FRTN10 Exercise 12. Synthesis by Convex Optimization

12.1 We want to design a controller C for the stable SISO process P_0 as shown in Figure 12.1 using the Youla parametrization and convex optimization. To do this, the control loop must first be transformed into the standard form of Figure 12.2, where z are the signals that we want to control, y are the signals available to the controller, w are the exogenous inputs and u is the control signal.

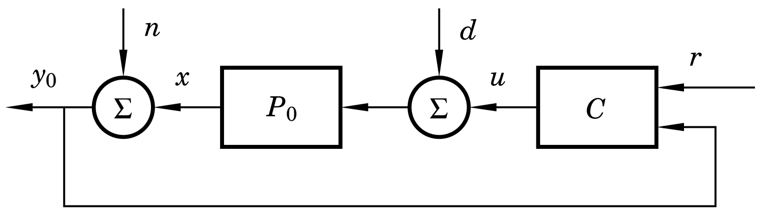

Figure 12.1 The control loop in problem 12.1

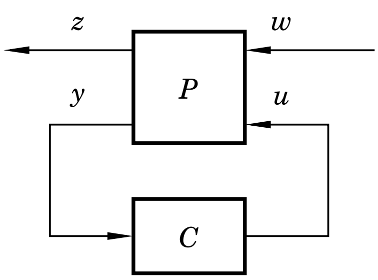

Figure 12.2 Desired form of the control loop in problem 12.1

The signals z and w are given by

$$z = \begin{pmatrix} e \\ u \end{pmatrix}, \qquad w = \begin{pmatrix} d \\ n \\ r \end{pmatrix},$$

where the control error is $e = r - x$. The controller C, which is a 1×2 transfer function, is the same in both figures, as is the control signal u.

a. What is the controller input y of Figure 12.2 according to Figure 12.1?
What is the size of the transfer matrix P?

b. Find the transfer matrix of the generalized plant P so that Figure 12.2 and Figure 12.1 describe the same control problem.

c. The Youla parametrization results in the closed-loop system

$$z = Hw$$

where the transfer function H is given by

$$H = P_{zw} + P_{zu} Q P_{yw}.$$

The control objectives are

(a) To make the gain $||H_{ij}|| \leq 10$ for all elements H_{ij}.

(b) During an impulse disturbance experiment in d, the control signal should satisfy $|u(t)| \leq 1$.

(c) During an impulse disturbance experiment in d, from two seconds onward the control error should be small: $|e(t)| \leq 0.75, t \geq 2$, if the impulse occurs at $t = 0$.

Two transfer functions Q_1 and Q_2 have been found that satisfy objective a). Figure 12.3 shows impulse responses from d to e and u when using the corresponding controllers C_1 and C_2. Find a Q that satisfies all three objectives a), b), c).

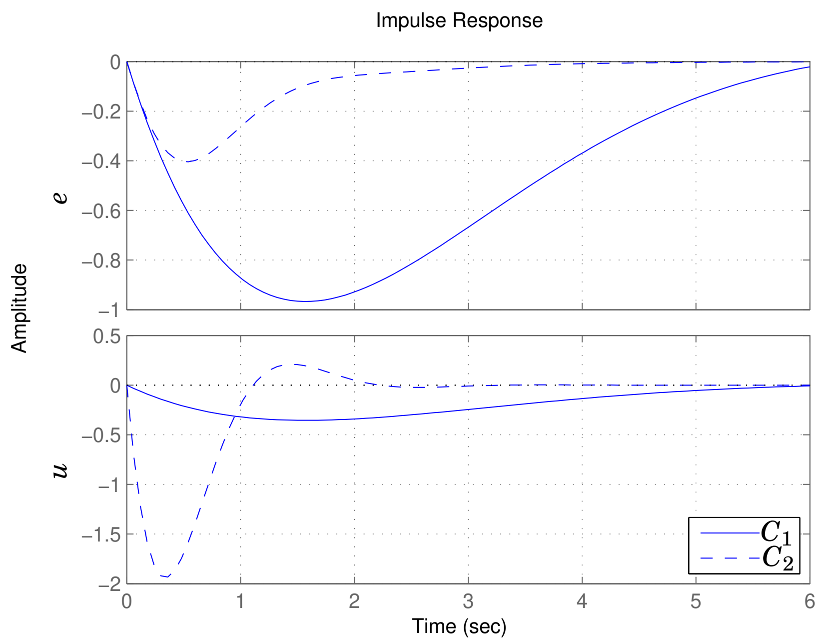

Figure 12.3 Impulse responses from disturbance d to control error e (top) and control signal u (bottom) for the controllers C_1 and C_2 in problem 12.1

12.2 *In Problem 11.2 we considered a mass-spring system. If you haven't completed that problem or don't remember it, you should go through it before solving this problem.*

In this problem we are going to find an (almost) optimal controller with respect to the cost function and constraints given in Problem 11.2. We are going to do this by reformulating the problem via the Youla parametrization, and then choose a finite basis for the Q parameter and a finite number of points where the time domain and frequency domain constraints should be enforced. The problem is then easy to solve using a convex optimization tool such as `sedumi` or `sdpt3`. Since it is time-consuming to interface directly with the solver, we will use cvx.

cvx is a Matlab-based modeling system for convex optimization. Before translating the problem to a suitable input to the solver, cvx checks if the problem satisfies the Disciplined Convex Programming (DCP) rule-set; if the DCP rules are satisfied the problem is guaranteed to be convex, and if they are not satisfied the problem is not processed. `cvx` is available on the lab computers, but it can also be downloaded at `http://cvxr.com/cvx/download/`, together with the solvers `sdpt3` and `sedumi`.

The files for the exercise are found on the course web page. The main file is `spring_mass_problem.m`, and the files `qcvx_*` help you to set up the Youla parametrization and recover the controller. In `spring_mass_problem.m`, you will be able to modify the objective function and the constraints.

We will formulate and solve the problem in discrete time, as it gives a straightforward parametrization of Q in the time-domain. If you are not used to working with discrete-time systems—don't worry! Just consider the discrete-time system an approximation of the continuous-time system.

To get a finite dimensional problem, the time-domain and frequency domain constraints can only be enforced in a finite number of points. For example, the constraints on the time domain signals are only enforced for the first 170 samples, which, given the sampling time of 0.2 seconds, amounts to the first 34 seconds. It is important that this interval covers the time-scale of the system dynamics.

The mass-spring system has two poles in $s = 0$ and is thus unstable. In order to get the Q optimization working we need to stabilize the plant with a nominal controller. The final controller will then be an augmented version of this stabilizing nominal controller. The function `qcvx_q_parametrization.m` designs this nominal controller and returns the transfer function matrices T_1, T_2, and T_3, for the Q-parametrization

$$T_1 + T_2 Q T_3,$$

as well as state space data required to recover the optimal controller from Q.

a. To get started you need to run `setup_cvx` and then `cvx_setup`. Then run `spring_mass_problem.m` and study the output. A solution satisfying the constraints of the problem has been computed. Look at the code in the script and try to understand it.

The constraints on the unit step response from r to d_1 are visualized in Figure 12.4. Which constraints are active? A constraint is active if the solution touches the constraint at any point.

b. Plot the Bode diagram of the controller. Can you recognize any resemblance with any other type of controller? Can you give some intuitive explanation to any of the dips in the magnitude plot?

c. The output from the solver looks something like this:

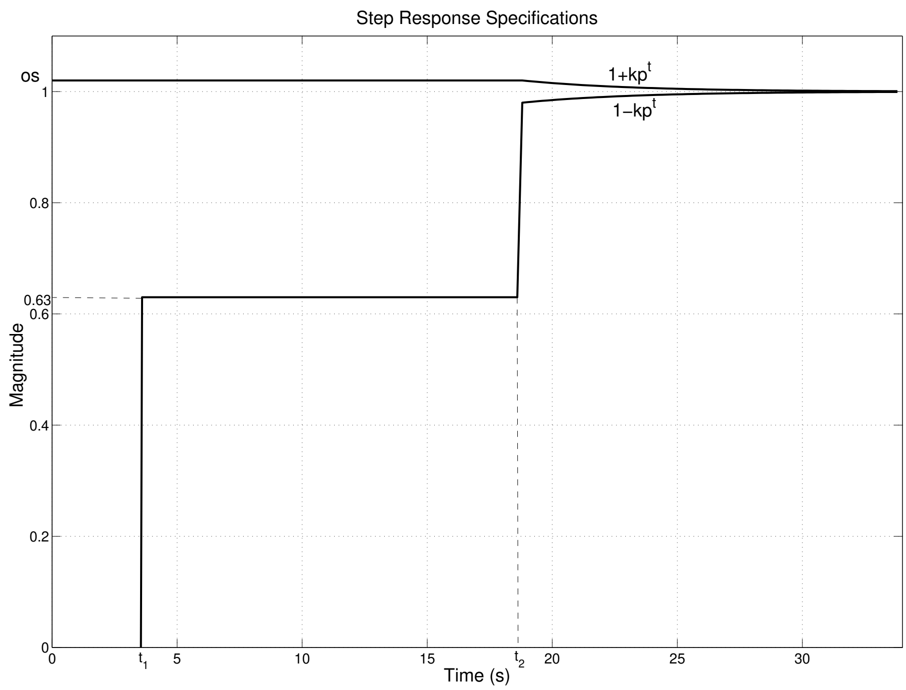

Figure 12.4 Constraints on the closed-loop step response from the reference, r, to the position of the first mass, d_1. os is the maximum allowed overshoot.

```
Calling SeDuMi 1.34: 1823 variables, 222 equality constraints
For improved efficiency, SeDuMi is solving the dual problem.
------------------------------------------------------------
SeDuMi 1.34 (beta) by AdvOL, 2005-2008 and Jos F. Sturm, 1998-2003.
Alg = 2: xz-corrector, Adaptive Step-Differentiation, theta = 0.250,
beta = 0.500, eqs m = 222, order n = 1285, dim = 1824, blocks = 202
nnz(A) = 24135 + 0, nnz(ADA) = 4682, nnz(L) = 2452
it :     b*y       gap    delta  rate   t/tP*  t/tD*   feas cg cg  prec
0 :            1.10E+03 0.000
1 :  -2.81E+00 3.99E+02 0.000 0.3628 0.9000 0.9000   3.89  1  1  1.3E+00
2 :  -8.46E+00 2.37E+02 0.000 0.5948 0.9000 0.9000   1.93  1  1  7.9E-01
...
29 :  -1.42E+02 1.41E-12 0.000 0.0450 0.9900 0.9900   1.00  5  5  1.9E-08
30 :  -1.42E+02 8.99E-14 0.000 0.0637 0.9900 0.9900   1.00  8  7  1.2E-09
...
------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +140.387
```

Let's not go into detail on what all columns stand for, but rather look only at the column feas. The number you end up with should ideally be 1, or at least close, in order for you to have a feasible solution (one that satisfies all constraints). If you end up with a value of −1, then you are certain that there is no solution. The higher you choose the order of your Q filters, the more likely it is that a solution is found. If you do not find a feasible solution, even though the order of the Q filters is large, this tells you that you need to relax the constraints in order to find a controller.

Take your code and decrease `NQ=n_q1=n_q2` until your problem is no longer feasible. What is the least order `NQ` you need in order to to obtain a feasible solution?

d. Increase the order of Q_1 and Q_2 simultaneously (`NQ=n_q1=n_q2`) from the value you obtained in the previous subproblem and plot `NQ` against the cost function value. Explain the shape of this plot and comment on how the max value of the control signal changes when `NQ` increases. *Hint: Comment out the line where* `NQ` *is set, and run the script* `spring_mass_problem.m` *from a new script where you change* `NQ` *and plot the value of* `optimal_cost`*.*

e. Change the weights in the objective function and see how this alters the solution. Also play around with the constraints to see if you can achieve an extremely fast step response. Explain your results.

f. Take a look in the other m-files and see if you can find the resulting closed-loop transfer function matrix. Plot all 9 closed-loop Bode magnitude diagrams. Point out at least one of these that you would have wished had a different appearance. How would you have preferred it to look and why?

g. Plot the effect of a step disturbance on the signal p_1. As you saw previously in part b, the low-frequency gain was very low, i.e., the controller did not have integral action. Is this consistent with the disturbance response? Add a constraint on the step response from d to p_1 and see if you can improve the disturbance rejection. Look at the gain curve of the new controller.

h. **(*)** Read more about convex optimization in

`http://www.stanford.edu/~boyd/cvxbook/`,

and more about finding the limits of performance of linear control systems using optimization in

`http://web.stanford.edu/~boyd/lcdbook/`.

Solutions to Exercise 12. Synthesis by Convex Optimization

12.1 a. The inputs to the controller are r and y_0, i.e.

$$y = \begin{pmatrix} r \\ y_0 \end{pmatrix}.$$

The input to P is $\begin{pmatrix} w \\ u \end{pmatrix}$, which contains 4 signals, and the output is $\begin{pmatrix} z \\ y \end{pmatrix}$, which contains 4 signals as well. Thus P must be 4×4.

b. We know that

$$\begin{pmatrix} z \\ y \end{pmatrix} = \begin{pmatrix} e \\ u \\ r \\ y_0 \end{pmatrix}, \qquad \begin{pmatrix} w \\ u \end{pmatrix} = \begin{pmatrix} d \\ n \\ r \\ u \end{pmatrix}.$$

Setting $C = 0$, the block diagram gives that

$$\begin{aligned} e &= r - x = r - P_0(d+u), \\ u &= u, \\ r &= r, \\ y_0 &= n + P_0(d+u). \end{aligned}$$

Arranging this into matrix form gives the generalized plant model

$$P = \begin{pmatrix} -P_0 & 0 & 1 & -P_0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ P_0 & 1 & 0 & P_0 \end{pmatrix}.$$

c. The control objective (a) is convex in H, and H is a linear function of Q, so the control objective a) is convex in Q. Since it is satisfied for Q_1 and Q_2, it is thus satisfied for any convex combination

$$Q = wQ_1 + (1-w)Q_2, \qquad w \in [0, 1].$$

We see from the impulse responses that neither Q_1 nor Q_2 satisfies (b) or (c). However, a convex combination of Q_1 and Q_2 will give the same convex combination of the disturbance responses. Taking e.g. $w = 0.7$,

- The control signal satisfies $|u(t)| \leq 0.7 \cdot 0.4 + 0.3 \cdot 2 = 0.88$, since $|u(t)| \leq 0.4$ with C_1 and $|u(t)| \leq 2$ with C_2.
- When $t \geq 2$, the control error satisfies $|e(t)| \leq 0.7 \cdot 1 + 0.3 \cdot 0.1 = 0.73$, since $|e(t)| \leq 1$ with C_1 and $|e(t)| \leq 0.1$ with C_2.

Thus we can use $Q = 0.7Q_1 + 0.3Q_2$.

12.2 a. See the plots in Figures 12.1–12.2 and the associated captions. We see that the only inactive constraint is the one on the control signal. Also see Figure 12.3 for a Nyquist plot with a circle for the M_s constraint.

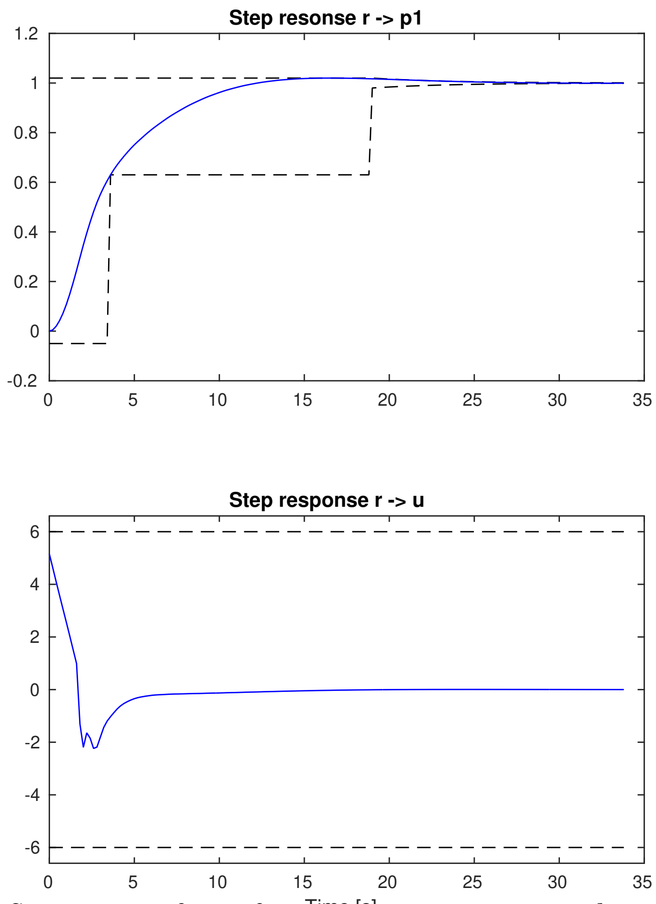

Figure 12.1 Step responses from reference r to mass position p_1 and to control signal u.

b. `bodemag(K2d)`

The gain curve is shown in Figure 12.4. The shape of the magnitude plot is very similar to that of a PD controller. Since the D-part of a PID controller acts to damp out oscillations, it makes sense that we have this kind of similarity.

The system has its resonance frequency at 5.8 rad/s, which is almost exactly at the same frequency as the deepest dip in the controller magnitude plot. The reason for the dip is that we do not want to amplify signals at this frequency. A PD controller does not have this kind of flexibility in its structure to damp out a certain frequency and is therefore not so well suited for highly oscillatory systems like this one.

c. The least value on `NQ`, for which our problem is feasible, is 7.

d. See Figure 12.5 for a plot of the cost function value versus the order of the Q filters. When `NQ` reaches around 20, the control will gain very little from increased complexity of the Q filter. We can then say that we have a good estimate of the *limit of performance*, i.e. lowest cost that linear controller can achieve given the problem setup.

The maximum value of the control signal will decrease as the order of the Q

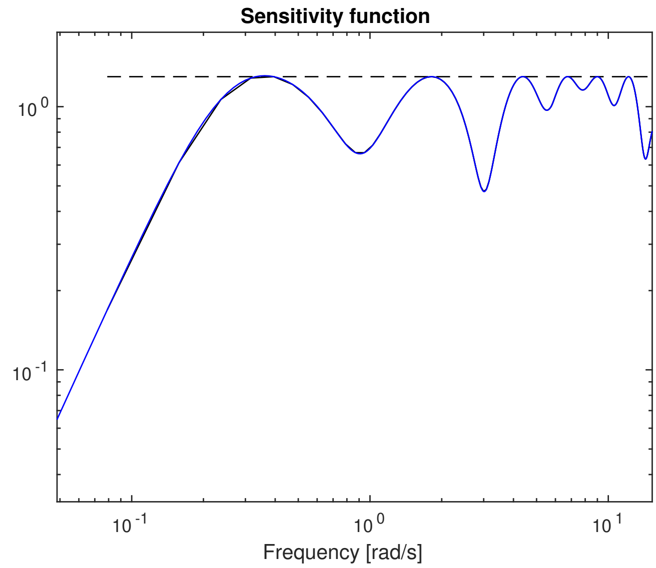

Figure 12.2 The sensitivity function of the system.

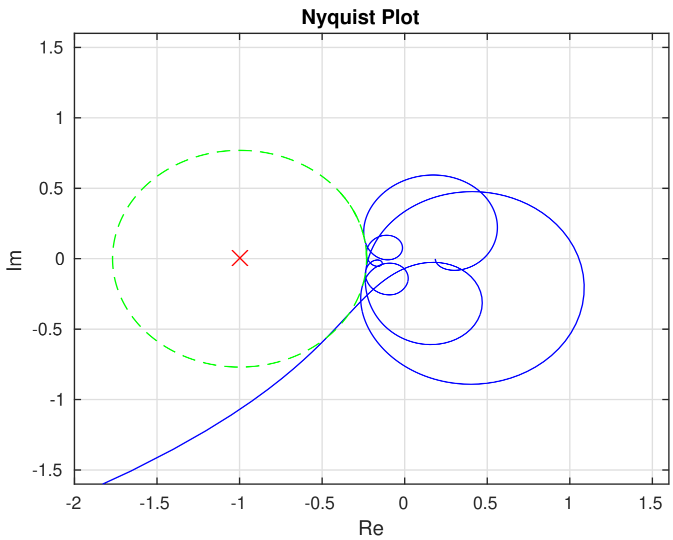

Figure 12.3 The open-loop Nyquist curve and the M_s circle when the optimal controller is used.

filters goes up. For `NQ`$= [7, 10, 15, 20]$, we get $u_{max} = [6.0, 4.9, 3.8, 3.7]$. This means that the more complex the controller becomes, the more freedom it will have to choose its control signal. As it is good to have a control signal that is low on energy (due to the cost function), it is also likely that it goes down in magnitude if it has the possibility.

e. If we start out by setting $\rho = 0$ (i.e. `weight_u = 0.0`), then we do not punish the control signal energy at all, which means that we may get very aggressive and poorly damped control. The constraint on u_{max} will to some extent prevent this, but if u_{max} is made arbitrarily large then we can get a step responses like the one in Figure 12.6. If instead $\gamma = 0$ while ρ remains 1, then the solution will remain fairly unchanged. The reason is that both the constraints on rise time and the cost of having a large e will force u to be quite active

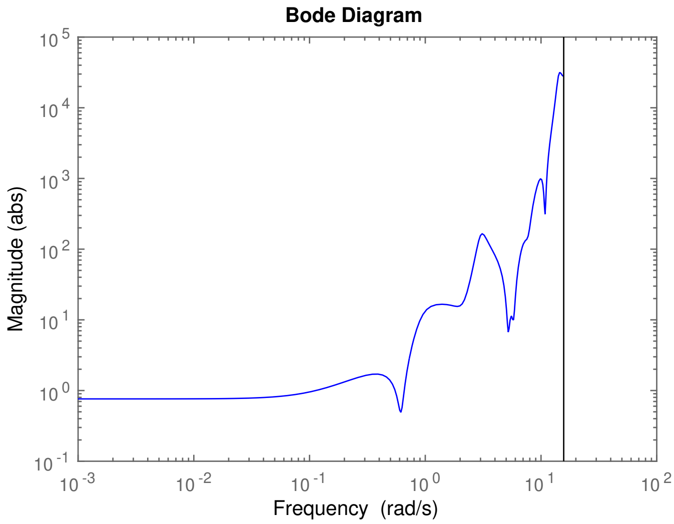

Figure 12.4 Bode diagram of the controller.

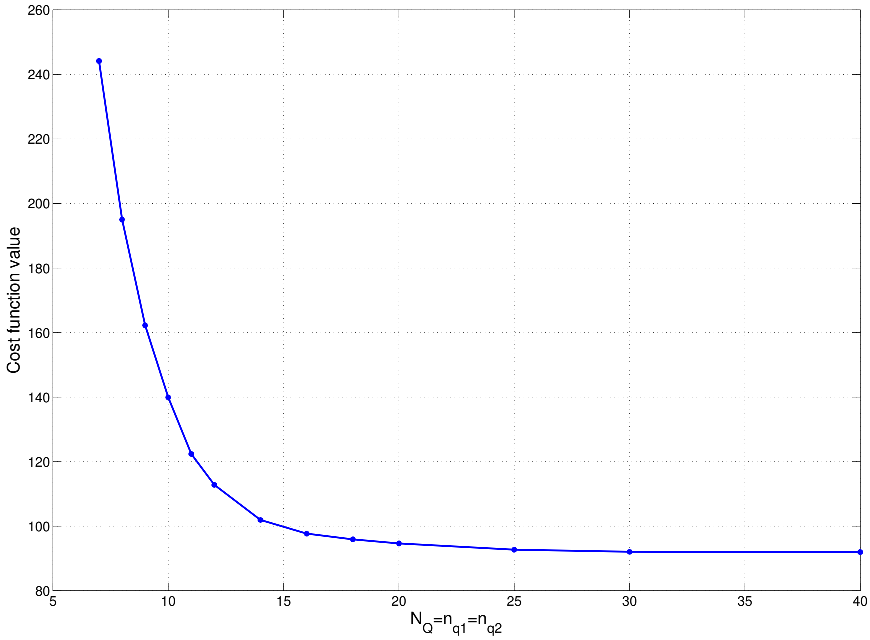

Figure 12.5 Cost function value plotted against the complexity of the Q filters.

still. If the step response constraints are made inactive, the solution will be fairly close to the one of the nominal LQG controller.

f. Looking in the files we find that `Gcl` corresponds to the closed loop transfer matrix. We can plot the magnitude curves with the command

```
bodemag(Gcl).
```

We see that the transfer function in the middle, $H_{u_o n}$, does not have high

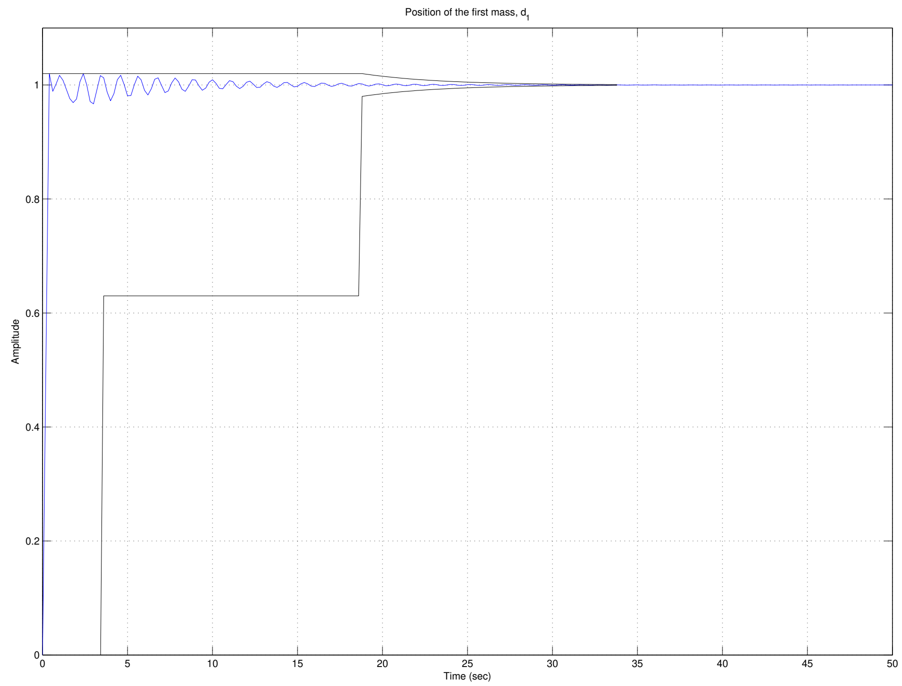

Figure 12.6 p_1 due to a reference step when $\rho = 0$. The control is very aggressive.

frequency roll-off. High frequency measurement noise, n, may therefore lead to a very noise control signal. This shows that it is very important to take all signals in a system into consideration and that a solution, even though “optimal”, might not be good. Remember, “you get what you ask for”. If we were to modify the problem, a good idea would be to put constraints on this closed-loop transfer function as well.

g. First define the constraint for the step response $d \to p_1$,

```
ub_cl_diststep = min(1, exp(-0.2*(t-5)));
```

and then add it to the optimization problem with

```
-ub_cl_diststep <= cl_stepresp(:,1,3) <= ub_cl_diststep;
```

If we try to solve the problem, we will see that it is has become infeasible. By increasing the order of the Q-filters by putting `NQ=30`, you will be able to find a solution. See figures 12.7 and 12.8.

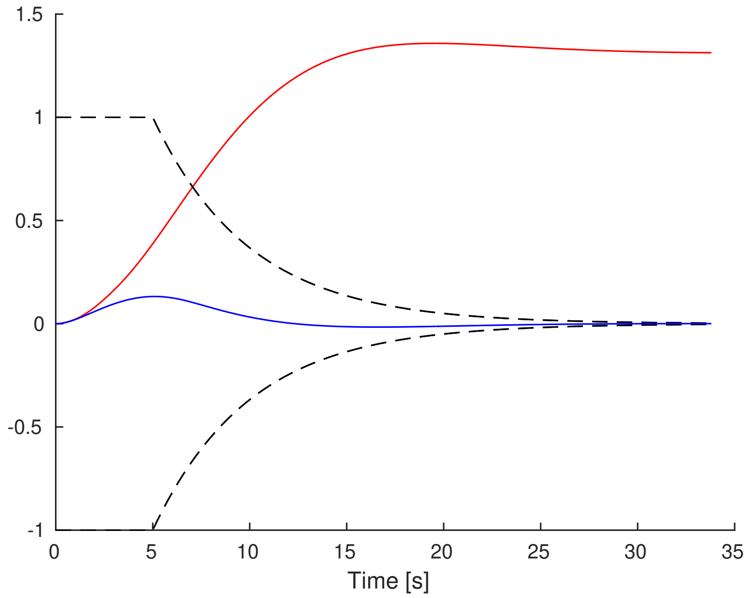

Figure 12.7 Disturbance step when the controller has been designed with respect to the disturbance rejection constraint (blue) and the original controller (red).

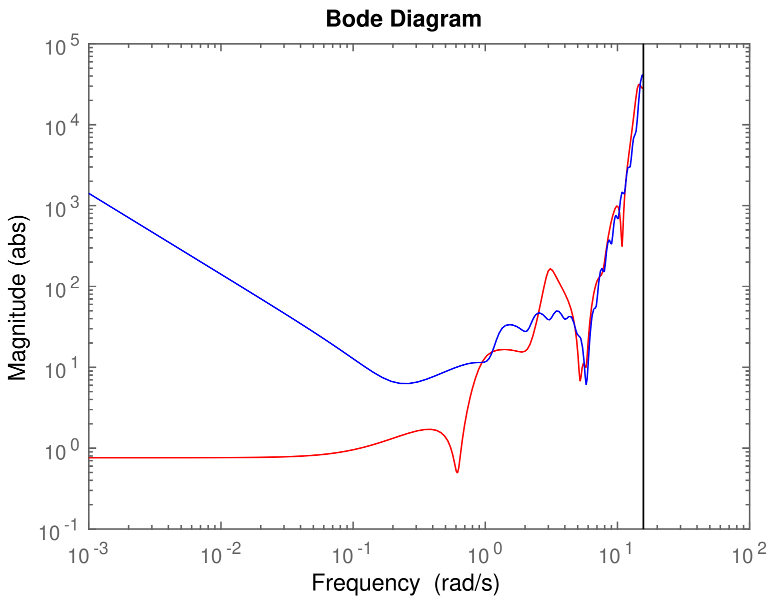

Figure 12.8 Controller gain when the controller has been designed with respect to a disturbance rejection constraint (blue) and original controller (red). Note that the low frequency gain is much higher for the controller that was designed for step disturbance rejection, this can be seen as added integral action.