A Comparison of State Space LQG, Wiener IMC and Polynomial LQG Discrete Time Feedback Control for Active Vibration Control Purposes

J. Morkholt, S.J. Elliott and T.C. Sors

ISVR Technical Memorandum 823

December 1997

UNIVERSITY OF SOUTHAMPTON

INSTITUTE OF SOUND AND VIBRATION RESEARCH

SIGNAL PROCESSING & CONTROL GROUP

A Comparison of State Space LQG, Wiener IMC and Polynomial LQG Discrete Time Feedback Control for Active Vibration Control Purposes

by

J.Mørkholt, S.J.Elliott and T.C.Sors

ISVR Technical Memorandum No. 823

December 1997

Authorised for issue by
Prof S J Elliott
Group Chairman

Contents

Appendices

1. Introduction

The objective of the work presented in this report has been to compare, through computer simulations, three methods of designing discrete time feedback controllers for active vibration control purposes: State space Linear Quadratic Gaussian (LQG) control theory, Internal Model Control (IMC) using an FIR Wiener approach and Internal Model Control using an LQG polynomial approach.

The three design methods are similar in that they are all based on the minimisation of a quadratic cost function under certain assumptions about the disturbance noise and sensor noise in the system to be controlled. They are also based on different models of the plant under control and the disturbance to be suppressed by the controllers.

The state space LQG method assumes a discrete time state space model of the plant, the filter generating the disturbance noise and the filter generating the sensor noise. The resulting controller consists of an optimal state estimator (in essence a model of the plant under control) combined with an optimal state feedback controller. The LQG controller is in state space form and therefore effectively an IIR-type controller. The controller design involves solving the discrete time algebraic Riccati equation.

The IMC formulation transforms the feedback problem into a feedforward one and expresses the optimal controller as a combination of a model of the plant under control and an optimal filter. The optimal filter can be found using a conventional FIR Wiener filter approach. The Wiener filter is determined from the impulse or frequency response of the plant and the autocorrelation function or power spectral density of the disturbance and sensor noise. The resulting IMC controller consists of a model of the plant (FIR or IIR) combined with the FIR Wiener filter. Alternatively an IIR solution to the optimal filter can be found if the plant, disturbance noise filter and sensor noise filter are initially represented as polynomial transfer functions.

Controllers based on the state space LQG method and the FIR Wiener IMC method have been designed from a model of a lightly damped, rectangular plate fitted with a piezoceramic patch control actuator and a point velocity sensor and excited by a point force actuator driven by white noise acting as the primary source. The design objective has been to suppress the effect of the primary disturbance on the output by minimising the mean square value of the output. The resulting controllers have then been evaluated in computer simulations, by letting them control the plant from which they were designed. Both minimum phase and non-minimum phase plant models have been used, with the latter being potentially more difficult to control than the former.

A basic comparison of the IMC and LQG controllers resulting from the two different methods has been carried out by designing a controller using each method. The controllers were designed to minimise the same quadratic cost function under the same assumptions about disturbance and sensor noise, and for the same plant and disturbance filter. Apart from this a number of different design aspects have been addressed such as the effect of varying the number of coefficients in the FIR Wiener filter, using different types of state estimators in the LQG design and the effect of delays in the control path.

The issue of stability robustness has also been addressed in the simulations. Both design methods incorporate two basic design parameters that can be adjusted to weight performance against robust stability: The amount of control effort used and the assumed ratio between the level of sensor noise in the feedback loop and the level of the disturbance to be suppressed. The effect of varying these parameters in the two design methods has been examined. The possibility of reducing the order of the LQG controllers has also been investigated. Although the LQG controller can be implemented with an IIR filter using potentially far less filter coefficients than the direct implementation of the corresponding FIR Wiener IMC controller, it may be possible to reduce the order of the LQG controllers even further. An alternative way of finding reduced order IIR controllers by fitting low order IIR filters to the frequency response of the desired controller has also been examined. Finally the use of the polynomial LQG/IMC approach in calculating controllers for the resonant system has been examined and it has been found that whereas the resulting controllers are identical to those obtained using the other two methods, numerical computation problems may arise rendering the method impractical.

Chapter 2 describes the IMC and LQG design methods. In chapter 3 the plant and disturbance model used in the simulations is described and finally chapter 4 presents the results of the simulations.

2. Feedback control

2.1 Disturbance suppression using feedback control

In the following section q^{-1} will be used to denote the time shift operator defined by $q^{-n}x(k) = x(k-n)$, where k is the discrete time. When interchanged with the z-transform variable in transfer functions the q operator allows algorithmic manipulation of transfer functions with time sequence signals, under certain conditions (see e.g. [1]).

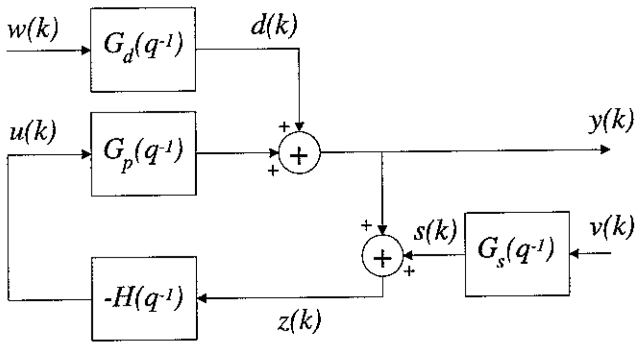

Figure 2.1. Disturbance suppression using feedback control.

A standard feedback control arrangement for disturbance suppression is shown in Figure 2.1. The physical output of the system under control is denoted $y(k)$. A certain amount of sensor noise $s(k)$ is assumed to be present, so that the measured system output is $z(k) = y(k) + s(k)$. A feedback controller $H(q^{-1})$ takes the measured system output as input, and determines a control signal $u(k)$ to the system, in order to suppress the effect of the primary disturbance $d(k)$ at the physical system output $y(k)$. The signal $z(k)$ measured on the system output can be written as

$$z(k) = y(k) + s(k) = G_p(q^{-1})u(k) + G_d(q^{-1})w(k) + G_s(q^{-1})v(k) \ , \tag{2.1}$$

i.e. as the sum of the effect on the output of the primary disturbance acting on the system and the effect on the output of the applied control actuation, plus measurement or sensor noise. The contribution from the control actuation is the control signal filtered by the plant transfer function $G_p(q^{-1})$. The disturbance contribution is assumed to be a white noise signal $w(k)$ filtered by the disturbance transfer function $G_d(q^{-1})$. The sensor noise contribution is assumed to be a white noise signal $v(k)$, filtered by the sensor noise transfer function $G_s(q^{-1})$.

The ability of the feedback controller to suppress the effect of the primary disturbance on the physical system output (the *performance* of the controller) can be evaluated from the transfer function between the disturbance and the system output. This transfer function is termed the *sensitivity function* and is given by

$$S\left(e^{-j\omega T}\right)=\frac{1}{1+G_p\left(e^{-j\omega T}\right)H\left(e^{-j\omega T}\right)} \quad . \tag{2.2}$$

A small value of S at a given frequency corresponds to a large attenuation of the disturbance at the physical system output. This requires a large open loop gain $G_p\left(e^{-j\omega T}\right)H\left(e^{-j\omega T}\right)$.

The ability of the controller to maintain stability if the plant changes is an important aspect in feedback control. One way of describing plant uncertainty is via unstructured multiplicative uncertainty, where the plant is written as [2]

$$G_p\left(e^{-j\omega T}\right)=\left[1+\Delta G_p\left(e^{-j\omega T}\right)\right]G_{p0}\left(e^{-j\omega T}\right) . \tag{2.3}$$

Here $G_{p0}\left(e^{-j\omega T}\right)$ is the nominal plant frequency response and $\Delta G_p\left(e^{-j\omega T}\right)$ is the fractional uncertainty. It can be shown [3] that if the magnitude of the fractional uncertainty is bounded by a real but potentially frequency dependent quantity $B(\omega)$ as

$$\left|\Delta G_p\left(e^{-j\omega T}\right)\right| < B(\omega) , \tag{2.4}$$

then *robust stability* of the controller (that is the control system stays stable if the plant response varies within the fractional uncertainty limit) is guaranteed if

$$\left|T_0\left(e^{-j\omega T}\right)\right| < \frac{1}{B(\omega)} , \tag{2.5}$$

where $T_0\left(e^{-j\omega T}\right)$ is the *complementary sensitivity function*

$$T_0\left(e^{-j\omega T}\right)=1-S_0\left(e^{-j\omega T}\right)=1-\frac{1}{1+G_{p0}\left(e^{-j\omega T}\right)H\left(e^{-j\omega T}\right)}=\frac{G_{p0}\left(e^{-j\omega T}\right)H\left(e^{-j\omega T}\right)}{1+G_{p0}\left(e^{-j\omega T}\right)H\left(e^{-j\omega T}\right)} , \tag{2.6}$$

corresponding to the nominal plant. Thus the robust stability of a given control configuration to unstructured multiplicative plant uncertainties can be evaluated by inspecting the corresponding complementary sensitivity function. A very high value of the complementary sensitivity function at a given frequency thus corresponds to a very low stability margin at that frequency.

2.2 Internal Model Control (IMC) using a FIR Wiener filter approach

By assuming a certain structure of the controller $H(q^{-1})$ the feedback control problem of Figure 2.1 can be transformed into a feedforward one [2]. This is shown in Figure 2.2.

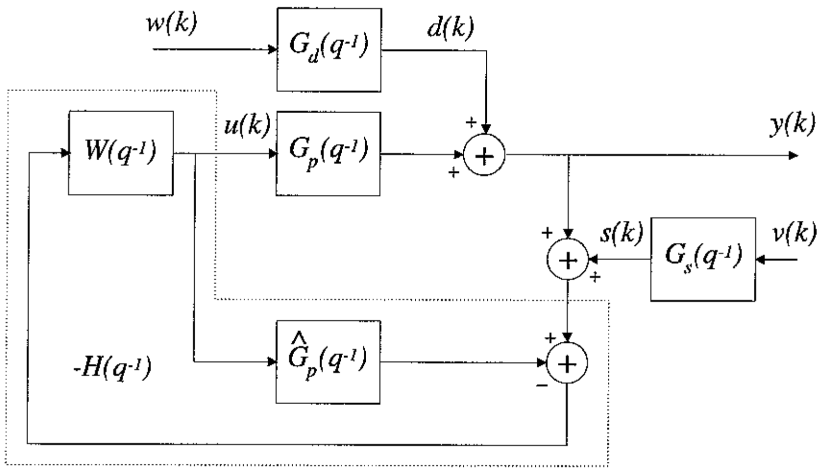

Figure 2.2. Internal Model Control.

This control structure is termed Internal Model Control [3]. The controller (contained inside the dashed box) consists of a model $\hat{G}_p(q^{-1})$ of the plant and a control filter $W(q^{-1})$. Passing the control signal through the plant model provides an estimate of the control contribution to the system output, and by subtracting this estimate from the output, an estimate of the disturbance is found. Assuming that the plant model is perfect, i.e. that $\hat{G}_p(q^{-1}) = G_p(q^{-1})$, the control contribution will be completely cancelled, and the block diagram of Figure 2.3 results

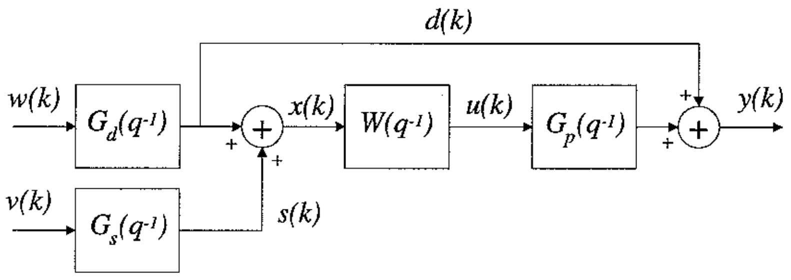

Figure 2.3. Equivalent feedforward system.

The problem is now a standard feedforward problem, and the filter $W(q^{-1})$ can be designed to minimise a steady state quadratic cost function of the form

$$J = E\{y^2(k)\} + \rho E\{u^2(k)\}, \tag{2.7}$$

using a Wiener approach. Here $E\{\bullet\}$ denotes expectation. The term $\rho E\{u^2(k)\}$ puts a penalty on very large control signals, and the effort weighting term ρ thus determines the amount of control effort used in the minimisation of the cost function. Assuming that the plant and control filter are both linear and time invariant, the block diagram can be rearranged as shown in Figure 2.4, and the filtered reference $r(k)$ can be introduced [4].

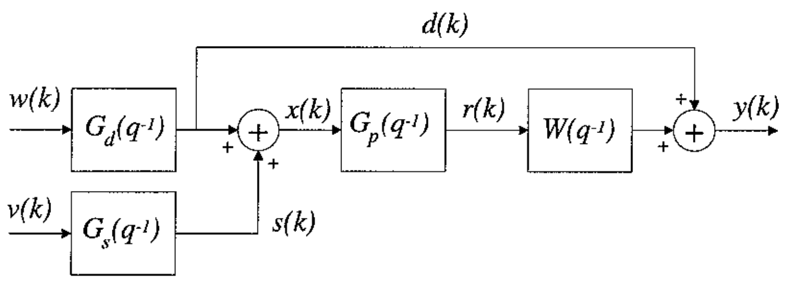

Figure 2.4. Rearranged feedforward system introducing the filtered reference.

Assuming that the control filter is implemented as an FIR filter, the system output can be written as [4]

$$y(k) = d(k) + \mathbf{r}^T(k)\mathbf{w} \tag{2.8}$$

where the filtered reference vector $\mathbf{r}(k)$ and the control filter impulse response vector $\mathbf{w}$ are defined as

$$\mathbf{r}(k) = [r(k) \quad r(k-1) \quad r(k-2) \quad \cdots \quad r(k-I)]^T \tag{2.9}$$

and

$$\mathbf{w} = [w_0 \quad w_1 \quad w_2 \quad \cdots \quad w_I]^T \quad \text{where} \quad W(q^{-1}) = w_0 + w_1 q^{-1} + w_2 q^{-2} + \ldots + w_I q^{-I} \,. \tag{2.10}$$

In a similar way the control signal can be written as

$$u(k) = \mathbf{x}^T(k)\mathbf{w}\,, \tag{2.11}$$

where the signal vector $\mathbf{x}(k)$ is given by

$$\mathbf{x}(k) = [x(k) \quad x(k-1) \quad x(k-2) \quad \cdots \quad x(k-I)]^T, \tag{2.12}$$

with the signal $x(k)$ defined as

$$x(k) = d(k) + s(k)\,. \tag{2.13}$$

Now the cost function can be written as

$$\begin{aligned} J &= E\{y^2(k)\} + \rho E(u^2(k)) = E\{[d(k) + \mathbf{r}^T(k)\mathbf{w}]^T[d(k) + \mathbf{r}^T(k)\mathbf{w}]\} + \rho E\{[\mathbf{x}^T(k)\mathbf{w}]^T[\mathbf{x}^T(k)\mathbf{w}]\} \\ &= E\{d^2(k) + 2\mathbf{w}^T\mathbf{r}(k)d(k) + \mathbf{w}^T\mathbf{r}(k)\mathbf{r}^T(k)\mathbf{w} + \rho\,\mathbf{w}^T\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{w}\} = \mathbf{w}^T\mathbf{R}\,\mathbf{w} + 2\mathbf{w}^T\mathbf{p} + c \end{aligned} \qquad (2.14)$$

where the matrix **R**, the vector **p** and the scalar c are given as

$$\mathbf{R} = E\{\mathbf{r}(k)\mathbf{r}^T(k) + \rho\,\mathbf{x}(k)\mathbf{x}^T(k)\} = \mathbf{R}_r + \rho\mathbf{R}_x \quad , \quad \mathbf{p} = E\{\mathbf{r}(k)d(k)\} \quad , \quad c = E\{d^2(k)\} \qquad (2.15)$$

The matrix $\mathbf{R}_r$ contains the auto correlation $R_{rr}(n) = E\{r(k)r(k+n)\}$ of $r(k)$ evaluated at $n = 0,1,\ldots,I$, the matrix $\mathbf{R}_x$ contains the auto correlation $R_{xx}(n) = E\{x(k)x(k+n)\}$ of $x(k)$ evaluated at $n = 0,1,\ldots,I$ and the vector **p** the cross correlation $R_{rd}(n) = E\{r(k)d(k+n)\}$ between $r(k)$ and $d(k)$ evaluated at $n = 0,1,\ldots,I$ as

$$\mathbf{R}_r = \begin{bmatrix} R_{rr}(0) & R_{rr}(1) & \cdots & R_{rr}(I) \\ R_{rr}(1) & R_{rr}(0) & \cdots & R_{rr}(I-1) \\ \vdots & \vdots & \ddots & \vdots \\ R_{rr}(I) & R_{rr}(I-1) & \cdots & R_{rr}(0) \end{bmatrix}, \mathbf{R}_x = \begin{bmatrix} R_{xx}(0) & R_{xx}(1) & \cdots & R_{xx}(I) \\ R_{xx}(1) & R_{xx}(0) & \cdots & R_{xx}(I-1) \\ \vdots & \vdots & \ddots & \vdots \\ R_{xx}(I) & R_{xx}(I-1) & \cdots & R_{xx}(0) \end{bmatrix}, \mathbf{b} = \begin{bmatrix} R_{rd}(0) \\ R_{rd}(1) \\ \vdots \\ R_{rd}(I) \end{bmatrix} \qquad (2.16)$$

The optimal control filter impulse response $\mathbf{w}_0$ that minimises the quadratic cost function is given by [4]

$$\mathbf{w}_0 = -\mathbf{R}^{-1}\mathbf{p} = -[\mathbf{R}_r + \rho\mathbf{R}_x]^{-1}\mathbf{p} \;, \qquad (2.17)$$

and involves inversion of the joint autocorrelation matrix **R**. In the classic Wiener approach the effort weighting term is not included, so that $\mathbf{R} = \mathbf{R}_r$. To overcome potential problems of ill conditioning of the auto correlation matrix with respect to inversion in this case, the cost function is usually modified to include a regularisation term proportional to the sum of the squared values of the control filter coefficients [2]

$$J = E\{y^2(k)\} + \beta\,\mathbf{w}^{\mathrm{T}}\mathbf{w} \;, \qquad (2.18)$$

where β is a regularisation parameter. Inclusion of a regularisation term in the cost function introduces a penalty on very large control filter coefficients, and subsequently on large control signals. The effect of the regularisation term is thus similar to (but not the same as) the effect of the "effort weighting" term included in the above derivation. In practise if measurement noise is present the regularisation term may not be required. The revised cost function (2.18) results in the **R** matrix in this case being given by [2]

$$\mathbf{R} = \mathbf{R}_r + \beta \mathbf{I} \ . \tag{2.19}$$

If $x(k)$ in the above derivation is a white noise sequence of unit intensity, the auto correlation matrix $\mathbf{R}_x$ is the identity matrix, so that in this case the effect of the effort weighting term is identical to the effect of the regularisation term.

The elements of the auto correlation matrices and the cross correlation vector can be calculated via the transfer functions of the plant, disturbance and sensor noise transfer functions and from knowledge of the statistical properties of the noise sequences $w(k)$ and $v(k)$. The auto spectrum $P_{rr}\left(e^{-j\omega T}\right)$ of the reference (the letter *P* is used to denote power spectra instead of the commonly used *S* to avoid confusion with the sensitivity function) can be written as

$$P_{rr}\left(e^{-j\omega T}\right) = \left|G_p\left(e^{-j\omega T}\right)\right|^2 P_{xx}\left(e^{-j\omega T}\right) \ , \tag{2.20}$$

where $P_{xx}\left(e^{-j\omega T}\right)$ is the auto spectrum of $x(k) = d(k) + s(k)$. The disturbance noise $d(k)$ is assumed to be the output of the transfer function $G_d\left(q^{-1}\right)$ driven by white noise $w(k)$ with auto spectrum $P_{ww}\left(e^{-j\omega T}\right) = Q_w$. In a similar manner the measurement noise $s(k)$ is assumed to be the output of the transfer function $G_s\left(q^{-1}\right)$ driven by white noise $v(k)$ with auto spectrum $P_{vv}\left(e^{-j\omega T}\right) = Q_v$. The signals $d(k)$ and $s(k)$ are furthermore assumed to be uncorrelated with each other. The auto spectrum $P_{xx}\left(e^{-j\omega T}\right)$ is thus given by

$$P_{xx}\left(e^{-j\omega T}\right) = P_{dd}\left(e^{-j\omega T}\right) + P_{ss}\left(e^{-j\omega T}\right) = \left|G_d\left(e^{-j\omega T}\right)\right|^2 Q_w + \left|G_s\left(e^{-j\omega T}\right)\right|^2 Q_v \ , \tag{2.21}$$

and so the auto spectrum $P_{rr}\left(e^{-j\omega T}\right)$ of the reference is given by

$$P_{rr}\left(e^{-j\omega T}\right) = \left|G_p\left(e^{-j\omega T}\right)\right|^2 \left(\left|G_d\left(e^{-j\omega T}\right)\right|^2 Q_w + \left|G_s\left(e^{-j\omega T}\right)\right|^2 Q_v\right). \tag{2.22}$$

The cross spectrum $P_{rd}\left(e^{-j\omega T}\right)$ between reference and disturbance is found by

$$P_{rd}\left(e^{-j\omega T}\right) = G_p\left(e^{+j\omega T}\right)\left[P_{dd}\left(e^{-j\omega T}\right) + P_{sd}\left(e^{-j\omega T}\right)\right] = G_p\left(e^{+j\omega T}\right) P_{dd}\left(e^{-j\omega T}\right) \ , \tag{2.23}$$

since $s(k)$ and $d(k)$ are uncorrelated. Finally expressing $P_{dd}\left(e^{-j\omega T}\right)$ via the disturbance transfer function gives

$$P_{rd}\left(e^{-j\omega T}\right) = G_p\left(e^{+j\omega T}\right)\left|G_d\left(e^{-j\omega T}\right)\right|^2 Q_w \ . \tag{2.24}$$

The auto- and cross correlation functions $R_{rr}(k)$, $R_{xx}(k)$ and $R_{rd}(k)$ can now be found from the auto and cross spectra $P_{rr}\left(e^{-j\omega T}\right)$, $P_{xx}\left(e^{-j\omega T}\right)$ and $P_{rd}\left(e^{-j\omega T}\right)$ via the inverse Fourier transform. It is clear that the effect of the sensor noise is to add a term to the elements of the auto correlation matrix **R**, whereas the cross correlation vector **p** is unaffected by the level of sensor noise. The level of the white disturbance noise affects both **R** and **p** in a linear way.

The transfer function $H_{IMC}\left(q^{-1}\right)$ of the IMC feedback controller can be found from the transfer functions $G_p\left(q^{-1}\right)$ and $W_0\left(q^{-1}\right)$ of the plant and the optimal Wiener filter as

$$H_{IMC}\left(q^{-1}\right) = -\frac{W_0\left(q^{-1}\right)}{1+G_p\left(q^{-1}\right)W_0\left(q^{-1}\right)} \quad , \tag{2.25}$$

where the sign of the controller has been chosen in correspondence with the notation in Figure 2.2. Note that since the Wiener filter of equation (2.10) has a direct link, this controller utilises plant output samples up to and including the k'th sample to calculate the k'th control signal sample. In a practical implementation the k'th control signal sample will not be read out until one sample later. To evaluate the performance of the controller, this one sample control delay should be included in the plant model.

The sensitivity function $S_0\left(e^{-j\omega T}\right)$ of the corresponding closed loop system is the transfer function from the disturbance input $d(k)$ to the output $y(k)$ and is thus given by

$$S_0\left(e^{-j\omega T}\right) = 1 + G_p\left(e^{-j\omega T}\right)W_0\left(e^{-j\omega T}\right) \quad . \tag{2.26}$$

The complementary sensitivity is thus given by

$$T_0\left(e^{-j\omega T}\right) = 1 - S_0\left(e^{-j\omega T}\right) = -G_p\left(e^{-j\omega T}\right)W_0\left(e^{-j\omega T}\right) \quad . \tag{2.27}$$

The auto spectrum of the closed loop system output is found as

$$\begin{aligned} P_{yy}\left(e^{-j\omega T}\right) &= \left|1+G_p\left(e^{-j\omega T}\right)W_0\left(e^{-j\omega T}\right)\right|^2\left|G_d\left(e^{-j\omega T}\right)\right|^2 Q_w + \left|G_p\left(e^{-j\omega T}\right)W_0\left(e^{-j\omega T}\right)\right|^2\left|G_s\left(e^{-j\omega T}\right)\right|^2 Q_v \\ &= \left|S_{IMC}\left(e^{-j\omega T}\right)G_d\left(e^{-j\omega T}\right)\right|^2 Q_w + \left|T_{IMC}\left(e^{-j\omega T}\right)G_s\left(e^{-j\omega T}\right)\right|^2 Q_v \end{aligned} \quad , \tag{2.28}$$

which shows that low sensitivity results in good disturbance reduction whereas low complementary sensitivity results in good sensor noise reduction. As mentioned earlier low complementary sensitivity also results in robust stability against unstructured plant uncertainties. Very low sensitivity and complementary sensitivity cannot be achieved at the same time since $T_0\left(e^{-j\omega T}\right) + S_0\left(e^{-j\omega T}\right) = 1$.

Due to Parseval's theorem the cost function of (2.7) can be written in the frequency domain as

$$E\{y^2(k)+\rho u^2(k)\} = \int_{-\omega_s/2}^{\omega_s/2}\left[P_{yy}\left(e^{-j\omega T}\right)+\rho P_{uu}\left(e^{-j\omega T}\right)\right]d\omega \ , \tag{2.29}$$

where ω_s is the angular sampling frequency and where $P_{uu}\left(e^{-j\omega T}\right)$ is the auto spectrum of the control signal given by

$$P_{uu}\left(e^{-j\omega T}\right)=\left|G_s\left(e^{-j\omega T}\right)W_0\left(e^{-j\omega T}\right)\right|^2 Q_v+\left|G_d\left(e^{-j\omega T}\right)W_0\left(e^{-j\omega T}\right)\right|^2 Q_w \ . \tag{2.30}$$

If the sensor noise is assumed to be zero and a certain amount of effort weighting is used in the control design, the cost function can be written as

$$E\{y^2(k)+\rho u^2(k)\}=Q_w\int_{-\omega_s/2}^{\omega_s/2}\left|S_0\left(e^{-j\omega T}\right)G_d\left(e^{-j\omega T}\right)\right|^2+\rho\left|W_0\left(e^{-j\omega T}\right)G_d\left(e^{-j\omega T}\right)\right|^2 d\omega \ , \tag{2.31}$$

which by using the relation (2.27) can be written as

$$E\{y^2(k)+\rho u^2(k)\}=Q_w\int_{-\omega_s/2}^{\omega_s/2}\left|S_0\left(e^{-j\omega T}\right)G_d\left(e^{-j\omega T}\right)\right|^2+\rho\left|T_0\left(e^{-j\omega T}\right)G_d\left(e^{-j\omega T}\right)G_p^{-1}\left(e^{-j\omega T}\right)\right|^2 d\omega \ . \tag{2.32}$$

The first term of this cost function is related to the performance (i.e. disturbance attenuation) of the controller whereas the last term is related to the robustness of the controller, in that it contains the complementary sensitivity weighted by the ratio between the disturbance filter and plant frequency response.

If on the other hand the effort weighting is set to zero and a certain amount of sensor noise is assumed to be present, the cost function becomes

$$E\{y^2(k)\}=Q_w\int_{-\omega_s/2}^{\omega_s/2}\left|S_0\left(e^{-j\omega T}\right)G_d\left(e^{-j\omega T}\right)\right|^2+\left(Q_v/Q_w\right)\left|T_0\left(e^{-j\omega T}\right)G_s\left(e^{-j\omega T}\right)\right|^2 d\omega \ . \tag{2.33}$$

The right hand side of this cost function is similar to the cost function of (2.32) and becomes identical to it if $Q_v/Q_w=\rho$ and $G_s\left(e^{-j\omega T}\right)=G_d\left(e^{-j\omega T}\right)/G_p\left(e^{-j\omega T}\right)$. Thus assuming a certain amount of sensor noise in the controller design has an effect on controller robustness which is similar to the effect of including an effort weighting term in the cost function. The important difference between the two methods of introducing robustness into the controller design is that by using the "assumed sensor noise" method, the frequency weighting of the complementary sensitivity in the cost function can be altered easily by assuming a certain shape of the filter $G_s\left(e^{-j\omega T}\right)$, i.e. by assuming a certain frequency spectrum of the sensor noise. If for instance the sensor noise is assumed to be white the complementary sensitivity is weighted equally for all frequencies. Alternatively if $\left|G_s\left(e^{-j\omega T}\right)\right| \propto B(\omega)$, where $B(\omega)$ is the fractional uncertainty bound defined in (2.4), the complementary sensitivity is weighted where the plant uncertainty is high. When using effort weighting the same design freedom does not exist since the complementary sensitivity is weighted by a ratio pre-determined by the nominal plant and disturbance dynamics. The idea of using "assumed" or "artificial" sensor noise to improve robustness in feedback control designs is described in [2] and is furthermore similar to the "fictitious noise" approach used in state space LQG loop recovery techniques [5],[6].

2.3 Linear Quadratic Gaussian (LQG) Control using a state space model

The Linear Quadratic Gaussian (LQG) control approach is based on the minimisation of a quadratic cost function of the form (2.7) under the assumption that the disturbance and sensor noise is modelled as filtered Gaussian white noise.

The state space LQG approach is based on a discrete time state space model of plant, disturbance and sensor noise dynamics. In the general case the plant, disturbance noise filter and sensor noise filter can be written in state space form as

$$\left.\begin{aligned} \mathbf{x}_p(k+1) &= \mathbf{A}_p\mathbf{x}_p(k)+\mathbf{B}_p u(k) \\ y(k) &= \mathbf{C}_p\mathbf{x}_p(k)+d(k) \\ z(k) &= y(k)+s(k) \end{aligned}\right\} \text{Plant} \tag{2.34}$$

$$\left.\begin{aligned} \mathbf{x}_d(k+1) &= \mathbf{A}_d\mathbf{x}_d(k)+\mathbf{B}_d w(k) \\ d(k) &= \mathbf{C}_d\mathbf{x}_d(k) \end{aligned}\right\} \text{Disturbance noise filter} \tag{2.35}$$

$$\left.\begin{aligned} \mathbf{x}_s(k+1) &= \mathbf{A}_s\mathbf{x}_s(k)+\mathbf{B}_s v(k) \\ s(k) &= \mathbf{C}_s\mathbf{x}_s(k)+v(k) \end{aligned}\right\} \text{Sensor noise filter} \tag{2.36}$$

where the connection to the transfer functions $G_p\left(q^{-1}\right)$, $G_d\left(q^{-1}\right)$ and $G_s\left(q^{-1}\right)$ introduced in the previous section is given by

$$G_p\left(q^{-1}\right)=\mathbf{C}_p\left[q\mathbf{I}-\mathbf{A}_p\right]^{-1}\mathbf{B}_p \tag{2.37}$$

$$G_d\left(q^{-1}\right) = \mathbf{C}_d\left[q\mathbf{I} - \mathbf{A}_d\right]^{-1}\mathbf{B}_d \tag{2.38}$$

$$G_s\left(q^{-1}\right) = \mathbf{C}_s\left[q\mathbf{I} - \mathbf{A}_s\right]^{-1}\mathbf{B}_s + 1 \ . \tag{2.39}$$

Again $w(k)$ and $v(k)$ are assumed to be white noise processes with variances Q_w and Q_v. Note that the plant transfer function does not include a direct path. This is a reasonable assumption for the transfer function of a physical plant sampled with a zero order hold circuit. The disturbance transfer function does not include a direct path either. This is not a limitation since any delay in this transfer function does not alter the power spectrum of the output signal. Moreover the notation is in correspondence with the notation used in [7]. The sensor noise filter does include a direct path, since this is necessary in order to be able to solve the LQG problem. The three state space descriptions (2.34), (2.35) and (2.36) can be gathered into one joint state space description by introducing the joint state vector

$$\mathbf{x}(k) = \left[\mathbf{x}_p^T(k) \quad \mathbf{x}_d^T(k) \quad \mathbf{x}_s^T(k)\right]^T . \tag{2.40}$$

Now the joint state equation can be written as

$$\begin{bmatrix} \mathbf{x}_p(k+1) \\ \mathbf{x}_d(k+1) \\ \mathbf{x}_s(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_p & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_d & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_s \end{bmatrix} \begin{bmatrix} \mathbf{x}_p(k) \\ \mathbf{x}_d(k) \\ \mathbf{x}_s(k) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_p \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} u(k) + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_d \\ \mathbf{B}_s & \mathbf{0} \end{bmatrix} \begin{bmatrix} v(k) \\ w(k) \end{bmatrix} . \tag{2.41}$$

The physical output before sensor noise $y(k)$ (also termed the *controlled* output) can be written as

$$y(k) = \begin{bmatrix} \mathbf{C}_p & \mathbf{C}_d & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}_p(k) \\ \mathbf{x}_d(k) \\ \mathbf{x}_s(k) \end{bmatrix} . \tag{2.42}$$

whereas the measured output $z(k)$ (also termed the *observed* output) is given by

$$z(k) = \begin{bmatrix} \mathbf{C}_p & \mathbf{C}_d & \mathbf{C}_s \end{bmatrix} \begin{bmatrix} \mathbf{x}_p(k) \\ \mathbf{x}_d(k) \\ \mathbf{x}_s(k) \end{bmatrix} + v(k) \tag{2.43}$$

The total state space description can be written in compact notation as

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k)+\mathbf{B}u(k)+\mathbf{w}(k) \\ y(k) &= \mathbf{C}_y\mathbf{x}(k) \\ z(k) &= \mathbf{C}_z\mathbf{x}(k)+v(k) \end{aligned} \tag{2.44}$$

where the matrices **A, B, C**$_y$ and **C**$_z$ are given by

$$\mathbf{A}=\begin{bmatrix} \mathbf{A}_p & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_d & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_s \end{bmatrix} \;,\; \mathbf{B}=\begin{bmatrix} \mathbf{B}_p \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \;,\; \mathbf{C}_y=\begin{bmatrix} \mathbf{C}_p & \mathbf{C}_d & \mathbf{0} \end{bmatrix} \;,\; \mathbf{C}_z=\begin{bmatrix} \mathbf{C}_p & \mathbf{C}_d & \mathbf{C}_z \end{bmatrix} \tag{2.45}$$

and $\mathbf{w}(k)$ is a vector white noise sequence with variance matrix

$$\mathbf{Q}_{\mathrm{w}}=E\{\mathbf{w}(k)\mathbf{w}^T(k)\}=\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_d Q_w \mathbf{B}_d^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}_s Q_v \mathbf{B}_s^T \end{bmatrix} \tag{2.46}$$

The vector noise sequence is correlated with $v(k)$ with covariance matrix

$$\mathbf{Q}_{\mathrm{w}v}=E\{\mathbf{w}(k)v(k)\}=\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{B}_s Q_v \end{bmatrix}. \tag{2.47}$$

Figure 2.5 shows the state space LQG feedback control structure. The output feedback controller corresponding to the transfer function $H(q^{-1})$ introduced in the previous section consists of an optimal state feedback law combined with an optimal state estimator. If the full state of the physical system is available and can be measured directly without noise, the linear optimal state feedback law is given by [7]

$$u(k)=-\mathbf{F}\,\mathbf{x}(k) \;, \tag{2.48}$$

where $\mathbf{F}$ is the optimal feedback gain matrix designed in order to minimise the steady state quadratic cost function for stationary signals

$$J=E\{y^2(k)+\rho\, u^2(k)\} \;. \tag{2.49}$$

Here ρ is again the effort weighting parameter.

The feedback gain matrix that minimises (2.49) is found from $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}_y$ as [8]

$$\mathbf{F} = \left[\mathbf{B}^T\mathbf{S}\mathbf{B} + \rho\right]^{-1}\mathbf{B}^T\mathbf{S}\mathbf{A} \ , \tag{2.50}$$

where the matrix $\mathbf{S}$ is the unique positive semidefinite solution to the discrete time Riccati equation [8]

$$\mathbf{S} - \mathbf{A}^T\mathbf{S}\mathbf{A} + \mathbf{A}^T\mathbf{S}\mathbf{B}\left[\mathbf{B}^T\mathbf{S}\mathbf{B} + \rho\right]^{-1}\mathbf{B}^T\mathbf{S}\mathbf{A} - \mathbf{C}_y^T\mathbf{C}_y = \mathbf{0} \ . \tag{2.51}$$

The numerical solution of this type of equation is treated in appendices A and B.

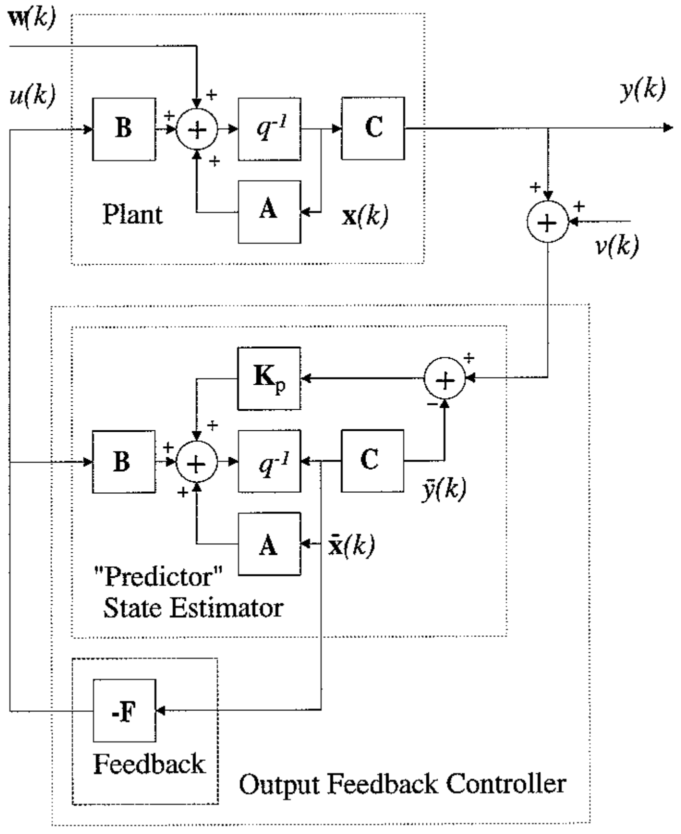

Figure 2.5. LQG state space feedback control structure.

Since a direct noise-free measurement of the internal state of the plant is usually not available for feedback, a state estimator is connected in parallel with the plant and used to provide an estimate of the state. The state estimate is thus found from measurements on the system input and system output contaminated by measurement/sensor noise. This state estimate is then used with the feedback law instead of the actual state. If the estimator takes the form of an optimal estimator, then the mean square error between the actual and estimated state will be minimal. The separation theorem [7] then states that the optimal LQG output feedback controller is found by using the optimal state estimate with the optimal feedback gain. The *prediction* state estimator takes the form [8]

$$\overline{\mathbf{x}}(k+1)=\mathbf{A}\,\overline{\mathbf{x}}(k)+\mathbf{B}\,u(k)+\mathbf{K}_p\left[z(k)-\mathbf{C}_z\,\overline{\mathbf{x}}(k)\right] \ , \tag{2.52}$$

and is essentially a model of the plant with an extra input given by the error between the plant and estimator outputs, scaled by the estimator gain $\mathbf{K}_p$. The estimator is termed a prediction estimator since a measurement at time k results in an estimate of the state that is valid at time k+1. The state estimate $\overline{\mathbf{x}}(k)$ is therefore termed the *prediction estimate.*

The estimator gain can be chosen in order to minimise the mean square error between the actual and estimated state

$$J_e=E\left\{\left[\mathbf{x}(k)-\overline{\mathbf{x}}(k)\right]^T\left[\mathbf{x}(k)-\overline{\mathbf{x}}(k)\right]\right\}. \tag{2.53}$$

The optimal prediction estimator gain that minimises the state error is given by [7]

$$\mathbf{K}_p=\left[\mathbf{A}\,\mathbf{P}\,\mathbf{C}_z^T+\mathbf{Q}_{wv}\right]\left[\mathbf{C}_z\,\mathbf{P}\,\mathbf{C}_z^T+Q_v\right]^{-1} \ , \tag{2.54}$$

where $\mathbf{P}$ is the unique positive semidefinite solution to the discrete time algebraic Riccati equation [7]

$$\mathbf{P}-\mathbf{A}_m\,\mathbf{P}\,\mathbf{A}_m^T+\mathbf{A}_m\,\mathbf{P}\,\mathbf{C}_z^T\left[\mathbf{C}_z\,\mathbf{P}\,\mathbf{C}_z^T+Q_v\right]^{-1}\mathbf{C}_z\,\mathbf{P}\,\mathbf{A}_m^T-\mathbf{Q}_m=\mathbf{0} \ , \tag{2.55}$$

with the matrices $\mathbf{A}_m$ and $\mathbf{Q}_m$ given by [7]

$$\mathbf{A}_m=\mathbf{A}-\mathbf{Q}_{wv}Q_v^{-1}\mathbf{C}_z \quad , \quad \mathbf{Q}_m=\mathbf{Q}_w-\mathbf{Q}_{wv}Q_v^{-1}\mathbf{Q}_{wv}^T \ . \tag{2.56}$$

The optimal gain thus depends on the variances Q_w and Q_v of the white noise signals $w(k)$ and $v(k)$. By inserting the optimal feedback law (2.48) (with the actual state replaced by the estimated state) into the estimator equation (2.52) a state equation for the output feedback controller can be found as

$$\begin{aligned}\overline{\mathbf{x}}(k+1)&=\left[\mathbf{A}-\mathbf{BF}-\mathbf{K}_p\mathbf{C}_z\right]\overline{\mathbf{x}}(k)+\mathbf{K}_p z(k)\\ u(k)&=-\mathbf{F}\,\overline{\mathbf{x}}(k)\end{aligned} \ , \tag{2.57}$$

and from this the transfer function of the resulting optimal feedback controller can be found as

$$H_{LQG}\left(q^{-1}\right)=\mathbf{F}\left[q\,\mathbf{I}-\mathbf{A}+\mathbf{B}\,\mathbf{F}+\mathbf{K}_p\mathbf{C}_z\right]^{-1}\mathbf{K}_p. \qquad (2.58)$$

and since $\mathbf{F}$ and $\mathbf{K}_p$ are functions of ρ, Q_v and Q_w thus shows that the dynamics of the controller are affected by all these parameters. Note that no direct link is present in this type of controller, since only output samples up to and including the k-1'th sample are used in the controller.

The sensitivity function of the feedback control loop is the closed loop transfer function from a disturbance $d_0(k)$ added to the controlled variable $y(k)$ to the controlled variable itself. In this case the controlled variable and the observed variable can be written as

$$y(k)=\mathbf{C}_y\mathbf{x}(k)+d_0(k) \quad , \quad z(k)=\mathbf{C}_z\mathbf{x}(k)+d_0(k) \qquad (2.59)$$

By combining the state estimator of (2.52) , the optimal feedback law of (2.48) (with the actual state replaced by the estimated one) and the state equation of (2.44) a state equation for the error between the physical and estimated state $\mathbf{e}(k)=\mathbf{x}(k)-\overline{\mathbf{x}}(k)$ can be found as

$$\begin{aligned}\mathbf{e}(k+1)&=\mathbf{x}(k+1)-\overline{\mathbf{x}}(k+1)\\&=\mathbf{A}\,\mathbf{x}(k)+\mathbf{B}\,u(k)-\mathbf{A}\,\overline{\mathbf{x}}(k)-\mathbf{B}\,u(k)-\mathbf{K}_p z(k)+\mathbf{K}_p\mathbf{C}_z\overline{\mathbf{x}}(k)\\&=\mathbf{A}\,\mathbf{e}(k)-\mathbf{K}_p\left(\mathbf{C}_z\mathbf{x}(k)+d_0(k)\right)+\mathbf{K}_p\mathbf{C}_z\overline{\mathbf{x}}(k)\\&=\left[\mathbf{A}-\mathbf{K}_p\mathbf{C}_z\right]\mathbf{e}(k)-\mathbf{K}_p d_0(k)\end{aligned} \qquad (2.60)$$

Accordingly the equation governing the physical state can be written as

$$\begin{aligned}\mathbf{x}(k+1)&=\mathbf{A}\,\mathbf{x}(k)-\mathbf{B}\,\mathbf{F}\,\overline{\mathbf{x}}(k)\\&=\mathbf{A}\,\mathbf{x}(k)-\mathbf{B}\,\mathbf{F}\left(\mathbf{x}(k)-\mathbf{e}(k)\right)\\&=\left[\mathbf{A}-\mathbf{B}\,\mathbf{F}\right]\mathbf{x}(k)+\mathbf{B}\,\mathbf{F}\,\mathbf{e}(k)\end{aligned} \qquad (2.61)$$

By combining (2.60) and (2.61) into one state equation the sensitivity function can be written in state space form as

$$\begin{bmatrix}\mathbf{x}(k+1)\\ \mathbf{e}(k+1)\end{bmatrix}=\begin{bmatrix}\mathbf{A}-\mathbf{B}\,\mathbf{F} & \mathbf{B}\,\mathbf{F}\\ \mathbf{0} & \mathbf{A}-\mathbf{K}_p\mathbf{C}_z\end{bmatrix}\begin{bmatrix}\mathbf{x}(k)\\ \mathbf{e}(k)\end{bmatrix}+\begin{bmatrix}\mathbf{0}\\ -\mathbf{K}_p\end{bmatrix}d_0(k) \quad , \quad y(k)=\begin{bmatrix}\mathbf{C}_y & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{x}(k)\\ \mathbf{e}(k)\end{bmatrix}+d_0(k) \qquad (2.62)$$

The closed loop poles are the eigenvalues of the joint state matrix. These are made up by the eigenvalues of $\mathbf{A}-\mathbf{B}\,\mathbf{F}$ (the regulator poles) and $\mathbf{A}-\mathbf{K}_p\mathbf{C}_z$ (the estimator poles) respectively, due to the special form of the joint state matrix. This means that the closed loop characteristic values are equal to the characteristic values corresponding to the optimal state feedback controller plus the characteristic values of the optimal state estimator [7]. It is a fundamental property of the optimal feedback law and of the optimal state estimator that if the system to be controlled is stable (all eigenvalues of **A** are inside the unit circle), then the regulator and estimator poles will also be stable [7] which in turn means that the closed loop will be stable. However there is no guarantee that the controller itself will be stable, and moreover no guaranteed stability margins exist for the LQG controller [6].

The prediction estimator provides an estimate of the state at time *k* from measurements up to and including time *k*-1. If a measurement of the plant output at time *k* is assumed available when calculating the *k*'th control signal sample (as in the case of the IMC/Wiener based controller), the prediction estimate can be modified to include this last piece of information. Figure 2.6 shows an LQG feedback controller using this *current* estimator.

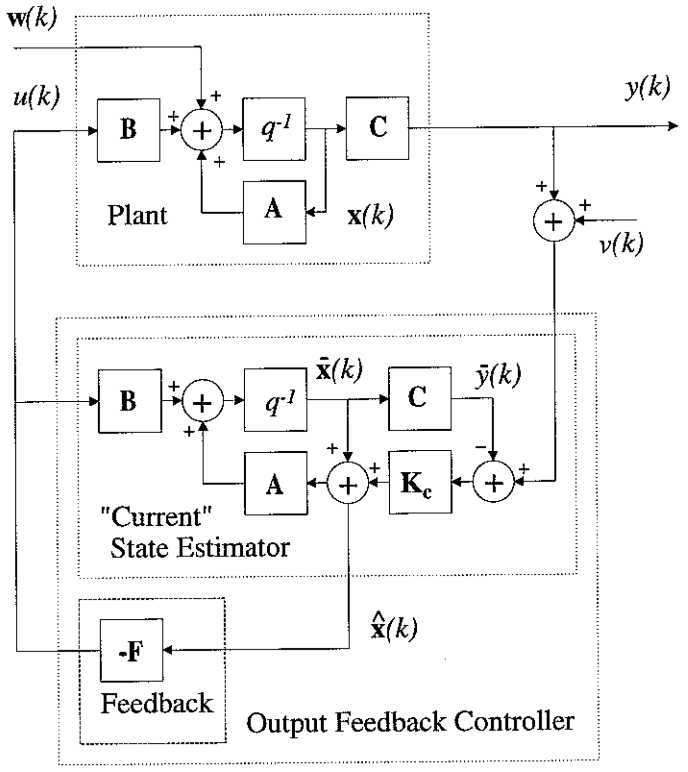

Figure 2.6. LQG feedback control with "current" estimator.

The resulting state estimate $\hat{\mathbf{x}}(k)$ is called the *current* estimate and is found as [8]

$$\hat{\mathbf{x}}(k) = \bar{\mathbf{x}}(k) + \mathbf{K}_c \left[z(k) - \mathbf{C}_z\, \bar{\mathbf{x}}(k) \right] \quad , \tag{2.63}$$

where $\mathbf{K}_c$ is the current estimator gain matrix. The prediction update is given by [8]

$$\overline{\mathbf{x}}(k+1) = \mathbf{A}\,\hat{\mathbf{x}}(k) + \mathbf{B}\,u(k) \ , \tag{2.64}$$

and by inserting the expression for the current estimate a state equation for the prediction update is found as

$$\overline{\mathbf{x}}(k+1) = \mathbf{A}\,\overline{\mathbf{x}}(k) + \mathbf{B}\,u(k) + \mathbf{A}\,\mathbf{K}_c\left[z(k) - \mathbf{C}_z\,\overline{\mathbf{x}}(k)\right] \ . \tag{2.65}$$

By comparing with equation (2.52) it is clear that the current estimator gain is related to the prediction estimator gain by $\mathbf{K}_p = \mathbf{A}\mathbf{K}_c$.The optimal current estimator gain that minimises

$$J_e = E\left\{\left[\mathbf{x}(k) - \hat{\mathbf{x}}(k)\right]^T\left[\mathbf{x}(k) - \hat{\mathbf{x}}(k)\right]\right\} \tag{2.66}$$

is thus given by

$$\mathbf{K}_c = \left[\mathbf{P}\,\mathbf{C}_z^T + \mathbf{A}^{-1}\mathbf{Q}_{wv}\right]\left[\mathbf{C}_z\,\mathbf{P}\,\mathbf{C}_z^T + Q_v\right]^{-1} \tag{2.67}$$

where again $\mathbf{P}$ is the unique positive semidefinite solution to equation (2.55). Note that in this case the system matrix needs to be inverted if $\mathbf{w}(k)$ and $v(k)$ are correlated. The state equation for the predictor estimate can be rearranged as

$$\overline{\mathbf{x}}(k+1) = \left[\mathbf{A} - \mathbf{A}\,\mathbf{K}_c\,\mathbf{C}_z\right]\overline{\mathbf{x}}(k) + \mathbf{B}\,u(k) + \mathbf{A}\,\mathbf{K}_c\,z(k) \tag{2.68}$$

i.e. the state differential equation of a two input system. If the current estimate of the state is used in the feedback law , the control signal is given by

$$u(k) = -\mathbf{F}\,\hat{\mathbf{x}}(k) = -\mathbf{F}\left(\overline{\mathbf{x}}(k) + \mathbf{K}_c\left[z(k) - \mathbf{C}_z\,\overline{\mathbf{x}}(k)\right]\right) = -\left[\mathbf{F} - \mathbf{F}\,\mathbf{K}_c\,\mathbf{C}_z\right]\overline{\mathbf{x}}(k) - \mathbf{F}\,\mathbf{K}_c\,z(k) \tag{2.69}$$

which is the output equation of the “current state” LQG output feedback controller. Inserting this expression into equation (2.68) yields

$$\overline{\mathbf{x}}(k+1) = \left[\mathbf{A} - \mathbf{A}\,\mathbf{K}_c\,\mathbf{C}_z - \mathbf{B}\,\mathbf{F} + \mathbf{B}\,\mathbf{F}\,\mathbf{K}_c\,\mathbf{C}_z\right]\overline{\mathbf{x}}(k) + \left[\mathbf{A}\,\mathbf{K}_c - \mathbf{B}\,\mathbf{F}\,\mathbf{K}_c\right]z(k) \tag{2.70}$$

which is the state equation of the “current state” LQG output feedback controller. Finally from these equations the transfer function $H_{LQG}\left(q^{-1}\right)$ of the controller can be written as

$$H_{LQG}\left(q^{-1}\right) = \left[\mathbf{F} - \mathbf{F}\,\mathbf{K}_c\,\mathbf{C}_z\right]\left[q\,\mathbf{I} - \mathbf{A} + \mathbf{A}\,\mathbf{K}_c\,\mathbf{C}_z + \mathbf{B}\,\mathbf{F} - \mathbf{B}\,\mathbf{F}\,\mathbf{K}_c\,\mathbf{C}_z\right]^{-1}\left[\mathbf{A}\,\mathbf{K}_c - \mathbf{B}\,\mathbf{F}\,\mathbf{K}_c\right] + \mathbf{F}\,\mathbf{K}_c \ . \tag{2.71}$$

This controller has a direct link as a result of assuming that the k'th output sample is available when calculating the k'th control signal sample. As in the case of the IMC/Wiener controller one sample calculation delay should be included in the plant model when evaluating the performance of this type of controller. A state equation for the sensitivity function can be found in the same way as for the predictor estimator controller and is given by

$$\begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{e}(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A}-\mathbf{B}\mathbf{F} & \mathbf{B}\mathbf{F}[\mathbf{I}-\mathbf{K}_c\mathbf{C}_z] \\ \mathbf{0} & \mathbf{A}-\mathbf{A}\mathbf{K}_c\mathbf{C}_z \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{e}(k) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ -\mathbf{A}\mathbf{K}_c \end{bmatrix} d_0(k) \quad , \quad y(k) = \begin{bmatrix} \mathbf{C}_y & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{e}(k) \end{bmatrix} + d_0(k) \tag{2.72}$$

This equation shows that the closed loop characteristic values are the same as for the predictor estimator controller, since $\mathbf{K}_p = \mathbf{A}\mathbf{K}_c$.

A special case of the LQG problem corresponding to the control problem studied in the simulations later in this report will be discussed here. In this case the sensor noise is assumed to be white, so that no sensor noise filter needs to be included in the joint state space description of (2.44). The poles of the plant and the disturbance noise filter are furthermore assumed to be identical. so that $\mathbf{A}_p = \mathbf{A}_d = \mathbf{A}$. The corresponding system description becomes

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\,\mathbf{x}(k) + \mathbf{B}\,u(k) + \mathbf{L}\,w(k) \\ y(k) &= \mathbf{C}\,\mathbf{x}(k) \qquad , \\ z(k) &= y(k) + v(k) \end{aligned} \tag{2.73}$$

The variance matrix $\mathbf{Q}_{\mathbf{w}}$ of (2.46) and covariance matrix $\mathbf{Q}_{\mathbf{w}v}$ of (2.47) thus become

$$\mathbf{Q}_{\mathbf{w}} = \mathbf{L} Q_w \mathbf{L}^T \quad , \quad \mathbf{Q}_{\mathbf{w}v} = \mathbf{0} , \tag{2.74}$$

so that in this case the disturbance $\mathbf{w}(k) = \mathbf{L}\,w(k)$ and the sensor noise $v(k)$ is not correlated. The optimal predictor estimator and current estimator gains are then given by

$$\mathbf{K}_p = \mathbf{A}\mathbf{P}\mathbf{C}^T\left[\mathbf{C}\mathbf{P}\mathbf{C}^T + Q_v\right]^{-1} \quad \text{and} \quad \mathbf{K}_c = \mathbf{P}\mathbf{C}^T\left[\mathbf{C}\mathbf{P}\mathbf{C}^T + Q_v\right]^{-1} , \tag{2.75}$$

with the matrix $\mathbf{P}$ being the unique positive semidefinite solution to the Riccati equation

$$\mathbf{P} - \mathbf{A}\mathbf{P}\mathbf{A}^T + \mathbf{A}\mathbf{P}\mathbf{C}^T\left[\mathbf{C}\mathbf{P}\mathbf{C}^T + Q_v\right]^{-1}\mathbf{C}\mathbf{P}\mathbf{A}^T - \mathbf{L}Q_w\mathbf{L}^T = \mathbf{0} . \tag{2.76}$$

Note that in this case no inversion of the state matrix is needed to form the optimal current estimator gain.

2.4 Linear Quadratic Gaussian (LQG) Control using a polynomial model

Another approach which can be taken to LQG control when the plant and disturbance model are specified in terms of their poles and zeros is based on the direct manipulation of polynomials. This method implicitly uses the IMC structure under the assumption that the plant is stable.

The plant and disturbance model transfer functions $G_p(z^{-1})$ and $G_d(z^{-1})$, shown in Figure 2.1, are assumed to be represented as a ratio of polynomials in z^{-1} such that

$$G_p(z^{-1}) = z^{-n}\frac{N_p(z^{-1})}{D_p(z^{-1})} \quad , \quad G_d(z^{-1}) = \frac{N_d(z^{-1})}{D_d(z^{-1})} \tag{2.77}$$

where z^{-n} implies a pure delay of n samples and $N_p(z^{-1})$, $N_d(z^{-1})$, $D_p(z^{-1})$ and $D_d(z^{-1})$ are polynomials in z^{-1}. For example

$$\begin{aligned} N_p(z^{-1}) &= \left(n_{p0} + n_{p1}z^{-1} + n_{p2}z^{-2} + \ldots\ldots + n_{pn_{np}}z^{-n_{np}}\right) \\ D_p(z^{-1}) &= \left(1 + d_{p1}z^{-1} + d_{p2}z^{-2} + \ldots\ldots d_{pn_{dp}}z^{-n_{dp}}\right) \end{aligned} \quad . \tag{2.78}$$

The integers n_{dp}, n_{np} define the degree or order of the polynomials and $D_p(z^{-1})$ is monic since its first coefficient is unity. The roots of $D_p(z^{-1})$ and $N_p(z^{-1})$ respectively define the poles and zeros of the plant and the two polynomials are assumed to be relatively prime so that they do not have any common roots. If the roots of $D_p(z^{-1})$ all lie within the unit circle, ($|z|=1$), the plant is stable. If the roots of $N_p(z^{-1})$ all lie within the unit circle then the plant is also minimum phase. In structural problems such as those discussed within this report, the plant is usually stable but not always minimum phase.

The system output $y(k)$ before measurement noise can be written

$$y(k) = G_p(q^{-1})u(k) + G_d(q^{-1})w(k) \quad , \tag{2.79}$$

which, by using (2.77), can be written

$$D_p(q^{-1})D_d(q^{-1})y(k) = q^{-n}N_p(q^{-1})D_d(q^{-1})u(k) + N_d(q^{-1})D_p(q^{-1})w(k) \quad . \tag{2.80}$$

This is more commonly expressed in the ARMAX (Auto Regressive Moving Average with eXogenous variable) form

$$A(q^{-1})y(k) = q^{-n}B(q^{-1})u(k) + C(q^{-1})w(k) \quad , \tag{2.81}$$

where

$$A\left(q^{-1}\right)=D_p\left(q^{-1}\right)D_d\left(q^{-1}\right) \quad , \quad B\left(q^{-1}\right)=N_p\left(q^{-1}\right)D_d\left(q^{-1}\right) \quad , \quad C\left(q^{-1}\right)=N_d\left(q^{-1}\right)D_p\left(q^{-1}\right) \quad . \tag{2.82}$$

When a cost function of the form

$$J=E\left[y^2(k)+\rho u^2(k)\right] \tag{2.7}$$

is used, and the optimal controller is assumed to have the IMC form shown in Figure 2.2 and given by equation (2.25) (with a plant model $\hat{G}_p\left(z^{-1}\right)$), the control filter $W_0\left(z^{-1}\right)$ is then found to take the form [9]

$$W_0\left(z^{-1}\right)=\frac{-Q_w A\left(z^{-1}\right)A(z)}{D_c\left(z^{-1}\right)D_f\left(z^{-1}\right)}\left\{\frac{z^k B(z)C(z)C\left(z^{-1}\right)}{D_c(z)D_f(z)A\left(z^{-1}\right)}\right\}_+ , \tag{2.83}$$

and is expressed in terms of the ARMAX polynomials (2.82), the spectral factors $D_c\left(z^{-1}\right)$ and $D_f\left(z^{-1}\right)$ and their inverses, which are discussed below, and the variance of the white noise disturbance signal Q_w. The notation $\{\}_+$ implies that the causal part of the enclosed polynomial in z should be taken. The derivation of this form of controller is described in [9], and calculation of the filter $W_0\left(z^{-1}\right)$ can be found in [10].

There are four basic steps involved in the derivation of the LQG polynomial controller coefficients:

1) ARMAX representation of the system
2) Spectral factorisation
3) Solution of Diophantine equations
4) Calculation of the optimal filter

These are discussed in detail in [10]. Steps 1 and 4 are relatively straightforward. Steps 2 and 3 are reviewed below.

Spectral factorisation.

This refers to the process of splitting a rational power spectral density into a causal, stable part and an anti-causal stable part such that

$$S_{xx}\left(z^{-1}\right)=X\left(z^{-1}\right)X(z) \quad , \tag{2.84}$$

where $X\left(z^{-1}\right)$ is causal and stable and its inverse $X^{-1}\left(z^{-1}\right)$ is also causal and stable. Alternative forms of spectral factorisation arise from the representation of the spectra in different domains and are discussed in [11].

There are several methods of performing the spectral factorisation of polynomials. The simplest of these can be performed in MATLAB by finding the roots of a polynomial. As these roots occur in pairs which are reciprocals reflected in the unit circle, the roots inside the unit circle correspond to the new roots of $X\left(z^{-1}\right)$ and these can be used to recreate the new polynomial required. This method is simple and fast but inaccuracies in finding the roots of a polynomial lead to numerical difficulties, especially when roots are close to the unit circle and in these cases, numerical algorithms which are less prone to numerical errors must be used. Some of these are discussed in [12]and the simplest of these, Peterka's algorithm, is implemented in MATLAB in [13]. This implementation was used in basic simulations for this report and was found to work well. However, as it is an iterative process, convergence can be time-consuming when high accuracy is required or where polynomials have high degrees. More efficient algorithms are also discussed in [12].

The specific spectral factors required from equation (2.83) are $D_c\left(z^{-1}\right)$ and $D_f\left(z^{-1}\right)$ where

$$\begin{aligned} D_c(z)D_c\left(z^{-1}\right) &= B(z)B\left(z^{-1}\right) + \rho A(z)A\left(z^{-1}\right) \\ D_f(z)D_f\left(z^{-1}\right) &= Q_w C(z)C\left(z^{-1}\right) + Q_v A(z)A\left(z^{-1}\right) \end{aligned} . \tag{2.85}$$

Q_v is the measurement noise intensity as described earlier in this chapter. The spectral factorisation can be performed as the right hand sides of the above equations form real, even, and non-negative spectra in z^{-1}.

Solution of the Diophantine equations.

When the IMC formulation is used the controller is assumed to take the form given in (2.25). The polynomial LQG problem can then be solved using a discrete-time Wiener-Hopf equation (appendices 1-3, [9]), and the optimal filter $W_0\left(z^{-1}\right)$ takes the form shown in equation (2.83). To find the optimal filter coefficients, it is then necessary to find the causal part of the term in brackets, i.e. to find

$$\left\{\frac{z^n B(z)C(z)C\left(z^{-1}\right)}{D_c(z)D_f(z)A\left(z^{-1}\right)}\right\}_+ . \tag{2.86}$$

The causal part of this term is found by solving a Diophantine equation [14] which is an equation of the form $A(x)X(x) + B(x)Y(x) = C(x)$ where *A, B* and *C* are known polynomials in *x* and we wish to obtain the minimal degree solution for the X and Y polynomials.

After substituting the spectral factors (2.85) into (2.86), it can be rewritten as

$$\left\{\frac{z^n B(z)C(z)C(z^{-1})}{D_c(z)D_f(z)A(z^{-1})}\right\}_+ = \frac{1}{Q_w}\left\{\frac{z^n B(z)D_f(z^{-1})}{D_c(z)A(z^{-1})} - \frac{Q_w B(z)A(z)}{D_c(z)D_f(z)}\right\}_+ . \tag{3.87}$$

The second term here is entirely non-causal and, by using the Diophantine equation

$$D_c(z)G(z^{-1}) + z^g F(z^{-1})A(z^{-1}) = z^n B(z)D_f(z^{-1}) \ , \tag{3.88}$$

it is found that

$$\frac{z^n B(z)D_f(z^{-1})}{D_c(z)A(z^{-1})} = \frac{G(z^{-1})}{A(z^{-1})} + \frac{z^g F(z^{-1})}{D_c(z)} \ , \tag{3.89}$$

and, when g and the order of $F(z^{-1})$ are chosen appropriately, the second term again becomes entirely non-causal, leaving only $G(z^{-1})/A(z^{-1})$ such that

$$\left\{\frac{z^n B(z)C(z)C(z^{-1})}{D_c(z)D_f(z)A(z^{-1})}\right\}_+ = \frac{1}{Q_w}\frac{G(z^{-1})}{A(z^{-1})} . \tag{3.90}$$

Now, substituting this back into (2.83), the equation for the optimal filter $W_0(z^{-1})$ required in (2.25) to form the optimal controller is given by

$$W_0(z^{-1}) = -\frac{A(z^{-1})G(z^{-1})}{D_c(z^{-1})D_f(z^{-1})} . \tag{3.91}$$

MATLAB code for solving Diophantine equations can be found in [14].

3. Plant and disturbance model

3.1 General description

This chapter describes the system model used in the simulations of IMC and LQG feedback control. The modelled physical system is a simply supported, rectangular thin plate, excited by a piezoceramic patch control actuator and a disturbance point force. The system output is the transversal velocity at a given point on the plate. A continuous time state space description of the physical system is derived and converted to discrete time by assuming zero-order hold D/A conversion. The discrete time model is further modified to include a pure time delay in the control signal path.

3.2 Equations of motion

The transversal displacement distribution $\xi(x,y,t)$ of a rectangular plate to time t can be approximated by the truncated sum of an infinite series of modal contributions [15]

$$\xi(x,y,t)=\sum_{m=1}^{M}\sum_{n=1}^{N}\Phi_{mn}(x,y)\xi_{mn}(t) \ , \tag{3.1}$$

where $\Phi_{mn}(x,y)$ is the (m,n)'th mode shape function and $\xi_{mn}(t)$ is the corresponding instantaneous amplitude. M and N are the number of x-direction and y-direction modes included in the modal sum. The time dependence of the (m,n)'th modal amplitude is governed by the second order differential equation [15]

$$\ddot{\xi}_{mn}(t)+2\zeta_{mn}\omega_{mn}\dot{\xi}_{mn}(t)+\omega_{mn}^2\xi_{mn}(t)=\frac{1}{ab\rho h\Lambda_{mn}}\int_0^a\int_0^b\Phi_{mn}(x,y)p(x,y,t)dxdy=\Gamma_{mn}(t) \ , \tag{3.2}$$

where a dot denotes differentiation with respect to time, ω_{mn} is the natural frequency associated with the (m,n)'th eigenmode and ζ_{mn} is a damping term included to account for various damping mechanisms of the (m,n)'th mode. The pressure distribution acting upon the plate is denoted $p(x,y,t)$, corresponding to a generalised forcing function $\Gamma_{mn}(t)$ associated with the corresponding modal amplitude. The length, width and thickness of the plate are denoted a, b and h, while ρ is the density of the plate material and Λ_{mn} is the mode normalisation factor associated with the mode shape function. For a simply supported panel the resonance frequencies are given by [15]

$$\omega_{mn}=\sqrt{\frac{EI}{\rho h}}\left(k_m^2+k_n^2\right) \quad \text{where} \quad k_m=\frac{m\pi}{a} \quad , \quad k_n=\frac{n\pi}{b} \quad \text{and} \quad m,n=1,2,3\ldots \ , \tag{3.3}$$

where E is the Young's modulus of the plate material and I is the plate moment of inertia given by $I = h^3 / [12(1-\nu^2)]$, where ν is the Poisson's ratio of the plate material. The mode shape functions are given by

$$\Phi_{mn}(x,y) = \varphi_m(x)\psi_n(y) \quad , \quad \varphi_m(x) = \sin(k_m x) \quad , \quad \psi_n(y) = \sin(k_n y) \quad , \quad m,n = 1,2,3\ldots \ , \tag{3.4}$$

and the mode normalisation factor is $\Lambda_{mn} = 1/4$.

3.3 Excitation by a point force

The plate pressure distribution corresponding to a point force acting on the plate in the point (x_p, y_p) is given by

$$p(x,y,t) = f(t)\delta(x - x_p)\delta(y - y_p), \tag{3.5}$$

where $f(t)$ is the time varying amplitude of the point force and $\delta(x)$ is the spatial Dirac delta function. The corresponding generalised forcing function becomes

$$\Gamma_{mn}(t) = \frac{1}{ab\rho h\Lambda_{mn}} \int_0^a \int_0^b f(t)\delta(x - x_p)\delta(y - y_p)\varphi_m(x)\psi_n(y)dx\,dy = \frac{4}{ab\rho h}\sin(k_m x_p)\sin(k_n y_p) f(t). \tag{3.6}$$

Introducing the modal excitation factor α_{mn} the generalised forcing function can be written as

$$\Gamma_{mn}(t) = \alpha_{mn} f(t) \quad \text{where} \quad \alpha_{mn} = \frac{4}{ab\rho h}\sin(k_m x_p)\sin(k_n y_p) \ . \tag{3.7}$$

The magnitude of the excitation factor depends on the point force position and the modal index.

3.4 Excitation by a pair of piezoceramic patches

A pair of piezoceramic patches bonded symmetrically on each side of the plate with their edges parallel to the plate edges and driven 180 degrees out of phase with each other, will induce internal moments in the plate given by [16]

$$m_x(x,y,t) = m_y(x,y,t) = C_0 \varepsilon_{pe}(t)[h(x - x_1) - h(x - x_2)][h(y - y_1) - h(y - y_2)], \tag{3.8}$$

where $h(x)$ is the Heaviside step function, (x_1, x_2) and (y_1, y_2) are the edge positions of the actuator pair, C_0 is the piezoelectric strain-plate moment coupling term and $\varepsilon_{pe}(t)$ is the time varying induced strain proportional to the applied voltage to the patch pair.

The induced moments correspond to a forcing pressure distribution given by [16]

$$p(x,y,t) = \frac{\partial^2 m_x(x,y,t)}{\partial x^2} + \frac{\partial^2 m_y(x,y,t)}{\partial y^2} \ . \tag{3.9}$$

The second order derivatives of the moments can be found from equation (3.8) as

$$\begin{aligned} \frac{\partial^2 m_x(x,y,t)}{\partial x^2} &= C_0 \varepsilon_{pe}(t)[\delta'(x-x_1) - \delta'(x-x_2)][h(y-y_1) - h(y-y_2)] \\ \frac{\partial^2 m_y(x,y,t)}{\partial y^2} &= C_0 \varepsilon_{pe}(t)[h(x-x_1) - h(x-x_2)][\delta'(y-y_1) - \delta'(y-y_2)] \end{aligned} \ , \tag{3.10}$$

where $\delta'(x)$ is the derivative of the Dirac delta function with respect to x. The generalised forcing function is now found by

$$\begin{aligned} \Gamma_{mn}(t) \ &= \ \frac{C_0 \varepsilon_{pe}(t)}{abph\Lambda_{mn}} \left[\int_0^a \varphi_m(x)[\delta'(x-x_1) - \delta'(x-x_2)]dx \right] \left[\int_0^b \psi_n(y)[h(y-y_1) - h(y-y_2)]dy \right] \\ &+ \ \frac{C_0 \varepsilon_{pe}(t)}{abph\Lambda_{mn}} \left[\int_0^a \varphi_m(x)[h(x-x_1) - h(x-x_2)]dx \right] \left[\int_0^b \psi_n(y)[\delta'(y-y_1) - \delta'(y-y_2)]dy \right] \end{aligned} \ , \tag{3.11}$$

where the linear independence of the beam functions has been exploited in the evaluation of the double integrals. The integrals involving derivatives of delta functions can be solved by making use of the fundamental sampling property of the n'th derivative $\delta^{(n)}(x)$ of the delta function [17]

$$\int_{x_0}^{x_2} f(x)\delta^{(n)}(x-x_1)dt = (-1)^n f^{(n)}(x_1) \ , \tag{3.12}$$

where $x_0 < x_1 < x_2$ and $f^{(n)}(x)$ is assumed to be continuous in the neighbourhood of x_1. The generalised forcing function thus becomes

$$\Gamma_{mn}(t) = \beta_{mn} \, \varepsilon_{pe}(t) \quad , \tag{3.13}$$

where β_{mn} is the modal excitation factor given by

$$\beta_{mn}=\frac{C_0}{ab\rho h\Lambda_{mn}}\left[\left[\varphi_m'(x_2)-\varphi_m'(x_1)\right]\int_{y_1}^{y_2}\psi_n(y)dy+\left[\psi_n'(y_2)-\psi_n'(y_1)\right]\int_{x_1}^{x_2}\varphi_m(x)dx\right] \quad . \tag{3.14}$$

Inserting the mode shape functions of equation (3.4) yields the modal excitation factor for a simply supported plate

$$\beta_{mn}=-\frac{4C_0}{ab\rho h}\left[\frac{k_m}{k_n}+\frac{k_n}{k_m}\right]\left[\cos(k_m x_2)-\cos(k_m x_1)\right]\left[\cos(k_n y_2)-\cos(k_n y_1)\right] \quad . \tag{3.15}$$

3.5 State space description of structural dynamics

Introducing state space notation, the differential equation for the *(m,n)*'th modal amplitude of a plate excited by a disturbance point force and a control piezo actuator can be written as

$$\begin{bmatrix}\dot{\xi}_{mn}(t)\\ \ddot{\xi}_{mn}(t)\end{bmatrix}=\begin{bmatrix}0 & 1\\ -\omega_{mn}^2 & -2\zeta_{mn}\omega_{mn}\end{bmatrix}\begin{bmatrix}\xi_{mn}(t)\\ \dot{\xi}_{mn}(t)\end{bmatrix}+\begin{bmatrix}0\\ \beta_{mn}\end{bmatrix}\varepsilon_{pe}(t)+\begin{bmatrix}0\\ \alpha_{mn}\end{bmatrix}f(t) \tag{3.16}$$

Using a more compact notation this can be written as

$$\dot{\mathbf{x}}_{mn}(t)=\mathbf{A}_{mn}\mathbf{x}_{mn}(t)+\mathbf{B}_{mn}\varepsilon_{pe}(t)+\mathbf{L}_{mn}f(t) \quad , \tag{3.17}$$

where the modal state $\mathbf{x}_{mn}(t)$ is given by

$$\mathbf{x}_{mn}(t)=\left[\xi_{mn}(t) \quad \dot{\xi}_{mn}(t)\right]^T \quad , \tag{3.18}$$

and where superscript *T* denotes transposition. The modal state matrix $\mathbf{A}_{mn}$, control input matrix $\mathbf{B}_{mn}$ and disturbance input matrix $\mathbf{L}_{mn}$ are

$$\mathbf{A}_{mn}=\begin{bmatrix}0 & 1\\ -\omega_{mn}^2 & -2\zeta_{mn}\omega_{mn}\end{bmatrix} \quad , \quad \mathbf{B}_{\mathbf{mn}}=\begin{bmatrix}0\\ \beta_{mn}\end{bmatrix} \quad , \quad \mathbf{L}_{mn}=\begin{bmatrix}0\\ \alpha_{mn}\end{bmatrix}. \tag{3.19}$$

Gathering the state space descriptions of each modal contribution into a total state space formulation yields

$$\dot{\mathbf{x}}_c(t)=\mathbf{A}_c\mathbf{x}_c(t)+\mathbf{B}_c\varepsilon_{pe}(t)+\mathbf{L}_cf(t) \quad , \tag{3.20}$$

where

$$\mathbf{x}_c(t)=\begin{bmatrix}\mathbf{x}_{11}(t)\\ \mathbf{x}_{21}(t)\\ \vdots\\ \mathbf{x}_{MN}(t)\end{bmatrix} \quad , \quad \mathbf{A}_c=\begin{bmatrix}\mathbf{A}_{11} & \mathbf{0} & \cdots & \mathbf{0}\\ \mathbf{0} & \mathbf{A}_{21} & \cdots & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_{MN}\end{bmatrix} \quad , \quad \mathbf{B}_c=\begin{bmatrix}\mathbf{B}_{11}\\ \mathbf{B}_{21}\\ \vdots\\ \mathbf{B}_{MN}\end{bmatrix} \quad , \quad \mathbf{L}_c=\begin{bmatrix}\mathbf{L}_{11}\\ \mathbf{L}_{21}\\ \vdots\\ \mathbf{L}_{MN}\end{bmatrix} . \tag{3.21}$$

The elements of the full state vector can be linearly combined to form the transversal velocity at a point $(x_s y_s)$ on the plate as

$$\dot{\xi}(x_s,y_s,t)=\mathbf{C}_c\mathbf{x}_c(t) \quad \text{where} \quad \mathbf{C}_c=\begin{bmatrix}0 & \Phi_{11}(x_s,y_s) & 0 & \Phi_{21}(x_s,y_s) & \cdots & 0 & \Phi_{MN}(x_s,y_s)\end{bmatrix} . \tag{3.22}$$

The subscript c indicates that these are the state space matrices of the continuous time system model.

3.6 Discrete time state space description

By assuming that system under consideration is sampled with a zero-order hold (ZOH) circuit (i.e. that the control signal is held piece-wise constant between the sample instants), an equivalent discrete time state space description can be found by "sampling of the continuous time system". The discrete time state space description is given as [1],[8]

$$\mathbf{x}(k+1)=\mathbf{A}\,\mathbf{x}(k)+\mathbf{B}\,u(k)+\mathbf{d}(k) \quad , \quad y(k)=\mathbf{C}\mathbf{x}(k) \tag{3.23}$$

where k is the discrete time variable. The discrete time state, control-, output- and disturbance signals are given by [1],[8]

$$\mathbf{x}(k)=\mathbf{x}_c(kT) \quad , \quad u(k)=\varepsilon_{pe}(kT) \quad , \quad y(k)=\dot{\xi}(x_s,y_s,kT) \quad , \quad \mathbf{d}(k)=\left[\int_{kT}^{kT+T}e^{\mathbf{A}_c\tau}f(\tau)d\tau\right]\mathbf{L}_c \quad , \tag{3.24}$$

where T is the sampling interval, and the matrices **A, B** and **C** are given by

$$\mathbf{A}=e^{\mathbf{A}_cT} \quad , \quad \mathbf{B}=\left[\int_0^T e^{\mathbf{A}_c\tau}d\tau\right]\mathbf{B}_c \quad , \quad \mathbf{C}=\mathbf{C}_c \ . \tag{3.25}$$

This transformation does not involve other approximations than the assumption of a piece-wise constant control signal, and thus gives the exact values of the state variables and the output *at the sampling instants* provided this assumption is true [1]. For simplicity it will furthermore be assumed that the continuous time disturbance $f(t)$ is also piece-wise constant, so that

$$\mathbf{d}(k)=\left[\int_{kT}^{kT+T} e^{\mathbf{A}_c\tau} f(\tau)\,d\tau\right]\mathbf{L}_c=\left[\int_{0}^{T} e^{\mathbf{A}_c\tau}\,d\tau\right]\mathbf{L}_c f(kT)=\mathbf{L}\,w(k) \tag{3.26}$$

where

$$\mathbf{L}=\left[\int_{0}^{T} e^{\mathbf{A}_c\tau}\,d\tau\right]\mathbf{L}_c \quad , \quad w(k)=f(kT) \tag{3.27}$$

3.7 Systems with pure control delay

If for some reason (finite computation time, finite conversion time, filter delays etc.) a pure time delay λ is present in the control input, the continuous time state space description takes the form [8]

$$\dot{\mathbf{x}}_c(t)=\mathbf{A}_c\mathbf{x}_c(t)+\mathbf{B}_c\varepsilon_{pe}(t-\lambda)+\mathbf{L}_c f(t) \quad , \quad \xi(t)=\mathbf{C}_c\mathbf{x}_c(t) \ . \tag{3.28}$$

To convert this system to discrete time it is convenient to divide the pure time delay into an integer number of samples minus a fraction of a sample [8]

$$\lambda=lT-\eta T \quad \text{where} \quad l=1,2,\ldots \quad , \quad 0\le\eta<1 \ . \tag{3.29}$$

Using a zero order hold D/A converter the equivalent discrete time state space description is found as [8]

$$\mathbf{x}(k+1)=\mathbf{A}\mathbf{x}(k)+\mathbf{B}_1 u(k-l)+\mathbf{B}_2 u(k-l+1)+\mathbf{L}w(k) \quad , \quad y(k)=\mathbf{C}\mathbf{x}(k) \tag{3.30}$$

where [8]

$$\mathbf{A}=e^{\mathbf{A}_cT} \quad , \quad \mathbf{B}_1=e^{\mathbf{A}_c\eta T}\left[\int_0^{T-\eta T} e^{\mathbf{A}_c\tau}d\tau\right]\mathbf{B}_c \quad , \quad \mathbf{B}_2=\left[\int_0^{\eta T} e^{\mathbf{A}_c\tau}d\tau\right]\mathbf{B}_c \quad , \quad \mathbf{L}=\left[\int_0^{T} e^{\mathbf{A}_c\tau}d\tau\right]\mathbf{L}_c \quad , \quad \mathbf{C}=\mathbf{C}_c \tag{3.31}$$

A convenient way of calculating these matrices is shown in appendix C.

To convert the state space description into standard form, the two cases $l=1$ and $l>1$ are dealt with separately. For $l=1$ the general state space description becomes

$$\mathbf{x}(k+1)=\mathbf{A}\mathbf{x}(k)+\mathbf{B}_1 u(k-1)+\mathbf{B}_2 u(k)+\mathbf{L}w(k) \quad , \quad y(k)=\mathbf{C}\mathbf{x}(k) \quad . \tag{3.32}$$

To eliminate the $u(k-1)$ input the state is augmented with a new variable $x_{n+1}(k)=u(k-1)$ so that

$$\begin{bmatrix} \mathbf{x}(k+1) \\ x_{n+1}(k+1) \end{bmatrix}=\begin{bmatrix} \mathbf{A} & \mathbf{B}_1 \\ \mathbf{0} & 0 \end{bmatrix}\begin{bmatrix} \mathbf{x}(k) \\ x_{n+1}(k) \end{bmatrix}+\begin{bmatrix} \mathbf{B}_2 \\ 1 \end{bmatrix}u(k)+\begin{bmatrix} \mathbf{L} \\ 0 \end{bmatrix}w(k) \quad , \quad y(k)=\begin{bmatrix} \mathbf{C} & 0 \end{bmatrix}\begin{bmatrix} \mathbf{x}(k) \\ x_{n+1}(k) \end{bmatrix} \quad . \tag{3.33}$$

Note that the state matrix of this state space description is singular. The other case $l>1$ is treated in a similar manner by augmenting the state with l new state vectors given by

$$x_{n+1}(k)=u(k-l) \quad , \quad x_{n+2}(k)=u(k-l+1) \quad , \quad \cdots \quad , \quad x_{n+l}(k)=u(k-1) \tag{3.34}$$

so that the standard form state space description becomes

$$\begin{bmatrix} \mathbf{x}(k+1) \\ x_{n+1}(k+1) \\ x_{n+2}(k+1) \\ x_{n+3}(k+1) \\ \vdots \\ x_{n+l}(k+1) \end{bmatrix}=\begin{bmatrix} \mathbf{A} & \mathbf{B}_1 & \mathbf{B}_2 & 0 & \cdots & 0 \\ \mathbf{0} & 0 & 1 & 0 & \cdots & 0 \\ \mathbf{0} & 0 & 0 & 1 & \cdots & 0 \\ \mathbf{0} & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & & 1 \\ \mathbf{0} & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}\begin{bmatrix} \mathbf{x}(k) \\ x_{n+1}(k) \\ x_{n+2}(k) \\ x_{n+3}(k) \\ \vdots \\ x_{n+l}(k) \end{bmatrix}+\begin{bmatrix} \mathbf{0} \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}u(k)+\begin{bmatrix} \mathbf{L} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}w(k) \quad , \tag{3.35}$$

and the output is given by

$$y(k)=\begin{bmatrix} \mathbf{C} & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}\begin{bmatrix} \mathbf{x}(k) & x_{n+1}(k) & x_{n+2}(k) & x_{n+3}(k) & \cdots & x_{n+l}(k) \end{bmatrix}^T \quad . \tag{3.36}$$

Again the state matrix becomes singular. The Mathworks MATLAB Control System Toolbox [18] provides an M-file c2dt.m that implements the zero-order hold discretisation with a pure control delay. The control delay is divided into an integer and fractional contribution, the continuous time matrices are converted to discrete time equivalents using (3.31) and the standard state space form is found using (3.33) or (3.35) and (3.36).

4. Simulation results

4.1 Introduction

A number of simulations were carried out to compare the use of the FIR Wiener IMC (from now on referred to simply as IMC) and state space LQG (from now on referred to simply as LQG) feedback control approach to actively suppress the vibration of a highly resonant structure. IMC and LQG feedback controllers were designed from a model of a lightly damped rectangular aluminium plate $(\zeta = 0.025)$ with dimensions 430x360x1.5 mm., fitted with a piezoceramic control actuator and a point velocity sensor. The primary disturbance excitation was modelled as a point force driven by white noise, and the feedback controllers were designed to control the piezo actuator to minimise the mean square out-of-plane velocity measured by the sensor. A detailed list of the model parameters can be found in appendix D.1 and D.2.

4.2 Plant configurations

Two different plant configurations were used in the simulations. In the first configuration the sensor and control actuator were located close to each other to minimise phase accumulation in the plant response. Seven modes were included in the modal model of the plate response. The control actuator was placed near the lower right corner of the panel, and the velocity sensor was placed within the area of the actuator. A sampling rate of 2 kHz and zero-order hold D/A-conversion was assumed in the simulations. A first order lowpass filter with a cut-off frequency of 600 Hz, bilinearly transformed into digital form, was used to filter the sensor output to decrease aliasing effects.

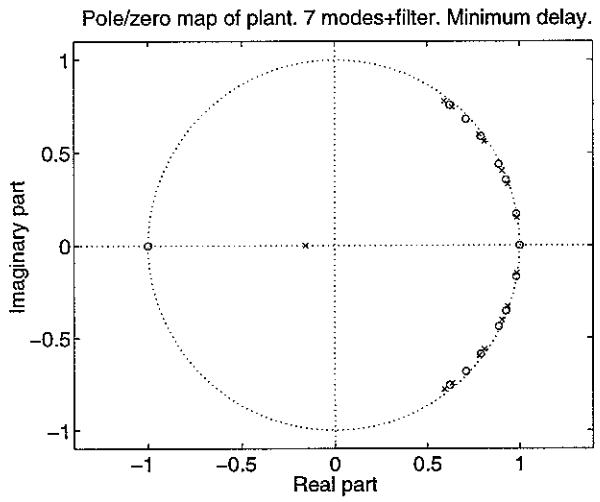

Figure 4.1. Map of poles (cross) and zeros (circle) of the minimum delay plant. The pole and zero with negative real parts belong to the LP filter.

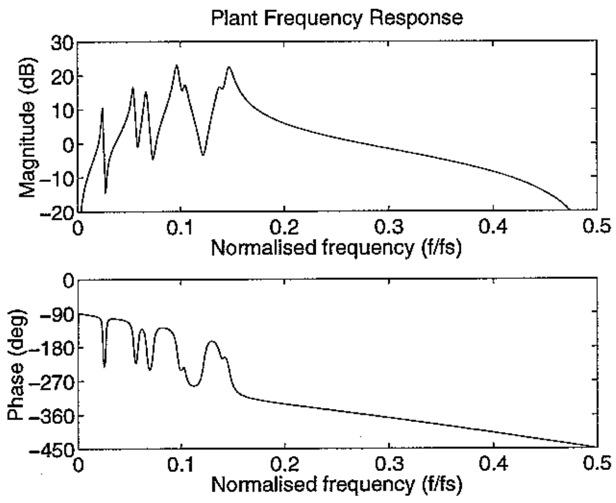

Figure 4.2. Frequency response (gain and phase) of minimum delay plant. Note that delay is the only source of phase accumulation.

Figure 4.1 shows a plot of the poles and zeros of the equivalent discrete time plant transfer function. The transfer function has a total of 15 poles and 14 zeros. Seven complex pole pairs correspond to the seven modes included in the plant model while the remaining real pole at (-0.15,0) belongs to the lowpass filter. Note that the poles corresponding to the structural modes are all very close to the unit circle, indicating a very lightly damped system. Most of the zeros are also placed very close to the unit circle and two zeros are actually placed on the unit circle (one at (1,0) and one at (-1,0)) corresponding to the plant having zero gain at DC and at the Nyquist frequency. Figure 4.2 shows the corresponding frequency response of the discrete time plant model. The magnitude response shows the resonant behaviour of the plant. The phase response shows that no phase accumulation occurs due to the positioning of the sensor/actuator but that approximately one sample time delay is present (the phase response has an overall linear slope of $d\theta/d\omega \approx \pi/(2\pi 0.5 f_s) = 1/f_s$), of which half a sample delay is due to the zero-order hold circuit [20], [21]. Due to the delay the plant is not strictly minimum phase and will from now on be referred to as the "minimum delay" plant. Note that no processing delay has been included in the plant model at this point.

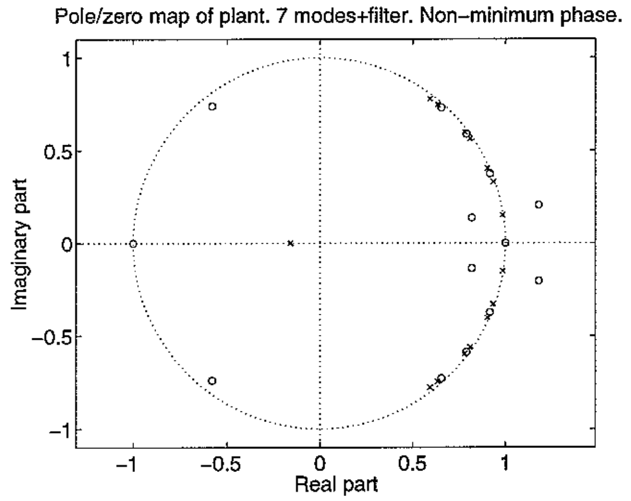

Figure 4.3. Map of poles (cross) and zeros (circle) of the non-minimum phase plant. The pole and zero with negative real parts belong to the LP filter.

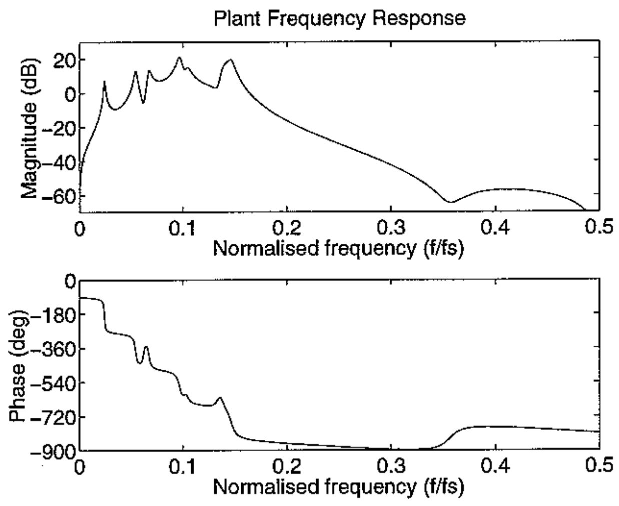

Figure 4.4. Frequency response (gain and phase) of the non-minimum phase plant. Notice the phase accumulation due to the actuator/sensor position.

In the second plant configuration the control actuator was moved away from the sensor to introduce further phase accumulation in the plant response. The control actuator was now placed in the upper left corner of the plate, whereas the position of the sensor was unchanged. Figure 4.3 and Figure 4.4 show the pole/zero plot and the frequency response of this plant configuration. Clearly this plant is non-minimum phase with zeros placed outside the unit circle causing phase accumulation, as seen in the frequency response. The locations of the system poles are of course not changed. Again the system has a zero at (1,0) and (-1,0) corresponding to zero gain at DC and the Nyquist frequency.

4.3 Comparing LQG and IMC controller performance

Feedback controllers were now designed from the plant and disturbance models using the IMC and LQG design approach. The LQG controllers were designed directly from discrete time state space models of plant and disturbance, the resulting controllers being also on state space form. The "current" estimator was used in the controller design. Closed loop sensitivity, controller frequency response etc. was then calculated from the state space description as outlined in chapter 2.

The IMC controllers were designed from the frequency responses of plant and disturbance, which in this case were calculated directly from the plant and disturbance state space models. In a practical situation they could have been obtained otherwise - for instance by direct measurements. Wiener FIR filters were designed in the time domain and their frequency responses were then combined with the plant frequency responses to form the resulting IMC controller frequency responses, closed loop sensitivities etc.

To compare the two types of feedback control the cost functions minimised in the two design approaches had to be identical. The two controller types were initially compared in the "minimum variance limit", that is with a very small effort weight (10^{-9}) in the cost function. The same level of sensor noise ($Q_v = 10^{-9}$) and disturbance noise ($Q_w = 1$) was assumed in both designs, and a very long Wiener filter (700 taps) was used in the IMC design to avoid the effect of filter truncation.

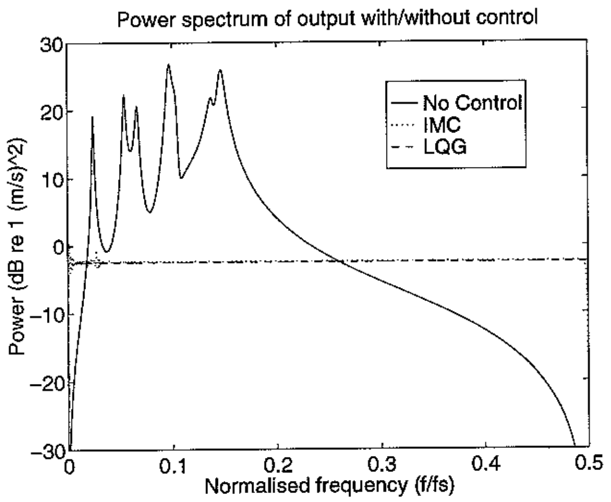

Figure 4.5. Power spectrum of output without control (solid) and with LQG (dashed and IMC (dotted) control. Minimum delay plant.

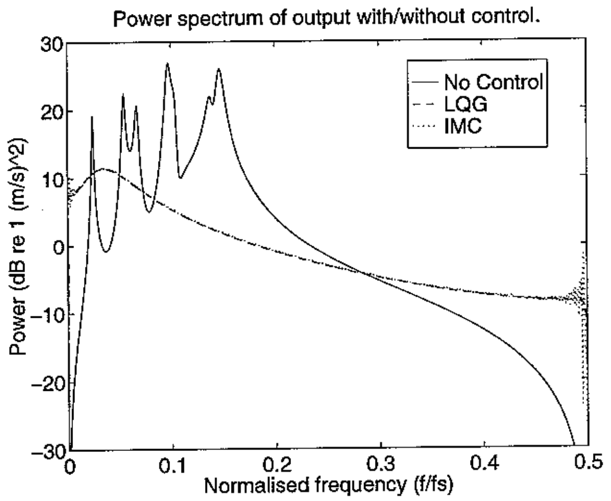

Figure 4.6. Power spectrum of output without control (solid) and with LQG (dashed) and IMC (dotted) control. Non-minimum phase plant.

Figure 4.5 shows the performance of the two types of control when tested with the minimum delay plant system model. The figure shows the power spectrum of the system output without control (disturbance only) and with IMC and LQG feedback control. In this plot a small amount of ripple is visible on the output power spectrum with IMC control. This is probably due to the finite length of the Wiener filter used in the IMC controller. Apart from this the performance of the controllers appears identical. The output power spectrum with control is totally flat throughout the entire frequency range, indicating that only a totally unpredictable part of the disturbance is left on the system output. The fact that the attenuation is not infinite is due to the delay in the discrete time plant equivalent. Attenuations of up to 30 dB are obtained in the resonant frequency range whereas disturbance enhancement is seen at low and high frequencies.

Controllers were also designed from the non-minimum phase plant model. A performance plot is shown in Figure 4.6. Ripple is now clearly visible in the frequency ranges 0-50 Hz and 950-1000 Hz, whereas the performance of the two controllers is nearly identical from 50 Hz to 950 Hz. The output power spectrum with control is no longer flat, due to the plant being non-minimum phase and therefore not fully invertible, and the obtained disturbance reductions are of course smaller than in the minimum delay plant case. Although the closed loop is stable, the controller itself is unstable in this case.

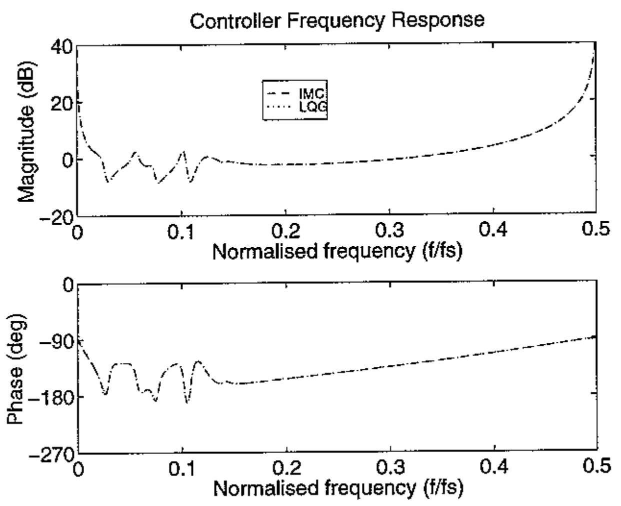

Figure 4.7. Controller frequency responses of LQG (dashed) and IMC (dotted) controllers, designed with identical effort weight and noise parameters. Minimum delay plant.

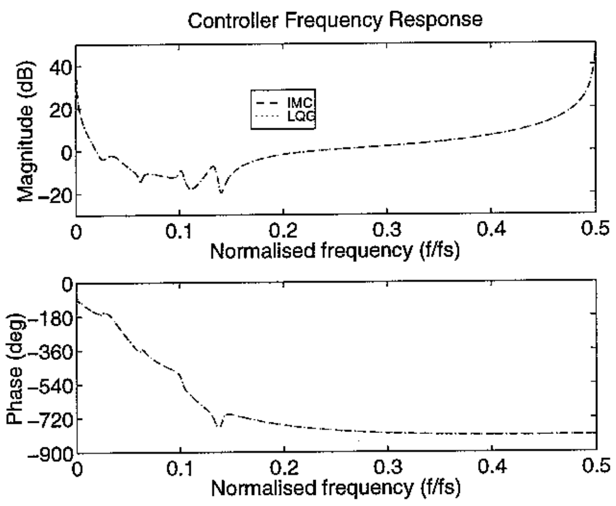

Figure 4.8. Controller frequency responses of LQG (dashed) and IMC (dotted) controllers, designed with identical effort weight and noise parameters. Non-minimum phase plant.

To emphasise the fact that the two different design methods yield controllers with identical input/output properties, the frequency responses of controllers designed with identical effort weights (ρ=1) and noise parameters (Q_v=10^{-9}, Q_w=1) using the two different methods were calculated. The effort weight was chosen so that the controller for the non-minimum phase plant was now stable. If this was not the case, it would not be possible to calculate the controller frequency response. Figure 4.7. shows the frequency responses of the controllers designed from the minimum delay plant. The frequency responses are identical throughout the entire frequency range within the precision of the plot. Frequency responses of the controllers for the non-minimum phase plant are shown in Figure 4.8. Again the responses are identical.

4.4 Stability robustness

Using very little or no effort weighting and assuming that very little or no sensor noise is present in the control loop when minimising the mean square value of the plant output is a very "aggressive" form of control, which can lead to controllers with very little robustness to plant changes. The stability robustness of the feedback controllers can be evaluated from the complementary sensitivity and the open loop gain.

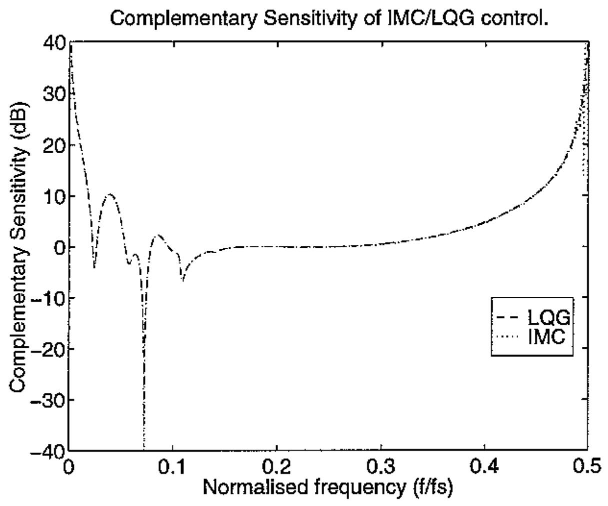

Figure 4.9. Complementary sensitivity of IMC (dot) and LQG (dash) control. Non-minimum phase plant.

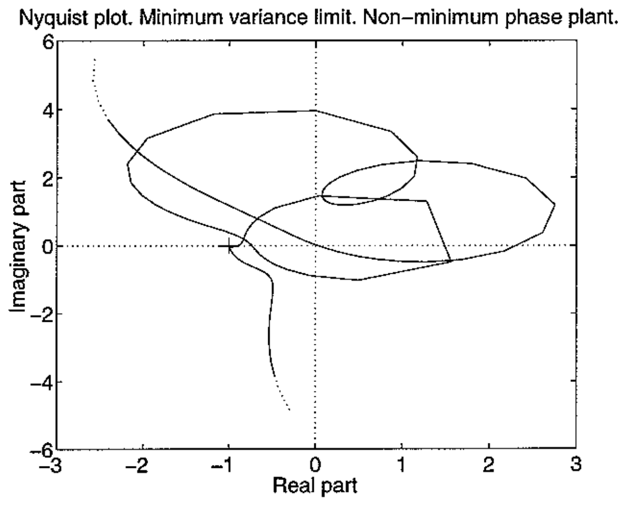

Figure 4.10. Nyquist plot of open loop gain for non-minimum phase plant. The plot must not encircle the (-1,0) point if the closed-loop system is to be stable.

Figure 4.9 shows a plot of the complementary sensitivity of the IMC and LQG controllers for the non-minimum phase plant. In the low and high frequency range the complementary sensitivity very large (greater than 30 or approximately 30 dB), indicating that very small plant changes (less than 3%) in these frequency regions can lead to instability. In the resonant frequency range the complementary sensitivity is smaller. Figure 4.10 shows a Nyquist plot of the corresponding open loop gain $G_p\left(e^{-j\omega T}\right)H\left(e^{-j\omega T}\right)$. The open loop gain must not encircle the (-1,0) point if the closed-loop system is to be stable. It is clear that the open loop gain is very close to the (-1,0) point in parts of the frequency range (mainly at low and high frequencies) and that the closed-loop system is thus very close to the stability limit. All in all this indicates a control strategy with very little stability robustness, and ways of modifying the designs to increase the robustness are therefore desirable.

4.5 Control effort weighting

Including a control effort weighting term in the cost function puts a penalty on large control signals when minimising the mean square value of the output. This corresponds to limiting the controller gain.

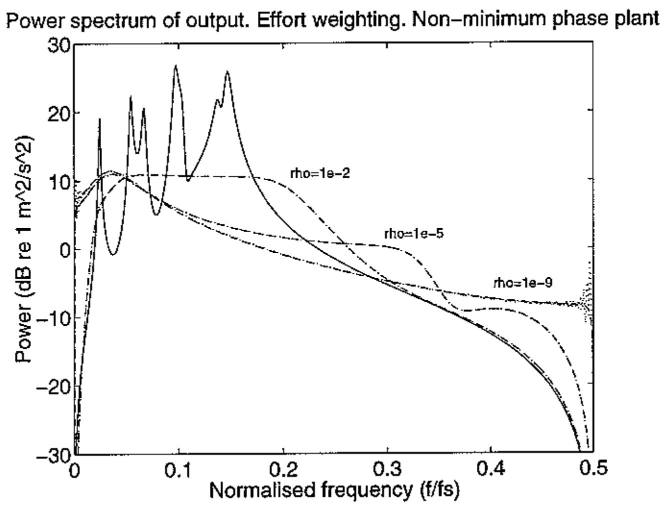

Figure 4.11. Power spectrum of output without control (solid) and with LQG (dash) and IMC (dot) control using effort weight parameters 1e-9, 1e-5 and 1e-2. Non-minimum phase plant.

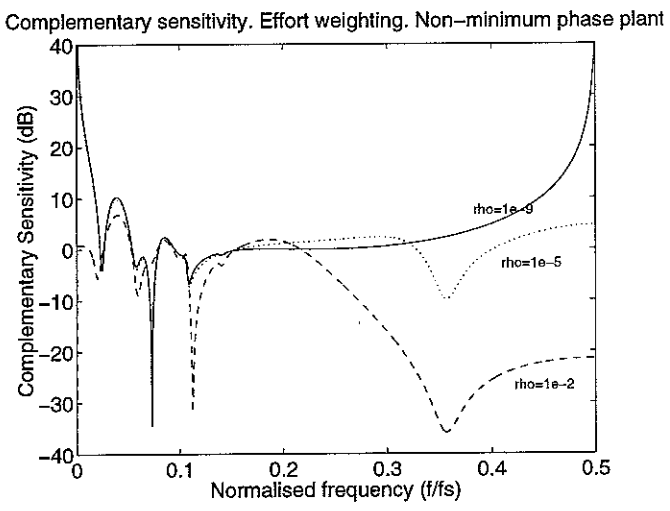

Figure 4.12. Complementary sensitivity with effort weight parameter 1e-9 (solid), 1e-5 (dot) and 1e-2 (dash). Non-minimum phase plant. Only LQG results are shown.

Figure 4.11 shows the performance of the LQG and IMC control with the non-minimum phase plant for effort weighting parameter (ρ) values of 10^{-9} , 10^{-5} and 10^{-2} . The performance in the resonant frequency region clearly deteriorates as ρ becomes larger than 10^{-5} . The disturbance enhancement at high and low frequencies decreases, however. Increasing ρ also has a positive effect on the stability robustness of the controller. Figure 4.12 shows the complementary sensitivity for increasing values of ρ. The complementary sensitivity is clearly brought down in the low and high frequency limit, but is also limited in the resonant frequency range, indicating an overall increased stability robustness of the controller.

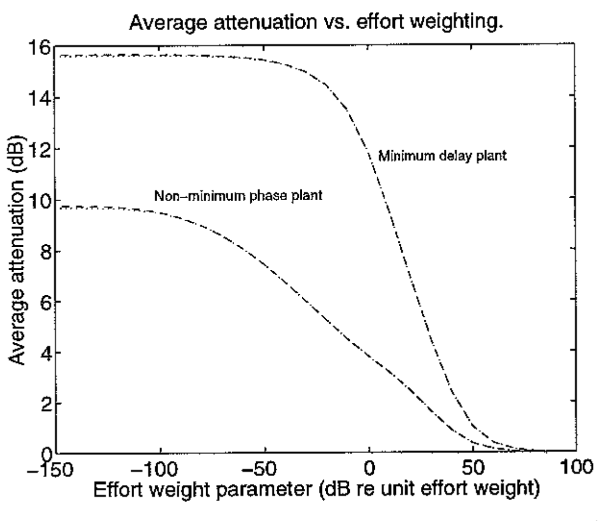

Figure 4.13. Average attenuation as a function of the effort weighting parameter for IMC (dot) and LQG (dash) control. Results for both plants are shown.

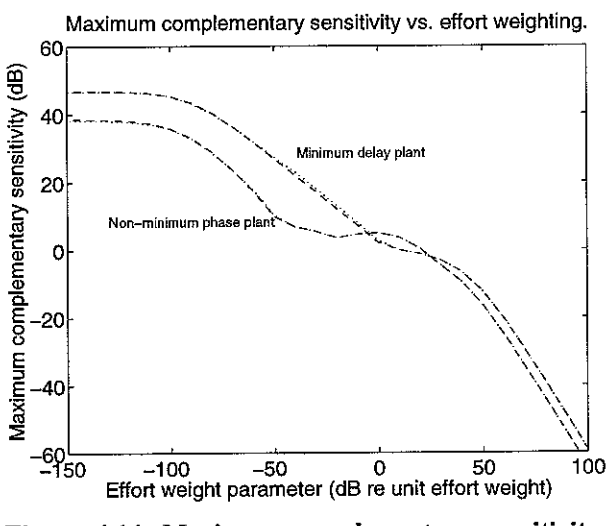

Figure 4.14. Maximum complementary sensitivity as a function of the effort weight parameter for IMC (dot) and LQG (dash) control. Results for both plants are shown.

To further illustrate the effect of effort weighting on the controller performance, the average power attenuation as a function of the effort weighting parameter is plotted in Figure 4.13. The plot shows the expected correspondence for both the minimum delay plant and the non-minimum phase plant. Note that the two controller designs yield identical results. Note also that in the minimum delay plant case the average attenuation drops from 15.5 dB to 0.5 dB as the effort weighting parameter is increased from 10^{-3} to 10^{3} (-60 dB to 60 dB re 1.0), whereas for the non-minimum phase plant the drop from 9.7 dB to 0.5 dB happens over the wider parameter range 10^{-6} to 10^{3} (-120 to 60 dB re 1.0).

The effect of effort weighting on controller robustness is summarised in Figure 4.14. This plot shows the maximum value of the complementary sensitivity function over the whole frequency range as a function of the effort weighting parameter for both plants and for both controller designs. As could be expected the maximal complementary sensitivity decreases as the effort weighting is increased, indicating increased robustness with increased effort weighting. It is worth noting that for a given amount of effort weighting the controller for the minimum delay plant is generally less robust than the controller for the non-minimum phase plant.

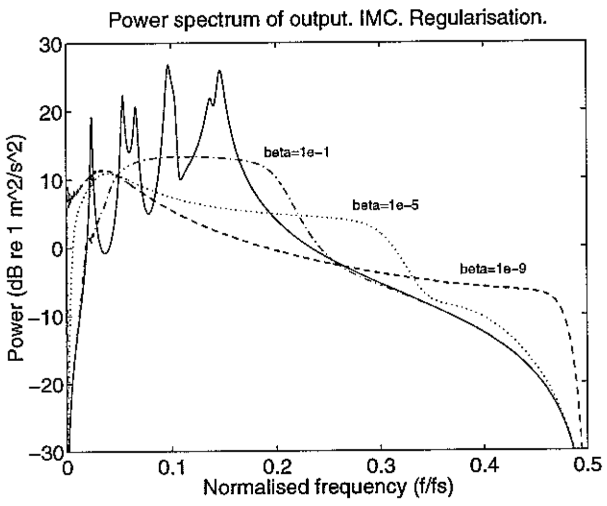

Figure 4.15. Performance of IMC controller with regularisation parameter 1e-9 (dash), 1e-5 (dot) and 1e-1 (dash/dot). Non-minimum phase plant.

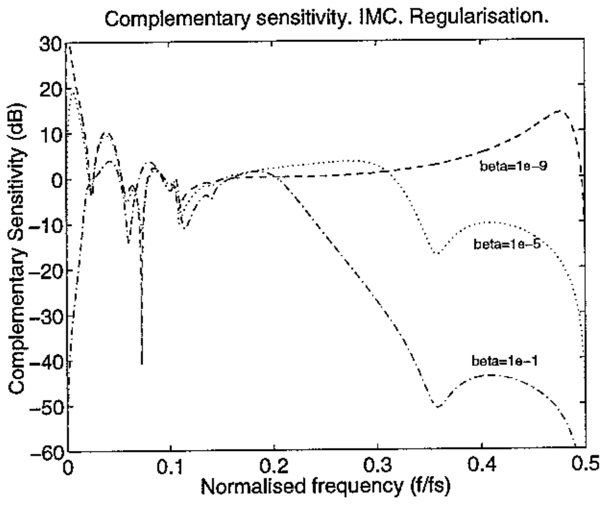

Figure 4.16. Complementary sensitivity of IMC control with regularisation parameter 1e-9 (dash), 1e-5 (dot) and 1e-1 (dash/dot). Non-minimum phase plant.

Including a regularisation term when calculating the IMC Wiener filter has an effect similar to the effort weighting term, since the regularisation term imposes a limit on the magnitude of the Wiener filter coefficients. Figure 4.15 shows the performance of the IMC controller with the non-minimum phase plant for β-values of 10^{-9} , 10^{-5} and 10^{-1}. The effect of including the regularisation term is very similar to that of including the effort term in the LQG design. The effect on the complementary sensitivity is also nearly the same as in the LQG case as seen in Figure 4.16.

4.6 The assumed ratio between sensor noise and disturbance noise as a design parameter

The assumed ratio between sensor noise and disturbance noise in the feedback loop can be looked upon as a design parameter in both the IMC and the LQG design method. Controllers designed for systems with a high level of sensor noise will have a lower gain than controllers designed for systems with negligible sensor noise, since sensor noise will be amplified by the controller and affect performance if the gain is too high, but because of the lower gain the controllers will also be more robust to plant changes. Similarly controllers designed for systems with high levels of disturbance noise will have high gains in order to suppress the effect of the disturbance on the output. Increasing the assumed level of disturbance noise in the controller design will therefore lead to controllers with high gains resulting in good performance but poor robustness properties.

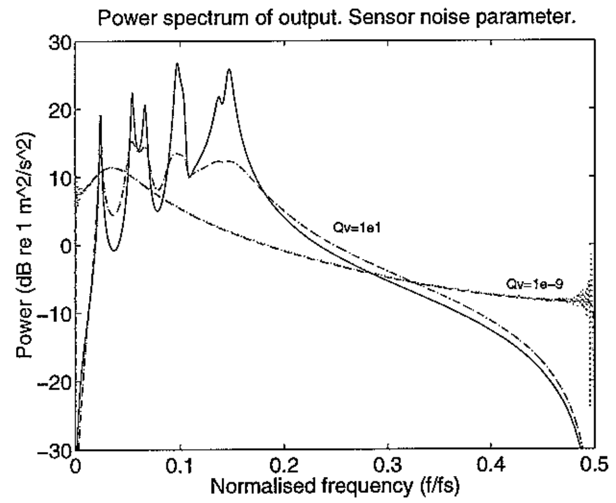

Figure 4.17. Performance of LQG (dash) and IMC (dot) controllers for high and low assumed levels of sensor noise. Non-minimum phase plant. Disturbance noise intensity is Q_w=1.

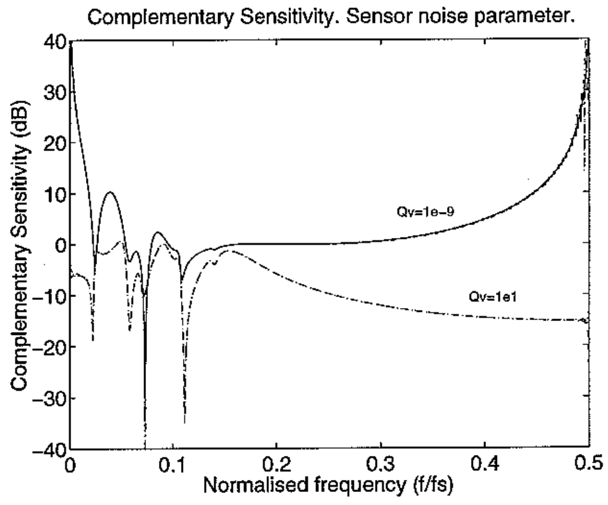

Figure 4.18. Complementary sensitivity for IMC and LQG control with a very low assumed level of sensor noise (solid) and for LQG (dashed) and IMC (dotted) with high assumed level of sensor noise.

Figure 4.17 shows the effect on controller performance of assuming different sensor noise levels in the IMC and LQG designs for a fixed assumed level of disturbance noise ($Q_w = 1$). The sensor noise was assumed to be white (i.e. $|G_s| = 1$). As could be expected, assuming the same noise levels in the two design methods resulted in nearly identical performance by the LQG and IMC controller. Assuming a high level of sensor noise in the design clearly led to reduced performance. However the stability robustness of the controllers was increased as can be seen from the complementary sensitivity plot in Figure 4.18.

The effect on controller performance of varying the assumed level of sensor noise is summarised in Figure 4.19. The figure shows the average disturbance attenuation as a function of the assumed level of sensor noise for both the IMC and LQG controller design and for both plants. The results for the two design methods are again identical. As could be expected the average attenuation decreased as the assumed level of sensor noise was increased. By comparing with Figure 4.13 it is seen that the correspondence between average attenuation and assumed sensor noise level is very similar to the correspondence between average attenuation and effort weight level for the minimum delay plant. However in the non-minimum phase case the slope of the curves in Figure 4.13 and Figure 4.19 is quite different.

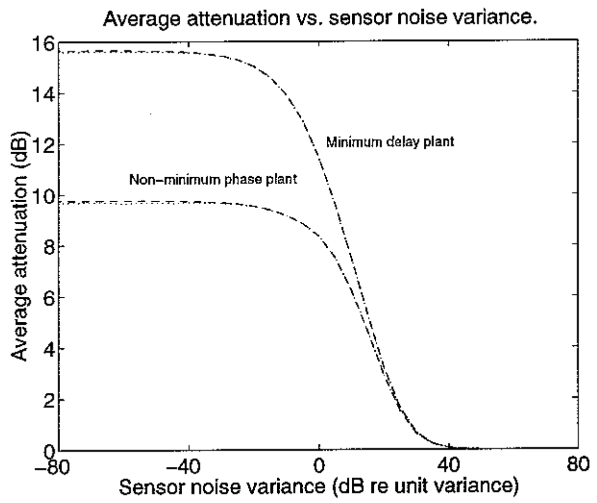

Figure 4.19. Average attenuation as a function of the assumed level of sensor noise for IMC (dot) and LQG (dash) control. Results for both plants are shown.

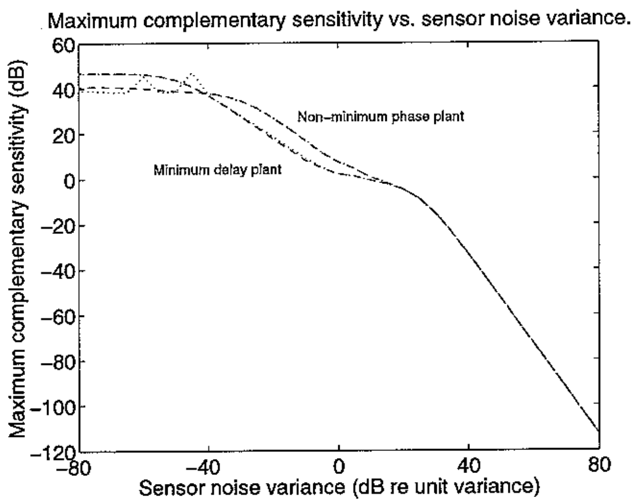

Figure 4.20. Maximum complementary sensitivity as a function of the assumed level of sensor noise for IMC (dot) and LQG (dash) control. Results for both plants are shown.

The effect on controller robustness is shown in Figure 4.20, where the maximum complementary sensitivity is plotted against the assumed level of sensor noise for both controller designs and both plants. The results for the IMC and LQG controllers are almost identical. The differences in the low sensor noise limit are due to numerical problems. Not surprisingly, the maximum complementary sensitivity decreases as the assumed level of sensor noise is increased, indicating overall increased robustness with increased assumed sensor noise level. The results for the two plants are closer to each other than in Figure 4.14.

To further compare the effect on performance and robustness of varying the effort weighting and the assumed level of sensor noise, the average attenuation is plotted against the maximal allowed multiplicative plant uncertainty $B_{\max} = 1/T_{\max}$, where $T_{\max}$ is the maximum complementary sensitivity (a high value of $B_{\max}$ indicates a high level of robustness). The result is seen in Figure 4.21. It is clear that the effect on performance and robustness of varying the effort weighting is nearly the same as the effect of varying the assumed level of sensor noise in the case of the minimum delay plant. By increasing the effort weighting or assumed level of sensor noise the robustness could be increased to some extent without reducing the performance significantly. However, above a certain level of effort weighting or assumed sensor noise the performance deteriorated very quickly with increased robustness. In the case of the non-minimum phase plant the effect of varying the two different parameters was not entirely the same. By varying the assumed level of sensor noise a slightly higher level of robustness could be achieved for a given disturbance attenuation than by varying the effort weighting. Above a certain level of effort weighting or sensor noise the performance again deteriorated rapidly with increased robustness.

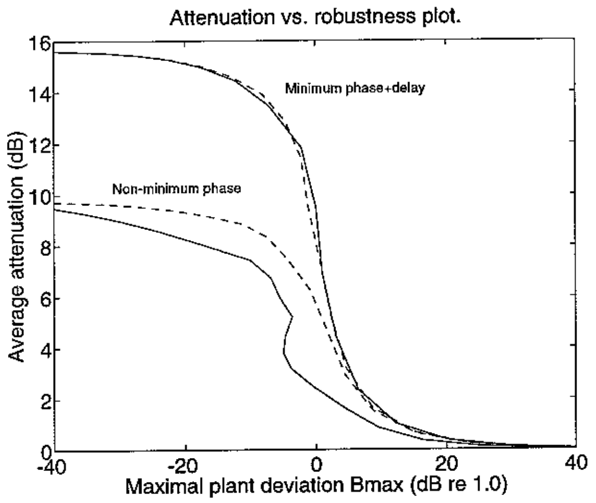

Figure 4.21. Plot of average attenuation against maximal allowed multiplicative plant uncertainty $B_{max} = 1/T_{max}$ for different levels of assumed sensor noise (dashed) and effort weighting (solid). Results for both plants are shown.

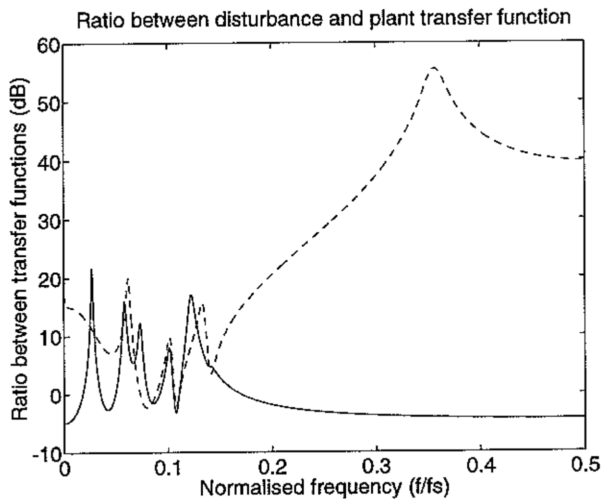

Figure 4.22. Ratio between disturbance noise filter transfer function and plant transfer function for the minimum delay plant (solid) and for the non-minimum phase plant (dashed).

The fact that the effect of varying the effort weighting was very similar to the effect of varying the assumed level of sensor noise in the case of the minimum delay plant but not in the case of the non-minimum phase plant can be understood via the frequency domain interpretation of the cost function described in chapter two. The frequency domain version of the cost function with effort weighting (equation 2.32) includes the complementary sensitivity weighted by the ratio between the disturbance filter transfer function and the plant transfer function, whereas in the "assumed sensor noise" case it includes the complementary sensitivity weighted by the transfer function of the sensor noise filter only (equation 2.33). The difference between the two cost functions thus depends on the disturbance filter and plant frequency response.

The ratio between the disturbance filter and plant transfer function is plotted in Figure 4.22 for both plants. The sensor noise filter transfer function is a constant since the sensor noise is assumed to be white. The ratio of frequency responses are clearly complicated functions of frequency, indicating that the use of effort weighting corresponds to limiting the complementary sensitivity in certain frequency bands. The use of an assumed white sensor noise, however, corresponds to limiting complementary sensitivity equally at all frequencies. The large peak in the ratio of frequency responses for the non-minimum phase plant explains why the "assumed sensor noise method" gave the best trade-off between attenuation and robustness in this case. The peak for the non-minimum phase plant in Figure 4.22 has the effect of over-emphasising the complementary sensitivity function for normalised frequencies above 0.5, while allowing quite large complementary sensitivities in other (more crucial) frequency bands without these adding significantly to the cost function. The difference between the cost functions is also evident from comparing Figure 4.12 and Figure 4.18. The effect of the peak on the complementary sensitivity is clear in Figure 4.12 where it results in a large dip at frequencies around $0.35f_s$. The same effect is not visible in Figure 4.18, since the complementary sensitivity was here weighted equally at all frequencies.

4.7 Delays

The IMC and LQG controller designs examined so far assumed that past data up to and including the k'th system output sample could be used to produce the k'th control signal sample, i.e. that no computation delay was present in the control loop. The IMC controllers were designed using Wiener filters having a direct link, and thus clearly assumed that control could be invoked without delay. When designing the LQG controllers a choice was made between using a "predictor" or "current" type state estimator. To be able to compare controllers designed using the two different methods, the LQG controllers had to be designed using the "current" type state estimator, so that this type of controller also made use of the k'th output signal sample to produce the k'th control signal sample. This was done in all the previous simulations, and apart from numerical errors the controllers resulting from these two design methods were identical.

In most practical control implementations the control signal sample found from past data up to and including the k'th output signal sample cannot be invoked until at least one sample later, to allow for finite computation time. The effect of this controller time delay can be investigated by including a pure control time delay in the plant model as described in chapter 2, and by designing and testing the controllers using this plant model. This was done using the minimum phase plant model with zero, one, two and three samples delay in the control path.

The performance of IMC and LQG controllers with such delays is shown in Figure 4.23. The controllers were designed with effort weights of $\rho = 10^{-9}$ and noise levels $Q_v = 10^{-9}$ and $Q_w = 1$. Again the two design methods yielded identical controllers, apart from numerical errors in the IMC design, which is visible as ripple in the corresponding power spectrum. It is clear that a one sample control delay has affected the controller performance considerably. Approximately 10 dB of disturbance attenuation is lost throughout the resonant frequency range. Note that the effect of further delays is not as considerable as the effect of "loosing the first sample".

Figure 4.24 shows the average attenuation as a function of the pure time delay in the control path. Results are shown for both the minimum delay plant and the non-minimum phase plant. Again it is clear how a delay of one sample in the control loop affects the performance considerably, especially in the minimum delay plant case where attenuation drops rapidly from 15.5 dB to 9.5 dB. Performance is also affected in the non-minimum phase plant case, but not as much as with the minimum delay plant.

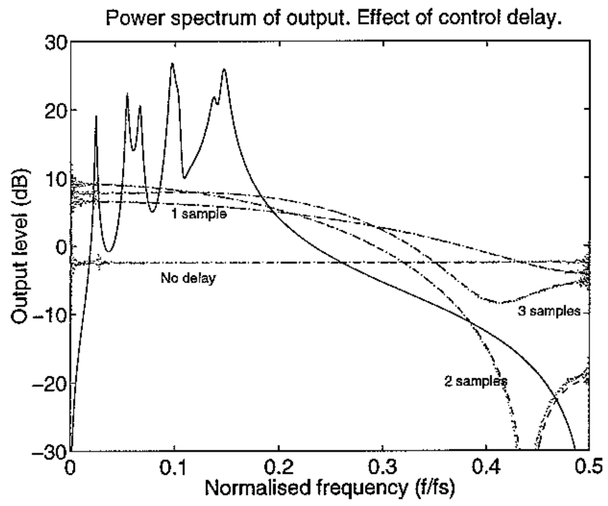

Figure 4.23. Performance IMC (dotted) and LQG current estimator (dashed) controllers for various pure time delays in the control path. Minimum delay plant.

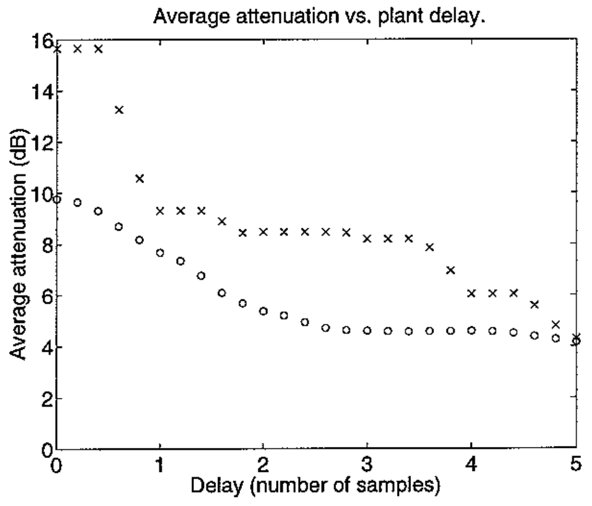

Figure 4.24. Average attenuation as a function of pure time delay in the control path. Results shown for the minimum delay plant (crosses) and the non-minimum phase plant (circles) LQG results only.

It is worth noticing that designing the LQG controller with a "predictor" type state estimator and no pure computation delay included in the plant model results in a controller that is identical to the one obtained by designing with a "current" estimator" and one sample computation delay in the plant. This is not surprising since this is merely two different ways of accounting for the same computation delay in the controller design.

4.8 Reduced order LQG controllers

The IMC controllers investigated in the previous simulations made use of FIR filters with several hundred coefficients, whereas the LQG controllers were implemented with considerably fewer coefficients, due to their IIR nature. It was still of interest to examine whether the order of the LQG controllers could be further reduced by pole/zero cancellation, and what effect this order reduction had on the controller performance and robustness.

A pole/zero map of a full order LQG controller designed from the minimum delay plant model is shown in Figure 4.25. The controller was designed using very little effort weighting ($\rho = 10^{-9}$) but with a relatively high assumed level of sensor noise ($Q_v/Q_w = 0.6/1 = 0.6$) in order to introduce a certain amount of robustness into the design. This was done in order to prevent the control loop from becoming unstable when inevitable uncertainties were introduced via the pole/zero cancellations in the controller. The chosen ratio between assumed sensor and disturbance noise resulted in a maximal complementary sensitivity $T_{\max}$ of 3 dB re 1.0 ≈ 1.41 (see Figure 4.20), thus allowing unstructured multiplicative uncertainty of 70% over the whole frequency range while maintaining closed loop stability. The corresponding average attenuation of the full order controller was approximately 12 dB (see Figure 4.19). A number of the controller poles and zeros were gathered in pairs as marked on the figure. The pairs 1 and 2a/2b were very close to cancelling each other. The pairs 3a/3b and 4a/4b were a bit further from cancelling each other, but still very close and so on. Reduced order LQG controllers could thus be found by removing progressive numbers of the pole/zero pairs that were close together from the controller transfer function.

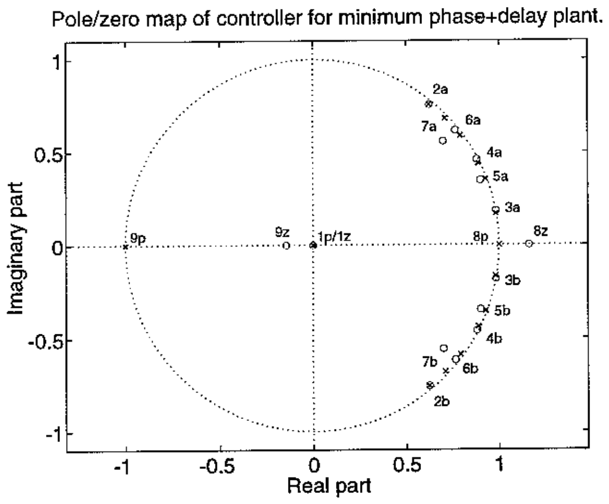

Figure 4.25. Pole/zero plot of LQG controller for the minimum delay plant. Poles are marked by (x), zeros by (o).

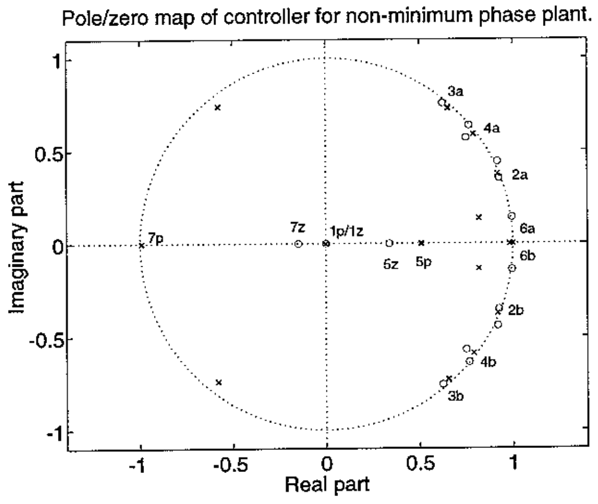

Figure 4.26. Pole/zero plot of LQG controller for the non-minimum phase plant. Poles are marked by (x), zeros by (o).

Figure 4.26 shows the corresponding pole/zero map of a full order LQG controller designed for the non-minimum phase plant. In this design a sensor/disturbance noise ratio of ($Q_v/Q_w = 5/1 = 5$) was used to again obtain a maximum complementary sensitivity of 3 dB (see Figure 4.20). Although a few poles and zeros were close to cancelling each other there seemed to be fewer obvious candidates for pole/zero cancellation than in the case of the minimum delay plant controller.

Figure 4.27 shows the performance of the full and reduced order controllers for both plants. For each cancellation the controller gain was re-adjusted to maintain the maximum complementary sensitivity of 3 dB. If this was not done, cancelling a pole/pair could sometimes lead to increased attenuation but also to reduced robustness. The poles and zeros where cancelled in the order indicated in Figure 4.25 and Figure 4.26.

The results for the minimum delay plant controller (indicated with crosses) show that in this case the order of the controller could be reduced quite drastically without significantly affecting the control performance and while maintaining the fixed robust stability margin. The order of the controller could be reduced from fifteen to six without reducing the average attenuation and controllers with eight or ten poles actually performed better than the full order controller. This somewhat surprising result is probably due to the procedure of readjusting the gain to keep a fixed maximum complementary sensitivity. This procedure corresponds to putting a constraint on the maximal value of the complementary sensitivity when minimising the squared plant output, as opposed to putting a constraint on the squared average of the complementary sensitivity as is done in the full order LQG design (with assumed sensor noise as the robustness parameter). Further reducing the controller order from six down to one resulted in a loss of attenuation of approximately 1 dB. A zero order controller corresponds to a simple gain and it is interesting to note that approximately 6 dB of attenuation along with a maximum complementary sensitivity of 3 dB could be achieved using this the simplest of all feedback controllers. The limit to the amount of attenuation that could be achieved with this controller while maintaining a certain robustness was of course set by the delay in the discrete time plant model. If the actuator and sensor were collocated and no delay was present, the inverse plant transfer function would be stable and thus the closed loop would be stable for all gains [22]. In this case perfect control would be possible with "infinite gain".

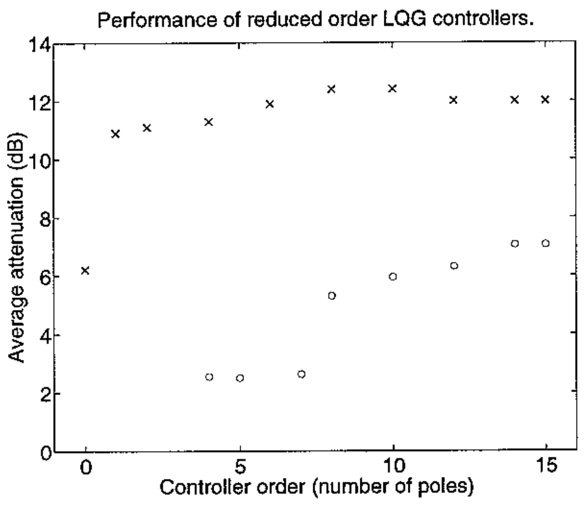

Figure 4.27. Average attenuation vs. controller order for reduced order LQG controllers. Results are shown for the minimum delay plant (x) and the non-minimum phase plant (o).

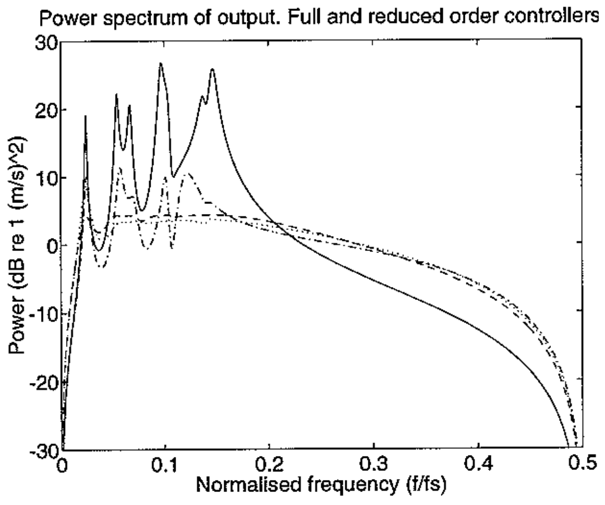

Figure 4.28. Performance of reduced order LQG controllers for the minimum delay plant. Results shown for controllers with 15 (dashed), 10 (dotted) and 1 (dash-dotted) pole(s) Minimum delay plant.

Figure 4.28 shows the output power spectrum without control and with control using the full order controller and reduced order controllers with ten poles and one pole. The effect of cancelling all but one pole and zero in the controller is seen as ripple in the otherwise rather flat power spectrum of the output with control.

The order of the controller for the non-minimum phase plant could not be reduced in the same way without significantly affecting the performance, as shown by open circles in Figure 4.30. The redundant pole/zero pair at (0,0) which was also present in the controller for the minimum delay plant could of course be removed without affecting the control performance at all, but further pole/zero cancellations affected the performance in a negative way. Nevertheless the controller order could be reduced from fifteen to ten without loosing more than 1 dB attenuation. Further reductions of the controller order led to serious deterioration of the performance and stable controllers of orders less than four could not be found by pole/zero cancellation.

The above results show that despite of the IIR nature of the optimal state space LQG controller, reduced order controllers could, in some cases, be found that performed as well as or better than their full order counterparts. Thus the state space LQG design method does not always result in controllers that are optimal with respect to controller order even if the model from which they are designed is perfect, as in this case. The state space LQG design method is based on a model of plant and disturbance and results in controllers of roughly (in this case exactly) the same order as the plant/disturbance model. Usually it is reasonable to attempt to reduce the order of controller, if this can be done without significantly affecting the performance and robustness of the resulting design [6],[7]. This order reduction can be carried out in a number of ways. The controller can be designed from a reduced order model of plant and disturbance. The major drawback of this method is that uncertainties are introduced early in the design, and dynamics that are not accounted for in the model can in the end render the resulting closed loop system unstable [6],[7]. Another method is to formulate the controller design as a quadratic optimisation problem with an order constraint and a closed loop stability constraint. This method can lead to very complex optimisation problems and to difficulties in obtaining closed loop design goals (for instance robustness) other than the goal of minimising a given cost function [6]. A third method -the method recommended in for instance [6]- is to design a full order LQG controller from an accurate high order model of the plant/disturbance and then approximate this controller by a controller of lower order. This method was used here (in a somewhat ad hoc manner) where reduced order controllers were designed by removing parts from the full order controller by pole/zero cancellation. More structured ways of performing this approximation can be found in [6].

4.9 Frequency response fitted IIR controllers

An alternative way of designing IIR controllers of relatively low orders would be to calculate the frequency response of the desired controller using the IMC Wiener approach and then to fit an IIR filter (polynomial or state space) to this response. This method would bypass the need to fit a model to the plant and disturbance since the IMC Wiener method is based on frequency responses and would postpone the process of fitting to the final stage of the controller design. Since the LQG design method does not guarantee controllers of minimal order anyway, this seems like a reasonable approach. One limitation to the method is that only stable controllers can be designed this way since the desired controller is specified entirely via its frequency response.

The controllers from which the reduced order LQG controllers were found in section 4.8 were both stable and so the described design method could be tested with these controllers. The controller frequency responses were calculated using the IMC Wiener approach and IIR filters of various orders were then fitted to the frequency responses. Only filters with an equal number of poles and zeros were used. The filters were fitted to the responses using a frequency-weighted least squares fit with a stability restriction on the resulting filters. The INVFREQZ algorithm in the MATLAB Signal Processing Toolbox [24] was used to carry out this fitting..

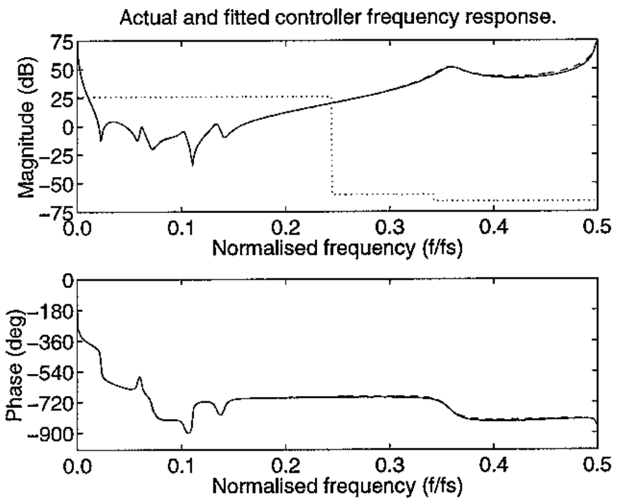

Figure 4.29. Frequency response of controller calculated using the IMC Wiener approach (solid) and of IIR filter fitted to frequency response (dashed). The weight curve is also shown (dotted).

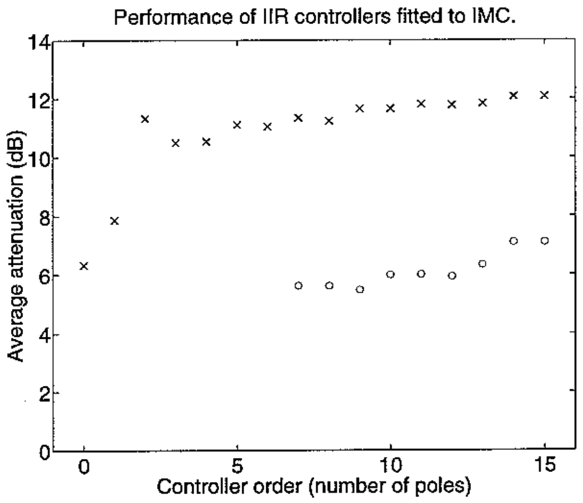

Figure 4.30. Average attenuation vs. filter order for IIR control filters fitted to IMC Wiener controller frequency response. Results for minimum delay (x) and non-minimum phase plant (o)

Figure 4.29 shows the calculated frequency response of the non-minimum phase plant controller along with the response of a fitted filter with 15 poles and 15 zeros (i.e. of the same order as the full order state space LQG controllers). The fit is very close to perfect. The figure also shows the weighting function used in determining the fit. The quality of the fit was found to depend very much on the shape of this weighting function and the weighting function shown in Figure 4.29 proved to work very well. Figure 4.30 shows the performance of the fitted IIR control filters as a function of filter order for both plants. As with the reduced order state space LQG controllers the gain of the fitted controllers was adjusted slightly in each case to maintain a maximum complementary sensitivity of 3 dB. The results are very similar to those presented in Figure 4.27. Fitted IIR control filters of orders down to two were found to perform almost as well as fitted control filters of order fifteen (the same order as the full order LQG controllers) in the minimum delay plant case. IIR control filters of orders down to seven could be determined for the non-minimum phase plant but any attempt to fit a filter of lower order than seven resulted in control filters that were closed loop unstable. It is worth noticing that the 7'th order fitted IIR controller in this case performed better than the corresponding 7'th order reduced LQG controller.

4.10 Reduced number of taps in IMC FIR filters

From an implementation point of view it is also interesting to examine the effect on controller performance of reducing the length of the FIR Wiener filters used in the IMC design.

Figure 4.31 and Figure 4.32 show how the IMC controller performed compared to the full order LQG controller, when Wiener filters with 200 taps were used in the IMC controllers instead of the 700 taps used in the previous simulations. The controllers were designed using the same effort weights and noise parameters as in section 4.3 ($\rho = 10^{-9}$, $Q_v = 10^{-9}$, $Q_w = 1$) and the figures should be compared with Figure 4.5 and Figure 4.6. The effect of the reduced number of taps is clear , especially for the minimum delay plant, where ripple in the output power spectrum of up to 20 dB peak to peak is seen with IMC control.

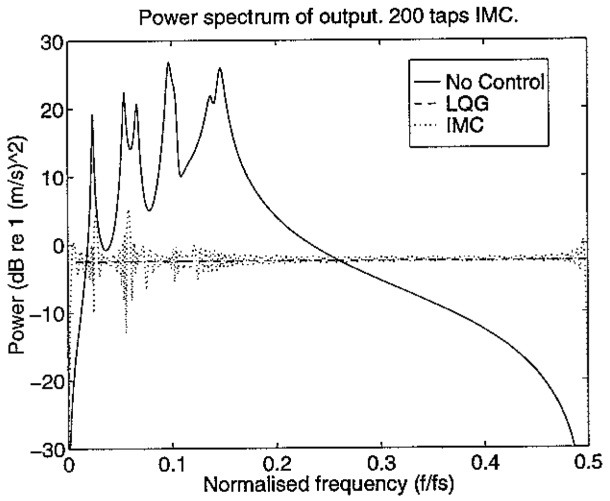

Figure 4.31. Performance of IMC control with 200 taps in the Wiener filter (dotted) compared to performance of LQG control (dashed). Minimum delay plant.

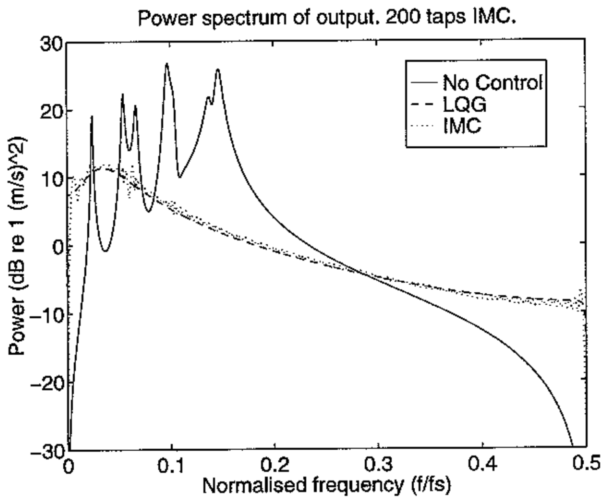

Figure 4.32. Performance of IMC control with 200 taps in the Wiener filter (dotted) compared to LQG control (dashed). Non-minimum phase plant.

The effect of reducing the number of filter coefficients was smaller in the non-minimum phase plant case. The fact that the effect of reducing the number of taps in the Wiener filter was more pronounced with the minimum delay plant can be explained as follows: The task of the Wiener filter in the IMC controller is to "invert the plant" as well as possible so that the control signal produces the "negative disturbance" on the physical output of the system under control. This means that zeros of the plant are recreated as poles by the

Wiener filter and vice versa. What causes problems for the short Wiener filter in the minimum delay plant case is thus the deep zeros of the plant transfer function, since emulating poles with an FIR filter requires a very long filter.

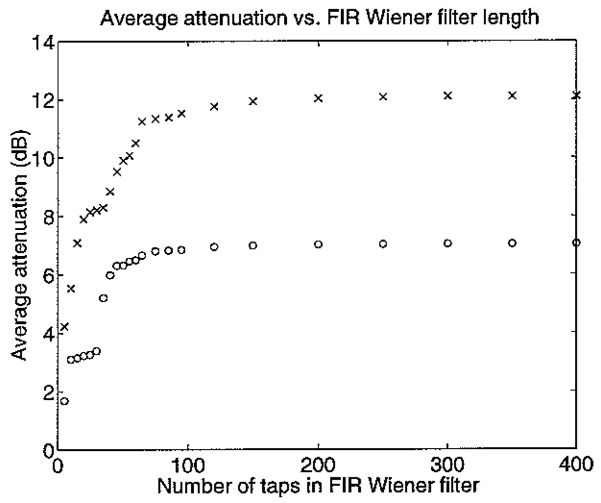

Figure 4.33. Average attenuation as a function of the number of FIR Wiener filter taps. Results for minimum delay plant (x) and non-minimum phase plant (o).

The effect of the Wiener filter length on IMC controller performance is summarised in Figure 4.33. Here the average attenuation is plotted against FIR Wiener filter length for both plants. In this case the controllers were designed with the relatively high assumed level of sensor noise ($Q_v = 0.6$ for the minimum delay plant and $Q_v = 1$ for the non-minimum phase plant) that was also used in sections 4.8 and 4.9. Clearly the length of the FIR Wiener filter has an effect on the average attenuation achievable with the IMC controllers.

In the minimum delay plant case (marked with crosses on Figure 4.33) the average attenuation deteriorates for filter lengths below 200 taps, whereas the performance of controllers for the non-minimum phase plant deteriorates for filter lengths below 100 taps. Note that closed loop stability is ensured via the IMC controller formulation independent of the number of coefficients in the FIR filters [3].

4.11 The LQG polynomial design approach

Simulations were carried out to compare the LQG polynomial technique with the LQG state-space and IMC techniques. In the first case, a single mode was included in the plant model and the three different types of feedback controllers were calculated. The effort weighting in the cost function was chosen to be 10^{-12}, the disturbance noise intensity was chosen to be 1 and, the measurement noise intensity was 10^{-11}. The results are shown in Figure 4.34 and it is clear that the LQG polynomial and LQG state-space techniques result in almost exactly the same control performance. This would be expected as they are based on the same cost function and plant and disturbance dynamics. However, it was found that a very high degree of accuracy was required in the calculation of the spectral factors and, as a result of this, carrying out the spectral factorisation was very time-consuming. It turned out that performing the spectral factorisation with higher order polynomials (for example in the seven mode simulations carried out in other sections of this report) with the high degree of accuracy required, was impractical. This is a commonly encountered problem and more efficient algorithms are currently being investigated following [12].

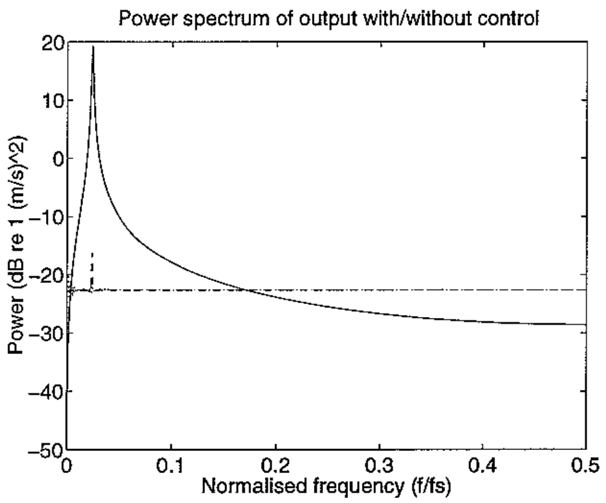

Figure 4.34. Power spectrum of output without control (solid) and with control using the LQG polynomial approach (dashed) and the IMC Wiener approach (dotted). Single mode minimum delay system.

5. Conclusion

A comparison of three different ways of designing discrete time feedback controllers has been carried out. Computer simulations are presented of controllers designed for a highly resonant plate fitted with a control actuator and a sensor. The main focus has been on two of the methods: The state space LQG and the Wiener IMC method. The polynomial LQG approach has also been investigated briefly but was found to have numerical problems.

The comparison has shown that, although quite different in their approach to solving the control problem, the different design methods yield controllers with practically identical performance (disturbance suppression) and robustness properties, when designed from the same plant and disturbance model and under the same assumptions about disturbance and sensor noise. This has been shown for both minimum delay (i.e. minimum phase plus delay) and non-minimum phase plants for two of the design methods. This is not surprising, since essentially the same cost function is minimised in the three designs.

All three design methods allow the adjustment of two basic parameters: The amount of control effort used and the ratio between the assumed level of the disturbance noise and the assumed level of sensor noise. Simulations have shown that varying these parameters in the different design methods again results in controllers with identical performance and robustness. Both parameters weight controller performance against controller robustness. Penalising large control signals results in reduced performance but increased robustness. The simulations have shown that a similar effect can be obtained by assuming a relative high level of sensor noise in the design.

The state space LQG and the Wiener IMC controller designs have been shown to deal equally well with plants having a pure time delay in the control path, although dealing with delays in the LQG design has required a generalisation of the method of solving the involved Riccati equations, compared to the method used in for instance the MATLAB Control Toolbox. The inclusion of time delay in the control path has been shown to affect the controller performance quite drastically.

The simulations have shown that quite long FIR Wiener filters are required in the IMC approach to match the performance of the LQG controllers. Furthermore the choice of estimator used in the state space LQG design has been shown to be crucial when comparing the performance of the two types of controllers.

The issue of reduced order controllers has also been addressed in the simulations. It has been shown that in some cases it is possible to reduce the order of LQG controllers without affecting the performance or robustness in any significant way. This has especially been found to be true for minimum phase plants. Indeed the order of a specific controller was reduced by 60% in the simulations without changing the corresponding performance and robustness. Furthermore an alternative way of designing reduced order IIR controllers has been investigated. IIR filters of varying orders have been fitted to frequency responses of the full order controller, and these control filters have been shown to perform equally well or even better than the reduced order LQG controllers. This observation suggests that a particularly efficient and insightful method of designing a practical IIR controller may be to use IMC with a long FIR control filter to determine the frequency response of the controller, assuming it is stable, and then to use IIR filter fitting to synthesise a numerically efficient implementation of the controller.

Finally a controller has been designed using the LQG polynomial approach to verify this design approach. Unfortunately numerical problems has prevented the testing of this design method with the plant model used in the simulations with the other two design methods.

References

[1] K.J. Åström, B. Wittenmark
Computer Controlled Systems - Theory and Design
Prentice Hall Inc., 1984

[2] S.J. Elliott, T.J. Sutton
Performance of Feedforward and Feedback Systems for Active Control
IEEE Trans. Speech Audio Proc., Vol. 4, No. 3, May 1996, pp 214-223

[3] M. Morari, E. Zafirou
Robust Process Control
Prentice Hall, 1989

[4] P.A. Nelson, S.J. Elliott
Active Control of Sound
Academic Press, 1992

[5] J.C. Doyle, G. Stein
Robustness with Observers
IEEE Trans. Automat. Contr., Vol. AC-24, No. 4, August 1979, pp. 607-611

[6] B.D.O. Anderson, J.B. Moore
Optimal Control - Linear Quadratic Methods
Prentice Hall, 1989

[7] H. Kwakernaak, R. Sivan
Linear Optimal Control Systems
John Wiley & Sons, 1972

[8] G.F. Franklin, J.D. Powell, M.L. Workman
Digital Control of Dynamic Systems
Addison-Wesley Publishing Company, 1990

[9] P.A. Nelson, D.R. Thomas
Discrete time LQG feedback control of sound radiation
ISVR Technical Memorandum No.775, October 1996, University of Southampton

[10] T.C. Sors, P.A. Nelson
Discrete Time LQG Feedback Control of Vibrations
ISVR Technical Memorandum No.815, February 1997, University of Southampton

[11] M.J. Grimble, M.A. Johnson
Optimal Control and Stochastic Estimation, Theory and Applications, Volume 1
John Wiley & Sons, 1988

[12] C. Mohtabi
Numerical algorithms in self-tuning control
Implementation of self-tuning controllers, edited by K. Warwick
Peter Peregrinus, 1988

[13] D.S. Shook, C. Mohtabi, S.L. Shah
Identification for long range predictive control
IEE Proc. Vol. 138, No. 1, January 1991, pp. 75-84.

[14] M.C.M. Wright
An Introduction to Diophantine Equations for Control Theory
ISVR Technical Memorandum No.804, September 1996, University of Southampton

[15] L. Meirovitch
Analytical Methods in Vibrations
The MacMillan Company, 1967

[16] E.K. Dimitriadis, C.R. Fuller, C.A. Rogers
Piezoelectric Actuators for Distributed Vibration Excitation of Thin Plates
Transactions of the ASME, Vol. 113, January 1991, pp 100-107

[17] R.F. Hoskins
Generalised Functions
Ellis Horwood Limited, 1979

[18] A. Grace, A.J. Laub, J.N. Little, C.M. Thompson
Control Systems Toolbox User's Guide.
The Mathworks Inc, 1992

[19] C.B. Moler, C.F. Van Loan
Matrix Computations
Johns Hopkins University Press, 1983

[20] B.C. Kuo
Digital Control Systems
Holt, Rinehart & Winston, 1980

[21] G.F. Franklin, J.D. Powell, A. Emami-Naeini
Feedback Control of Dynamic Systems
Addison-Wesley, 1994

[22] M.J. Balas
Direct Velocity Feedback Control of Large Space Structures
J. Guidance Control, 2, 1979, pp232-253

[23] T.P.Krauss, L.Shure, J.N. Little
MATLAB Signal Processing Toolbox - User's Guide
The Mathworks Inc., 1994

[24] G.J.Gaalman
Comments on "A Nonrecursive Algebraic Solution for the Discrete Riccati Equation"
IEEE Trans. Automat. Contr., Vol. AC-25, No. 3, June 1980, pp. 610-612

[25] T.Pappas, A.J.Laub, N.R.Sandell Jr
On the Numerical Solution of the Discrete-Time Algebraic Riccati Equation
IEEE Trans. Automat. Contr., Vol. AC-25, No. 4, August 1980, pp. 631-641

[26] P.Van Dooren
A Generalized Eigenvalue Approach for Solving Riccati Equations
SIAM J. Sci. Stat. Comput., Vol. 2, No. 2, June 1981, pp. 121-135

[27] D.R.Vaughan
A Nonrecursive Algebraic Solution for the Discrete Riccati Equation
IEEE Trans. Automat. Contr., October 1970, pp 597-599

[28] A.J. Laub
A Schur Method for Solving Algebraic Riccati Equations
IEEE Trans. Automat. Contr., Vol. AC-24, No. 6, December 1979, pp. 913-921

[29] G.W.Stewart
Error and Perturbation Bounds for Subspaces Associated with Certain Eigenvalue Problems
SIAM Review, Vol. 15, No. 4, October 1973, pp. 727-764

[30] A.Emami-Naeini, G.F.Franklin
Comments on "On the Numerical Solution of the Discrete-Time Algebraic Riccati Equation"
IEEE Trans. Automat. Contr., Vol. AC-25, No. 5, October 1980, pp. 1015-1016

[31] C.B.Moler, G.W.Stewart
An Algorithm for Generalized Matrix Eigenvalue Problems
SIAM J. Numer. Anal., Vol. 10, No. 2, April 1973, pp. 241-256

[32] Matlab Reference Guide
The Mathworks Inc, 1992

[33] R.C.Ward
Balancing the Generalized Eigenvalue Problem
SIAM J. Sci. Stat. Comput., Vol. 2, No. 2, June 1981, pp. 141-152

[34] W.F.Arnold, A.J.Laub
Generalised Eigenproblem Algorithms and Software for Algebraic Riccati Equations
Proceedings of the IEEE, Vol. 72, No. 12, December 1984, pp. 1746-1754

[35] T.Gudmundsson, C.Kenney, A.J.Laub
Scaling of the Discrete-Time Algebraic Riccati Equation to Enhance Stability of the Schur Solution Method
IEEE Trans. Automat. Contr., Vol. 37, No. 4, April 1992, pp. 513-51

[36] G.A. Hewer
An Iterative Technique for the Computation of the Steady State Gains for the discrete Optimal Regulator
IEEE Trans. Automat. Contr., August 1971, pp. 382-384.

Appendix A: Solving the discrete time algebraic Riccati equation.

Determining the optimal feedback gain and estimator gain in the LQG controller requires the solution of discrete time algebraic Riccati equations of the form

$$\mathbf{A}^T\mathbf{X}\mathbf{A}-\mathbf{X}-\mathbf{A}^T\mathbf{X}\mathbf{B}\left[\mathbf{B}^T\mathbf{X}\mathbf{B}+\mathbf{R}\right]^{-1}\mathbf{B}^T\mathbf{X}\mathbf{A}+\mathbf{C}^T\mathbf{C}=\mathbf{0} \ , \tag{A.1}$$

where $\mathbf{A},\mathbf{X}\in\Re^{n\times n}$, $\mathbf{B}\in\Re^{n\times m}$, $\mathbf{C}\in\Re^{p\times n}$ and $\mathbf{R}\in\Re^{m\times m}$ with $\mathbf{R}=\mathbf{R}^T>\mathbf{0}$. The pair $(\mathbf{A},\mathbf{B})$ is assumed to be stabilisable and the pair $(\mathbf{A},\mathbf{C})$ is assumed to be detectable. This means that if the system

$$\mathbf{x}(k+1)=\mathbf{A}\,\mathbf{x}(k)+\mathbf{B}\,\mathbf{u}(k) \quad , \quad \mathbf{y}(k)=\mathbf{C}\,\mathbf{x}(k) \tag{A.2}$$

has states that cannot be reached from the input or seen from the output (or both) then these states are stable (i.e. will decay with time) [7] . A stable system of course has this property since all states are stable. In this case equation (A.1) has a unique non-negative definite solution [8]

Except for very low order systems the solution has to be found numerically, and numerical tools to carry out this task are readily available for instance as part of the MATLAB Control Systems Toolbox [18]. Unfortunately the solution method used by these tools requires inversion of the state matrix, and so are not able to handle the case where this matrix is singular. As shown earlier modelling a pure time delay in the control path results in the state matrix becoming singular. Thus dealing with this type of system requires an alternative way of solving the Riccati equation than the one provided in the toolbox. This appendix describes a general method of solving the discrete time algebraic Riccati equation used in the simulations presented in this report, and also describes the method used in the MATLAB Control Systems Toolbox as a specific case of this general method.

Associated with equation (A.1) is the generalised eigenproblem [24], [25]

$$\mathbf{M}\mathbf{z}=\lambda\mathbf{N}\mathbf{z} \ , \tag{A.3}$$

where the $2n\times 2n$ matrices $\mathbf{M}$ and $\mathbf{N}$ are given by [25]

$$\mathbf{M}=\begin{bmatrix}\mathbf{A} & \mathbf{0}\\ -\mathbf{C}^T\mathbf{C} & \mathbf{I}\end{bmatrix} \ , \ \mathbf{N}=\begin{bmatrix}\mathbf{I} & \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\\ \mathbf{0} & \mathbf{A}^T\end{bmatrix} \tag{A.4}$$

and $\mathbf{I}$ is the $n \times n$ unity matrix. The $2n$ generalised eigenvalues corresponding to (A.3) are the solutions λ to $\det[\lambda\mathbf{N} - \mathbf{M}] = 0$ and are such that the reciprocal of an eigenvalue is also an eigenvalue, so that n of the generalised eigenvalues will be stable (inside the unit circle) [25], [26]. It can be shown (see [24] or [25]) that the unique non-negative definite solution to the Riccati equation can be found from a set of n vectors spanning the generalised eigenspace corresponding to the n stable generalised eigenvalues of (A.3).

If $\mathbf{A}$ is invertible the generalised eigenproblem can be converted into the standard eigenproblem form $\mathbf{N}^{-1}\mathbf{M}\mathbf{z} = \lambda\,\mathbf{z}$ where

$$\mathbf{N}^{-1}\mathbf{M} = \begin{bmatrix} \mathbf{I} & -\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{A}^{-T} \\ \mathbf{0} & \mathbf{A}^{-T} \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ -\mathbf{C}^T\mathbf{C} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{A} + \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{A}^{-T}\mathbf{C}^T\mathbf{C} & -\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{A}^{-T} \\ -\mathbf{A}^{-T}\mathbf{C}^T\mathbf{C} & \mathbf{A}^{-T} \end{bmatrix} \tag{A.5}$$

The solution to the Riccati equation can then be found from n vectors spanning the eigenspace corresponding to the n stable eigenvalues of to this eigenproblem (see [27] or [28]). This is the method used in the MATLAB Control Systems Toolbox [18]. It is clear that the generalised eigenvalue approach is a more general approach to the solution of the Riccati equation. This method was therefore used in the simulations presented in this report, and the method will be described in further detail here.

The set of vectors spanning the stable eigenspace can be found as the n eigenvectors of the generalised eigenproblem (A.3) corresponding to the n stable generalised eigenvalues. Unfortunately if the eigenproblem has repeated or nearly repeated eigenvalues, finding a full set of independent eigenvectors is either not possible or possibly numerically very difficult [29]. To overcome this problem the set of vectors can instead be chosen as the n generalised Schur vectors corresponding to the stable generalised eigenvalues [25], [30]. The generalised Schur decomposition method of solving the Riccati equation is based on the fact that there exist unitary matrices $\mathbf{Q}$ and $\mathbf{Z}$ so that [26]

$$\mathbf{QMZ} = \begin{bmatrix} \alpha_{11} & \cdots & * & * & \cdots & * \\ & \ddots & \vdots & \vdots & \ddots & \vdots \\ & & \alpha_{nn} & * & \cdots & * \\ & & & \alpha_{(n+1)(n+1)} & \cdots & * \\ & & & & \ddots & \vdots \\ 0 & & & & & \alpha_{2n2n} \end{bmatrix} \quad \mathbf{QNZ} = \begin{bmatrix} \beta_{11} & \cdots & * & * & \cdots & * \\ & \ddots & \vdots & \vdots & \ddots & \vdots \\ & & \beta_{nn} & * & \cdots & * \\ & & & \beta_{(n+1)(n+1)} & \cdots & * \\ & & & & \ddots & \vdots \\ 0 & & & & & \beta_{2n2n} \end{bmatrix} \tag{A.6}$$

are both upper triangular matrices. The ratios $\lambda_i = \alpha_{ii}/\beta_{ii}$ between the diagonal elements of the two matrices are the generalised eigenvalues corresponding to equation (A.3). Furthermore the upper left n ratios comprise the stable generalised eigenvalues. If the matrix $\mathbf{Z}$ is now partitioned as

$$\mathbf{Z} = \left[\begin{array}{c|c} \mathbf{Z}_{11} & \mathbf{Z}_{12} \\ \mathbf{Z}_{21} & \mathbf{Z}_{22} \end{array}\right] \tag{A.7}$$

then the n column vectors of $[\mathbf{Z}_{11} \quad \mathbf{Z}_{21}]^T$ are the generalised Schur vectors associated with the stable generalised eigenvalues [26]. These vectors span the stable generalised eigenspace, and the unique, positive semidefinite solution $\mathbf{X}$ to the algebraic Riccati equation can be calculated as [26]

$$\mathbf{X} = \mathbf{Z}_{21}\mathbf{Z}_{11}^{-1}. \tag{A.8}$$

When $\mathbf{A}$ is non-singular, all generalised eigenvalues are nonzero. However if $\mathbf{A}$ is singular, at least one generalised eigenvalue will be zero, corresponding to $\alpha_i = 0 \wedge \beta_i \neq 0$ [25]. Since the reciprocal of an eigenvalue is also an eigenvalue, l zero eigenvalues will be matched by l "infinite eigenvalues", simply corresponding to the case where $\alpha_i \neq 0 \wedge \beta_i = 0$ [25]. Because the LQG problem can never be degenerate [30], the undetermined case $\alpha_i = 0 \wedge \beta_i = 0$ does not appear.

The transformation of the generalised eigenvalue problem to upper triangular form can be found with the so called QZ-algorithm [31], which for instance is implemented as a built-in function in MATLAB [32]. Unfortunately the QZ-algorithm returns the generalised eigenvalues in arbitrary order. The eigenvalues can be reordered to any desired order by applying appropriate unitary transformations $\mathbf{Q}_r$ and $\mathbf{Z}_r$ [26], so that applying the QZ-algorithm followed by a reordering results in the overall transformations

$$\mathbf{Q} = \mathbf{Q}_r\mathbf{Q}_u \quad , \quad \mathbf{Z} = \mathbf{Z}_u\mathbf{Z}_r \tag{A.9}$$

where $\mathbf{Q}_u$ and $\mathbf{Z}_u$ are the unitary transformations resulting from the QZ-algorithm. The reordering can be carried out by sorting the generalised eigenvalues via successive interchanging of consecutive eigenvalues [26], so that a single general transformation type (i.e. a transformation that interchanges two consecutive eigenvalues) can be used to reorder the whole problem. If N interchanges are needed to reorder the eigenvalues and the n'th interchange is carried out by applying the transformation $\mathbf{Q}_{i,n}$ and $\mathbf{Z}_{i,n}$, the total reordering transformation can be written as

$$\mathbf{Q}_r = \mathbf{Q}_{i,N}\mathbf{Q}_{i,N-1}\cdots\mathbf{Q}_{i,2}\mathbf{Q}_{i,1} \quad , \quad \mathbf{Z}_r = \mathbf{Z}_{i,1}\mathbf{Z}_{i,2}\cdots\mathbf{Z}_{i,N-1}\mathbf{Z}_{i,N} \ . \tag{A.10}$$

A numerically stable way of implementing the transformation $\mathbf{Q}_{i,n}$ and $\mathbf{Z}_{i,n}$ can be found in [26], and a standard sorting algorithm can then be used to determine the order of the transformations.

If the magnitude of the elements in the **M** or **N** matrices vary over a wide range then inaccuracies can occur when calculating the Schur vectors and the generalised eigenvalues with the QZ-algorithm [33],[34]. To improve the performance of the QZ-algorithm, balancing can be applied to the eigenproblem prior to applying the QZ-algorithm. Ward [33] has developed a balancing procedure (known as Ward Balancing) that attempts to scale **M** and **N** so that their elements are as close to unity as possible. The balancing procedure consists of three steps: If possible, perturbation matrices $\mathbf{P}_1$ and $\mathbf{P}_2$ are applied to the original matrices **M** and **N** to bring them to the form

$$\mathbf{P}_1\mathbf{M}\mathbf{P}_2 = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} & \mathbf{M}_{13} \\ \mathbf{0} & \mathbf{M}_{22} & \mathbf{M}_{23} \\ \mathbf{0} & \mathbf{0} & \mathbf{M}_{33} \end{bmatrix} \quad , \quad \mathbf{P}_1\mathbf{N}\mathbf{P}_2 = \begin{bmatrix} \mathbf{N}_{11} & \mathbf{N}_{12} & \mathbf{N}_{13} \\ \mathbf{0} & \mathbf{N}_{22} & \mathbf{N}_{23} \\ \mathbf{0} & \mathbf{0} & \mathbf{N}_{33} \end{bmatrix} \tag{A.11}$$

where $\mathbf{M}_{11}, \mathbf{M}_{33}, \mathbf{N}_{11}, \mathbf{N}_{33}$ are upper triangular matrices. This reduces the QZ-transformation problem to the submatrices $\mathbf{M}_{22}$ and $\mathbf{N}_{22}$. These matrices are now scaled so that their elements are close to unity by applying two-sided diagonal transformations $\mathbf{Q}_{s,22}$ and $\mathbf{Z}_{s,22}$, and thereafter graded in a way that enhances the performance of the QZ-algorithm by the transformations $\mathbf{Q}_{g,22}$ and $\mathbf{Z}_{g,22}$. The scaled and graded submatrices now have the form

$$\mathbf{M}'_{22} = \mathbf{Q}_{g,22}\mathbf{Q}_{s,22}\mathbf{M}_{22}\mathbf{Z}_{s,22}\mathbf{Z}_{g,22} \quad , \quad \mathbf{N}'_{22} = \mathbf{Q}_{g,22}\mathbf{Q}_{s,22}\mathbf{N}_{22}\mathbf{Z}_{s,22}\mathbf{Z}_{g,22} \tag{A.12}$$

and the QZ-algorithm can now be applied to yield accurate transformations $\mathbf{Q}_{u,22}$ and $\mathbf{Z}_{u,22}$. The eigenproblem has now been brought to unsorted upper triangular form, and what's left is to apply the reordering transformation.

To summarise, the initial matrices **M** and **N** are brought to sorted upper triangular form by

$$\mathbf{QMZ} = \mathbf{Q}_r\mathbf{Q}_u\mathbf{Q}_g\mathbf{Q}_s\mathbf{P}_1\mathbf{M}\mathbf{P}_2\mathbf{Z}_s\mathbf{Z}_g\mathbf{Z}_u\mathbf{Z}_r \quad , \quad \mathbf{QNZ} = \mathbf{Q}_r\mathbf{Q}_u\mathbf{Q}_g\mathbf{Q}_s\mathbf{P}_1\mathbf{N}\mathbf{P}_2\mathbf{Z}_s\mathbf{Z}_g\mathbf{Z}_u\mathbf{Z}_r \tag{A.13}$$

where the scaling transformations $\mathbf{Q}_s$ and $\mathbf{Z}_s$ are given by

$$\mathbf{Q}_s = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{s,22} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \quad , \quad \mathbf{Z}_s = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}_{s,22} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}, \tag{A.14}$$

the grading transformations $\mathbf{Q}_g$ and $\mathbf{Z}_g$ are given by

$$\mathbf{Q}_g = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{g,22} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \quad , \quad \mathbf{Z}_g = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}_{g,22} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} , \tag{A.15}$$

the QZ-transformation matrices $\mathbf{Q}_u$ and $\mathbf{Z}_u$ are given by

$$\mathbf{Q}_u = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{u,22} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \quad , \quad \mathbf{Z}_u = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}_{u,22} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} , \tag{A.16}$$

and $\mathbf{Q}_r$ and $\mathbf{Z}_r$ are the full reordering transformations. The final Schur vector matrix $\mathbf{Z}$ from which the solution is directly derived is given by

$$\mathbf{Z} = \mathbf{P}_2 \mathbf{Z}_s \mathbf{Z}_g \mathbf{Z}_u \mathbf{Z}_r . \tag{A.17}$$

Gudmundsson, Kenney and Laub [35] suggest a different scaling procedure, where the solution $\hat{\mathbf{X}}$ to the scaled problem

$$\hat{\mathbf{M}} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ -\mathbf{C}^T\mathbf{C}/\rho & \mathbf{I} \end{bmatrix} \quad , \quad \hat{\mathbf{N}} = \begin{bmatrix} \mathbf{I} & \rho\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T \\ \mathbf{0} & \mathbf{A}^{\mathrm{T}} \end{bmatrix} \tag{A.18}$$

is found. By choosing the scalar parameter ρ appropriately, the conditioning of the corresponding generalised Schur problem is improved, and the solution $\mathbf{X}$ to the unscaled problem is found from the solution to the scaled problem as $\mathbf{X} = \hat{\mathbf{X}}\rho$. In [35] it is pointed out that there are two possible origins of numerical instabilities in the generalised Schur method. One is the ill-conditioning of $\mathbf{Z}_{11}$ with respect to inversion, and the other is that even though $\mathbf{Z}$ is well determined as a whole, its small elements are in general less accurate than its large ones, and therefore the solution $\mathbf{X}$ may still be inaccurate if the elements of $\mathbf{Z}$ have widely varying magnitudes.

It is shown in [35] that $\rho = \|\mathbf{X}\|^2$ is close to an optimal scaling factor in the sense that it minimises the chance of the smaller elements of $\mathbf{Z}$ being inaccurately determined and simultaneously ensures that $\mathbf{Z}_{11}$ is well-conditioned with respect to inversion. Since the optimal scaling factor depends on the solution itself, [35] suggests that the (inaccurate) solution to the unscaled problem is first calculated directly and used to scale the problem. The scaled problem is then solved to provide an accurate solution. The drawback of this method is that the solution to the Riccati equation has to be calculated twice.

The accuracy of the solution found with the generalised Schur method can sometimes be further improved by iterative refinement [34], utilising the Newton method [36]

$$\begin{aligned}
&\text{(I)}\quad \mathbf{X_k} = (\mathbf{A} - \mathbf{BK_k})^T \mathbf{X_k} (\mathbf{A} - \mathbf{BK_k}) + \mathbf{K_k^T R K_k} + \mathbf{C}^T\mathbf{C} \\
&\text{(II)}\quad \mathbf{K_k} = \left(\mathbf{R} + \mathbf{B}^T\mathbf{X_{k-1}B}\right)^{-1} \mathbf{B}^T\mathbf{X_{k-1}A}
\end{aligned} \tag{A.19}$$

Each iteration requires the solution of a discrete Lyapunov equation (I). The Mathworks Control Systems Toolbox [18] provides a solving routine for this.

To summarise, the steps involved in calculating an accurate solution to the discrete time Riccati equation using the generalised Schur approach are:

1) Set up the **M** and **N** matrices of the corresponding generalised eigenvalue problem
2a) Ward Balancing if elements of **M** and **N** vary much in magnitude (optional)
2b) Optimal Scaling with norm of solution (optional)
3) QZ algorithm to transform to unsorted upper triangular form
4) Ordering algorithm to transform to sorted upper triangular form
5a) Backtransform to unbalanced problem from Ward Balancing
5b) Backtransform to unbalanced problem from Optimal Scaling
6) Extract Schur vectors corresponding to stable subspace
7) Solve set of linear equations to find **X**
8) Newton iteration (optional)

A generalised discrete time Riccati equation solver incorporating these steps has been implemented in MATLAB and is listed in appendix B. Part of the implementation (the ordering algorithm and the Ward balancing algorithm) is based on the FORTRAN software package RICPACK, described in [34], whereas central parts like the QZ-algorithm, the linear equation solver and the discrete time Lyapunov equation solver are either built-in MATLAB functions [32] or part of the Mathworks Control Systems Toolbox [18]. Once again it is emphasised that the discrete time Riccati equation solver in the Control Toolbox 3.0b is *not* able to handle systems with singular system matrices and/or systems that result in Hamiltonian matrices with repeated eigenvalues.

Appendix B: Riccati equation solver (MATLAB)

Main Riccati equation solver GDARE.M

```
function[X]=gdare(A,B,S,R,scal,bal,itera)

% GDARE.M Generalised Discrete Algebraic Riccati Equation solution. This
% function finds the stablising solution (if it excists) to the discrete
% time algebraic Riccati equation A'*X*A-X-A'*X*B*inv(R+B'*X*B)*B'*X*A+S=0
% where R is positive definite and symmetric, (A,B) is stabilisable and
% (A,C) is detectable (S=C'*C), and where G=B*inv(R)*B'. The function uses
% the generalised Schur method.
%
% Syntax: X=gdare(A,B,S,R,scal,bal,itera)
% where bal=1 indicates optional Ward Balancing before the QZ-transformation
% and itera indicates the number of optional Newton iterations to perform in
% order to refine the solution.
%
% References:
% P. Van Dooren: A Generalised Eigenvalue Approach for Solving Riccati Equations
% SIAM J. Sci. Stat. Comp., Vol 2, No 2, June 1981, pp 121-135.
% R.C.Ward: Balancing the Generalised Eigenvalue Problem
% SIAM J. Sci. Stat. Comp., Vol 2, No 2, June 1981, pp 141-152
%
% ISVR, University of Southampton
% 05-08-1997, by Jakob Mørkholt

%-----------------------------------------------------------------------
% SETTING UP MATRIX PENCIL
%-----------------------------------------------------------------------
syssize=size(A,1);
M=[A zeros(syssize);-S/scal eye(syssize)];
L=[eye(syssize) (B/R*B')*scal;zeros(syssize) A'];

%-----------------------------------------------------------------------
% WARD BALANCING
%-----------------------------------------------------------------------
if bal
  [M,L,Low,High,Qr,Zr]=reduce(M,L);
  [M,L,Qs,Zs]=scaleg(M,L,Low,High);
  [M,L,Qg,Zg]=gradeq(M,L,Low,High);
end

%-----------------------------------------------------------------------
% QZ TRANSFORMATION
%-----------------------------------------------------------------------
[M,L,Q,Z,dummy]=qz(M,L);

%-----------------------------------------------------------------------
% TRANSFORMATION TO STABLE/UNSTABLE SUBSPACE
%-----------------------------------------------------------------------
[M,L,Q,Z]=order(M,L,Q,Z);

%-----------------------------------------------------------------------
% WARD BACKTRANSFORMATION
%-----------------------------------------------------------------------
if bal
  Z=Zr*Zs*Zg*Z;
end

%-----------------------------------------------------------------------
% EXTRACT SOLUTION FROM STABLE SUBSPACE
%-----------------------------------------------------------------------
Z11=Z(1:syssize,1:syssize);
Z21=Z((syssize+1):(2*syssize),1:syssize);
X=real(Z21/Z11);
X=scal*X;

%-----------------------------------------------------------------------
% PERFORM NEWTON ITERATIONS
%-----------------------------------------------------------------------
if itera>0
  X=newt(A,B,S,R,X,itera);
end
```

Eigenvalue ordering routine ORDER.M

```
function[M,L,Q,Z]=order(M,L,Q,Z)

% ORDER.M reorders the diagonal elements along with the generalised
% eigenvalues corresponding to the generalised eigenvalue problem
% M*X=lambda*L*X by constructing row and column equivalence transfor-
% mations to Q and Z. The eigenvalues are ordered by increasing magnitude.
% This routine is used after QZ.M.
%
% Syntax: [M,L,Q,Z]=order(M,L,Q,Z)
%
% where on input M,L,Q,Z are the outputs of QZ.M and on output are
% the matrices corresponding to the sorted stable/unstable problem.
%
% Reference:
% Van Dooren, P.: A generalised eigenvalue approach for solving Riccati eq's
% SIAM J. Sci. Stat. Comp., Vol 2, No 2, June 1981, pp 121-135.
%
% Algorithm uses EXCHQZ.M adopted from EXCHQZ.F in the RICPACK package.
%
% ISVR, University of Southampton
% 18-08-1997, by Jakob Mørkholt

%-----------------------------------------------------------------------------
% INITIALISATION
%-----------------------------------------------------------------------------
pensize=size(M,1);
tol=1e-10;

%-----------------------------------------------------------------------------
% TRANSFORMATION TO STABLE/UNSTABLE SUBSPACE
%-----------------------------------------------------------------------------
alfa=abs(diag(M));                  % alfa values are diag elements of A
beta=abs(diag(L));                  % beta values are diag elements of B
infpos=find(beta<tol);              % positions of infinite eigenvalues
numinf=length(infpos);              % number of infinite eigenvalues

for k=1:numinf                      % deal with infinite eigenvalues
  start=infpos(numinf+1-k);         % start position of current infinite
  stop=pensize-k;                   % destination of current infinite
  params=[k start stop]
  for n=start:stop                  % move!
    [M,L,Q,Z]=exchqz(M,L,Q,Z,n);    % ...using consecutive swaps
  end
end

for k=1:(pensize/2)                 % deal with finite eigenvalues
  alfa=abs(diag(M));
  alfa=alfa(k:(pensize-numinf));    % alfa's from k to end of finites
  beta=abs(diag(L));
  beta=beta(k:(pensize-numinf));    % beta's from k to end of finites
  minpos=find(alfa./beta<1);        % list of remaining stable eigvals
  if isempty(minpos)
    error('Unable to sort generalised eigenvalues')
  end
  start=k+minpos(1)-2;              % start position of closest stable
  stop=k;                           % destination of closest stable
  params=[k start stop]
  for n=start:-1:stop               % move!
    [M,L,Q,Z]=exchqz(M,L,Q,Z,n);    % ...using consecutive swaps
  end
end

k=pensize/2+1;
alfa=abs(diag(M));
alfa=alfa(k:(pensize-numinf));
beta=abs(diag(L));
beta=beta(k:(pensize-numinf));
minpos=find(alfa./beta<1);
if isempty(minpos)
  disp('Sorting of generalised eigenvalues succesfull')
else
  error('Unable to sort generalised eigenvalues')
end
```

Similarity transform routine to interchange two eigenvalues EXCHQZ.M

```
function[M,L,Q,Z]=exchqz(M,L,Q,Z,pos)

% EXCHQZ.M produces equivalence transformations Qch and Zch that transform
% the complex upper triangular matrix L and the complex upper triangular
% matrix M, (for example as output from the QZ algorithm) in the generalised
% eigenvalue problem M*X=lambda*L*X so that the pos and pos+1 generalised
% eigenvalues are interchanged.
%
% Reference:
% Van Dooren, P.: A generalised eigenvalue approach for solving Riccati eq's
% SIAM J. Sci. Stat. Comp., Vol 2, No 2, June 1981, pp 121-135.
%
% Syntax: [M,L,Q,Z]=exchqz(M,L,Q,Z,pos)
%
% On input:  M,L complex upper triangular matrices in M-lambda*L
%            Q,Z transformation matrix from QZ algorithm.
% On output: M,L complex upper triangular matrices Qch*M*Zch and Qch*L*Zch
%            Eigenvalue pos and pos+1 interchanged.
%            Z transformation matrix Z*Zch.
%            Q transformation matrix Qch*Q.
%
% ISVR, University of Southampton
% 22-07-1997, by Jakob Mørkholt

%-----------------------------------------------------------------------------
% CALCULATING POST MULTIPLICATION MATRIX Zch - APPLYING IT TO M & L
%-----------------------------------------------------------------------------
c=norm([M(pos+1,pos+1) L(pos+1,pos+1)],inf);
altl=1;                                      % altb true: L-elem largest
if abs(M(pos+1,pos+1))>=c
  altl=0;                                    % altb false: M-elem largest
end
sm=M(pos+1,pos+1)/c;                         % normalsd upper M diag elem
sl=L(pos+1,pos+1)/c;                         % normalsd upper L diag elem
f=sm*L(pos,pos)-sl*M(pos,pos);               % Z rotation parameters
g=sm*L(pos,pos+1)-sl*M(pos,pos+1);
c=norm([f g],inf);
f=f/c;
g=g/c;
c=norm([f g],2);
f=f/c;
g=g/c;

Zch=eye(size(M,1));                          % Postmultiplication matrix Z
Zch(pos:pos+1,pos:pos+1)=[g f;-f g];

M=M*Zch;                                     % Post Transform M matrix
L=L*Zch;                                     % Post Transform B matrix
Z=Z*Zch;                                     % Post Transform Z matrix

%-----------------------------------------------------------------------------
% CALCULATING PRE MULTIPLICATION MATRIX Qch - APPLYING IT TO M & L
%-----------------------------------------------------------------------------
if altl                                      % Rotation parameters for Q
  c=norm([L(pos,pos) L(pos+1,pos)],inf);     % Infinity norm
  f=L(pos,pos)/c;
  g=L(pos+1,pos)/c;
else                                         % Rotation parameters for Q
  c=norm([M(pos,pos) M(pos+1,pos)],inf);     % Infinity norm
  f=M(pos,pos)/c;
  g=M(pos+1,pos)/c;
end
c=norm([f g],2);
f=f/c;
g=g/c;

Qch=eye(size(M,1));                          % Premultiplication matrix Q
Qch(pos:pos+1,pos:pos+1)=[f g;-g f];

M=Qch*M;
L=Qch*L;
M(pos+1,pos)=0;                              % They're zero anyway so...
L(pos+1,pos)=0;
Q=Qch*Q;
```

Ward Balancing reducing routine REDUCE.M

```
function[M,L,low,high,Q,Z]=reduce(M,L)

% REDUCE.M reduces, if possible, the order of the generalised eigenvalue
% problem M*X=lambda*L*X by permuting the rows and columns of M and L so
% that they each have the form
%
%       U X Y
%       O C V
%       O O R
%
% where U and R are upper triangular and C, X, Y and V are arbitrary.
% Thus, the isolated eigenvalues corresponding to the triangular matrices
% are obtained by a division, leaving only eigenvalues corresponding to
% the center matrices to be computed.
%
% Syntax: [M,L,low,high,Q,Z]=reduce(M,L)
%
% where low and high is the beginning and ending index of the submatrices
% of M and L containing the non-isolated eigenvalues, and Q and Z are
% the row and column permutation matrices, so that Q*M*Z and Q*L*Z have
% the desired form. If high=1 the permuted M and L matrices are upper
% triangular.
%
% Reference:
% Ward, R.C.: Balancing the generalised eigenvalue problem
% SIAM J. Sci. Stat. Comp., Vol 2, No 2, June 1981, pp 141-152.
%
% Algorithm adopted from REDUCE.F in the FORTRAN package RICPACK.
%
% ISVR, University of Southampton
% 25-07-1997, by Jakob Mørkholt


%------------------------------------------------------------------------
% INITIALISATION
%------------------------------------------------------------------------
pensize=size(M,1);                            % size of matrix pencil
Q=eye(pensize);                               % row transform matrix
Z=eye(pensize);                               % column transform matrix
low=1;                                        % lower block index
high=pensize;                                 % upper block index

%------------------------------------------------------------------------
% FIND ROW WITH ONE NONZERO IN COLUMNS 1 THROUGH high AND PERMUTE
%------------------------------------------------------------------------
found=1;

while (high>1)&found
  found=0;
  for row=high:-1:1                           % loop through rows high to 1
    nonz_M=find(M(row,1:high));               % index to nonz in row of M
    nonz_L=find(L(row,1:high));               % index to nonz in row of L
    totnonz=length(nonz_M)+length(nonz_L);    % total number of nonzeros
    if totnonz==0                             % both rows all zero
      col=high;found=1;break;
    elseif (totnonz==1)&isempty(nonz_L)
      col=nonz_M(1);found=1;break;            % one nonz in row of M only
    elseif (totnonz==1)&isempty(nonz_M)
      col=nonz_L(1);found=1;break;            % one nonz in row of L only
    elseif (totnonz==2)&(length(nonz_M)==1)
      if nonz_M(1)==nonz_L(1)                 % same pos nonz in M and L
        col=nonz_M(1);found=1;break;
      end
    end
  end
  if found                                    % a row with one nonz found
    if row~=high
      M([row high],low:pensize)=M([high row],low:pensize);  % swap rows
      L([row high],low:pensize)=L([high row],low:pensize);
      Q([row high],:)=Q([high row],:);
    end
    if col~=high
      M(1:high,[col high])=M(1:high,[high col]);            % swap cols
      L(1:high,[col high])=L(1:high,[high col]);
      Z(:,[col high])=Z(:,[high col]);
    end
  else                                        % no row found -> col mode
    break
  end
  high=high-1;                                    % decrease upper bound
end

if high==1                                        % M and L upper triangular
  return;
end


%--------------------------------------------------------------------------
% FIND COLUMN WITH ONE NONZERO IN ROWS low THROUGH pensize AND PERMUTE
%--------------------------------------------------------------------------
found=1;

while (low<high)&found
  found=0;
  for col=low:high
    nonz_M=find(M(low:pensize,col))+low-1;
    nonz_L=find(L(low:pensize,col))+low-1;
    totnonz=length(nonz_M)+length(nonz_L);
    if totnonz==0
      row=high;found=1;break;
    elseif (totnonz==1)&isempty(nonz_L)
      row=nonz_M(1);found=1;break;
    elseif (totnonz==1)&isempty(nonz_M)
      row=nonz_L(1);found=1;break;
    elseif (totnonz==2)&(length(nonz_M)==1)
      if nonz_M(1)==nonz_L(1)
        row=nonz_M(1);found=1;break;
      end
    end
  end
  if found
    if row~=low
      M([row low],low:pensize)=M([low row],low:pensize);     % swap rows
      L([row low],low:pensize)=L([low row],low:pensize);
      Q([row low],:)=Q([low row],:);
    end
    if col~=low
      M(1:high,[col low])=M(1:high,[low col]);              % swap cols
      L(1:high,[col low])=L(1:high,[low col]);
      Z(:,[col low])=Z(:,[low col]);
    end
  else                                            % no col found - stop!
    break
  end
  low=low+1;                                      % increase lower bound
end
```

Ward Balancing Scaling routine SCALEG.M

```
function[M,L,Q,Z]=scaleg(M,L,low,high)

% SCALEG.M scales the matrices M and L in the generalised eigenvalue
% problem M*X=LAMBDA*L*X such that the magnitudes of the elements of
% the submatrices of M and L (as specified by low and high) are close
% to unity in the least squares sense. The matrices Q and Z are the
% row and column scaling matrices so that Q*M*Z and Q*L*Z are the
% scaled matrices.
%
% Syntax: [M,L,Q,Z]=scaleg(M,L,low,high)
%
% Reference:
% Ward, R.C.: Balancing the generalised eigenvalue problem
% SIAM J. Sci. Stat. Comp., Vol 2, No 2, June 1981, pp 141-152.
%
% Algorithm adopted from SCALEG.F in the FORTRAN package RICPACK.
%
% ISVR, University of Southampton
% 28-07-1997, by Jakob Mørkholt

%-------------------------------------------------------------------
% INITIALISATION
%-------------------------------------------------------------------
pensize=size(M,1);
nr=high-low+1;
coef=1/(2*nr);
coef2=coef*coef;
coef5=0.5*coef2;
nrp2=nr+2;
beta=0;
it=1;
cmax=1;

if low==high
  return
end

WK(low:high,1:6)=zeros(nr,6);
CS(low:high,1)=zeros(nr,1);
RS(low:high,1)=zeros(nr,1);


%-------------------------------------------------------------------
% COMPUTE RIGHT SIDE VECTOR IN RESULTING LINEAR EQUATIONS
%-------------------------------------------------------------------
TM=M(low:high,low:high);
TM=reshape(TM,1,nr*nr);
indxM=find(TM);
TM(indxM)=log10(abs(TM(indxM)))/log10(2);
TM=reshape(TM,nr,nr);

TL=L(low:high,low:high);
TL=reshape(TL,1,nr*nr);
indxL=find(TL);
TL(indxL)=log10(abs(TL(indxL)))/log10(2);
TL=reshape(TL,nr,nr);

WK(low:high,5)=-sum(TM')'-sum(TL')';
WK(low:high,6)=-sum(TM)'-sum(TL)';


%------------------------------------------------------------------------
% START GENERALIZED CONJUGATE GRADIENT ITERATION
%------------------------------------------------------------------------
while (cmax>=0.5)&(it<=nrp2)
  gamma=WK(low:high,5)'*WK(low:high,5)+WK(low:high,6)'*WK(low:high,6);
  ew=sum(WK(low:high,5));
  ewc=sum(WK(low:high,6));
  gamma=coef*gamma-coef2*(ew^2+ewc^2)-coef5*(ew-ewc)^2;
  if it~=1
    beta=gamma/pgamma;
  end
  WK(low:high,2)=beta*WK(low:high,2)+coef*WK(low:high,5)+coef5*(ewc-3*ew);
  WK(low:high,1)=beta*WK(low:high,1)+coef*WK(low:high,6)+coef5*(ew-3*ewc);
```

```
    % APPLY MATRIX TO VECTOR

    for row=low:high
      indxM=find(M(row,low:high));                  % index to nonzero elements in row'th row of M
      indxL=find(L(row,low:high));                  % index to nonzero elements in row'th row of L
      kount=length(indxM)+length(indxL);
      WK(row,3)=kount*WK(row,2)+sum(WK(indxM,1))+sum(WK(indxL,1));
    end
    for col=low:high
      indxM=find(M(low:high,col));                  % index to nonzero elements in col'th col of M
      indxL=find(L(low:high,col));                  % index to nonzero elements in col'th col of L
      kount=length(indxM)+length(indxL);
      WK(col,4)=kount*WK(col,1)+sum(WK(indxM,2))+sum(WK(indxL,2));
    end
    summ=WK(low:high,2)'*WK(low:high,3)+WK(low:high,1)'*WK(low:high,4);
    if summ==0
      return
    end
    alpha=gamma/summ;


    % DETERMINE CORRECTION TO CURRENT ITERATE

    corwk=alpha*WK(low:high,2);
    RS(low:high)=RS(low:high)+corwk;
    corcs=alpha*WK(low:high,1);
    CS(low:high)=CS(low:high)+corcs;
    cmax=max(abs([corwk;corcs]));

    if cmax>=0.5
      WK(low:high,5)=WK(low:high,5)-alpha*WK(low:high,3);
      WK(low:high,6)=WK(low:high,6)-alpha*WK(low:high,4);
      pgamma=gamma;
      it=it+1;
    end
  end


%------------------------------------------------------------------------
% END GENERALIZED CONJUGATE GRADIENT ITERATION
%------------------------------------------------------------------------
indxgez=find(RS(low:high)>=0);
indxltz=find(RS(low:high)<0);
RS(indxgez)=RS(indxgez)+0.5;
RS(indxltz)=RS(indxltz)-0.5;
indxgez=find(CS(low:high)>=0);
indxltz=find(CS(low:high)<0);
CS(indxgez)=CS(indxgez)+0.5;
CS(indxltz)=CS(indxltz)-0.5;


%------------------------------------------------------------------------
% SCALE M AND L
%------------------------------------------------------------------------
Q=diag([ones(low-1,1) ; 2.^RS(low:high) ; ones(pensize-high,1)]);
Z=diag([ones(low-1,1) ; 2.^CS(low:high) ; ones(pensize-high,1)]);
M=Q*M*Z;
L=Q*L*Z;
```

Ward Balancing Grading routine GRADEQ.M

```
function[M,L,Q,Z]=gradeq(M,L,low,high)

% GRADEQ.M grades the submatrices of M and L given by starting index
% low and ending index high in the generalised eigenvalue problem
% M*X=LAMBDA*L*X, by permuting rows and columns such that the norm
% of the I'th row (column) of the L submatrix becomes smaller as I
% increases.
%
% Syntax: [M,L,Q,Z]=scaleg(M,L,low,high)
% where Q and Z is the row and column permutation matrices such that
% Q*M*Z and Q*L*Z are the graded matrices.
%
% Reference:
% Ward, R.C.: Balancing the generalised eigenvalue problem
% SIAM J. Sci. Stat. Comp., Vol 2, No 2, June 1981, pp 141-152.
% Algorithm adopted from GRADEQ.F in the FORTRAN package RICPACK.
%
% ISVR, University of Southampton
% 28-07-1997, by Jakob Mørkholt

%---------------------------------------------------------------------
% INITIALISATION
%---------------------------------------------------------------------
pensize=size(M,1);
if low==high
  return
end
Q=eye(pensize);
Z=eye(pensize);

%---------------------------------------------------------------------
% COMPUTE ROW NORMS OF M / THOSE OF L
%---------------------------------------------------------------------
for row=low:high
  normM=norm(M(row,low:high),1);                    % sum of absolute values
  normL=norm(L(row,low:high),1);
  if normL==0
    WK(row,1)=1.0e38;
  else
    WK(row,1)=normM/normL;                          % norm ratios
  end
end

%---------------------------------------------------------------------
% CALCULATE ROW PERMUTATION MATRIX Q (DECREASING QUOTIENTS)
%---------------------------------------------------------------------
[WK(low:high) index]=sort(WK(low:high));            % sort by norm ratios
RP(low:high,1)=flipud(index)+(low-1);               % index decreasing ratios
Q=Q([1:(low-1) RP(low:high,1)' (high+1):pensize],:); % permute rows of Q

%---------------------------------------------------------------------
% COMPUTE COLUMN NORMS OF M / THOSE OF L
%---------------------------------------------------------------------
for col=low:high
  normM=norm(M(low:high,col),1);                    % sum of absolute values
  normL=norm(L(low:high,col),1);
  if normL==0
    WK(col,1)=1.0e38;
  else
    WK(col,1)=normM/normL;                          % norm ratios
  end
end

%---------------------------------------------------------------------
% CALCULATE COLUMN PERMUTATION MATRIX Z (DECREASING QUOTIENTS)
%---------------------------------------------------------------------
[WK(low:high) index]=sort(WK(low:high));            % sort by norm ratios
CP(low:high,1)=flipud(index)+(low-1);               % index decreasing ratios
Z=Z(:,[1:(low-1) CP(low:high,1)' (high+1):pensize]); % permute cols of Z

%---------------------------------------------------------------------
% APPLY Q AND Z TO M AND L
%---------------------------------------------------------------------
M=Q*M*Z;
L=Q*L*Z;
```

Appendix C: Calculation of discrete time state space matrices

The discrete time state space matrices are found from the continuous time matrices via

$$\mathbf{A} = e^{\mathbf{A}_c T} \quad , \quad \mathbf{B}_1 = e^{\mathbf{A}_c \eta T}\left[\int_0^{T-\eta T} e^{\mathbf{A}_c \tau} d\tau\right]\mathbf{B}_c \quad , \quad \mathbf{B}_2 = \left[\int_0^{\eta T} e^{\mathbf{A}_c \tau} d\tau\right]\mathbf{B}_c \quad , \quad \mathbf{C} = \mathbf{C}_c \tag{C.1}$$

where T is the sampling interval and ηT is the fractional sample delay. The exponential of the matrix $\mathbf{A}_c \tau$ is defined by the series

$$e^{\mathbf{A}_c \tau} = \sum_{k=0}^{\infty} \frac{\mathbf{A}_c^k \tau^k}{k!} = \mathbf{I} + \mathbf{A}_c \tau + \frac{\mathbf{A}_c^2 \tau^2}{2!} + \frac{\mathbf{A}_c^3 \tau^3}{3!} + \cdots \tag{C.2}$$

Using this expansion it is found that

$$\int_0^t e^{\mathbf{A}_c \tau} d\tau = \int_0^t \sum_{k=0}^{\infty} \frac{\mathbf{A}_c^k \tau^k}{k!} d\tau = \sum_{k=0}^{\infty} \frac{\mathbf{A}_c^k \tau^{k+1}}{(k+1)!} = \mathbf{I}\tau + \frac{\mathbf{A}_c \tau^2}{2!} + \frac{\mathbf{A}_c^2 \tau^3}{3!} + \cdots \tag{C.3}$$

Introducing now the matrix $\mathbf{S}_c$ given by

$$\mathbf{S}_c = \begin{bmatrix} \mathbf{A}_c & \mathbf{B}_c \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{C.4}$$

it is easily seen that

$$\mathbf{S}_c^2 = \begin{bmatrix} \mathbf{A}_c^2 & \mathbf{A}_c \mathbf{B}_c \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad , \quad \mathbf{S}_c^3 = \begin{bmatrix} \mathbf{A}_c^3 & \mathbf{A}_c^2 \mathbf{B}_c \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad , \quad \mathbf{S}_c^k = \begin{bmatrix} \mathbf{A}_c^k & \mathbf{A}_c^{k-1} \mathbf{B}_c \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{C.5}$$

The exponential of $\mathbf{S}_c \tau$ can thus be written as

$$e^{\mathbf{S}_c \tau} = \sum_{k=0}^{\infty} \frac{\mathbf{S}_c^k \tau^k}{k!} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} + \begin{bmatrix} \mathbf{A}_c & \mathbf{B}_c \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\tau + \begin{bmatrix} \mathbf{A}_c^2 & \mathbf{A}_c \mathbf{B}_c \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\frac{\tau^2}{2!} + \begin{bmatrix} \mathbf{A}_c^3 & \mathbf{A}_c^2 \mathbf{B}_c \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\frac{\tau^3}{3!} + \cdots = \begin{bmatrix} e^{\mathbf{A}_c \tau} & \int_0^{\tau} e^{\mathbf{A}_c t} dt \mathbf{B}_c \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{C.6}$$

Forming the exponentials of $\mathbf{S}_c(T - \eta T)$ and $\mathbf{S}_c \eta T$ gives

$$e^{\mathbf{S}_c(T-\eta T)} = \begin{bmatrix} e^{\mathbf{A}_c(T-\eta T)} & \int_0^{T-\eta T} e^{\mathbf{A}_c t} dt \, \mathbf{B}_c \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \quad , \quad e^{\mathbf{S}_c \eta T} = \begin{bmatrix} e^{\mathbf{A}_c \eta T} & \int_0^{\eta T} e^{\mathbf{A}_c t} dt \, \mathbf{B}_c \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{C.7}$$

and by comparing with (C.1) it is seen that the discrete time matrices can all be formed from the elements of these matrices. This method is used in the Mathworks Control Toolbox M-file c2dt.m. The matrix exponential is calculated by the built-in MATLAB function expm.m, which uses a Padé approximation as described in [6].

Appendix D1: Model parameters - minimum phase plant

PHYSICAL PARAMETERS	
Length of panel (m)	a=0.430
Width of panel (m)	b=0.360
Thickness of panel (m)	h=0.0015
Youngs modulus of panel material (aluminium) (Pa)	E=7.2e10
Poisson ratio of panel material (aluminium)	my=0.3;
Density of panel material (aluminium) (kg/m^3)	rho=2.7e3
Equivalent viscous damping ratio of panel	damp=0.025
ACTUATOR, SENSOR AND DISTURBANCE	
X coordinate of point force (m)	dpx=0.06
Y coordinate of point force (m)	dpy=0.06
Point force amplification factor	Dis_Poin_Amp=30
Length of piezo actuators (m)	a_piezo=0.050
Width of piezo actuators (m)	b_piezo=0.025
X coordinate of left side of piezo (m)	cpx1=0.34
X coordinate of right side of piezo (m)	cpx2=cpx1+b_piezo
Y coordinate of bottom side of piezo (m)	cpy1=0.04
Y coordinate of top side of piezo (m)	cpy2=cpy1+a_piezo
Piezo element amplification factor	Con_Piez_Amp=60
X coordinate of point sensor (m)	sx=0.35
Y coordinate of point sensor (m)	sy=0.05
Sensor amplification factor	Sens_Amp=100
MODEL PARAMETERS	
Number of modes included	NumModes=7
Boundary conditions ('simply','freely')	support='simply'
Sensor type ('displacement','velocity')	sensor='velocity'
Control delay (milli seconds)	delay=0
Sampling frequency (Hz)	fs=2000
Switch anti aliasing filter on/off	aliasingfilt='on'
LP-filter cutoff frequency (Hz)	cutoff=500
Order of LP-filter	filtord=1

CONTROL PARAMETERS

Number of taps in impulse responses	NI=2048
Number of taps in Wiener filter	NW=700
IMC Regularisation parameter	IMC_beta=0
LQG Effort weight	LQG_rho=1e-9
Disturbance noise intensity	Q=1
Sensor noise intensity	R=1e-9
LQG type of estimator ('current','predict')	estimator='current';

Appendix D2: Model parameters - non -minimum phase plant

The position of the piezo actuator is changed to

Length of piezo actuators (m)	a_piezo=0.050
Width of piezo actuators (m)	b_piezo=0.025
X coordinate of left side of piezo (m)	cpx1=0.05
X coordinate of right side of piezo (m)	cpx2=cpx1+a_piezo
Y coordinate of bottom side of piezo (m)	cpy1=0.30
Y coordinate of top side of piezo (m)	cpy2=cpy1+b_piezo
Piezo element amplification factor (m)	Con_Piez_Amp=60