

Praktikum “Simulation im Zustandsraum”

Berechnung und Darstellung in Scilab

Aufgabe 1 (Überführung Übertragungsfunktion in Zustandsraum). Nutzen Sie Scilab, um die Übertragungsfunktion

$$G(s) = \frac{3 + s}{10 + 1.1s + s^2}$$

in eine Zustandsraumdarstellung zu überführen. Nutzen Sie:

- `syslin`, um die Übertragungsfunktion zu definieren,
- `tf2ss` zur Überführung in den Zustandsraum und
- `csim`, um eine Sprungantwort des Systems aufzuzeichnen und mit `plot2d` zu plotten.

Aufgabe 2 (Steuerbarkeit und Beobachtbarkeit). Bestimmen Sie die Steuerbarkeits- und Beobachtbarkeitsmatrizen und ihre Ränge. Beurteilen Sie die Steuerbarkeit und Beobachtbarkeit.

Aufgabe 3 (Phasendiagramm). Zeichnen Sie ein Richtungsfeld mit

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = A \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

als Richtung eines Pfeils an der Koordinate (x_1, x_2) . Wählen Sie den Bereich $x_{1,2} = -1 \dots 1$.

Welche Bedeutung hat dieses Richtungsfeld? Welche Eigenschaften des Systems lassen sich ablesen?

Hinweise:

- Definieren Sie eine Funktion `xdot`, um die Richtung der Pfeile zu bestimmen.
- Definieren Sie eine globale Variable zur Speicherung der Systemmatrix.
- Nutzen Sie die Funktion `fchamp`, um ein Richtungsfeld zu plotten.
- Optional: Ergänzen Sie eine Trajektorie des System im Richtungsfeld.

Lösung 1 (Skript zur Lösung der Aufgaben 1, 2 und 3).

```

clear
clc
clear global
mode(2)
delete(gcf())
global A

// Aufgabe 1
// s als Variable festlegen
s = syslin('c',poly(0,'s'),1)
// Uebertragungsfunktion eingeben
G=(s+3)/(s^2+1.1*s+10);
// In Transferfunktion ueberfuehren
Sys=tf2ss(G)
// Zeitachse
t = 0:0.025:10;
// Sprungantwort
ystep = csim('step', t, Sys)
plot2d(t, ystep)

//Aufgabe 2
// Rang der Steuerbarkeits- und Beobachtbarkeitsmatrizen
Cont = rank([Sys.B, Sys.A*Sys.B])
ContU = rank([Sys.C*Sys.B, Sys.C*Sys.A*Sys.B])
Obs = rank([Sys.C; Sys.C*Sys.A])

// Aufgabe 3
// Auf globale Variable umspeichern
A = Sys.A;
// Funktion zur Berechnung der Steigung
function xv=xdot(t, x)
    xv = A*x;
endfunction
// Bereich und Aufloesung
x1 = -1:0.2:1; x2 = -1:0.2:1;
// System-Trajektorie
[ytra, xtra] = csim(zeros(t), t, Sys, rand(2,1))

// Plot
scf(2)
fchamp(xdot, 0, x1, x2, arfact = 0.7);
plot2d(xtra(1,:), xtra(2,:), 2)
a = gca();
a.isoview = 'on';
a.x_location = "middle";

```

```
a.y_location = "middle";
```

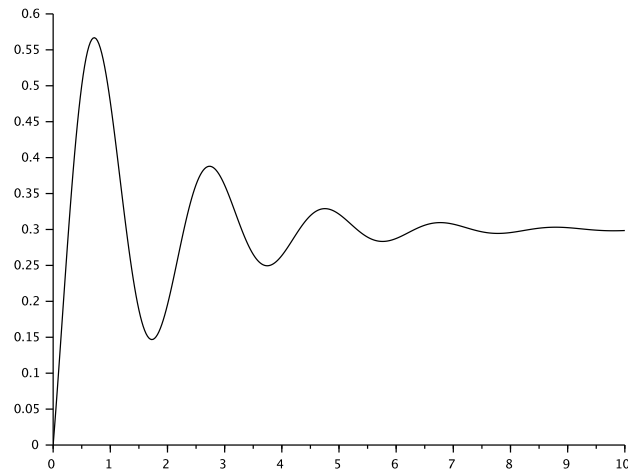


Abbildung 1: Ausgabe der Sprungantwort

Simulation in Scicos

Aufgabe 4 (Erstellen eines Blockdiagramms in Scicos). *Erstellen Sie ein Blockdiagramm eines Systems in Regelungsnormalform mit folgender Systembeschreibung*

$$\frac{d^n y}{dt^n} + a_1 \frac{dy}{dt} + a_0 y = b_0 u$$

und den Parametern $a_0 = 1$, $a_1 = 0,2$ und $b_0 = 1$.

- Die Eingangsgröße in das System sei ein Sprung bei $t = 0$.
- Nutzen Sie eine geeignete Senke, um die Antwort des Systems beobachten zu können.

Aufgabe 5 (Ausführung mittels Scilab). *Modifizieren Sie das Blockdiagramm aus Aufgabe 4 so, dass Sie*

- die Systemparameter und
- die Eingangsgröße

in Scilab definieren können.

Erstellen Sie in Programm in Scilab, das es Ihnen ermöglicht

- die Parameter und Eingangsgröße zu definieren,
- das Blockdiagramm auszuführen und
- Eingangsgröße und Zustandsgrößen in einen gemeinsamen Diagramm darzustellen.

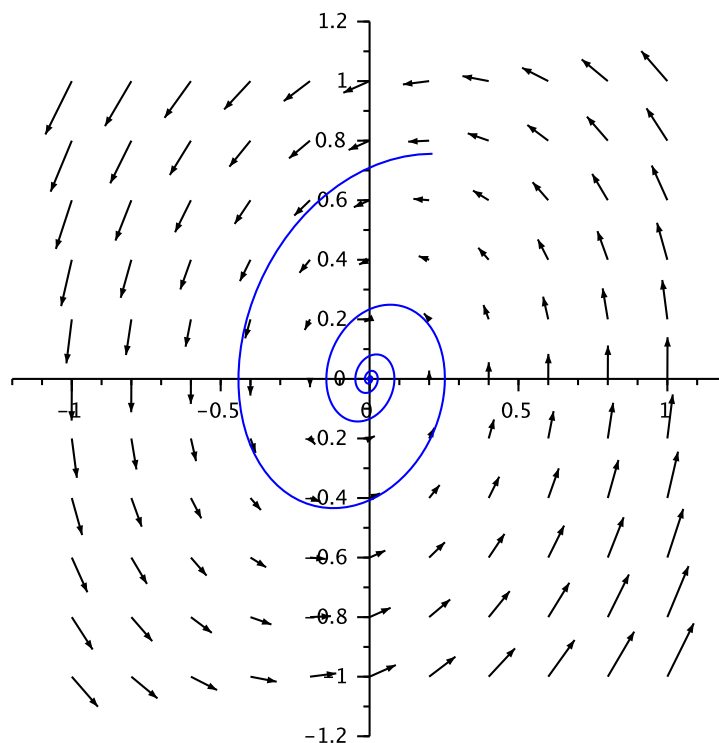


Abbildung 2: Ausgabe des Richtungsfeldes und der Trajektorie

Lösung 2 (Skript zur Lösung der Aufgaben 5).

```

clear
delete(gcf())

// load the blocks library and the simulation engine
loadXcosLibs(); loadScicos();
// load diagram
importXcosDiagram("P1.zcos")

N = 1000; // time steps
tmax = 50; // final simulation time
time = linspace(0,tmax,N)'; // time axis
// Define input data for fromworkspace block
myInput.time = time;
myInput.values = ones(time)+ 0*sin(time)+0*rand(time);
// System parameters
a1 = .2;
a0 = 1;
b0 = 1;
// Execute scicos diagram
scicos_simulate(scs_m)
// Plot data

```

```
plot2d(myOutput.time, myOutput.values)  
legend('u', 'x1(t)', 'x2(t)')
```

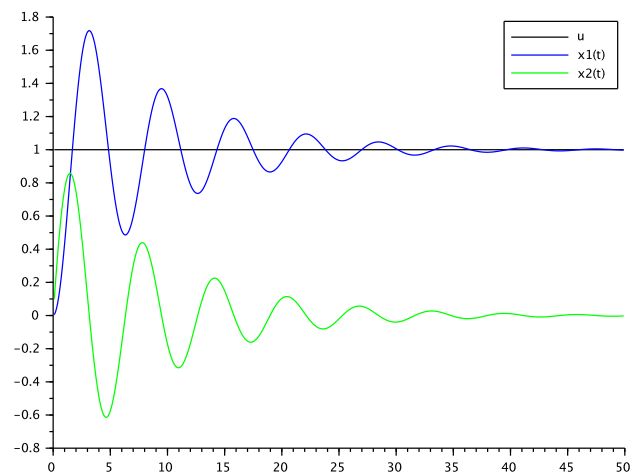


Abbildung 3: Ausgabe der Sprungantwort