Mec6313 Homework 4

H_∞-state feedback control for a P-P robot, designed using linear matrix inequalities

Stefan Bracher

(Version 2)

Presented to: El-Kébir Boukas

November 28, 2007

Index

1. Introduction

This is the last report in a series of homeworks [1, 2, 3] in which different types of controllers for a pick-and-place robot have been designed. This time, the system is stabilized by a state feedback controller, of which the gains are computed with an $\boldsymbol{H}_\infty$-approach using linear matrix inequalities (LMIs) to assure robustness as well as perturbation rejection.

2. The system

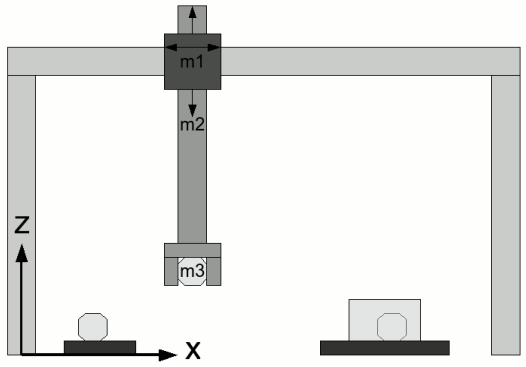

Figure 1: The pick and place robot

The system studied, a serial pick-and-place robot, is the same as in the previous homeworks. For details refer to [1, 2].

Its parameters are:
Nominal mass fist member: $m_1 = 1\,kg$
Nominal mass second member: $m_2 = 1\,kg$
Load: $m_3 = 0 - 0.5\,kg$
Nominal friction coefficients: $k_1 = k_2 = 0.1\,kg/s$

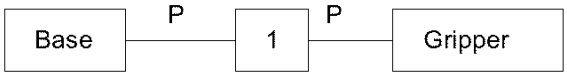

Figure 2: Kinematic chain

Like in the other homeworks, gravity is regarded as a perturbation and the load mass m_3 is an uncertainty.

In addition to the previous works, an other perturbation with limited energy, $\omega(t)$, which has to be rejected, is added to the system. It is supposed that $\omega(t) = \sin(10\pi \cdot t) \cdot e^{-0.8 \cdot t}$ is a parasite signal, due to vibration in the sensor of the x-position and is transmitted partially (factor 0.2) to the controller.

States: $\boldsymbol{x} = [\xi, x, \dot{x}, \zeta, z, \dot{z}]'$ $\quad \dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u} + B_\omega * \omega$
Inputs: $\boldsymbol{u} = [F_x, F_z]'$ $\quad \boldsymbol{y} = C_y \boldsymbol{x}$
Perturbation: $\omega(t) = \sin(10\pi \cdot t) \cdot e^{-0.8 \cdot t}$ $\quad \boldsymbol{z} = C_z \boldsymbol{x}$
Output: $\boldsymbol{y} = [\xi, x, \dot{x}, \zeta, z, \dot{z}]'$
Controlled output: $\boldsymbol{z} = [\xi_z, x_z, \dot{x}_z, \zeta_z, z_z, \dot{z}_z]'$

with

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{-k_1}{(m_1+m_2)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & \frac{-k_2}{m_2} \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{(m_1+m_2)} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{m_2} \end{bmatrix}, \quad B_w = \begin{bmatrix} 0 \\ 0.2 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad C_y = C_z = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

(2.1)

3. Control of the nominal System

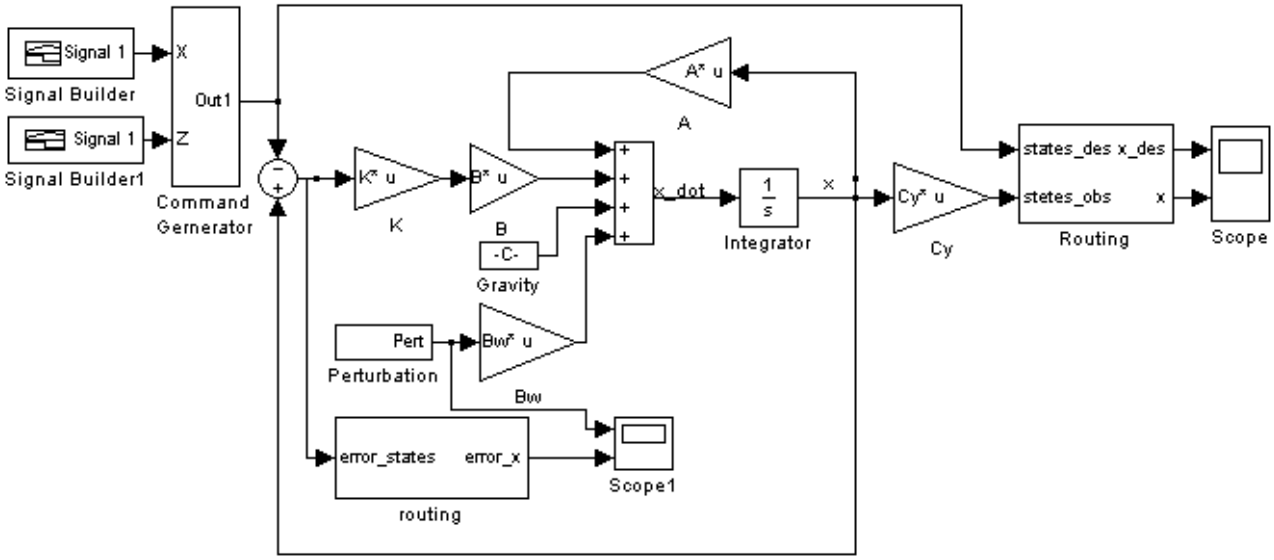

Figure 3: Simulink model of the nominal system

3.1 State feedback controller design

The system is stabilized with a state-feedback controller of the form $u = K\boldsymbol{x}$. The gains K are computed using the following H_∞ - LMI:

H_∞ - LMI stability theorem for nominal systems with state feedback controller [4]
Let γ be a positive constant. If there exist a symmetric positive-definite matrix X>0 and a matrix Y so that the following LMI holds: $\begin{bmatrix} J & B_w & X C_z^T + Y^T D_z^T \\ B_w^T & -\gamma^2 \boldsymbol{I} & B_z^T \\ C_z X + D_z Y & B_z & -\boldsymbol{I} \end{bmatrix} < 0$ with $J = X A^T + A X + Y^T B^T + B Y$ then, the nominal system under the controller $u = K\boldsymbol{x}$ with $K = Y X^{-1}$ is stable and, moreover, the closed-loop system satisfies the disturbance rejection of level γ.

For a rejection level of $\gamma = 0.8$ and a slightly changed matrix A ($A = A + 2\cdot\boldsymbol{I}$) to decrease reaction time (and the gains that otherwise would by physically impossible to attain), the computed solution using Yalmip [5] and Sedumi [6] is:

$$K = \begin{bmatrix} -724.6536 & -385.4958 & -50.0793 & -0.0024 & -0.0054 & -0.0015 \\ 0.0068 & 0.0038 & 0.0005 & -151.3128 & -83.4744 & -13.2542 \end{bmatrix} \quad (3.1.1)$$

As it can be seen, there is some coupling, what should not occur, as the x- and z-direction are completely independent. As it is very small, it might come from some numerical residual in the LMI solution algorithm.

3.2 Simulation

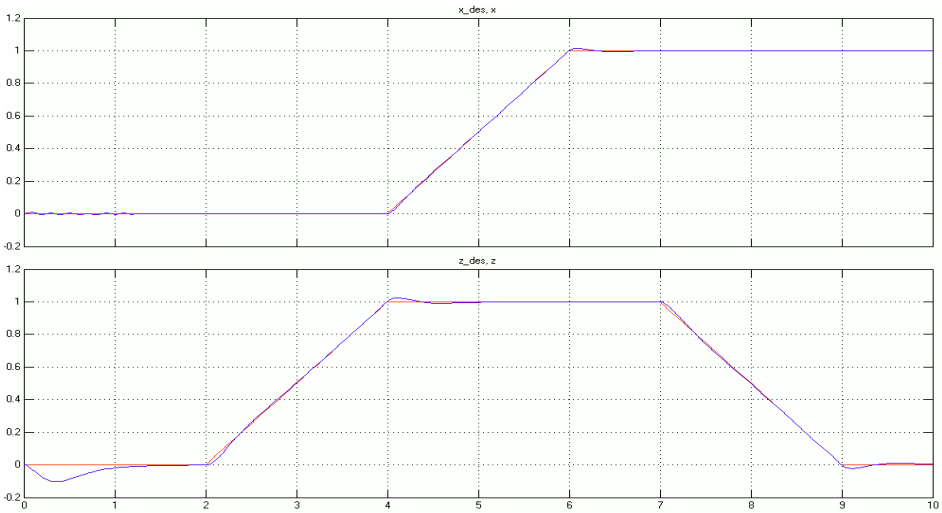

Figure 4: Simulink simulation result using K (3.1.1) and $\gamma = 0.8$. Commanded trajectory (red) and the real positions (blue).

As the simulation result (See figure 4) shows, the robot follows the trajectory as desired. Note that the difference in z for t<1s is due to the fact that the controller needs some time to compensate for the gravity, as the integral part first has to sum up.

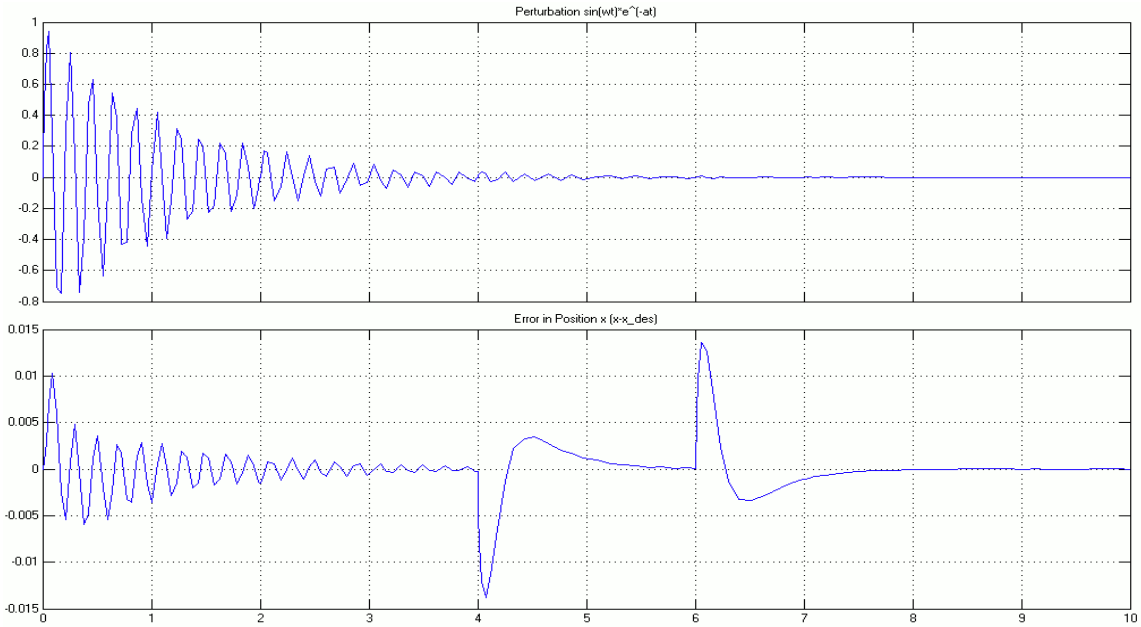

Figure 5: Simulink simulation result using K (3.1.1), a=0.8 and $\gamma = 0.8$. Top: Perturbation signal, Bottom: System error in x

The perturbation is rejected as desired. As it can be seen on the top, the perturbation was assumed to fade out in order to be sure it has limited energy. Note that the two error peaks at t=4s and t=6s are coming from the trajectory command change and not from the perturbation itself.

4. Control of the System with uncertainties

4.1 System uncertainties

The system uncertainties due to the variable load mass are norm bounded and modeled like in the previous homeworks [2, 3]:

$$A=\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -k_1*(\frac{1}{(m_1+m_2)}+\Delta_1) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -k_2*(\frac{1}{m_2}+\Delta_2) \end{bmatrix}=\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{-k_1}{(m_1+m_2)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & \frac{-k_2}{m_2} \end{bmatrix}+D_A F_A E_A$$

with $D_A=\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{(m_1+m_2+m_{3max})}-\frac{1}{(m_1+m_2)} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{(m_2+m_{3mac})}-\frac{1}{m_2} \end{bmatrix}$, $1\geqslant F_A(1,1)\geqslant 0$, $E_A=\begin{bmatrix} 0 & 0 & -k_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -k_2 \end{bmatrix}$

(4.1.1)

$$B=\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{(m_1+m_2)}+\Delta_3 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{m_2}+\Delta_4 \end{bmatrix}=\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{(m_1+m_2)} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{m_2} \end{bmatrix}+D_B*F_B*E_B \quad \text{with} \quad D_B=\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{(m_1+m_2+m_{3max})}-\frac{1}{(m_1+m_2)} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{(m_2+m_{3mac})}-\frac{1}{m_2} \end{bmatrix},$$

$1\geqslant F_B(1,1)\geqslant 0$, $E_B=\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

(4.1.2)

All C-matrices of the system (2.1) are not affected.

4.2 State feedback controller design

H_∞ - LMI taking in account the system uncertainties of section 4.1 is:

H_∞ - LMI stability theorem for systems with norm bounded uncertainties and state feedback controller [4]

Let γ be a positive constant. If there exist a symmetric positive-definite matrix X>0, a matrix Y and positive scalars $\epsilon_A>0$, $\epsilon_B>0$, $\epsilon_{Cz}>0$ and $\epsilon_{Dz}>0$ so that the following LMI holds for all admissible uncertainties:

$$\begin{bmatrix} J & B_w & X C_z^T + Y^T D_z^T & Z \\ B_w^T & -\gamma^2 \boldsymbol{I} & B_z^T & 0 \\ C_z X + D_z Y & B_z & -\boldsymbol{I} & 0 \\ Z^T & 0 & 0 & -V \end{bmatrix} < 0$$

with

$$J = X A^T + A X + Y^T B^T + B Y + \epsilon_A D_A D_A^T + \epsilon_B D_B D_B^T$$

$$U = \boldsymbol{I} - \epsilon_{Dz} D_{Dz} D_{Dz}^T - \epsilon_{Cz} D_{Cz} D_{Cz}^T$$

$$Z = \begin{bmatrix} X E_A^T & Y^T E_B^T & Y T E_{Cz}^T & Y^T E_{Dz}^T \end{bmatrix}$$

$$V = diag\begin{bmatrix} \epsilon_A \boldsymbol{I} & \epsilon_B \boldsymbol{I} & \epsilon_{Cz} \boldsymbol{I} & \epsilon_{Dz} \boldsymbol{I} \end{bmatrix}$$

then the uncertain system under the controller $u = K \boldsymbol{x}$ with $K = Y X^{-1}$ is stable and, moreover, the closed-loop system satisfies the disturbance rejection of level γ.

For a rejection level of $\gamma = 0.8$, a variable load of maximal 0.5kg and a slightly changed matrix A ($A = A + 2 \cdot \boldsymbol{I}$) to decrease reaction time (and thus the resulting gain), the computed solution using Yalmip [5] and Sedumi [6] is:

$$K = 1.0e{+}003 \cdot \begin{bmatrix} -3.3544 & -2.0014 & -0.2824 & -0.0000 & -0.0000 & -0.0000 \\ 0.0000 & 0.0000 & 0.0000 & -0.2502 & -0.1370 & -0.0221 \end{bmatrix} \quad (4.2.1)$$

Although the reaction time already has been reduced by a factor 2 in order to get reasonable gains, the gain K is still quite high.

4.3 Simulation

The simulation result for no load is:

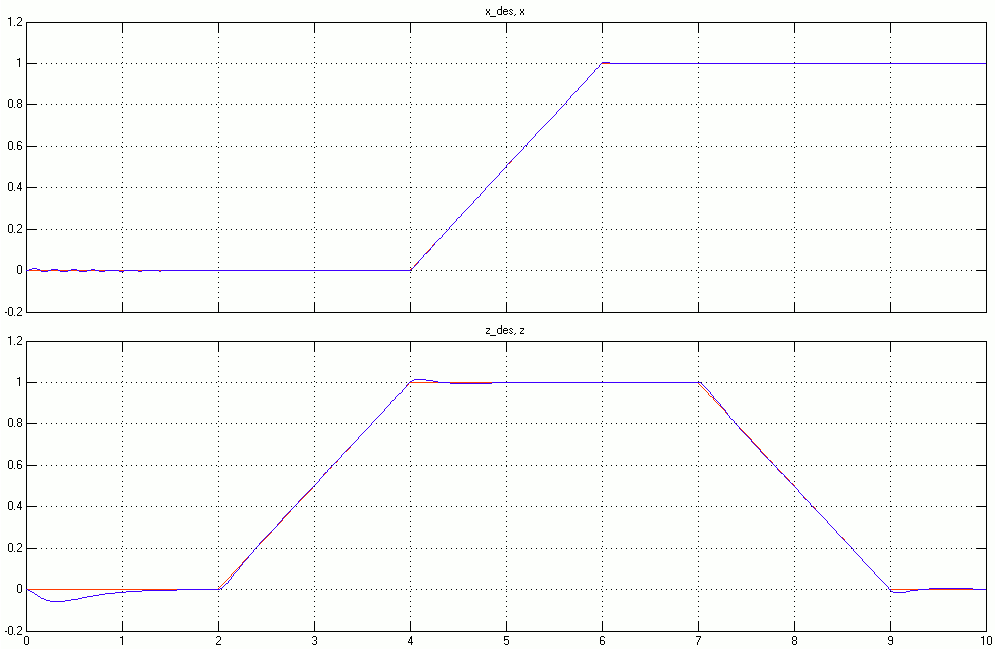

Figure 6: Simulink simulation result using K (4.2.1), $\gamma = 0.8$ and no load. Commanded trajectory (red) and the real positions (blue).

The result is comparable to the nominal case. Again, the robot follows the command as desired, but needs some time to adjust for gravity (that changes from zero to 1G at the beginning of the simulation), what creates the error in z for t<1s.

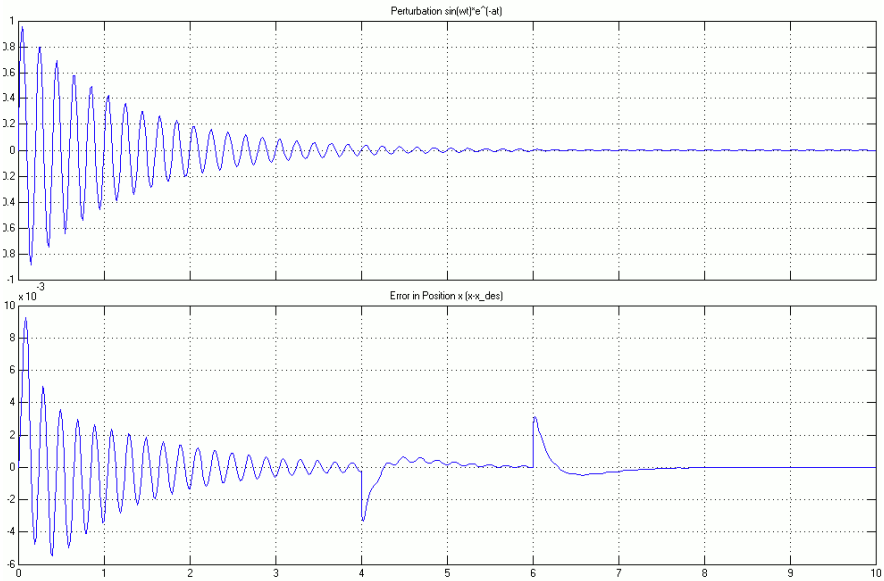

Figure 7: Simulink simulation result using K (4.2.1) ,a=.08, $\gamma = 0.8$ and no load. Top: Perturbation signal, Bottom: System error in x

The perturbation is again successfully rejected.

The simulation result for maximal load (0.5kg) is:

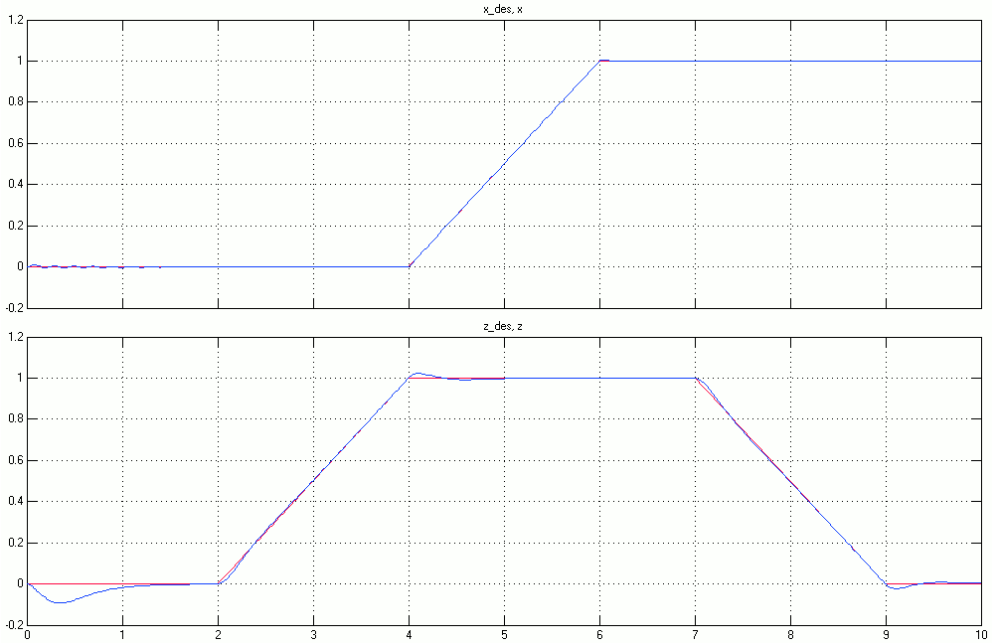

Figure 8: Simulink simulation result using K (4.2.1) and maximal load. Commanded trajectory (red) and the real positions (blue).

Robustness is assured, the change of the load does not destabilize the system. Only the peak errors, are, due to higher inertia and force due to gravity, slightly bigger.

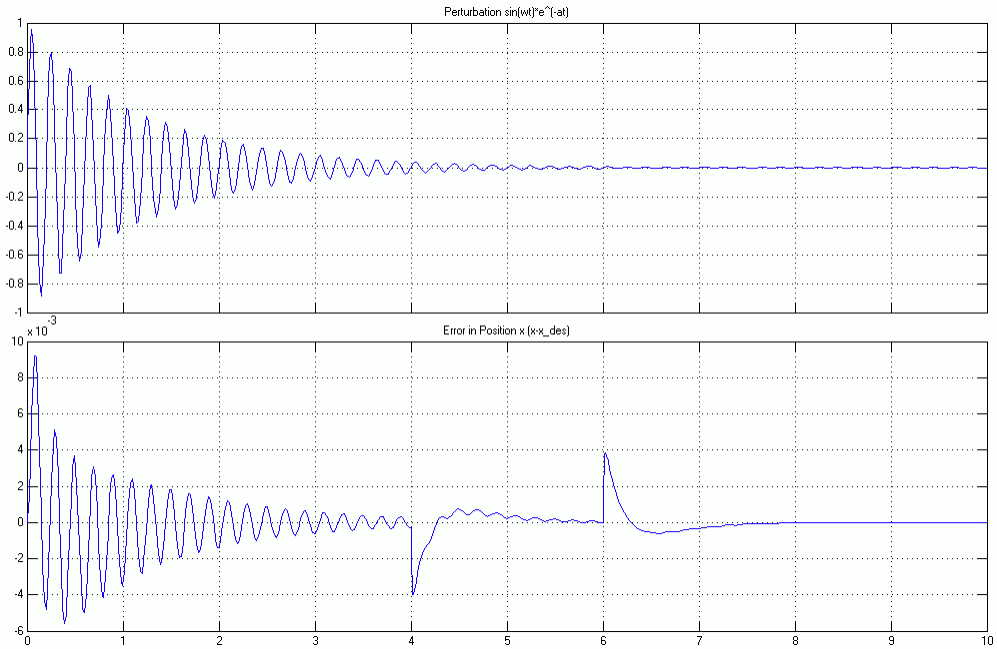

Figure 9: Simulink simulation result using K (4.2.1) and maximal load. Top: Perturbation signal, Bottom: System error in x

Perturbation rejection as well is maintained as desired.

5. Conclusion

The H_∞-LMI theorems worked for both, the nominal- and the case with variable load. Theoretically any perturbation signal $\omega(t)$ with finite energy, affecting the x-position of the robot, should be rejected by the obtained controllers. Simulations done with $\omega(t)=\sin(10\pi\cdot t)\cdot e^{-0.8\cdot t}$ showed that the rejection rate for the nominal and the uncertain system were even higher than what was demanded.

References

[1] Bracher Stefan,"State feedback controller and observer design for a pick and place robot", Homeworks MEC6313, Ecole Polytechnique de Montréal, 2007

[2] Bracher Stefan,"State feedback controller design for a system with uncertainties using linear matrix inequalities", Homeworks MEC6313, Ecole Polytechnique de Montréal, 2007

[3] Bracher Stefan,"Output feedback control for a P-P robot, designed using linear matrix inequalities", Homeworks MEC6313, Ecole Polytechnique de Montréal, 2007

[4] Boukas El-Kebir*, "Class Notes MEC6313"*, Ecole Polytechnique de Montréal, 2007

[5] Löfberg J.*, "YALMIP : A Toolbox for Modeling and Optimization in MATLAB*." In Proceedings of the CACSD Conference, Taipei, Taiwan, 2004.

[6] Sedumi: http://sedumi.mcmaster.ca

Appendix

LMI_hinf_nom.m

```
function [K, lmiok]=LMI_hinf_nom(A, B, Bw, Cz, Dz, Bz, gamma, speed)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function:     LMI_hinf_nom.m
% Description:  Calculates the gains K for state feedback using H-inf
% Author:       Stefan Bracher
% Requirements: Yalmip (http://control.ee.ethz.ch/~joloef/yalmip.php)
%               Sedumi (http://sedumi.mcmaster.ca)
%
% Inputs:       A:       System matrix A of x_dot=A*x+B*u+Bw*w
%               B:       System matrix B of x_dot=A*x+B*u+Bw*w
%               Bw:      System matrix Bw of x_dot=A*x+B*u+Bw*w
%               Cz:      System matrix Cz of z_dot=C_z*x+D_z*u+Bz*w
%               Dz:      System matrix Dz of z_dot=C_z*x+D_z*u+Bz*w
%               Bz:      System matrix Bz of z_dot=C_z*x+D_z*u+Bz*w
%               gamma:   Rejection rate
%               speed:   Add additional reaction time (speed > 0)
% Outputs:      K:       State Feedback gain u=K*x
%               lmiok:   0: Not successful, 1: successful
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% System dimensions %%%
lineA=size(A, 1);  % Lines of A
colB=size(B, 2);   % Colums of B
colBw=size(Bw, 2); % Colums of Bw
lineBz=size(Bz, 1); % Lines of Bz

A=A+speed*eye(size(A)); % Speed up the system

%%% LMI Variables %%%
X=sdpvar(lineA, lineA, 'symmetric');
Y=sdpvar(colB, lineA, 'full');
J=X*A'+A*X+Y'*B'+B*Y;

%%% LMI's %%%
F=set(X>0);            % X positive definit
F=F+set([J          Bw                      X*Cz'+Y'*Dz';
         Bw'        -gamma*gamma*eye(colBw) Bz'        ;
         Cz*X+Dz*Y  Bz                      -eye(lineBz)
         ]<0);         % Main LMI
Sol=solvesdp(F);       % Solving it


%%% Extract Data %%%
X=double(X);
Y=double(Y);
K=Y*inv(X);
[primal, dual]=checkset(F);   % Get primary and dual constraint residuals

if ((primal(1)>0)&&(primal(2)>0)&&(abs(dual(1))<1)&&(abs(dual(2))<1))
% The system is stable if the primal residuals are positive and the dual
% residuals are small
lmiok=1;
else
lmiok=0;
end
```

LMI_hinf_uncert.m

```
function [K, lmiok]=LMI_hinf_uncert(A, DA, EA, B, DB, EB, Bw, Cz, ECz, Dz, EDz, Bz, gamma, speed)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function:     LMI_hinf_uncert.m
% Description:  Calculates the gains K for state feedback using H-inf
%               with uncertainties
% Author:       Stefan Bracher
% Requirements: Yalmip (http://control.ee.ethz.ch/~joloef/yalmip.php)
%               Sedumi (http://sedumi.mcmaster.ca)
%
% Inputs:       A:       System matrix A of x_dot=A*x+B*u+Bw*w
%               DA:      Uncertainty matrix DA of A=A+DA*FA*EA
%               EA:      Uncertainty matrix EA of A=A+DA*FA*EA
%               B:       System matrix B of x_dot=A*x+B*u+Bw*w
%               DB:      Uncertainty matrix DB of B=B+DB*FB*EB
%               EB:      Uncertainty matrix EB of B=B+DB*FB*EB
%               Bw:      System matrix Bw of x_dot=A*x+B*u+Bw*w
%               Cz:      System matrix Cz of z_dot=C_z*x+D_z*u+Bz*w
%               ECz:     Uncertainty matrix ECz of Cz=Cz+DCz*FCz*ECz
%               Dz:      System matrix Dz of z_dot=C_z*x+D_z*u+Bz*w
%               EDz:     Uncertainty matrix EDz of Dz=Dz+DDz*FDz*EDz
%               Bz:      System matrix Bz of z_dot=C_z*x+D_z*u+Bz*w
%               gamma:   Rejection rate
%               speed:   Add additional reaction time (speed > 0)
% Outputs:      K:       State Feedback gain u=K*x
%               lmiok:   0: Not successful, 1: successful
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% System dimensions %%%
lineA=size(A, 1);  % Lines of A
colB=size(B, 2);   % Colums of B
colBw=size(Bw, 2); % Colums of Bw
lineBz=size(Bz, 1); % Lines of Bz
colBz=size(Bz, 2); % Lines of Bz

A=A+speed*eye(size(A)); % Speed up the system

%%% LMI Variables %%%
X=sdpvar(lineA, lineA, 'symmetric');
Y=sdpvar(colB, lineA, 'full');
epA=sdpvar(1, 1);
epB=sdpvar(1,1);
epCz=sdpvar(1, 1);
epDz=sdpvar(1,1);

J=X*A'+A*X+Y'*B'+B*Y+epA*DA*DA'+epB*DB*DB';
U=eye(lineBz);
Z=[X*EA', Y'*EB', Y'*ECz', Y'*EDz'];
V=[epA*eye(size(X*EA', 2))    zeros(size(Y'*EB', 2))    zeros(size(Y'*ECz', 2))
zeros(size(Y'*EDz', 2));
   zeros(size(X*EA', 2))      epB*eye(size(Y'*EB', 2))  zeros(size(Y'*ECz', 2))
zeros(size(Y'*EDz', 2));
   zeros(size(X*EA', 2))      zeros(size(Y'*EB', 2))    epCz*eye(size(Y'*ECz', 2))
zeros(size(Y'*EDz', 2));
   zeros(size(X*EA', 2))      zeros(size(Y'*EB', 2))    zeros(size(Y'*ECz', 2))
epDz*eye(size(Y'*EDz', 2))];


%%% LMI's %%%
F=set(X>0);            % X positive definit
F=F+set(epA>0);
F=F+set(epB>0);
F=F+set(epCz>0);
F=F+set(epDz>0);
F=F+set([J          Bw                      X*Cz'+Y'*Dz'                Z;
         Bw'        -gamma*gamma*eye(colBw) Bz'                         zeros(colBw, size(Z,2));
         Cz*X+Dz*Y  Bz                      -U                          zeros(lineBz, size(Z,2));
         Z'         zeros(size(Z,2), colBz) zeros(size(Z,2), lineBz)    -V
         ]<0);         % Main LMI
Sol=solvesdp(F);       % Solving it
```

```
%%% Extract Data %%%
X=double(X);
Y=double(Y);
K=Y*inv(X);
[primal, dual]=checkset(F);   % Get primary and dual constraint residuals

if ((primal(1)>0)&&(primal(2)>0)&&(abs(dual(1))<1)&&(abs(dual(2))<1))
% The system is stable if the primal residuals are positive and the dual
% residuals are small
lmiok=1;
else
lmiok=0;
end
```

system_initialisation.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Script:       MEC 6313 Homework 4, system_initialisation.m
% Description:  Initializes the system to be analyzed
% Author:       Stefan Bracher
% Requirements: none
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Parameters %%%
m1=1;        % Mass of the first moving member, including motor
m2=1;        % Mass of the second moving member
k1=0.1;      % Friction coefficient member 1
k2=0.1;      % Friction coefficient member 2

%%% System %%%
A=[ 0   1    0            0     0    0;
    0   0    1            0     0    0;
    0   0   -k1/(m1+m2)   0     0    0;
    0   0     0           0     1    0;
    0   0     0           0     0    1;
    0   0     0           0     0    -k2/m2];

B=[0            0;
   0            0;
   1/(m1+m2)    0;
   0            0;
   0            0;
   0            1/m2];

Bw=[0 0.2 0 0 0 0]';
Cy=eye(size(A));
Dy=zeros(size(B));
By=zeros(size(Bw));
Cz=Cy;
Dz=Dy;
Bz=zeros(size(Bw));
```

system_uncertainties.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Script:       MEC 6313 Homework 4, system_uncertainties.m
% Description:  Initializes the system uncertainties
% Author:       Stefan Bracher
% Requirements: none
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

EA=[0 0 -k1 0 0 0;
    0 0 0   0 0 -k2];

DA=[0                                   0;
    0                                   0;
    1/(m1+m2+m3max)-1/(m1+m2)           0;
    0                                   0;
    0                                   0;
    0                                   1/(m2+m3max)-1/m2];

EB=eye(2, 2);
DB=DA;
ECz=zeros(2,2);
EDz=zeros(2, 2);
```

simulation_init_nominal.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Script:       MEC 6313 Homework 4, simulation_init_nominal.m
% Description:  Initializes the simulation of the nominal system
% Author:       Stefan Bracher
% Requirements: system_initialisation.m
%               LMI_hinf_nom.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Some clearing up %%%
clear all;
clc;
yalmip('clear');

%%% Design Parameters %%%
gamma=0.8;      % Demanded rejection rate
speed=2;        % Additional reaction time speedup

%%% System Parameters %%%
system_initialisation

%%% Calculate gains with LMIs %%%
[K, lmiok]=LMI_hinf_nom(A, B, Bw, Cz, Dz, Bz, gamma, speed);
K               % Display K
if lmiok
disp('The controller part of the system is stable');
else
disp('No information about the stability of the system can be given');
end
```

simulation_init_uncertain.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Script:       MEC 6313 Homework 4, simulation_init_uncertain.m
% Description:  Initializes the simulation of the uncertain system
% Author:       Stefan Bracher
% Requirements: system_initialisation.m
%               system_uncertainties.m
%               LMI_hinf_nom.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Some clearing up %%%
clear all;
clc;
yalmip('clear');

%%% Design Parameters %%%
gamma=0.8;      % Demanded rejection rate
speed=2;        % Additional reaction time speedup
m3max=0.5;      % Maximal Load Mass
m3=0.5;           % Actual load max for this run

%%% System Parameters and uncertainties %%%
system_initialisation       % Defitition of system matrixes

system_uncertainties        % Definition of system uncertainties

%%% Calculate gains with LMIs %%%
[K, lmiok]=LMI_hinf_uncert(A, DA, EA, B, DB, EB, Bw, Cz, ECz, Dz, EDz, Bz, gamma, speed);
K             % Display K
if lmiok
disp('The controller part of the system is stable');
else
disp('No information about the stability of the system can be given');
end

%%% Add load to the system %%%
A=[ 0   1    0            0     0    0;
    0   0    1            0     0    0;
    0   0   -k1/(m1+m2+m3)   0     0    0;
    0   0     0           0     1    0;
    0   0     0           0     0    1;
    0   0     0           0     0    -k2/(m2+m3)];

B=[0            0;
   0            0;
   1/(m1+m2+m3)    0;
   0            0;
   0            0;
   0            1/(m2+m3)];
```