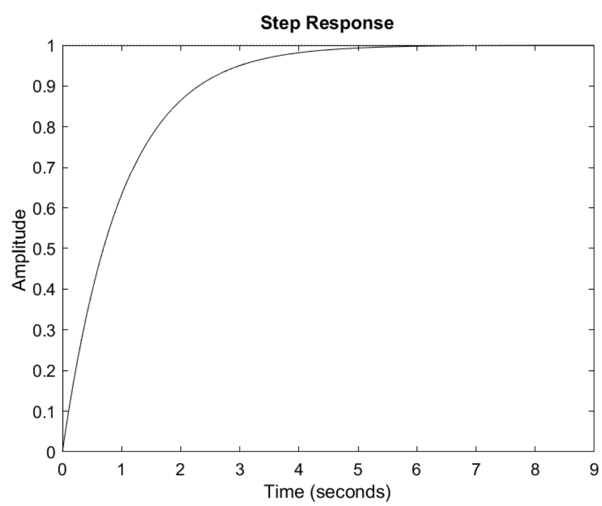

Abbildung 2: Sprungantwort: Plot mit Step Funktion

- *Plot mit Simulink*
  Gleichzeitig bekommt man ebenfalls dasselbe Ergebnis in Simulink mit folgender Schaltung:

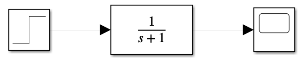

Abbildung 3: Sprungantwort: Simulink Schaltung

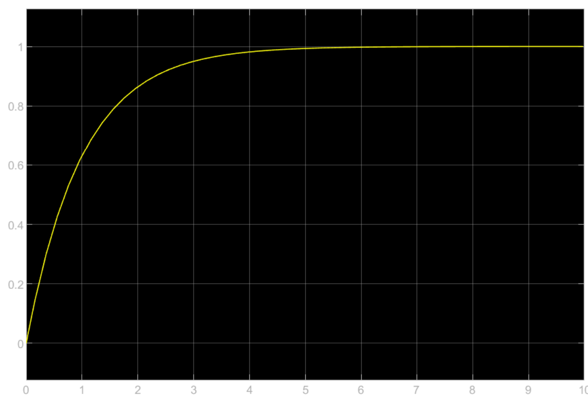

Abbildung 4: Sprungantwort: Plot mit Simulink

7. **Gewichtsfunktion / Impulsantwort**

   (a) **Parametrische Darstellung**
   Die parametrische Darstellung der Gewichtsfunktion $g(t)$ erhält man, indem man die Laplace-Rücktransformierte der Übertragungsfunktion bildet.
   Das macht man wie folgt:

```
syms s
syms t
G(s) = 1/(s+1)
g(t) = ilaplace(G(s))
```

   Matlab liefert hier $g(t) = e^{-t}$

   Alternativ kann die Gewichtsfunktion durch Ableiten der Übertragungsfunktion erhalten werden.

```
diff(h(t))
```

(b) **Nichtparametrische Darstellung**

- *Plot mit Matlab*
  Mit dem wissen der parametrischen Darstellung, können wir diese in Matlab plotten
  ```
  t=[0:0.1:10]
  plot(t, exp(-t))
  ```

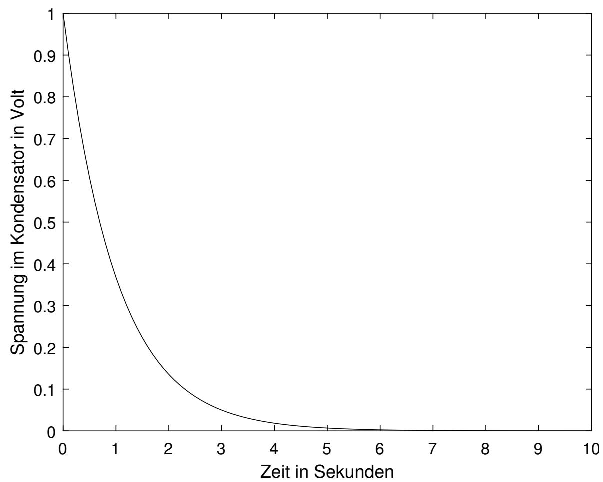

Abbildung 5: Impulsantwort: Plot mit Matlab

- *Plot über Impuls-Funktion*
  Den selben Plot bekommt man auch über die Matlab `impulse` Funktion
  ```
  sys = tf([1], [1 1])
  impulse(sys)
  ```

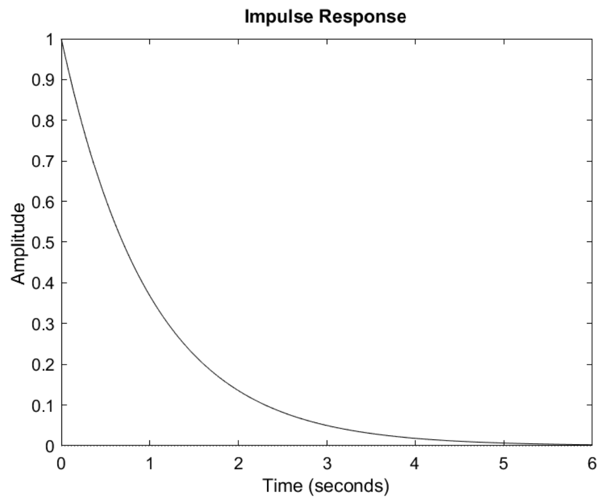

Abbildung 6: Impulsantwort: Plot mit Impuls Funktion

- *Plot mit Simulink*
  Multiplizieren wir in Simulink die Übertragungsfunktion mit s, so erhalten wir das Ergebnis eines Deltaimpulses.
  Die Schaltung und der dazugehörige Plot sehen in Simulink wie folgt aus:

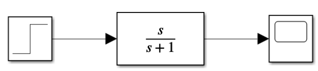

Abbildung 7: Impulsantwort: Simulink Schaltung

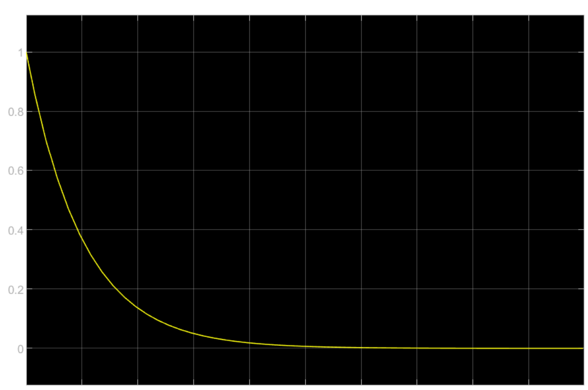

Abbildung 8: Impulsantwort: Plot mit Simulink

8. **Frequenzgang**

(a) **Parametrische Darstellung**
Den Frequenzgang erhalten wir, wenn wir die komplexe Variable s durch $i\omega$ ersetzen. Das ist gleichbedeutend mit einem Schnitt der komplexen Funktion $G(s)$ entlang der imaginären Achse.
Damit wird jeder Kreisfrequenz ω eine komplexe Zahl $G(i\omega)$ zugeordnet.

Bedingt durch den Schnitt hat der Frequenzgang eine geringere Aussagekraft als die Übertragungsfunktion. Die Bedeutung des Frequenzgangs ergibt sich aus der Tatsache, dass das stationäre Verhalten des Systems auf eine sinuidale (sinusartige) Anregung beschrieben wird.

Sei $u(t) = sin(\omega t)$, so ist $y_{stat}(t)$, d.h. das Ausgangssignal nach Abklingen der Anfangswerte, durch:

$$y_{stat}(t) = A \cdot |G(i\omega)| \cdot sin(\omega t + \varphi(G(i\omega)))$$

gegeben.

(b) **Nichtparametrische Darstellung**
Experimentell kann der Frequenzgang durch folgende Simulink Schaltung aufgenommen werden:

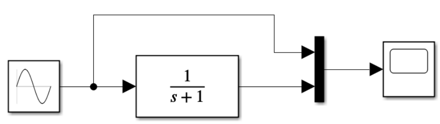

Abbildung 9: Frequenzgang: Simulink Schaltung

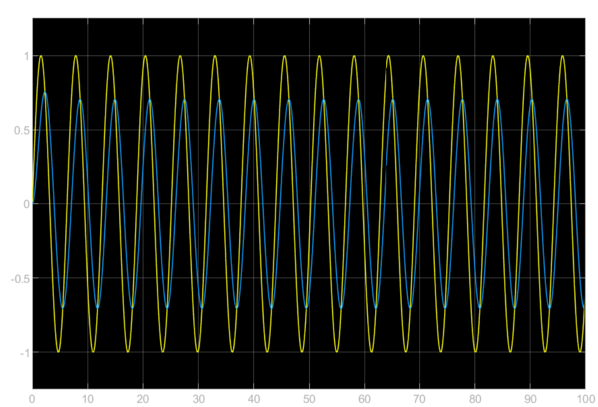

Abbildung 10: Frequenzgang: Plot mit Simulink

Am Eingangssinus wird eine Frequenz eingestellt und am Ausgang die Amplituden-Verstärkung und die Phasenverschiebung betrachtet.
Hat der Eingang die Amplitude 1, kann man die Ausgangsverstärkung direkt ablesen.
Das Experiment wird mit verschiedenen Frequenz wiederholt und die jeweiligen Ergebnisse werden dabei als Punkte in die folgenden Diagramme eingetragen.

- *Nyquist-Plot*
  Beim Nyquist-Plot ist auf der X-Achse der Realteil der gemessenen Punkte und auf der Y-Achse der Imaginärteil, bei steigender Frequenz ω zu sehen.
  Dieser lässt sich in Matlab mit folgendem Befehl generieren:
  `nyquist(sys)`

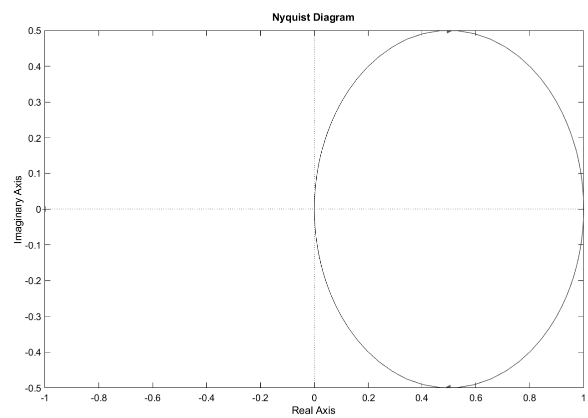

Abbildung 11: Frequenzgang: Nyquist Plot

- *Bode-Plot*
  Der Bode-Plot besteht aus zwei Graphen. Auf einem der Graphen zeigt der Plot an der Y-Achse die Amplituden-Verstärkung des Eingangs in dB, auf dem zweiten Graphen die Phasenverschiebung in °.
  Beides über der Frequenz ω.

  Die erhaltenen Punkte, des oben beschriebenen Vorgangs, trägt man in das Bode-Diagramm ein. Sind genügend Punkte vorhanden, verbindet man diese und erhält das Bode-Diagramm.
  Ebenso wie wir an den Verläufen der Sprungantwort auf die Differenzialgleichung schließen konnten, können Experten aus den Verläufen des Bode-Diagramms auf das Verhalten schließen, doch wie dies geht, ist nicht Gegenstand der Vorlesung.

Selbstverständlich können wir auch per Computeralgebra, Amplitude und Phasenverschiebung einer komplexen Zahl z berechnen.

```
angle(z)
abs(z)
```

In Matlab lässt sich der Bode Plot mit dem Befehl

```
bode(sys)
```

erzeugen.

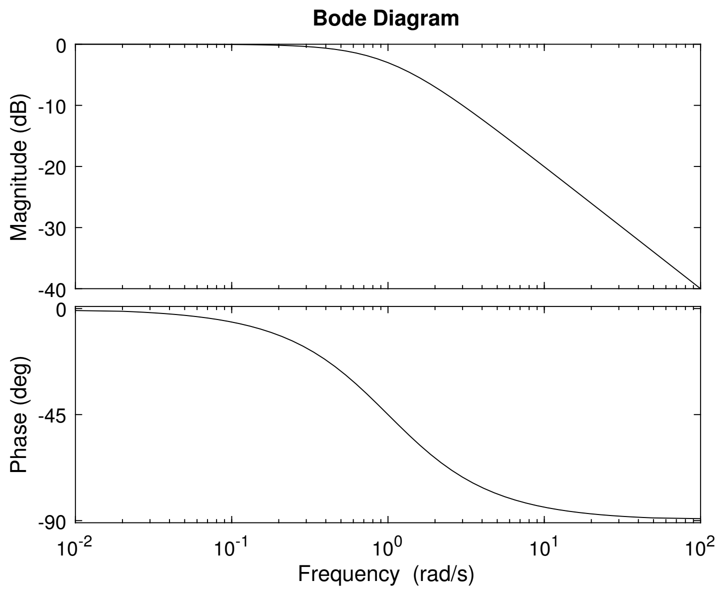

Abbildung 12: Frequenzgang: Bode-Plot

9. **Pol-Nullstellen-Plot**
   Der Pole-Zero-Plot stellt die Pole und Nullstellen der Übertragungsfunktion in der komplexen Ebene dar. Pole werden durch ein Kreuz dargestellt, Nullstellen durch einen Kreis. Ein Doppelpol/-nullstelle wird durch ein Doppelkreuz/-kreis gekennzeichnet.
   Der Pol-Nullstellen-Plot enthält keine Information über die statische Verstärkung, jedoch lassen sich aus dem PZ-Plot Eigenschaften wie Stabilität und Minimalphasigkeit ablesen.

   Ein System ist dann stabil, wenn alle Pole in der linken offenen Halbebene liegen. Sobald ein Pol auf der rechten Halbebene liegt, ist das System instabil. Es ist grenzstabil, wenn keine Pole auf der rechten Halbebene liegen, aber einige auf der imaginären Achse auftauchen.

   Ein System ist dann minimalphasig, wenn alle Nullstellen in der linken offenen Halbebene liegen. Es ist nicht minimalphasig sobald eine Nullstelle auf der rechten Halbebene liegt und letztlich ist es schwach minimalphasig, wenn keine Nullstelle auf der rechten

Halbebene liegt, aber einige auf der imaginären Achse liegen.

In Matlab lässt sich der PZ-Plot mit folgendem Befehl erzeugen:
`pzplot(sys)`

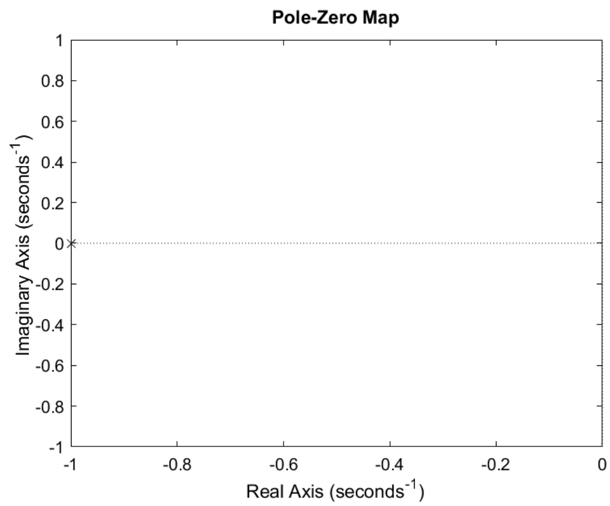

Abbildung 13: PZ-Plot

10. **Statische Kennlinie**
In der statischen Kennlinie wird der Ausgang über dem Eingang dargestellt. Bei einem linearen System ist die statische Kennlinie eine Ursprungsgerade, deren Anstieg der statischen Verstärkung entspricht. Die statische Verstärkung k kann aus $G(0) = k$ berechnet werden. Sie ist bei Systemen mit D-Verhalten 0 und bei Systemen mit I-Verhalten unendlich.

In der statischen Kennlinie ist keinerlei Dynamik zu erkennen und die Information über Pole und Nullstellen fehlen. Sie ist somit die schwächste der Modellbeschreibungen. Allerdings ist sie einfach zu bestimmen: Man stellt einen konstanten Wert u ein, wartet bis die Eigenvorgänge abgeklungen sind, und liest den y Wert ab. Diesen Punkt trägt man in das Diagramm ein und wiederholt das ganze für mehrere Punkte.

In der Praxis werden sich häufig nicht ideale Geraden ergeben. Bei Öfen beispielsweise verringert sich der Anstieg bei hohem u. Der Praktiker liest aus der statischen Kennlinie ab, wie gut seine Annahme eines linearen Modells ist. Sie ist gut, wenn die Kennlinie einer Geraden ähnelt oder zumindest in einem Bereich einer Geraden ähnelt.

Das gegebene System hat einen Eingang und einen Ausgang. Demnach reicht folgende Darstellung, die sich in Matlab mit dem Befehl
`fplot(dcgain(sys)*u, [0 10])`
erzeugen lässt. (Siehe Abbildung 14)

Diese können wir uns aber auch berechnen, indem wir alle Ableitungen 0 setzen und eine Beziehung zwischen y und u herstellen:

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

Somit ist unser statischer Verstärkungsfaktor $k = 1$ und liefert uns folgende statische Kennlinie:

```
u=[0:0.1:10]
y=u
plot(u, y)
```

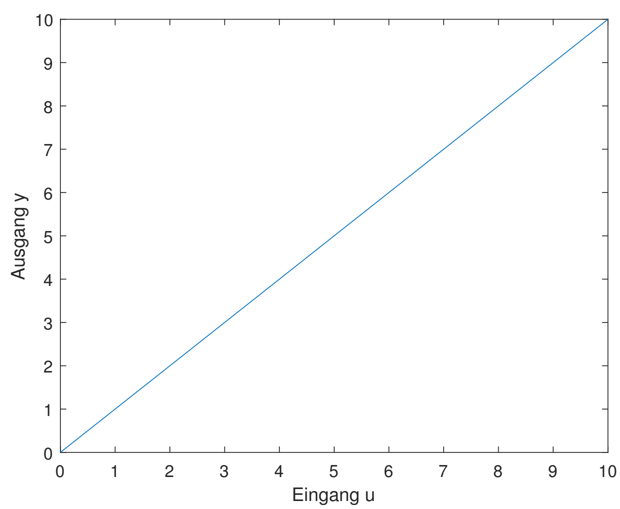

Abbildung 14: Statische Kennlinie