BALL AND BEAM 2: Robust Control

Mark Readman, Visiting Scientist, control-systems-principles.co.uk

ABSTRACT, This is one of a series of white papers on systems modelling, analysis and control, prepared by Control-Systems-Principles.co.uk to give insights into important principles and processes in control. In control systems there are a number of generic systems and methods which are encountered in all areas of industry and technology. These white papers aim to explain these important systems and methods in straightforward terms. The white papers describe what makes a particular type of system/method important, show how it works and then demonstrates how to control it. The control demonstrations are performed using models of real systems designed by Peter Wellstead, and which have been developed for manufacture by TQ Education and Training Ltd in their CE Range of equipment. This white paper is about the application of robust control to the Ball and Beam System.

1. What is this White Paper About?

Only a brief example of controller design was given in the ball and beam white paper published previously. To compensate for this omission, this further white paper has been written to give some better notes on control of the ball and beam. This white paper concerns one of the most popular robust controller design methods and describes its use for the classic ball and beam experiment. The controller design method used is the McFarlane-Glover technique for robust loop shaping control. This controller maximizes the closed-loop tolerance to coprime factor uncertainty. It has been argued that uncertainty modeling via coprime factorization is a practical approach to uncertainty modeling. Since the ball and beam contains a number of uncertainties, good results can be expected from the McFarlane-Glover method applied to this system.

2. Robust Loop Shaping Control System

For a single input-single output (SISO) control system, a loop shaping controller can be implemented as shown in Figure 1 below. The design of the robust controller $K_R(s)$ is based on the shaped plant enclosed in the red dotted line. Essentially all the design effort goes into choosing the pre-compensator $W(s)$ to achieve the desired performance specifications. The design of the robust controller $K_R(s)$ is based on the weighted plant PW. The controller is then implemented as shown enclosed by the black dotted line.
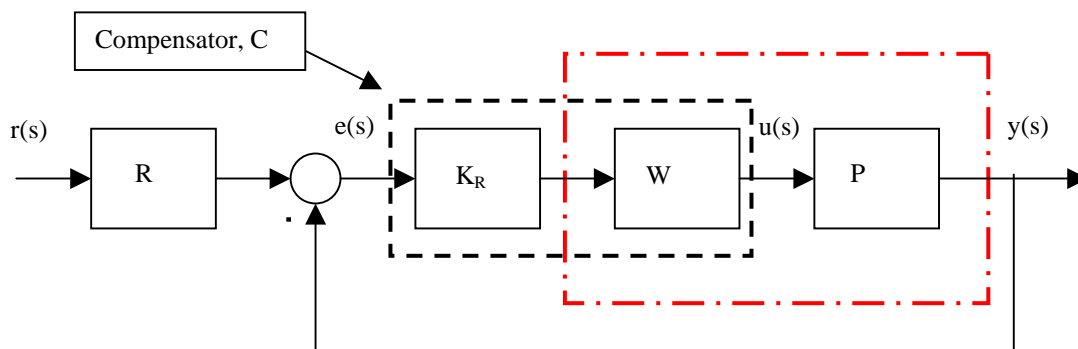
Figure 1. Loop Shaping Control System.

In the figure,

r(s) = reference signal
e(s) = r(s) - y(s) error signal
u(s) = controller output and plant input = beam angle
y(s) = plant output = ball position

P(s) is the plant e.g. the ball and beam

W(s) is a pre-compensator

$K_R(s)$ is the robust controller, and $K(s) = W(s) K_R(s)$ is the complete controller

R(s) is a pre-filter for the reference signal.

3. Ball and Beam Model

As explained in the ball and beam white paper, the ideal ball and beam experiment is modeled as the classic double integrator,

$$P(s) = \frac{V_x(s)}{V_\theta(s)} = \frac{K_{bb}}{s^2} \qquad K_{bb} \approx 0.63$$

The variables $V_x(s)$ and $V_g(s)$ are respectively the voltage from the ball position sensor and the voltage supplied to the beam angle position servo. The value of K_{bb} here is the measured value obtained from the experiment. Note the value of K_{bb} differs slightly from that used in the Ball and Beam white paper, because a different piece of ball and beam equipment was used. This is to be expected, because due to differences in component tolerances no two pieces of hardware are identical.

4. Explaining the Examples

Controller performance will be illustrated using examples of robust loop shaping of the ball and beam control system. The applications use different weighting functions, but otherwise follow the same basic method. The controllers are all implemented using the CE2000 software and its built-in real time interface is used to communicate with a CE106 Ball and Beam System. The results of the experiments were imported into MATLAB for plotting, and the controllers were designed using the MATLAB code given at the end of this white paper. Hopefully, this shows how the CE2000 Control Software can be used for real time implementation and MATLAB for design and presentation.

5. Example 1. Constant Weighting function W(s)=5.

The simplest pre-compensator is just a constant gain. With W(s) = 5 the crossover frequency is approximately 2rad/sec. Notice that the roll-off rate at the crossover frequency is –40db/decade which is higher than we would like. Specifically, we expect the robust loop shape to have a similar crossover frequency, but with a lower roll-off rate at the crossover frequency. Applying the McFarlane-Glover design method gives an optimal gamma = 2.6131. This indicates a good match between target and achieved Bode magnitude, and that the closed-loop system will tolerate approximately 33% coprime factor uncertainty.

The robust stabilizing controller (obtained using the MATLAB code given at the end of this white paper) is,

$$K_R = 30.322 \frac{(s + 0.7307)}{(s + 9.28)(s + 6.362)}$$

with DC gain of 0.3753. Note that the controller is stable, minimum phase and strictly proper. The controller is implemented in discrete time with a sampling time Ts=0.05 sec. The digital controller is obtained by adding zero order hold (ZOH) and discretising the above continuous time controller and weighting function. The discrete time controller in this case is,

$$K_R(z) = \frac{1.048z^{-1} - 1.009z^{-2}}{1 - 1.356z^{-1} + 0.4574z^{-2}}$$

and W(z)=5.

The step response of the robust ball and beam controller is shown in Figure 2. The overshoot is approximately 20% and the steady state error |e(t)|<0.1v. This is equivalent to a maximum error of 0.45cm on the beam.
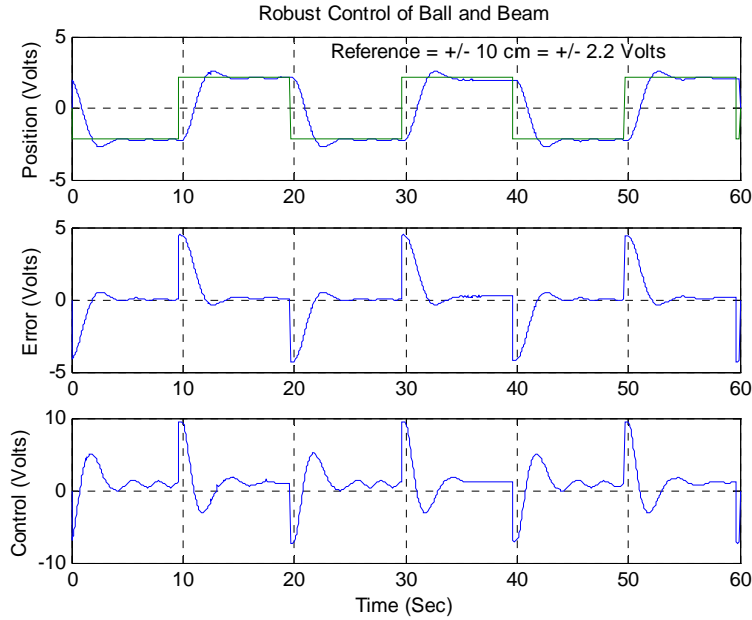
Figure 2. Results for Robust Loop Shaping Control with Constant Gain Pre-Compensator.

The disturbance response of the robust controller is shown in Figure 3 below. The reference is zero and the disturbance is generated by manually pushing the ball away from the steady state position.
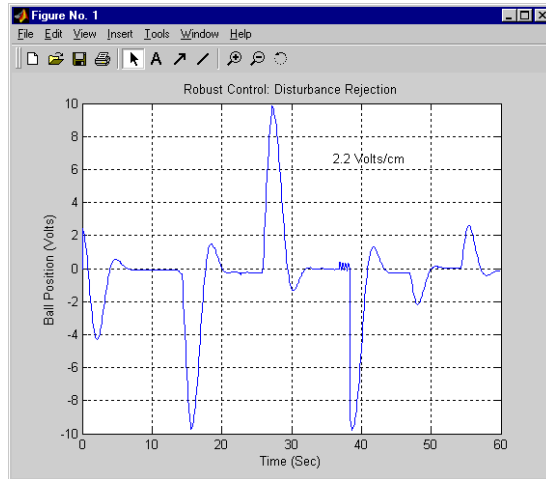
Figure 3. Disturbance Rejection Results for Robust Loop Shaping Control with Constant Gain Pre-Compensator

Example 1, Ramp tracking.

The double integrator is a type two system so the closed-loop system can track a ramp input with zero steady state error. The ability of the ball to track a ramp input is shown in Figure 4. In practice, the rolling friction of the ball, other nonlinearities and sensor noise will affect the ability of the ball to track a ramp. However in Figure 4 the experimental results show that reasonable ramp tracking can be achieved.
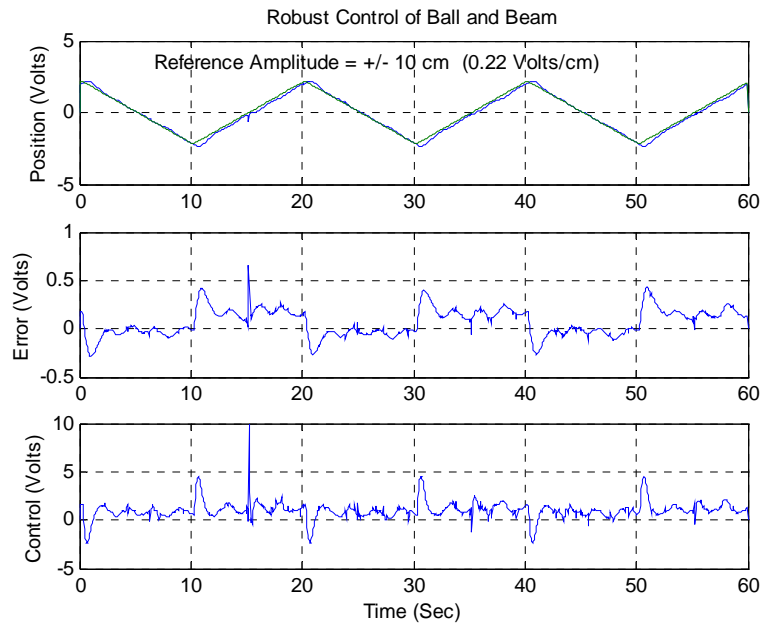
Figure 4. Ramp Tracking Results for Robust Loop Shaping Control with Constant Gain Pre-Compensator

Example 1, Experimental Results With Two Balls on the Beam.

This is a fun experiment which has an interesting and unexpected result. Two ball were placed on the beam, one at either end. The controller was then started. After a somewhat noisy start the two balls synchronize, move toward each other and then move together on the beam. The step response is slightly more damped than with one ball alone because of the increased friction of the balls rolling together – however it is not easy to see why the balls come together and roll together. The results of this experiment are shown in Figure 5 below.
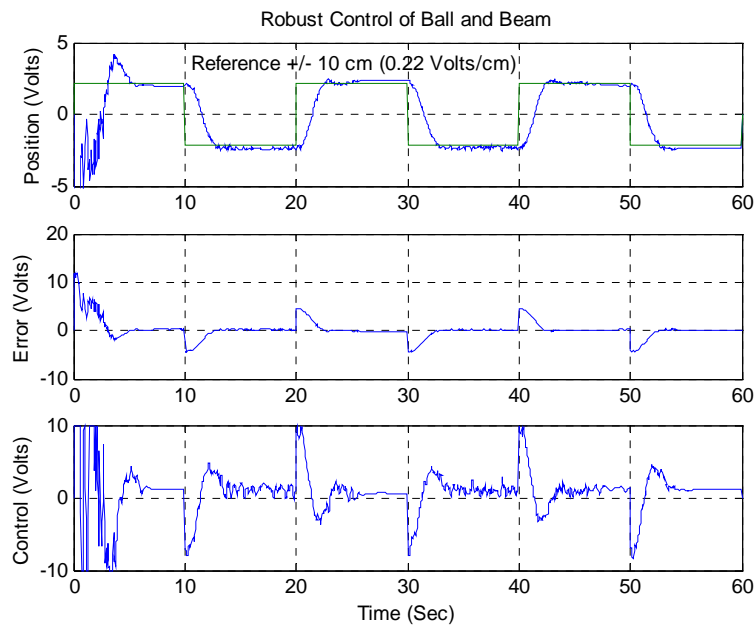
Figure 5. Results for Two Balls on the Beam and Robust Loop Shaping Control with Constant Gain Pre-Compensator

6. Example 2, Lead Weighting function W(s)=5*(s+1)/(0.3*s+1).

Like the constant gain, this weighting function was set to give the crossover frequency at about 2rad/sec. Notice now, however, that the roll-off rate at the crossover frequency is ~20db/decade. We expect the robust loop shape to have a similar crossover frequency. Applying the McFarlane-Glover design method gives an optimal gamma of 2.110. Note that this is a slightly smaller optimal gamma that in the previous example. This is because the lead compensator has reduced the rolloff around the cutoff frequency so the robust controller design does not have to work as hard to meet the target loop shape. The closed-loop system will now tolerate approximately 42% coprime factor uncertainty. Using the MATLAB code at the end of this white paper, the robust stabilizing controller is,

$$K_R = 31.1919 \frac{(s+3.523)(s+0.6338)}{(s+0.9933)(s^2+22.82s+146.1)}$$

with DC gain of 0.4802. The order of the controller has increased because the weighting function is 1st order, also the controller is stable, minimum phase, and strictly proper.

The controller is implemented in discrete time with a sampling time Ts=0.05 seconds by discretising the above continuous time controller and weighting function.The controller and weighting function can be implemented in the form,

$$K_R(z) = \frac{0.9601z^{-1} - 1.733z^{-2} + 0.7779z^{-3}}{1 - 2.06z^{-1} + 1.374z^{-2} - 0.304z^{-3}}$$

$$W(z) = 16.66 \frac{1 - 0.9539z^{-1}}{1 - 0.8465z^{-1}}$$

The experimental results are shown below in Figure 6. Notice that the actuator signal saturates causing a damping effect on the ball position.
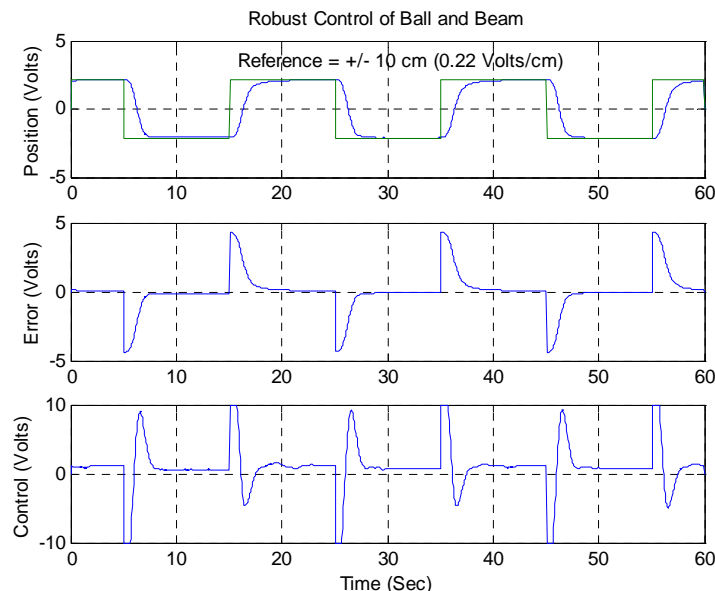

Figure 6. Results for Robust Loop Shaping Control with Lead Pre-Compensator

7. Sensitivity Functions

In the MATLAB code given later, the sensitivity functions for the closed loop systems are computed. These transfer functions determine the system closed loop response. With reference to Figure 7 they are,

1. The input disturbance sensitivity (Sio), which describes how the system output, y, responds to disturbances, d_i, added to the control signal, u.
2. The control sensitivity (Suo), which describes how the controller output, u, responds to disturbances, d, or measurement noise, n, added to the system output, y.
3. The complementary sensitivity function (To), which describes how the system output y responds to measurement noise, n, added to the system output. (It is also the closed loop transfer function for a unity precompensator R).
4. The nominal sensitivity (So).
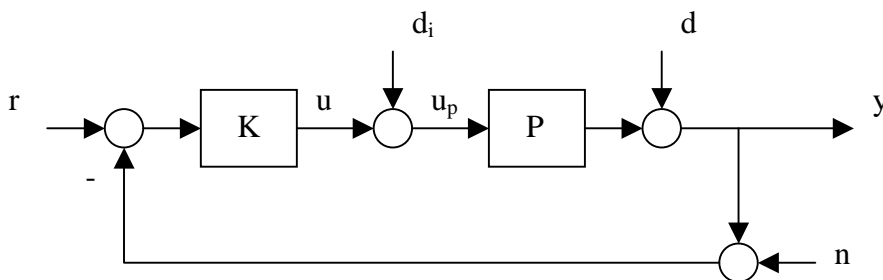
Figure 7. Feedback Diagram

The sensitivities are used to assess closed loop system performance as described in the book by Graham Goodwin and colleagues (see 'A Final Word' below). For the ball and beam the control sensitivity and complementary sensitivity are particularly interesting because of the noise associated with the meausurement of ball position. We would like the control signal and the ball position to be insensitive to measurement noise as well as giving good servo tracking response. It has been said that the McFarlane-Glover method implicitly minimise these sensitivity functions, so good results can be expected – better than with simple phase advance , proportional plus derivative or state feedback control. All the sensitivity functions are related so that we are always making a compromise in controller design – in the ball and beam the compromise is between good tracking response and good measurement noise rejection.

8. CE2000 Implementation

The robust control examples were implemented using the CE2000 software in which a built-in real time interface connects the software to the Ball and Beam System. Figure 8 shows the CE2000 implementation for a constant gain weighting function.

In the figure, the object Kr(z) is the compensator and the block marked Process is the connection to the built-in CE2000 external interface. Notice that a constant is subtracted from the measured ball position in this implementation. This is to compensate for the DC component or offset in the sensor noise. This will be slightly different for each Ball and Beam System. Note this assumes that we are modeling the sensor noise as, $y_m(t)=y(t)+y_n(t)+y_{dc}(t)$. Where y(t) is the true measurement and $y_n(t)$ and $y_{dc}(t)$ are the zero mean and DC component the sensor noise respectively
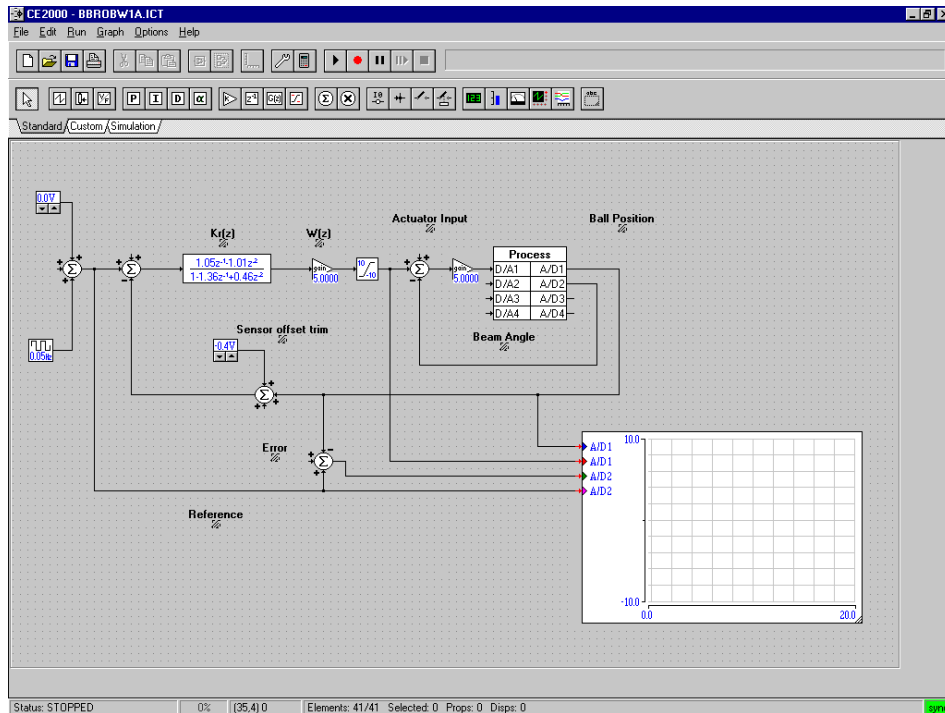
Figure 8. CE2000 Implementation of Robust Loop Shaping Control with Constant Pre-Compensator

9. The MATLAB Code used in the Examples.

The references given in the final section of this white paper describe robust controller design using the McFarlane-Glover method. The MATLAB script below is the one that was used to obtain the controller settings for the CE2000 implementations given above.

```
%Robust Loop Shaping
%Ball and Beam controller
%Strictly proper loop shape D=0

s=tf('s');

%Nominal Ball and Beam Model (or put your model here)
Kbb=0.66;
plant=Kbb/s^2;
pss=ss(plant);
[A1 B1 C1 D1]=ssdata(pss);

%Two different weighting function
%W1=tf(5); %Constant gain
%W1=5*(s+1)/(0.3*s+1);   %lead comp

%loop shape used in design
loopss=pss*W1;
[A B C D]=ssdata(loopss);
%
%solve two Ricatti equations
x=care(A,B,C'*C);
z=care(A',C',B*B');
%
%compute gamma
```

```
gamo=sqrt(1+max(eig(x*z)));
%
%Use suboptimal gamma for the design 10 % larger is usual
ten=1.1;
gam=ten*gamo;
R=eye(size(z*x))-(eye(size(z*x))+z*x)/gam^2;
%
%observer
L=B'*x;
K=inv(R)*z*C';
Ao=A-K*C;
%
%controller (A,B,C,D)
Ac=A-B*L-K*C;
Cc=L;
Dc=0;
Bc=K;
%
%the  robust controller
comp=ss(Ac,Bc,Cc,Dc);
loop1=plant*W1*comp;
loop1ss=ss(loop1);
%
%the sensitivity functions
T=minreal(feedback(loop1,1)); % T is the complementary sensitivity
So=minreal(feedback(1,loop1));% So is the sensitivity
Su=minreal(comp*W1*So);        % Su is the control sensitivity
Si=minreal(plant*So);          % Si is the input disturbance sensitivity
ltiview('sigma',T,So,Su,Si)
%
%Digital controller
Ts=0.05;
W1d=c2d(W1,Ts);
[num den]=tfdata(W1d,'v');
filt(num,den,Ts);
compd=c2d(comp,Ts);
[num den]=tfdata(compd,'v');
compz=filt(num,den,Ts);
```

10. A Final Word

It is hoped that you have learnt some ideas about robust control for the ball and beam system from this white paper. Unfortunately, it is not possible to answer general questions from students and engineers about the contents of this white paper, <u>unless</u> control-systems-principles has an arrangement with your organisation. The MATLAB code is supplied for illustration only, and is not intended for professional use. For more information about the CE106 Ball and Beam System and the CE2000 control software go to the TQ Education and Training Ltd web site using the links on our web site www.control-systems-principles.co.uk or use the email info@tq.com. To learn more about robust control, there are several very good references to the McFarlane Glover technique. These include,

1). McFarlane D.C. and Glover K., *Robust Controller Design Using Normalized Coprime Factorization Factor Plant Descriptions*. Springer 1989. This is by the developers of Robust Loop Shaping. It is clearly written with interesting examples to demonstrate the method.

2) Paattilammi J., and Makila P. M., *Fragility and Robustness, A case study on paper machine head box control*. Control Systems Magazine. Feb 2000. This paper discusses implementation issues for the McFarlane Glover method. In particular the importance of controller fragility.

3) Goodwin G. C., Graebe S. F., Salgado M. E., *Control System Design*. Prentice Hall, 2001. A good text on control theory with many good application notes. See this reference especially for sensitivity function explanations.

4). Glad, T., and Ljung, L., *Control Theory Multivariable and Nonlinear Methods*, Taylor and Francis, 2000. A comprehensive reference to control theory with a chapter on robust loop shaping.

Finally, do not forget the world wide web – with key phrases, Robust Loop Shaping etc...