Stochastic Adaptive Control (02421)

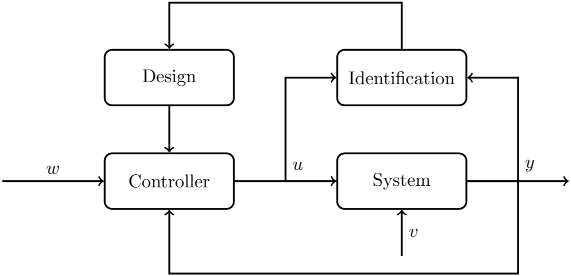
Lecture 12

Tobias K. S. Ritschel

Section for Dynamical Systems, DTU Compute

Lecture Plan

1. Systems theory
2. Stochastics
3. State estimation - Kalman filter 1
4. State estimation - Kalman filter 2
5. Optimal control 1 - internal models
6. External models
7. Prediction + optimal control 1 - external models
8. Optimal control 2 - external models
9. System identification 1
10. System identification 2
11. System identification 3 + model validation
12. **Adaptive control 1**
13. Adaptive control 2



Today's Agenda

- Follow-up from last lecture
- Adaptive control methods
- Self-tuners
    1. Explicit self-tuner
    2. Implicit self-tuner

Follow-up from last time

Questions?

Adaptive Control

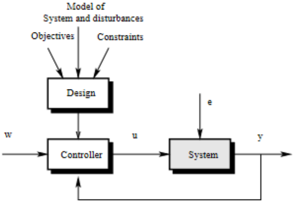
Stochastic control relies on a detailed model which might not be available.

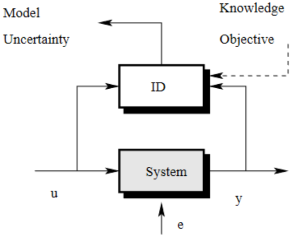
1. Parameter values cannot be measured.
2. The underlying physics is not known sufficiently well

Approach 1:
A model can be created using identification methods and a stochastic controller can be designed.

If the system varies in time, e.g., due to aging or wear, the identification will have to be repeated occasionally.

Approach 2:
Alternatively, we can combine online identification and control.





Alternative to adaptive control: Robust control

If the model of a system is uncertain, there also exist other methods than the adaptive control. One such method is the robust control:

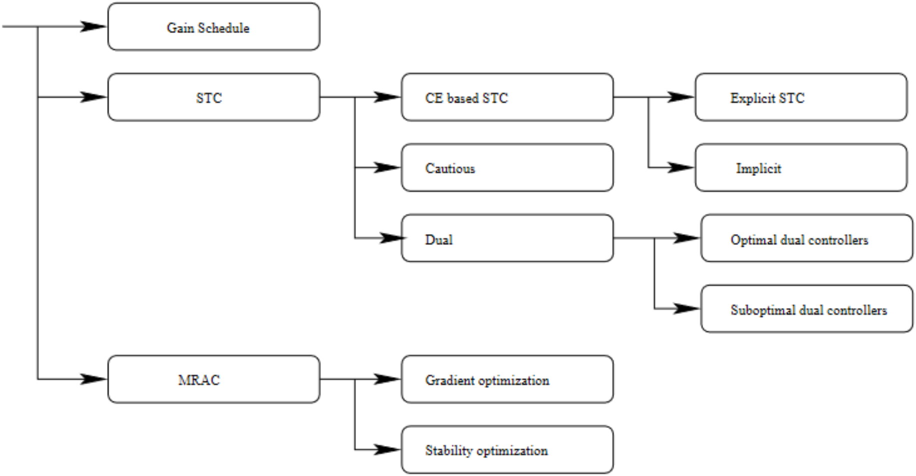
1. Robust control: Low sensitivity to the effect of uncertain system parts, a control that, in some sense, operates after worst-case scenario.
2. Adaptive control: Monitors/estimates the uncertain parts, a control law that changes with the identified system.

In some sense, robust control can be seen as the opposite method to adaptive control: Adapting the control usage (sensitivity) vs. adapting the control design.

That is the subject of the course 34746 Robust and fault-tolerant control.

Adaptive control - Method Overview

Several schools exist within adaptive control

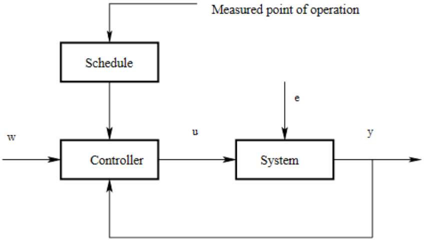


Adaptive control - Gain Scheduling Control

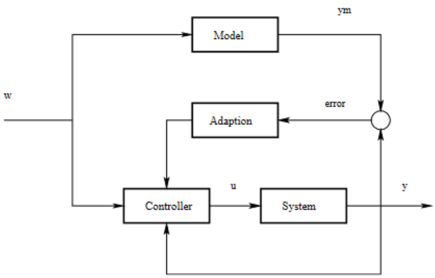
A simple approach is to manually change the model based on the operating point.

1. Linear control of non-linear system: Airplanes/robots.
2. Piecewise systems: Laws for behaviour at night vs day.

Adaptation is manual, so no performance feedback to the adaptations.



Adaptive control - Model Reference Adaptive Control (MRAC)



Another approach is to adapt the control until the output follows a desired transfer function with the least possible deviation.

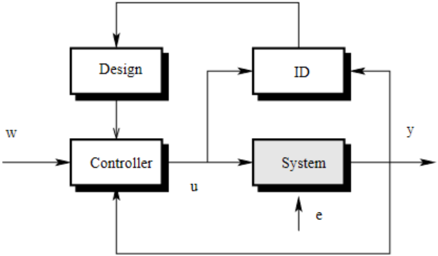
The focus is on the control problem and the adaptation is feedback on the model deviations.

The concept is similarly to that of an observer/Kalman filter.

Adaptive control - Self-tuning Control

The self-tuning methods are based on the combination of an identification algorithm, a design method, and a controller.

It is further assumed that the certainty equivalence principle holds.



Adaptive control - Certainty Equivalence Principle

The certainty equivalence principle, is the idea that the true value can be replaced by an estimate

$$\theta \to \hat{\theta} \tag{1}$$

For linear systems with additive noise, the principle holds, allowing for the state space control:

$$u_t = -Lx_t \quad \to \quad u_t = -L\hat{x}_t \tag{2}$$

This is the basis for the separation theorem between the design of Kalman filters and LQR.

In adaptive control, the principle is an assumption that allows us to formulate the minimum variance control as

$$C = AG + q^{-k}S \qquad \to \qquad \hat{C} = \hat{A}G + q^{-k}S \tag{3}$$

$$BGu_t = -Sy_t \qquad \to \qquad \hat{B}Gu_t = -Sy_t \tag{4}$$

In this case, the principle does not guarantee optimality, but it is assumed for convenience.

Basic Self-tuning

Let us now discuss the self-tuning methods in terms of the minimum variance controller, the so-called basic self-tuning controller.

We combine a recursive estimation approach for

$$A(q^{-1})y_t = q^{-k}B(q^{-1})u_t + C(q^{-1})e_t \tag{5}$$

$$B(q^{-1}) = b_0 + b_1 q^{-1} + \cdots + b_{n_b} q^{-n_b}, \quad b_0 \neq 0 \tag{6}$$

$$e_t \in \mathcal{F}(0, \sigma^2) \text{ and white} \tag{7}$$

with the design of the minimum variance controller for the objective

$$J = E\{y_{t+k}^2\} \tag{8}$$

$$u = func(Y_t) \tag{9}$$

Self-tuning methods come in two variants: Explicit and implicit.

1. Explicit: Estimation of model, used to design the control.
2. Implicit: Estimation of the control model (control parameters + $C(q^{-1})$).

The Basic Self tuner - Explicit

In the explicit method, we are interested in identifying the model

$$\hat{A}(q^{-1})y_t = q^{-k}\hat{B}(q^{-1})u_t + \hat{C}(q^{-1})\epsilon_t \tag{10}$$

to use in the control. We do this using a chosen estimation method:

$$y_t = \phi_t^T\theta_{t-1} + e_t \tag{11}$$

$$\hat{\theta}_t = \arg\min \sum_{i=1}^{t} \epsilon_i^2 \tag{12}$$

Using the estimate, we compute the control as

$$u_t = \arg\min E\{y_{t+k}^2\} \tag{13}$$

and we repeat at the next sampling time.

For a correct estimation ($\epsilon_t = e_t$), we have that the sum of control errors:

$$J_e(t) = \sum_{i=1}^{t} \epsilon_i^2 \simeq t\sigma^2 \tag{14}$$

The Basic Self tuner - Explicit: example

First, we apply RML estimation:

$$\phi_t = (-y_{t-1}, ..., u_{t-k}, ..., \epsilon_{t-1}, ...)^T, \quad \psi_t = \frac{1}{\hat{C}(q^{-1})}\phi_t \tag{15}$$

$$\theta = (a_1, ..., b_0, ..., c_1, ...)^T \tag{16}$$

$$P_t^{-1} = P_{t-1}^{-1} + \psi_t \psi_t^T \tag{17}$$

$$\epsilon_t = y_t - \phi_t^T \theta_{t-1} \tag{18}$$

$$\hat{\theta}_t = \hat{\theta}_{t-1} + P_t \psi_t \epsilon_t \tag{19}$$

Then, we obtain the control by solving the simple Diophantine equation,

$$\hat{C}(q^{-1}) = \hat{A}(q^{-1})G(q^{-1}) + q^{-k}S(q^{-1}) \tag{20}$$

$$R(q^{-1}) = \hat{B}(q^{-1})G(q^{-1}) \tag{21}$$

with the control law given by

$$u_t = -\frac{S}{R}y_t = -\frac{1}{r_0}(s_0 y_t + s_1 y_{t-1} + \cdots - r_1 u_{t-1} - r_2 u_{t-2} - \cdots) \tag{22}$$

The Basic Self tuner - Explicit: ARX example

Let us consider the simple ARX system with a single delay:

$$\hat{A}(q^{-1})y_t = q^{-1}\hat{B}(q^{-1})u_t + \epsilon_t \tag{23}$$

Our controller is then given by the Diophantine:

$$1 = \hat{A}(q^{-1}) + q^{-1}S(q^{-1}) \tag{24}$$

$$G(q^{-1}) = 1; \tag{25}$$

With the controller polynomials and law given as

$$S(q^{-1}) = q(1 - \hat{A}(q^{-1})) \tag{26}$$

$$R(q^{-1}) = B(q^{-1}) \tag{27}$$

$$u_t = -\frac{S}{R}y_t \tag{28}$$

The Basic Self tuner - Explicit: properties

The control loss function of the explicit self tuner:

$$J_r(t) = \sum_{i=1}^{t} y_t^2 \simeq E\{y_t^2\}t \tag{29}$$

$$J_u(t) = \sum_{i=1}^{t} u_t^2 \simeq E\{u_t^2\}t \tag{30}$$

For a correct estimate of the parameters, we have that $\epsilon_t = e_t$, therefore the residuals loss function follows

$$J_e(t) = \sum_{i=1}^{t} \epsilon_t^2 \simeq \sigma^2 t \tag{31}$$

Implicit Self tuning

In each step of the explicit version, we have to

1. estimate model parameters,
2. solve the Diophantine equation for the controller polynomials, R and S, and
3. compute the control, u_t.

To simplify the computation, we combine the Diophantine equation and the system equation:

$$Cy_{t+k} = [Ru_t + Sy_t] + CGe_{t+k} \tag{32}$$

to derive the implicit version. In each step of the implicit version, we

1. estimate the controller parameters and
2. compute the control, u_t.

Implicit Self tuning: ARX-model

For the ARX model, our control model is given by

$$y_{t+k} = [R(q^{-1})u_t + S(q^{-1})y_t] + G(q^{-1})e_{t+k} \tag{33}$$

The minimum-variance control drives the kth prediction to zero. Consequently,

$$R(q^{-1})u_t + S(q^{-1})y_t = 0 = \phi_{t+k}^T \theta \tag{34}$$

$$\theta = (s_0, s_1, ..., r_0, r_1, ...)^T \tag{35}$$

$$\phi_{t+k} = (y_t, y_{t-1}, ..., u_t, u_{t-1}, ...)^T \tag{36}$$

$$\phi_t = (y_{t-k}, y_{t-k-1}, ..., u_{t-k}, u_{t-k-1}, ...)^T \tag{37}$$

We can then apply an estimation method such as RLS:

$$\hat{\theta}_t : \quad y_t = \phi_t^T \hat{\theta} + \epsilon_t \tag{38}$$

$$u_t : \quad \phi_{t+k}^T \hat{\theta}_t = 0 \tag{39}$$

with the control being computed afterwards

Implicit Self tuning: ARMAX-model

For the ARMAX model, our control model is given by

$$y_{t+k} = \frac{1}{C(q^{-1})}[R(q^{-1})u_t + S(q^{-1})y_t] + G(q^{-1})e_{t+k} \tag{40}$$

The minimum-variance control drives the kth prediction to zero. Consequently,

$$R(q^{-1})u_t + S(q^{-1})y_t = \phi_t^T\theta = 0 \tag{41}$$

Furthermore, if the estimate $\hat{\theta}$ converges, it will reach parameters for which the regressors and residuals are uncorrelated. Therefore, we can formulate the model as

$$y_{t+k} = [R(q^{-1})u_t + S(q^{-1})y_t] + \epsilon_{t+k} = \phi_t^T\theta + \epsilon_{t+k} \tag{42}$$

$$\epsilon_t = G(q^{-1})e_t, \quad \epsilon_t \perp \phi_t \tag{43}$$

As it now has the shape of an ARX model, we can do the estimation and computation in the same manner which means that RLS can be used for the ARMAX-model as well.

Implicit Self tuning: ARMAX-model - simple proof

Consider control, system and model given by

$$R(q^{-1})u_{t-k} + S(q^{-1})y_{t-k} = \phi_t^T\theta = 0 \quad (44)$$

$$\mathcal{S} : y_t = \frac{1}{C(q^{-1})}\phi_t^T\theta + G(q^{-1})e_t \quad (45)$$

$$\mathcal{M} : y_t = \phi_t^T\hat{\theta} + \epsilon_t \quad (46)$$

If $\hat{\theta}$ converges to θ, we can formulate the model residual as

$$\epsilon_t = y_t - \phi_t^T\theta = \frac{1}{C(q^{-1})}\phi_t^T\theta + G(q^{-1})e_t - \phi_t^T\theta \quad (47)$$

$$= \frac{1 - C(q^{-1})}{C(q^{-1})}\phi_t^T\theta + G(q^{-1})e_t \quad (48)$$

Given the control design, the residuals become

$$\epsilon_t = G(q^{-1})e_t \quad (49)$$

$$E\{\phi_t\epsilon_t\} = 0 \quad (50)$$

where it can be seen that it is independent of the regressor.

Implicit self tuning: properties

Advantages:

1. Design is simple.
2. Can use RLS (even if $C \neq 1$).

Disadvantages:

1. More parameters to estimate $(k \gg 1)$.
2. Not all strategies can be transformed into an implicit strategy (model + design has to be combined).
3. Estimation must be restarted if the design choice changes.

QRS Controller Implementation

Consider a controller in the form

$$R(q^{-1})u_t = Q(q^{-1})w_t - S(q^{-1})y_t - \xi_0 \tag{51}$$

We can implement the controller using state-space representation, e.g. a canonical form:

$$X_{t+1}^r = A^r X_t^r + B^r \begin{bmatrix} y_t \\ w_t \end{bmatrix} \tag{52}$$

$$u_t = C^r X_t^r + D^r \begin{bmatrix} y_t \\ w_t \end{bmatrix} + u_0, \tag{53}$$

$$u_0 = (C_\xi (1 - A^r)^{-1} B_\xi + D_\xi)\xi_0 \tag{54}$$

where the matrices A^r, B^r, C^r, D^r can be computed using Q, R and S.

QRS Controller Implementation - direct approach

Alternatively, we can reformulate the controller in the form:

$$S(q^{-1})y_t + R(q^{-1})u_t - Q(q^{-1})w_t + \xi_0 = 0 \tag{55}$$

and rewriting it as

$$(u_t, \ldots, y_t, \ldots, -w_t, \ldots, 1) \times (r_0, \ldots, s_0, \ldots, q_0, \ldots, \xi_0)^T = 0 \tag{56}$$

the control is then found by isolating u_t:

$$u_t = -(0, u_{t-1}, \ldots, y_t, \ldots, -w_t, \ldots, 1) \times (r_0, \ldots, s_0, \ldots, q_0, \ldots, \xi_0)^T / r_0 \tag{57}$$

$$= -\phi_r^T \theta_r / r_0 \tag{58}$$

QRS Controller Implementation - direct approach

Recursively, the direct approach becomes

```
% Load system
[A, B, k, C, d, s2] = system();

% Design controller
[Q, R, S, G] = dsnmv0(A, B, k, C);

nr   = length([R, Q, S]) + 1;
phir = zeros(nr, 1);
thr  = [R, Q, S, G(1)*d]';
pil  = 1 + [0, length(R), length([R, Q])];

for it = 1:nstp,
    % Setpoint
    w = wt(it);

    % Measure output of sytem
    [y, t] = measure();

    % Ru = Q*w - S*y - G*d
    phir(2:end) =  phir(1:end-1);
    phir(pil)   =  [0, -w, y]
    u           = -phir'*thr/thr(1)
    phir(1)     = u;

    % Actuate control to the sytem
    actuate(u);
end
```

where the functions system, dsnmv0, measure, and actuate are pseudocode “placeholders”.

Explicit methods

We can now utilize the controllers we have discussed previously in the course to obtain more advanced adaptive controllers.

Explicit MV_0 Control

Identified system (general method):

$$\hat{A}(q^{-1})y_t = q^{-k}\hat{B}(q^{-1})u_t + \hat{C}(q^{-1})\epsilon_t + \hat{d} \tag{59}$$

Controller optimality criteria:

$$J = E\{(y_{t+k} - w_t)^2\} \tag{60}$$

Controller design:

$$\hat{B}(q^{-1})G(q^{-1})u_t = \hat{C}(q^{-1})w_t - S(q^{-1})y_t - G(1)\hat{d} \tag{61}$$

$$\hat{C}(q^{-1}) = \hat{A}(q^{-1})G(q^{-1}) + q^{-k}S(q^{-1}) \tag{62}$$

QRS form:

$$Q = \hat{C}(q^{-1}), \quad R = \hat{B}(q^{-1})G(q^{-1}), \quad \xi_0 = G(1)\hat{d} \tag{63}$$

Matlab

Let us look at some Matlab examples of the explicit MV_0 controller.

Explicit Pole Placement Control

Identified system (general method):

$$\hat{A}(q^{-1})y_t = q^{-k}\hat{B}(q^{-1})u_t + \hat{C}(q^{-1})\epsilon_t + \hat{d} \tag{64}$$

Controller optimality criteria:

$$J = E\{(A_m(q^{-1})y_{t+k} - B_{m1}(q^{-1})\hat{B}_-(q^{-1})w_t)^2\} \tag{65}$$

$$\hat{B}(q^{-1}) = \hat{B}_+(q^{-1})\hat{B}_-(q^{-1}) \tag{66}$$

Controller and Design:

$$\hat{B}_+(q^{-1})G(q^{-1})u_t = B_{m1}(q^{-1})\hat{C}(q^{-1})w_t - S(q^{-1})y_t - \frac{G(1)}{B_-(1)}\hat{d} \tag{67}$$

$$A_m(q^{-1})\hat{C}(q^{-1}) = \hat{A}(q^{-1})G(q^{-1}) + q^{-k}\hat{B}_-(q^{-1})S(q^{-1}) \tag{68}$$

QRS form:

$$Q = B_{m1}(q^{-1})\hat{C}(q^{-1}), \quad R = \hat{B}_+(q^{-1})G(q^{-1}), \quad \xi_0 = \frac{G(1)}{B_-(1)}\hat{d} \tag{69}$$

Explicit PZ Control - $B_- = 1$ & $B_+ = B$

Identified system (general method):

$$\hat{A}(q^{-1})y_t = q^{-k}\hat{B}(q^{-1})u_t + \hat{C}(q^{-1})\epsilon_t + \hat{d} \tag{70}$$

Controller optimality criteria:

$$J = E\{(A_m(q^{-1})y_{t+k} - B_m(q^{-1})w_t)^2\} \tag{71}$$

Controller and Design:

$$\hat{B}(q^{-1})G(q^{-1})u_t = B_m(q^{-1})\hat{C}(q^{-1})w_t - S(q^{-1})y_t - G(1)\hat{d} \tag{72}$$

$$A_m(q^{-1})\hat{C}(q^{-1}) = \hat{A}(q^{-1})G(q^{-1}) + q^{-k}S(q^{-1}) \tag{73}$$

QRS form:

$$Q = B_m(q^{-1})\hat{C}(q^{-1}), \quad R = \hat{B}(q^{-1})G(q^{-1}), \quad \xi_0 = G(1)\hat{d} \tag{74}$$

Explicit GMV Control

Identified system (general method):

$$\hat{A}(q^{-1})y_t = q^{-k}\hat{B}(q^{-1})u_t + \hat{C}(q^{-1})\epsilon_t + \hat{d} \tag{75}$$

Controller optimality criteria (monic denominators):

$$J = E\left\{\left(\frac{B_y(q^{-1})}{A_y(q^{-1})}y_{t+k} - \frac{B_w(q^{-1})}{A_w(q^{-1})}w_t\right)^2 + \rho\left(\frac{B_u(q^{-1})}{A_u(q^{-1})}u_t\right)^2\right\} \tag{76}$$

Controller and Design:

$$R(q^{-1})\breve{u}_t = Q(q^{-1})\breve{w}_t - S(q^{-1})\breve{y}_t - \xi_0 \tag{77}$$

$$B_y(q^{-1})\hat{C}(q^{-1}) = A_y(q^{-1})\hat{A}G(q^{-1}) + q^{-k}S(q^{-1}) \tag{78}$$

QRS form:

$$Q = \hat{C}, \quad R = A_u\hat{B}G + \frac{\rho}{\hat{b}_0}B_u\hat{C}, \quad \xi_0 = G(1)\hat{d} \tag{79}$$

$$\breve{u}_t = \frac{1}{A_u}u_t, \quad \breve{y}_t = \frac{1}{A_y}y_t, \quad \breve{w}_t = \frac{B_w}{A_w}w_t \tag{80}$$

Explicit LQG Control

Identified system (general method):

$$\hat{A}(q^{-1})y_t = q^{-k}\hat{B}(q^{-1})u_t + \hat{C}(q^{-1})\epsilon_t + \hat{d}, \quad \bar{B} = q^{-k}\hat{B} \tag{81}$$

Controller optimality criteria (monic denominators):

$$J = \lim_{N\to\infty} E\left\{\frac{1}{N}\sum_{i=t}^{N}(y_i - w_i)^2 + \rho u_t^2\right\} \tag{82}$$

Controller and Design:

$$R(q^{-1})u_t = Q(q^{-1})w_t - S(q^{-1})y_t - \xi_0 \tag{83}$$

$$P(q^{-1})P(q) = \bar{B}(q^{-1})\bar{B}(q) + \rho\hat{A}(q^{-1})\hat{A}(q) \quad \text{(spectral factorization)} \tag{84}$$

$$P(q^{-1})\hat{C}(q^{-1}) = \hat{A}R(q^{-1}) + q^{-k}\hat{B}(q^{-1})S(q^{-1}) \quad \text{(Diophantine)} \tag{85}$$

QRS form:

$$Q = \frac{P(1)}{\hat{B}(1)}\hat{C}, \quad R = A_u\hat{B}G + \frac{\rho}{\hat{b}_0}B_u\hat{C}, \quad \xi_0 = G(1)\hat{d} \tag{86}$$

Implicit methods

Similarly we can apply our controllers in an implicit manner.

Implicit MV_0 Control

System:

$$A(q^{-1})y_t = q^{-k}B(q^{-1})u_t + C(q^{-1})e_t + d \tag{87}$$

Controller optimality criteria:

$$J = E\{(y_{t+k} - w_t)^2\} \tag{88}$$

Estimation and control:

$$\zeta_t = y_t - w_{t-k} = \phi_t^T \hat{\theta}_{t-1} + \epsilon_t \tag{89}$$

$$\phi_{t+k}^T \hat{\theta}_t = 0 \tag{90}$$

$$\hat{\theta}_t^T = (s_0, ..., r_0, ..., q_0, ..., \xi) \tag{91}$$

$$\phi_t^T = (y_{t-k}, ..., u_{t-k}, ..., -w_{t-k}, ..., 1) \tag{92}$$

$$\phi_{t+k}^T = (y_t, ..., u_t, ..., -w_t, ..., 1) \tag{93}$$

The theoretical control law:

$$R(q^{-1})u_t = Q(q^{-1})w_t - S(q^{-1})y_t - \xi \tag{94}$$

$$R = BG, \quad Q = C, \quad S = q^k(C - AG), \quad \xi = G(1)d \tag{95}$$

Implicit PZ Control

System:

$$A(q^{-1})y_t = q^{-k}B(q^{-1})u_t + C(q^{-1})e_t + d \quad (96)$$

Controller optimality criteria:

$$J = E\{(A_m(q^{-1})y_{t+k} - B_m(q^{-1})w_t)^2\} \quad (97)$$

Estimation and control:

$$\zeta_t = A_m(q^{-1})y_t - B_m(q^{-1})w_{t-k} = \phi_t^T\hat{\theta}_{t-1} + \epsilon_t \quad (98)$$

$$\phi_{t+k}^T\hat{\theta}_t = 0 \quad (99)$$

$$\hat{\theta}_t^T = (s_0, ..., r_0, ..., q_0, ..., \xi) \quad (100)$$

$$\phi_t^T = (y_{t-k}, ..., u_{t-k}, ..., -w_{t-k}, ..., 1) \quad (101)$$

The theoretical control law:

$$R(q^{-1})u_t = Q(q^{-1})w_t - S(q^{-1})y_t - \xi \quad (102)$$

$$R = BG, \quad Q = B_mC, \quad S = q^k(A_mC - AG), \quad \xi = G(1)d \quad (103)$$

Implicit GMV Control

System:

$$A(q^{-1})y_t = q^{-k}B(q^{-1})u_t + C(q^{-1})e_t + d \tag{104}$$

Controller optimality criteria (monic denominators):

$$J = E\left\{\left(\frac{B_y(q^{-1})}{A_y(q^{-1})}y_{t+k} - \frac{B_w(q^{-1})}{A_w(q^{-1})}w_t\right)^2 + \rho\left(\frac{B_u(q^{-1})}{A_u(q^{-1})}u_t\right)^2\right\} \tag{105}$$

where we can define the variables:

$$\tilde{y}_t = \frac{B_y(q^{-1})}{A_y(q^{-1})}y_{t+k}, \quad \tilde{w}_t = \frac{B_w(q^{-1})}{A_w(q^{-1})}w_t, \quad \tilde{u}_t = \frac{\rho}{b_0}\frac{B_u(q^{-1})}{A_u(q^{-1})}u_t \tag{106}$$

Implicit GMV Control

Estimation and control:

$$\zeta_t = \tilde{y}_{t+k} - \tilde{w}_t + \frac{\rho}{\hat{b}_0}\tilde{u}_t = \phi_t^T \hat{\theta}_{t-1} + \epsilon_t \tag{107}$$

$$\phi_{t+k}^T \hat{\theta}_t = 0 \tag{108}$$

$$u_t = A_u(q^{-1})\check{u}_t \tag{109}$$

$$\hat{\theta}_t^T = (s_0, ..., r_0, ..., q_0, ..., \xi) \tag{110}$$

$$\phi_t^T = (\check{y}_{t-k}, ..., \check{u}_{t-k}, ..., -\check{w}_{t-k}, ..., 1) \tag{111}$$

The theoretical control law:

$$R(q^{-1})\check{u}_t = Q(q^{-1})\check{w}_t - S(q^{-1})\check{y}_t - \xi \tag{112}$$

$$R = A_u BG + \frac{\rho}{b_0} B_u C, \quad Q = C, \quad S = q^k(B_y C - A_y A G), \tag{113}$$

$$\xi = G(1)d \tag{114}$$

$$\check{u}_t = \frac{1}{A_u}u_t, \quad \check{y}_t = \frac{1}{A_y}y_t, \quad \check{w}_t = \frac{B_w}{A_w}w_t \tag{115}$$

Examples

Let us return to Matlab and look at the implicit implementation of a PZ controller.

Questions

Questions?