Controller Design

Tim Hesketh, Zhiyu Xi

April, 2010

Abstract

A number of controller design issues are covered in this course. This lecture brings them together integratively, and provides optional extensions.

Controller design is based on MPC. For an infinite control horizon, this converges to linear quadratic control. LQR is provided as an extension. Integral action yields robustness, introduced by differencing and embedding an integrator between controller and process. These notes show how an integrator may also be introduced explicitly. Explicit integrator introduction provides an opportunity to embed a process model within the controller (with robustness advantages). Set point shaping is achieved with a set-point filter, which further extends robustness and provides an opportunity for high-gain control.

Relevant Documents

The following documents cover relevant controller design methods:

design.pdf Explains the MPC design process step-by-step.

mpclec.pdf Outlines MPC design with NMSS.

lqr.pdf Gives a dynamic programming derivation of LQR design.

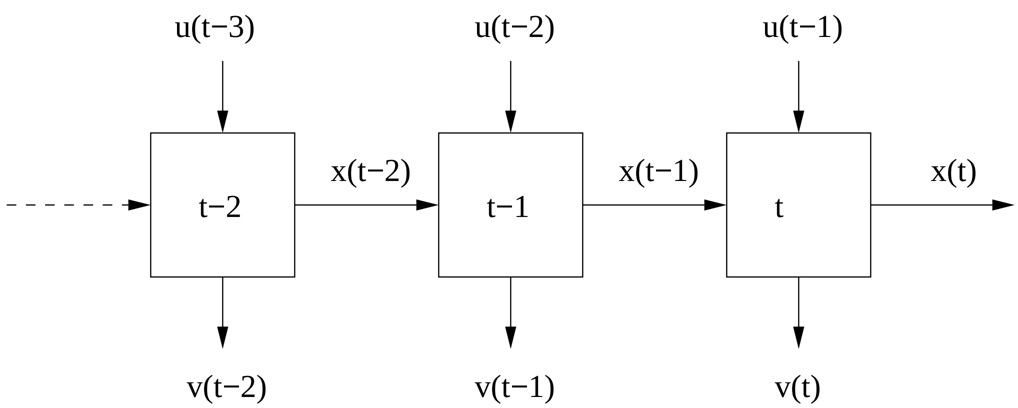
altnmss.pdf Illustrates design with explicit integrator introduction and process model embedding.

Linear Quadratic Regulator

A Linear Quadratic Controller minimises a quadratic cost function:

$$C = \sum_{t=1}^{N} X_t^\top P X_t + U_{t-1}^\top Q U_{t-1} = \sum_{t=1}^{N} v_t$$

This can be represented as a multi-stage optimisation problem, and solved by dynamic programming.



The state updating (constraint) is given by

$$x_{t+1} = Ax_t + Bu_t$$

1 Stage Optimisation

$$
\begin{aligned}
r_1 &= x_t^\top P x_t + u_{t-1}^\top Q u_{t-1} \\
&= (Ax_{t-1} + Bu_{t-1})^\top P (Ax_{t-1} + Bu_{t-1}) + u_{t-1}^\top Q u_{t-1}
\end{aligned}
$$

Optimising with respect to u_{t-1}

$$
\begin{aligned}
\frac{dr_1}{du_{t-1}} &= 2B^\top P_1 (Ax_{t-1} + Bu_{t-1}) + 2Qu_{t-1} = 0 \\
u_{t-1} &= -(Q + B^\top P_1 B)^{-1} B^\top P_1 A x_{t-1} = -K_{t-1} x_{t-1} \\
r_1^* &= (Ax_{t-1} - BK_{t-1}x_{t-1})^\top P_1 (Ax_{t-1} - BK_{t-1}x_{t-1}) \\
&\quad + (K_{t-1}x_{t-1})^\top Q (K_{t-1}x_{t-1})
\end{aligned}
$$

r_1^* is the optimised cost.

2 Stage Optimisation

The two-stage cost is the optimal one-stage cost plus the added cost of control for the preceding stage, i.e.

$$\begin{aligned} r_2 &= r_1^* + x_{t-1}^\top P x_{t-1} + u_{t-2}^\top Q u_{t-2} \\ &= x_{t-1}^\top [P + (A - BK_{t-1})^\top P_1 (A - BK_{t-1}) \\ &\quad + K_{t-1}^\top Q K_{t-1}] x_{t-1} + u_{t-2}^\top Q u_{t-2} \end{aligned}$$

This has exactly the same form as the cost for the one stage optimisation if we make the following replacement:

$$P_2 = P + (A - BK_{t-1})^\top P_1 (A - BK_{t-1}) + K_{t-1}^\top Q K_{t-1}$$

In this case

$$
\begin{aligned}
u_{t-2} &= -(Q + B^\top P_2 B)^{-1} B^\top P_2 A x_{t-2} \\
&= -K_{t-2} x_{t-2}
\end{aligned}
$$

General Solution

Iterating over more stages will obviously produce similar results, and we can see that the general solution can be given as:

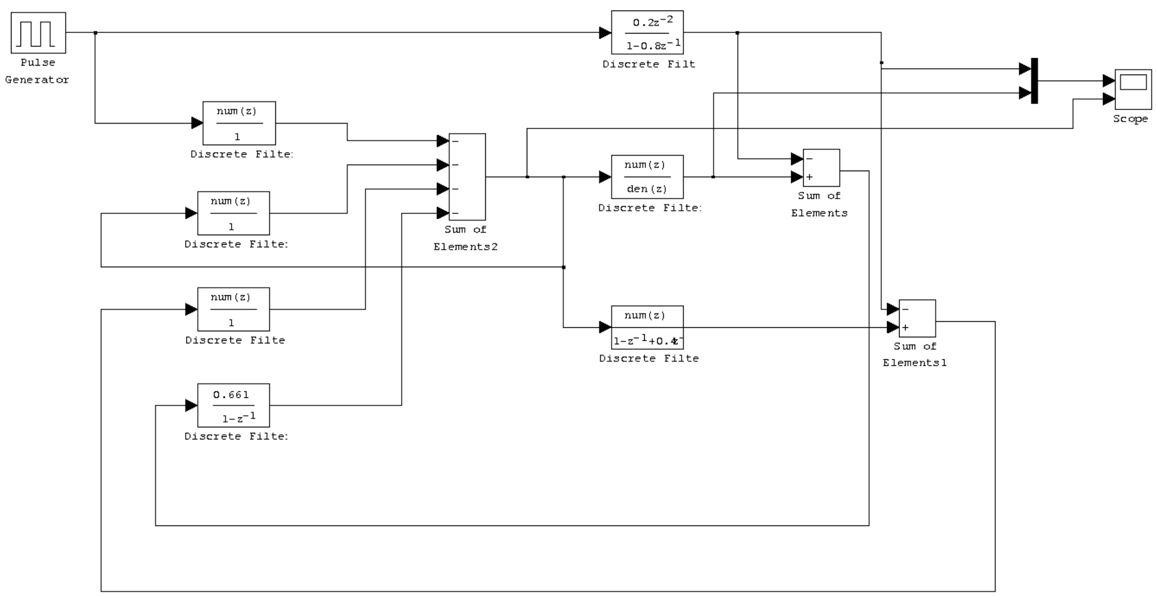
$$\begin{aligned} P_i &= P + (A - BK_{i-1})^\top P_{i-1}(A - BK_{i-1}) + K_{i-1}^\top Q K_{i-1} \\ K_i &= (Q + B^\top P_i B)^{-1} B^\top P_i A \\ u_i &= -K_i x_i \end{aligned}$$

Combining the first two equations yields what is known as the *Riccati Equation* which needs to be iterated to steady state to derive LQ control.

The file **altnmss.pdf** contains a function (dlqro.sci) which performs an efficient square root algorithm to solve the Riccati Equation.

Embedded Model Design, Explicit Integrator

The closed loop system structure is shown by means of a Simulink model.



Features of this Structure

- The setpoint is shaped by an M-box.
- Structure includes a real process (in this case with unmodelled dynamics) and a nominal process on which the controller design is based.
- The controller comprises:
  - Feedforward from the unconditioned setpoint signal.
  - Autoregressive feedback of the control signal.
  - Feedback of the error between the nominal model and the conditioned setpoint.
  - Feedback through an integrator of the error between the actual process output and the conditioned setpoint signal.

Actual and Nominal Processes

The actual process is

$$y(t) = \frac{0.2q^{-1} + 0.3q^{-2} + 0.1q^{-3}}{1 - 1.1q^{-1} + 0.5q^{-2} - 0.04q^{-3}} u(t)$$

The nominal process model is

$$\hat{y}(t) = \frac{0.2q^{-1} + 0.3q^{-2}}{1 - q^{-1} + 0.4q^{-2}} u(t)$$

The actual process thus contains unmodelled dynamics. The controller is designed for the nominal model. If $s(t)$ is the unconditioned setpoint signal, and the conditioning transfer function is $M = \frac{0.2q^{-2}}{1-0.8q^{-1}}$:

NMSS Model

$$
\begin{aligned}
e(t) &= \frac{0.2q^{-1} + 0.3q^{-2}}{1 - q^{-1} + 0.4q^{-2}} u(t) - \frac{0.2q^{-2}}{1 - 0.8q^{-1}} s(t) \\
(1 - 1.8q^{-1} &+ 1.2q^{-2} - 0.32q^{-3}) e(t) = \\
& (0.2q^{-1} + 0.14q^{-2} - 0.24q^{-3}) u(t) + \ldots \\
& \ldots + (0.2q^{-2} - 0.2q^{-3} + 0.08q^{-4}) s(t)
\end{aligned}
$$

Defining $esum(t) = \sum_{k=0}^{t} e(k)$ (i.e. integral of error):

$$
esum(t) = esum(t-1) + e(t)
$$

From the above difference equations, it is possible to write the non-minimal state-space model for the system (see file **altnmss.pdf**).

The Controller

The cost function to be minimised is

$$C = \sum_{t=0}^{\infty} (x(t)^\top C^\top C x(t) + u(t)^\top R u(t))$$

$$u(t) \quad = \quad -Kx(t)$$

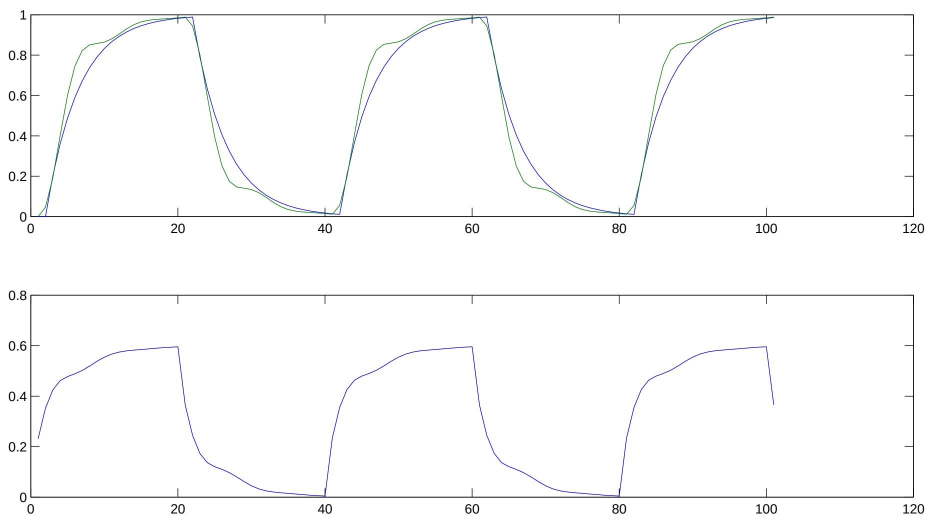
For the purposes of inclusion in the block diagram model, this controller may be expressed as:

$$u(t) \quad = \quad -K_1 \hat{e}(t) - K_2 u(t) - K_3 s(t) - \frac{K_4}{1 - q^{-1}} e(t)$$

Software for performing all the design steps is given in **altnmss.pdf**.

Closed Loop Performance

The closed loop performance was obtained from Simulink.



Summary

- The closed loop performance shows good model following and attainment of the correct steady-state setpoint.
- The controller is a compromise between error minimisation and actuation amplitude. It would be possible to weight the error more heavily and obtain closer setpoint following (see section below on *High Gain*).
- The effect of the unmodelled dynamics on the closed loop behaviour is apparent, but the controller compensates satisfactorily.
- In the above simulation, only the integral of error and control actuation signals play any part in the cost function. By choice of the vector C, other signals may be included in the cost function. Controllers designed in this way may display greater robustness (to disturbances or model uncertainty).

Choice of M for Robustness

- It is important not to ask the controller to try to make the process do something it cannot.
- Shape the setpoint so that it is something the process can follow.
- Where possible get the controller to do as little as possible.

High Gain Control

- If the process can easily follow the shaped setpoint, then control signals will not be excessive.
- It may be good policy to incorporate a delay into the shaped setpoint to permit feedforward action to anticipate required changes (see paper **design.pdf**).
- Regulation of the process output to follow the shaped setpoint is important.
- Tight regulation can be achieved by forcing high gain feedback control.
- Response to setpoint change need not require large actuation fluctuations.
- Other factors may require limiting the high gain (for example in the above example, unmodelled dynamics could produce large actuation changes if the gain is too high). Other factors include noise and model uncertainty.