
Versuch V: Aufschwingsteuerung des Doppelpendels

Andreas Jentsch, Ali Kerem Sacakli

Praktikumsbericht – Praktikum Matlab/Simulink II

04. Juli 2017



TECHNISCHE
UNIVERSITÄT
DARMSTADT

REGELUNGSTECHNIK **rtm**
UND MECHATRONIK

4.8 Berechnung der Trajektorien

In diesem Versuch geht es um den Entwurf und der Simulation einer Aufschwungsteuerung für den Doppelpendel. Hierzu werden zunächst Codes generiert, um die Trajektorien zu generieren. Im nächsten Kapitel folgen die Simulationsergebnisse.

Die relevanten Codes sind wie folgt:

Listing 4.1: Quellcode der Funktion **RandwertproblemDGL**

```
1 function dxdt = RandwertproblemDGL(t, x, stPendel, Q, R)
    % t:                Zeit
    % x:                Zustände x(1) und x(2)
    % Pendeldataen:    Pendeldataen
    % Q:                Gewichtungsmatrix für Zustände
6    % R:                Gewichtung für Stellgröße

    % l1,l2,m1,m2,g,Rp1,Rp2,q1,q2,q3,q4,sym_R
    l1 = stPendel.l1;
11    l2 = stPendel.l2;
    g = stPendel.g;
    m1 = stPendel.m1;
    m2 = stPendel.m2;
    Rp1 = stPendel.Rp1;
16    Rp2 = stPendel.Rp2;
    q1 = Q(1,1);
    q2 = Q(2,2);
    q3 = Q(3,3);
    q4 = Q(4,4);
21    sym_R = R;
    dxdt = 0;

    % Differentialgleichung 8ter Ordnung,
    % x1-x4: Originale Zustände,
26    % x5-x8: Lagrange-Multiplikatoren

    phi1 = x(1);
    dphi1 = x(2);
31    phi2 = x(3);
    dphi2 = x(4);
```

```

lambda1 = x(5);
lambda2 = x(6);
lambda3 = x(7);
lambda4 = x(8);

```

36

```

dx1 = dphi1;
dx2 = -(3*(4*l2*((12*lambda2)/(l1^2*sym_R*(4*m1 + 12*m2 - 9*m2*→
  ←cos(phi1 - phi2)^2)) - (18*lambda4*cos(phi1 - phi2))/(l1*l2*→
  ←sym_R*(4*m1 + 12*m2 - 9*m2*cos(phi1 - phi2)^2))) + 4*Rp1*→
  ←dphi1*l2 + 6*Rp2*dphi1*l1*cos(phi1 - phi2) - 6*Rp2*dphi2*l1*→
  ←cos(phi1 - phi2) + 2*dphi2^2*l1*l2^2*m2*sin(phi1 - phi2) + →
  ←2*g*l1*l2*m1*sin(phi1) + 4*g*l1*l2*m2*sin(phi1) + 3*dphi1^2*→
  ←l1^2*l2*m2*cos(phi1 - phi2)*sin(phi1 - phi2) - 3*g*l1*l2*m2*→
  ←cos(phi1 - phi2)*sin(phi2)))/(l1^2*l2*(4*m1 + 12*m2 - 9*m2*→
  ←cos(phi1 - phi2)^2));

```

41

```

dx3 = dphi2;
dx4 = (3*(6*l2*m2*cos(phi1 - phi2)*((12*lambda2)/(l1^2*sym_R*(4*→
  ←m1 + 12*m2 - 9*m2*cos(phi1 - phi2)^2)) - (18*lambda4*cos(→
  ←phi1 - phi2))/(l1*l2*sym_R*(4*m1 + 12*m2 - 9*m2*cos(phi1 - →
  ←phi2)^2))) + 4*Rp2*dphi1*l1*m1 + 12*Rp2*dphi1*l1*m2 - 4*Rp2*→
  ←dphi2*l1*m1 - 12*Rp2*dphi2*l1*m2 + 6*dphi1^2*l1^2*l2*m2^2*→
  ←sin(phi1 - phi2) - 6*g*l1*l2*m2^2*sin(phi2) + 6*Rp1*dphi1*l2*→
  ←m2*cos(phi1 - phi2) + 3*dphi2^2*l1*l2^2*m2^2*cos(phi1 - →
  ←phi2)*sin(phi1 - phi2) + 2*dphi1^2*l1^2*l2*m1*m2*sin(phi1 - →
  ←phi2) + 6*g*l1*l2*m2^2*cos(phi1 - phi2)*sin(phi1) - 2*g*l1*→
  ←l2*m1*m2*sin(phi2) + 3*g*l1*l2*m1*m2*cos(phi1 - phi2)*sin(→
  ←phi1)))/(l1*l2^2*m2*(4*m1 + 12*m2 - 9*m2*cos(phi1 - phi2)^2)→
  ←);

```

```

dx5 = lambda2*((3*(4*l2*((18*lambda4*sin(phi1 - phi2))/(l1*l2*→
  ←sym_R*...
  (4*m1 + 12*m2 - 9*m2*cos(phi1 - phi2)^2)) - (216*lambda2*m2*→
  ←cos(phi1 - phi2)...
  *sin(phi1 - phi2))/(l1^2*sym_R*(4*m1 + 12*m2 - 9*m2*cos(phi1→
  ← - phi2)^2)^2)...
  + (324*lambda4*m2*cos(phi1 - phi2)^2*sin(phi1 - phi2))/(l1*→
  ←l2*sym_R*...
  (4*m1 + 12*m2 - 9*m2*cos(phi1 - phi2)^2)^2)) - 6*Rp2*dphi1*→
  ←l1*sin(phi1 - phi2) ...

```

46

```

+ 6*Rp2*dphi2*l1*sin(phi1 - phi2) + 3*dphi1^2*l1^2*l2*m2*cos(
  phi1 - phi2)^2 ...
-3*dphi1^2*l1^2*l2*m2*sin(phi1 - phi2)^2 + 2*dphi2^2*l1*l2*
  l2^2*m2*cos(phi1 - phi2) ...
+ 2*g*l1*l2*m1*cos(phi1) + 4*g*l1*l2*m2*cos(phi1) + 3*g*l1*
  l2*m2*sin(phi1 - phi2)...
* sin(phi2)))/(l1^2*l2*(4*m1 + 12*m2 - 9*m2*cos(phi1 - phi2)*
  l2^2)) - (54*m2*cos(phi1 - phi2)*sin(phi1 - phi2)*(4*l2*
  l2^2*((12*lambda2)/(l1^2*sym_R*(4*m1 + 12*m2 - 9*m2*cos(phi1 -
  phi2)^2)) - (18*lambda4*cos(phi1 - phi2))/(l1*l2*
  sym_R*(4*m1 + 12*m2 - 9*m2*cos(phi1 - phi2)^2))) + 4*Rp1*
  dphi1*l2 + 6*Rp2*dphi1*l1*cos(phi1 - phi2) - 6*Rp2*
  dphi2*l1*cos(phi1 - phi2) + 2*dphi2^2*l1*l2^2*m2*sin(
  phi1 - phi2) + 2*g*l1*l2*m1*sin(phi1) + 4*g*l1*l2*m2*sin(
  phi1) + 3*dphi1^2*l1^2*l2*m2*cos(phi1 - phi2)*sin(phi1 -
  phi2) - 3*g*l1*l2*m2*cos(phi1 - phi2)*sin(phi2)))/(l1*
  l2^2*(4*m1 + 12*m2 - 9*m2*cos(phi1 - phi2)^2)^2) -
  phi1*q1 - lambda4*((3*(6*l2*m2*cos(phi1 - phi2)...
  *((18*lambda4*sin(phi1 - phi2))/(l1*l2*sym_R*(4*m1 + 12*m2 -
  9*m2*cos(phi1 - phi2)^2)) - (216*lambda2*m2*cos(phi1 -
  phi2)*sin(phi1 - phi2))/(l1^2*sym_R*(4*m1 + 12*m2 - 9*m2*
  cos(phi1 - phi2)^2)^2) + (324*lambda4*m2*cos(phi1 -
  phi2)^2*sin(phi1 - phi2))/(l1*l2*sym_R*(4*m1 + 12*m2 -
  9*m2*cos(phi1 - phi2)^2)^2) - 6*l2*m2*sin(phi1 - phi2)*
  *((12*lambda2)/(l1^2*sym_R*(4*m1 + 12*m2 - 9*m2*cos(phi1 -
  phi2)^2)) - (18*lambda4*cos(phi1 - phi2))/(l1*l2*
  sym_R*(4*m1 + 12*m2 - 9*m2*cos(phi1 - phi2)^2))) - 6*Rp1*
  dphi1*l2*m2*sin(phi1 - phi2) + 6*dphi1^2*l1^2*l2*m2^2*
  cos(phi1 - phi2) + 3*dphi2^2*l1*l2^2*m2^2*cos(phi1 -
  phi2)^2 - 3*dphi2^2*l1*l2^2*m2^2*sin(phi1 - phi2)^2 ...
+ 2*dphi1^2*l1^2*l2*m1*m2*cos(phi1 - phi2) + 6*g*l1*l2*m2^2*
  cos(phi1 - phi2)*cos(phi1) - 6*g*l1*l2*m2^2*sin(phi1 -
  phi2)*sin(phi1) + 3*g*l1*l2*m1*m2*cos(phi1 - phi2)*cos(
  phi1) - 3*g*l1*l2*m1*m2*sin(phi1 - phi2)*sin(phi1)))/(l1*
  l2^2*m2*(4*m1 + 12*m2 - 9*m2*cos(phi1 - phi2)^2) -
  (54*cos(phi1 - phi2)*sin(phi1 - phi2)*(6*l2*m2*cos(phi1 -
  phi2)*((12*lambda2)/(l1^2*sym_R*(4*m1 + 12*m2 - 9*m2*
  cos(phi1 - phi2)^2)) - (18*lambda4*cos(phi1 - phi2))/(l1*
  l2*sym_R*(4*m1 + 12*m2 - 9*m2*cos(phi1 - phi2)^2))) +
  4*Rp2*dphi1*l1*m1 + 12*Rp2*dphi1*l1*m2 - 4*Rp2*dphi2*l1*

```

```

        ←m1 - 12*Rp2*dphi2*l1*m2 + 6*dphi1^2*l1^2*l2*m2^2*sin(→
        ←phi1 - phi2) - 6*g*l1*l2*m2^2*sin(phi2) ...
+ 6*Rp1*dphi1*l2*m2*cos(phi1 - phi2) + 3*dphi2^2*l1*l2^2*m2→
    ←^2*cos(phi1 - phi2)*sin(phi1 - phi2) + 2*dphi1^2*l1^2*l2→
    ←*m1*m2*sin(phi1 - phi2) + 6*g*l1*l2*m2^2*cos(phi1 - phi2→
    ←)*sin(phi1) - 2*g*l1*l2*m1*m2*sin(phi2) + 3*g*l1*l2*m1*→
    ←m2*cos(phi1 - phi2)*sin(phi1)))/(l1*l2^2*(4*m1 + 12*m2 -→
    ← 9*m2*cos(phi1 - phi2)^2)^2));
dx6 = (3*lambda2*(6*dphi1*l2*m2*cos(phi1 - phi2)*sin(phi1 - phi2→
    ←)*l1^2 + 6*Rp2*cos(phi1 - phi2)*l1 + 4*Rp1*l2))/(l1^2*l2*(4*→
    ←m1 + 12*m2 - 9*m2*cos(phi1 - phi2)^2)) - dphi1*q2 - lambda1 →
    ←- (3*lambda4*(12*dphi1*l2*sin(phi1 - phi2)*l1^2*m2^2 + 4*→
    ←dphi1*l2*m1*sin(phi1 - phi2)*l1^2*m2 + 12*Rp2*l1*m2 + 4*Rp2*→
    ←m1*l1 + 6*Rp1*l2*cos(phi1 - phi2)*m2))/(l1*l2^2*m2*(4*m1 + →
    ←12*m2 - 9*m2*cos(phi1 - phi2)^2));
56 dx7 = lambda4*((3*(6*l2*m2*cos(phi1 - phi2))*((18*lambda4*sin(→
    ←phi1 - phi2))/(l1*l2*sym_R...
    *(4*m1 + 12*m2 - 9*m2*cos(phi1 - phi2)^2)) - (216*lambda2*m2→
    ←*cos(phi1 - phi2)*...
    sin(phi1 - phi2))/(l1^2*sym_R*(4*m1 + 12*m2 - 9*m2*cos(phi1 →
    ←- phi2)^2)^2) + (324*lambda4*m2...
    *cos(phi1 - phi2)^2*sin(phi1 - phi2))/(l1*l2*sym_R*(4*m1 + →
    ←12*m2 - 9*m2*cos(phi1 - phi2)^2)^2))...
    - 6*l2*m2*sin(phi1 - phi2)*((12*lambda2)/(l1^2*sym_R*(4*m1 +→
    ← 12*m2 - 9*m2*cos(phi1 - phi2)^2)) ...
61 - (18*lambda4*cos(phi1 - phi2))/(l1*l2*sym_R*(4*m1 + 12*m2 -→
    ← 9*m2*cos(phi1 - phi2)^2))) ...
    - 6*Rp1*dphi1*l2*m2*sin(phi1 - phi2) + 6*dphi1^2*l1^2*l2*m2→
    ←^2*cos(phi1 - phi2) + 6*g*l1*l2*m2^2*cos(phi2) + 3*dphi2→
    ←^2*l1*l2^2*m2^2*cos(phi1 - phi2)^2 - 3*dphi2^2*l1*l2^2*→
    ←m2^2*sin(phi1 - phi2)^2 + 2*dphi1^2*l1^2*l2*m1*m2*cos(→
    ←phi1 - phi2) + 2*g*l1*l2*m1*m2*cos(phi2) - 6*g*l1*l2*m2→
    ←^2*sin(phi1 - phi2)*sin(phi1) - 3*g*l1*l2*m1*m2*sin(phi1→
    ←- phi2)*sin(phi1)))/(l1*l2^2*m2*(4*m1 + 12*m2 - 9*m2*→
    ←cos(phi1 - phi2)^2)) - (54*cos(phi1 - phi2)*sin(phi1 - →
    ←phi2)*(6*l2*m2*cos(phi1 - phi2))*((12*lambda2)/(l1^2*→
    ←sym_R*(4*m1 + 12*m2 - 9*m2*cos(phi1 - phi2)^2)) - (18*→
    ←lambda4*cos(phi1 - phi2))/(l1*l2*sym_R*(4*m1 + 12*m2 - →
    ←9*m2*cos(phi1 - phi2)^2))) ...

```

```

+ 4*Rp2*dphi1*l1*m1 + 12*Rp2*dphi1*l1*m2 - 4*Rp2*dphi2*l1*m1→
  ← - 12*Rp2*dphi2*l1*m2 + 6*dphi1^2*l1^2*l2*m2^2*sin(phi1 →
  ← phi2) - 6*g*l1*l2*m2^2*sin(phi2) + 6*Rp1*dphi1*l2*m2*→
  ← cos(phi1 - phi2) + 3*dphi2^2*l1*l2^2*m2^2*cos(phi1 - →
  ← phi2)*sin(phi1 - phi2) + 2*dphi1^2*l1^2*l2*m1*m2*sin(→
  ← phi1 - phi2) + 6*g*l1*l2*m2^2*cos(phi1 - phi2)*sin(phi1)→
  ← - 2*g*l1*l2*m1*m2*sin(phi2) + 3*g*l1*l2*m1*m2*cos(phi1 →
  ← phi2)*sin(phi1)))/(l1*l2^2*(4*m1 + 12*m2 - 9*m2*cos(→
  ← phi1 - phi2)^2)^2) - phi2*q3 - lambda2*((3*(4*l2*((18*→
  ← lambda4*sin(phi1 - phi2))/(l1*l2*sym_R*(4*m1 + 12*m2 - →
  ← 9*m2*cos(phi1 - phi2)^2)) - (216*lambda2*m2*cos(phi1 - →
  ← phi2)*sin(phi1 - phi2))/(l1^2*sym_R...
*(4*m1 + 12*m2 - 9*m2*cos(phi1 - phi2)^2)^2) + (324*lambda4*→
  ← m2*cos(phi1 - phi2)^2*sin(phi1 - phi2))/(l1*l2*sym_R*(4*→
  ← m1 + 12*m2 - 9*m2*cos(phi1 - phi2)^2)^2) - 6*Rp2*dphi1*→
  ← l1*sin(phi1 - phi2) + 6*Rp2*dphi2*l1*sin(phi1 - phi2) + →
  ← 3*dphi1^2*l1^2*l2*m2*cos(phi1 - phi2)^2 - 3*dphi1^2*l1→
  ← ^2*l2*m2*sin(phi1 - phi2)^2 + 2*dphi2^2*l1*l2^2*m2*cos(→
  ← phi1 - phi2) + 3*g*l1*l2*m2*sin(phi1 - phi2)*sin(phi2) +→
  ← 3*g*l1*l2*m2*cos(phi1 - phi2)*cos(phi2)))/(l1^2*l2*(4*→
  ← m1 + 12*m2 - 9*m2*cos(phi1 - phi2)^2)) - (54*m2*cos(phi1→
  ← - phi2)*sin(phi1 - phi2)*(4*l2*((12*lambda2)/(l1^2*→
  ← sym_R*(4*m1 + 12*m2 - 9*m2*cos(phi1 - phi2)^2)) - (18*→
  ← lambda4*cos(phi1 - phi2))...
/(l1*l2*sym_R*(4*m1 + 12*m2 - 9*m2*cos(phi1 - phi2)^2))) + →
  ← 4*Rp1*dphi1*l2 + 6*Rp2*dphi1*l1*cos(phi1 - phi2) - 6*Rp2→
  ← *dphi2*l1*cos(phi1 - phi2) + 2*dphi2^2*l1*l2^2*m2*sin(→
  ← phi1 - phi2) + 2*g*l1*l2*m1*sin(phi1) + 4*g*l1*l2*m2*sin→
  ← (phi1) + 3*dphi1^2*l1^2*l2*m2*cos(phi1 - phi2)*sin(phi1 →
  ← phi2) - 3*g*l1*l2*m2*cos(phi1 - phi2)*sin(phi2)))/(l1→
  ← ^2*l2*(4*m1 + 12*m2 - 9*m2*cos(phi1 - phi2)^2)^2));
66 dx8 = (3*lambda4*(- 6*dphi2*l1*cos(phi1 - phi2)*sin(phi1 - phi2)→
  ← l2^2*m2^2 + 12*Rp2*l1*m2 + 4*Rp2*l1*m1))/(l1*l2^2*m2*(4*m1 →
  ← + 12*m2 - 9*m2*cos(phi1 - phi2)^2)) - dphi2*q4 - (3*lambda2→
  ← *(- 4*dphi2*l1*m2*sin(phi1 - phi2)*l2^2 + 6*Rp2*l1*cos(phi1 →
  ← phi2)))/(l1^2*l2*(4*m1 + 12*m2 - 9*m2*cos(phi1 - phi2)^2))→
  ← - lambda3;

dxdt = [dx1;dx2;dx3;dx4;dx5;dx6;dx7;dx8];

```

```
71 end % function RandwertproblemDGL
```

Listing 4.2: Quellcode der Funktion RandwertproblemRB

```
function [deltaRB ] = RandwertproblemRB( xa, xb )
%RANDWERTPROBLEMRB Summary of this function goes here
% Detailed explanation goes here
4 deltaRB = [xa(1);
             xa(2);
             xa(3);
             xa(4);
             xb(1)-pi;
9           xb(2);
             xb(3)-pi;
             xb(4)];

14 end
```

Listing 4.3: Quellcode der Funktion berechneTrajektorie

```
function [ stTraj ] = berechneTrajektorie( stPendel, Q, R, T )
%BERECHNETRAJEKTORIE Summary of this function goes here
% Detailed explanation goes here

5 RandwertproblemDGLhandle = @(t,x) RandwertproblemDGL(t,x,stPendel,Q,→
    ←R);

intervals = 1000;

solinit.x = linspace(0,T,intervals);
10 solinit.y = [linspace(0,pi,intervals);
               zeros(1,intervals);
               linspace(0,pi,intervals);
               zeros(1,intervals);
               zeros(1,intervals);
15              zeros(1,intervals);
               zeros(1,intervals);
               zeros(1,intervals);];
```

```

RelTol = 1e-10;
20 bvp4cOptions = bvpset('RelTol',RelTol,'Stats','on');

for ii = 1:15
    sol = bvp4c(RandwertproblemDGLhandle,@RandwertproblemRB,solinit,→
        ←bvp4cOptions);
    solinit = sol;
25 if isfield(sol,'stats')
        break
    end
end

30 [f, h] = nonlinear_model;

dfdu_symb = jacobian(f,sym('M'));

35 dfdu = double(subs(dfdu_symb,{sym('phi1'),sym('phi2')},{sol.y(1,:),→
    ←sol.y(3,:)})));

stTraj.T = T;
stTraj.vT = sol.x;
40 stTraj.vU = diag(- R^(-1)* dfdu' * sol.y(5:8,:))';
% vU = diag(stTraj.vU);
stTraj.mX = sol.y(1:4,:);

end

```

Lösungen können für den Wertebereich von $0,65\text{ s} \leq T \leq 1,9\text{ s}$ gefunden werden. Die Unterschreitung von $T = 0,65\text{ s}$ führt zu numerischen Problemen. Eine Überschreitung von $T = 1,9\text{ s}$ ist physikalisch unmöglich, da das zweite Pendel eine Mindestgeschwindigkeit braucht um in die aufrechte Lage aufgeschwungen werden zu können.

Die kleinste T , bei der die Stellgrößenbeschränkung $|M|_{\max} = 1,5\text{ Nm}$ noch eingehalten wird ist $T = 1,1185\text{ s}$.

Nachfolgend werden Plots der Zustände $\mathbf{x}^*(t)$ und $\mathbf{M}^*(t)$ für verschiedene Übergangszeiten gezeigt.

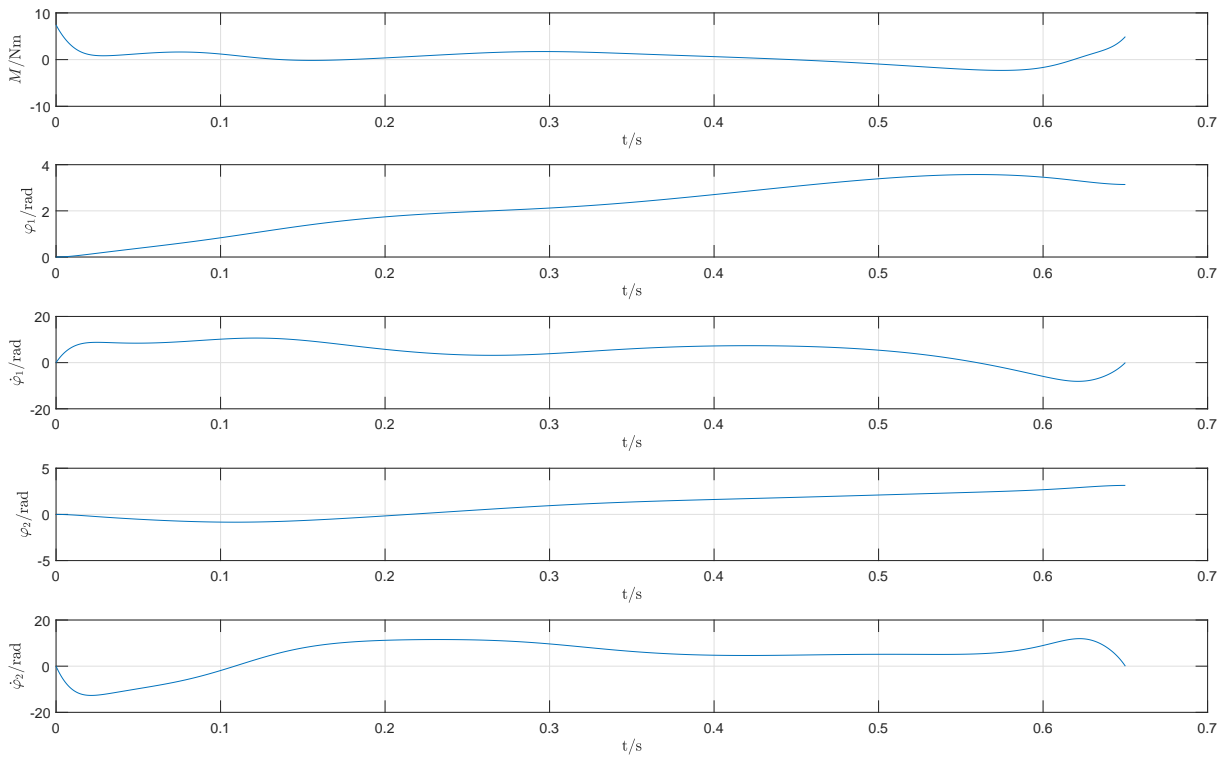


Figure 4.1: Plot der Zustände $\mathbf{x}^*(t)$ und $M^*(t)$ für $T = 0,65\text{ s}$

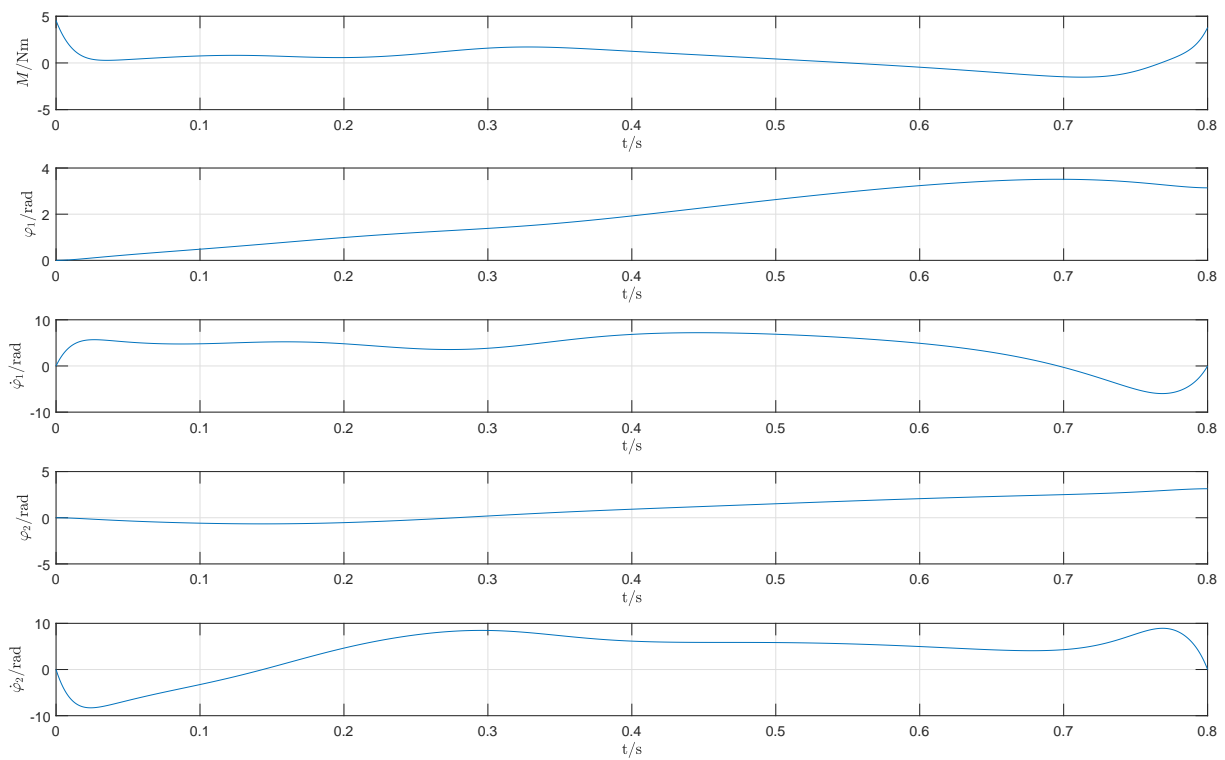


Figure 4.2: Plot der Zustände $\mathbf{x}^*(t)$ und $M^*(t)$ für $T = 0,8\text{ s}$

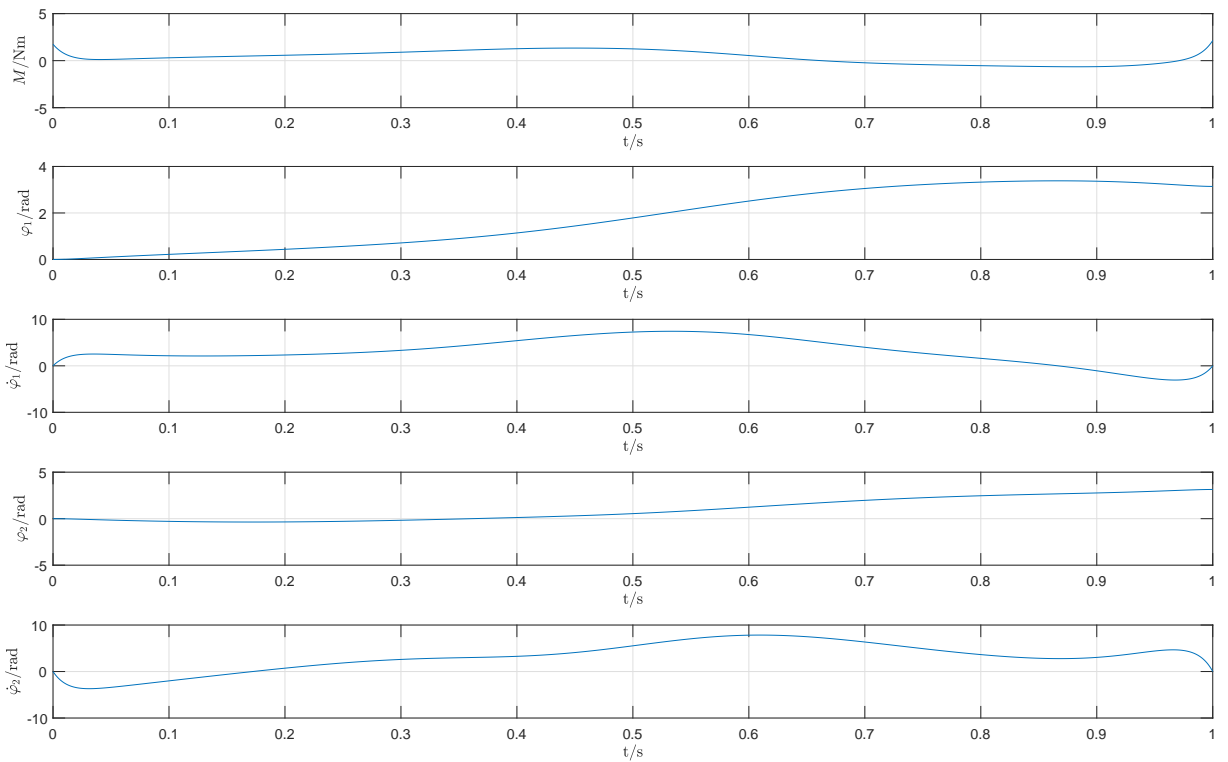


Figure 4.3: Plot der Zustände $\mathbf{x}^*(t)$ und $M^*(t)$ für $T = 1$ s

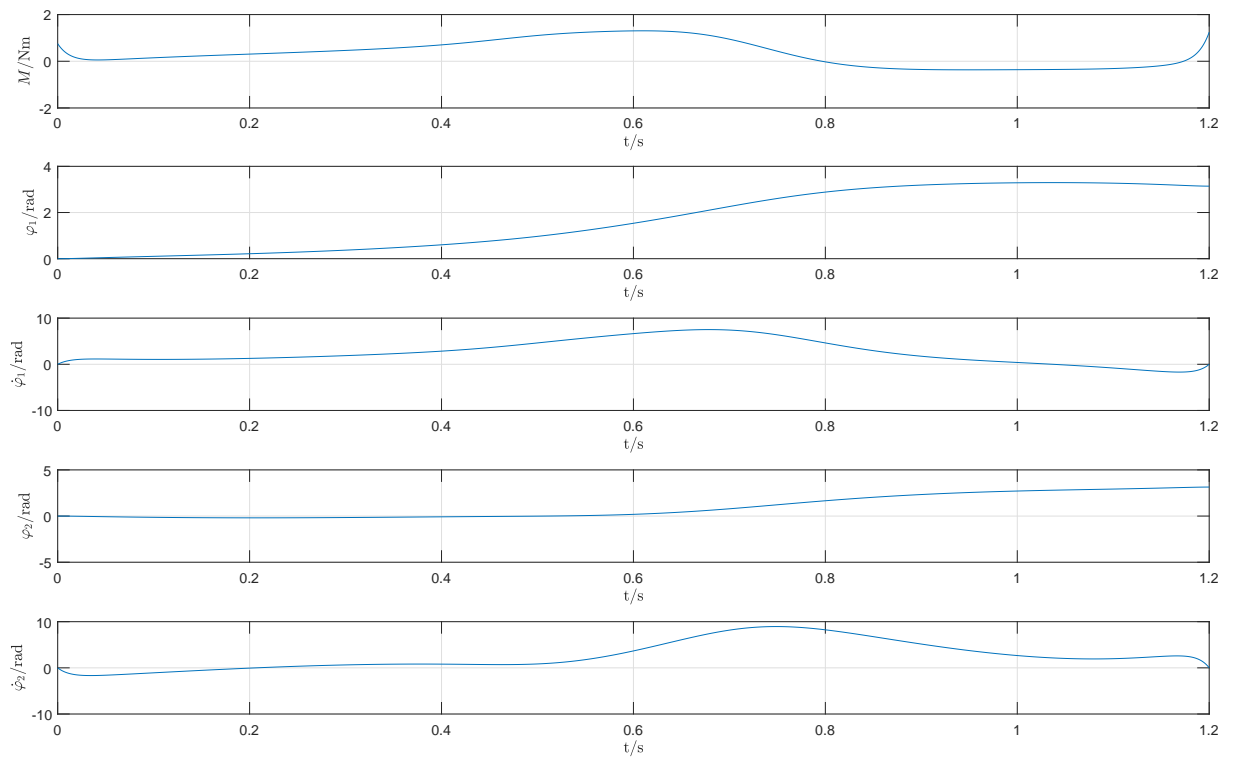


Figure 4.4: Plot der Zustände $\mathbf{x}^*(t)$ und $M^*(t)$ für $T = 1,2$ s

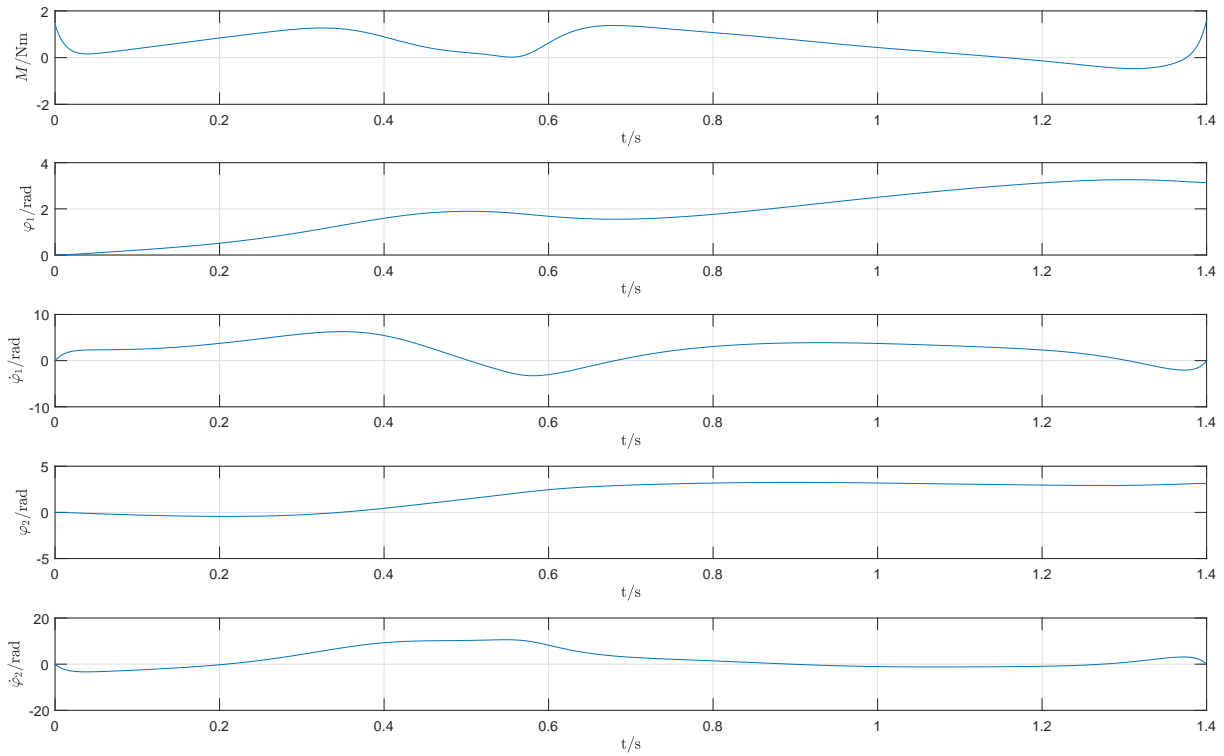


Figure 4.5: Plot der Zustände $\mathbf{x}^*(t)$ und $M^*(t)$ für $T = 1,4$ s

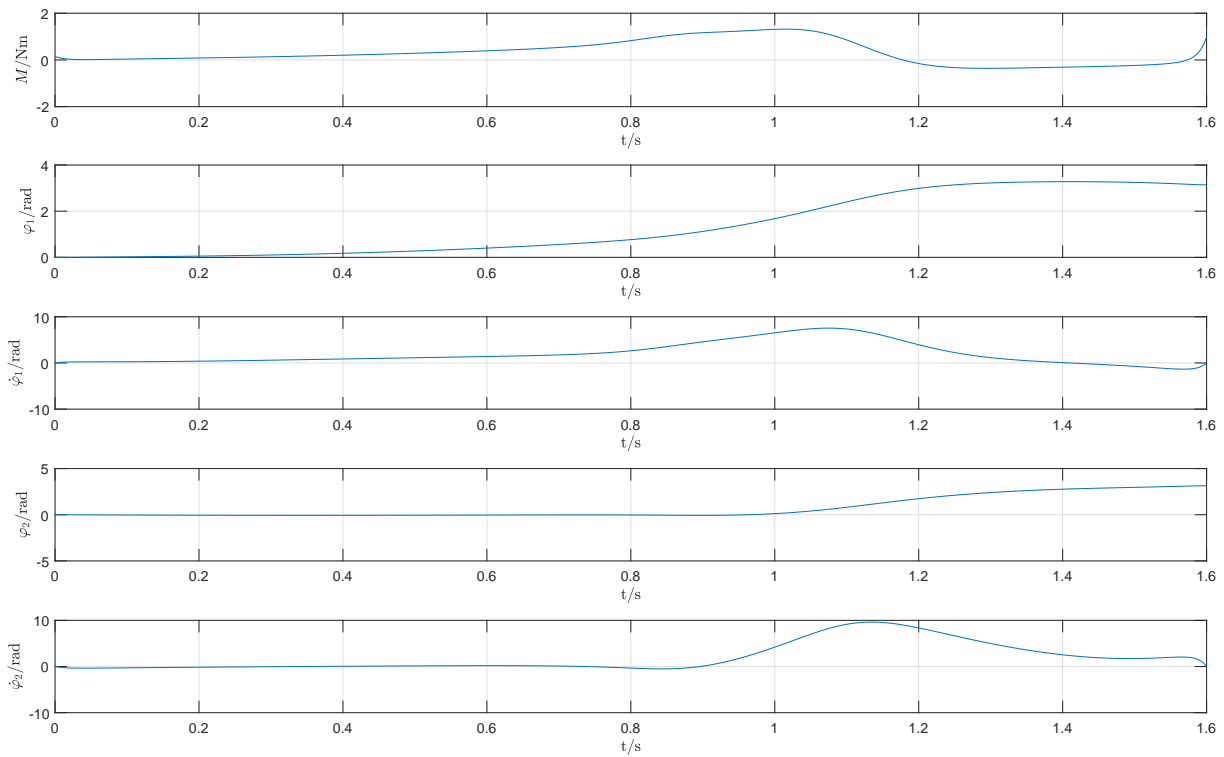


Figure 4.6: Plot der Zustände $\mathbf{x}^*(t)$ und $M^*(t)$ für $T = 1,6$ s

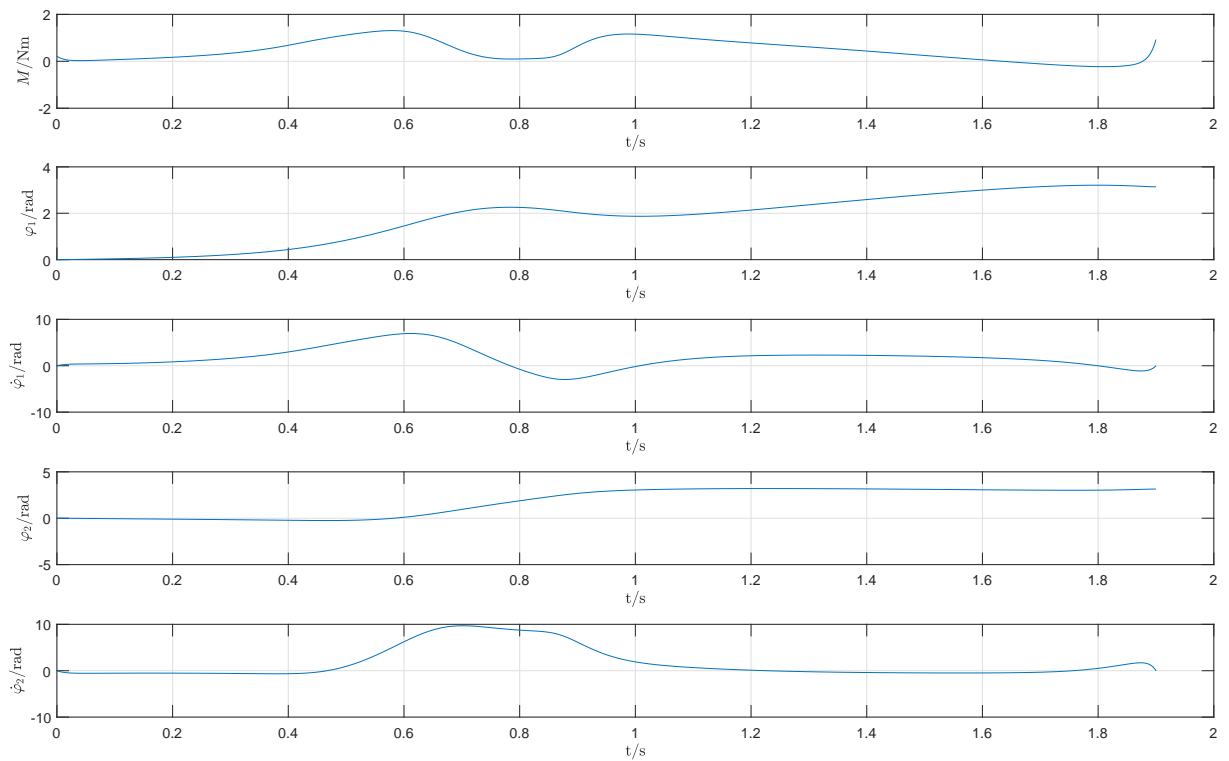


Figure 4.7: Plot der Zustände $\mathbf{x}^*(t)$ und $M^*(t)$ für $T = 1,9\text{s}$

Erklären Sie die Auswirkung der Übergangszeiten T auf das Ergebnis

- T größer \rightarrow M_{\max} kleiner
- ...

4.9 Simulation

Das Simulinkmodell der Aufschwungsteuerung sieht wie folgt aus:

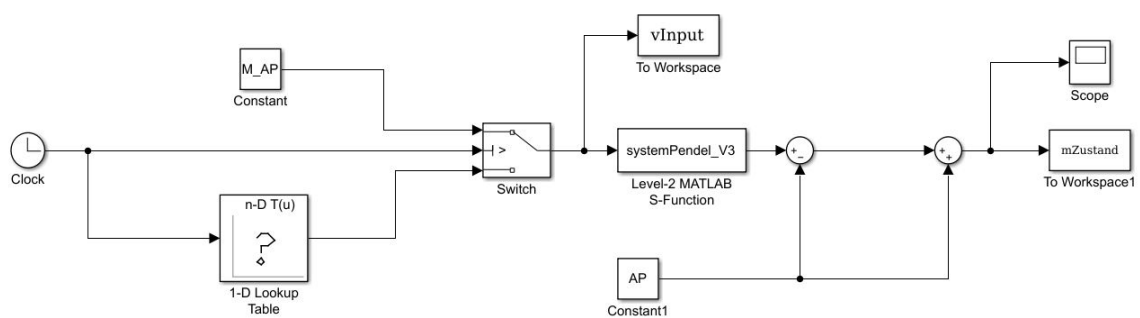


Figure 4.8: Simulink-Modell der Aufschwungsteuerung

Die folgende Abbildung veranschaulicht die Simulation und zusammen mit den vorgegebenen Trajektorien bei gegebenen Pendeldaten. Für die darauffolgenden Abbildungen wurden die Pendeldaten verändert ($l_1 = 0,3\text{ m}$, $l_2 = 0,3\text{ m}$, $R_p1 = 10^{-3}$, $R_p2 = 10^{-4}$).

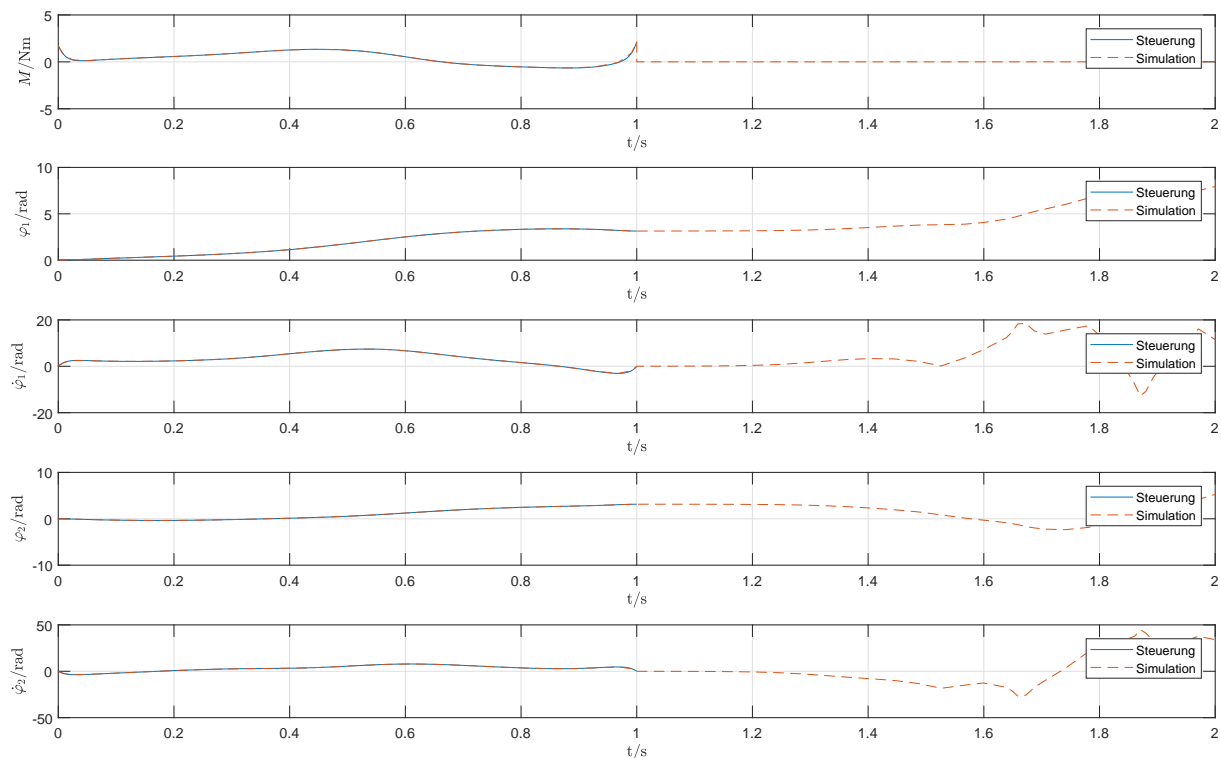


Figure 4.9: Plot der Zustände $\mathbf{x}^*(t)$ und $\mathbf{M}^*(t)$ für $T = 1\text{ s}$, $T_{sim} = 2\text{ s}$ mit gegebenen Pendeldaten

Sollen wir noch alles animieren und Screenshots einfügen? Siehe Aufgabenstellung

Der Zeitplot 4.9 zeigt eine Kongruenz zwischen den simulierten Verläufen und den berechneten Trajektorien. Die Zeitplots, die die Simulationen des Doppelpendels mit veränderten Pendeldaten zeigen, lassen jedoch eindeutig erkennen, dass die berechneten Trajektorien mit den oben angegebenen veränderten Pendeldaten für eine Aufschwungsteuerung nicht geeignet sind. Dies lässt auch vermuten, dass die berechneten Trajektorien empfindliche gegenüber einer Änderung der Pendeldaten sind. Schließlich ist eine Aufschwungsteuerung nur für die dafür berechneten Pendeldaten gültig.

check
please!

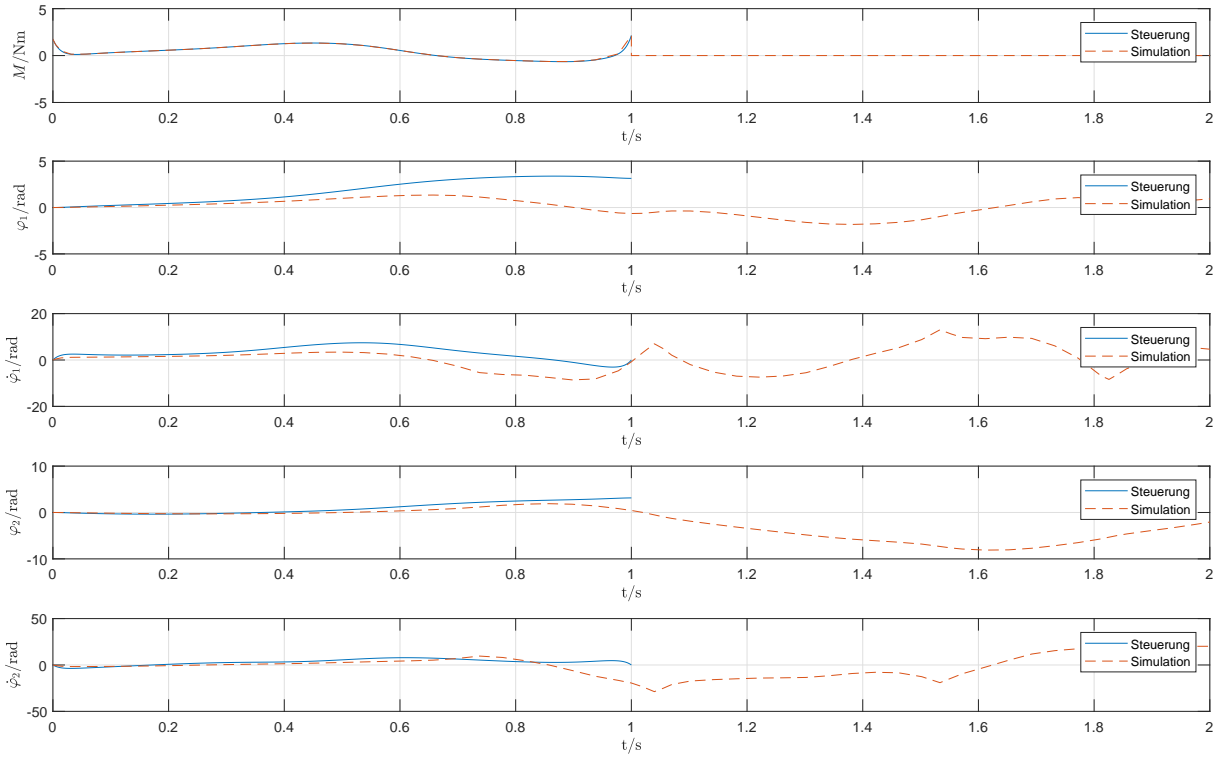


Figure 4.10: Plot der Zustände $\mathbf{x}^*(t)$ und $\mathbf{M}^*(t)$ für $T = 1\text{ s}$, $T_{sim} = 2\text{ s}$ ($l_1 = 0,3\text{ m}$, $l_2 = 0,3\text{ m}$, $R_{p1} = 10^{-3}$, $R_{p2} = 10^{-4}$)

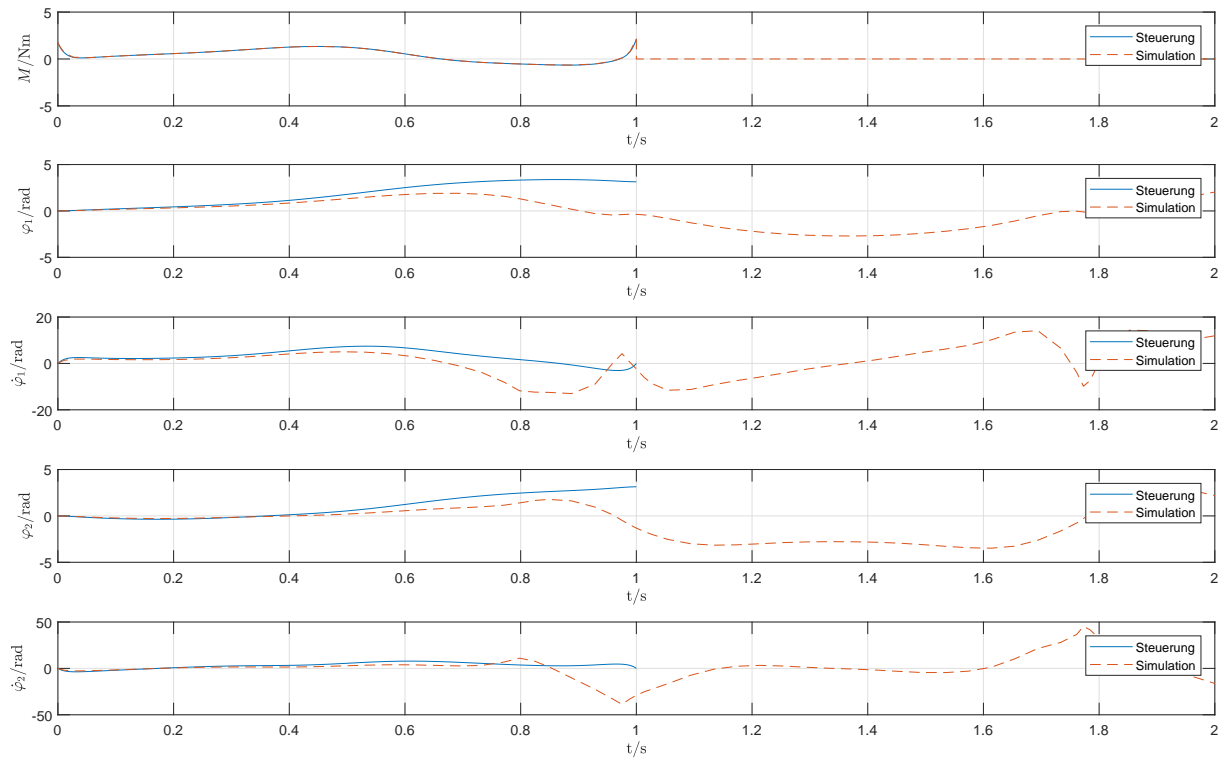


Figure 4.11: Plot der Zustände $\mathbf{x}^*(t)$ und $\mathbf{M}^*(t)$ für $T = 1\text{ s}$, $T_{sim} = 2\text{ s}$ ($m_1 = 0,4\text{ kg}$, $m_2 = 0,4\text{ kg}$)

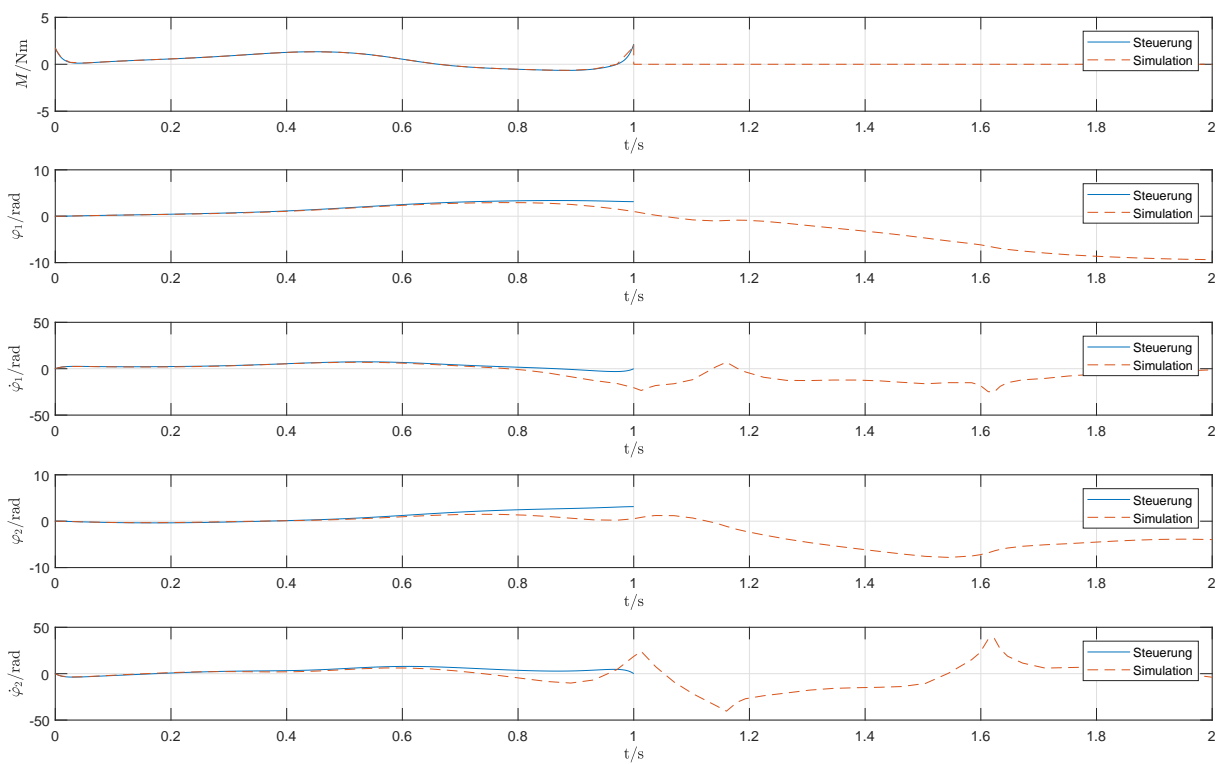


Figure 4.12: Plot der Zustände $\mathbf{x}^*(t)$ und $\mathbf{M}^*(t)$ für $T = 1\text{ s}$, $T_{sim} = 2\text{ s}$ ($g = 11\text{ N/kg}$)