Stochastic Adaptive Control (02421)

Lecture 12

Tobias K. S. Ritschel

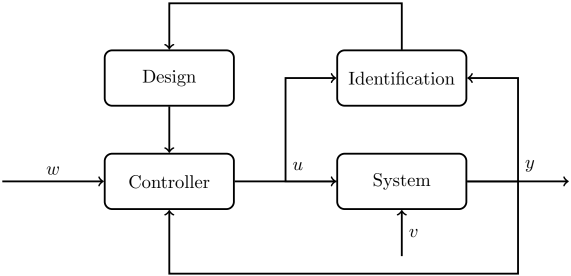
Section for Dynamical Systems

Department of Applied Mathematics and Computer Science

Technical University of Denmark

Lecture Plan

1. System theory
2. Stochastics
3. State estimation 1
4. State estimation 2
5. Optimal control 1
6. System identification 1 + adaptive control 1
7. External models + prediction
8. Optimal control 2
9. Optimal control 3
10. System identification 2
11. System identification 3 + model validation
12. **System identification 4 + adaptive control 2**
13. Adaptive control 3



Today's Agenda

- Closed-loop system identification
- Explicit self-tuning controllers (STCs)
- Implicit self-tuning controllers (STCs)

Follow-up from last time

Questions?

Closed-loop system identification

Closed-Loop Identification - why?

Why closed-loop identification?

1. Unstable open-loop system
2. Running process (production/economics)
3. Safety
4. Adaptive control

Closed-Loop Identification - Pitfalls

System and control law

$$y_t + a y_{t-1} = b u_{t-1} + e_t, \quad e_t \sim N_{iid}(0, \sigma^2) \tag{1}$$
$$u_t = -f y_t \tag{2}$$

Closed-loop system (time domain)

$$y_t + (a + bf) y_{t-1} = e_t \tag{3}$$

The parameter estimate is not unique

$$a = a_0 + \gamma f, \tag{4}$$
$$b = b_0 - \gamma \tag{5}$$

γ is arbitrary and the controller is therefore too simple

Closed-Loop Identification

Question: How complex should a controller be?

ARMAX model

$$A(q^{-1})y_t = q^{-k}B(q^{-1})u_t + C(q^{-1})e_t, \qquad e_t \sim N_{iid}(0,\sigma^2) \tag{6}$$

Control law

$$u_t = -\frac{S(q^{-1})}{R(q^{-1})}y_t \tag{7}$$

Closed-loop system

$$\left(A(q^{-1})R(q^{-1}) + q^{-k}B(q^{-1})S(q^{-1})\right)y_t = R(q^{-1})C(q^{-1})e_t \tag{8}$$

A controller is sufficiently complex if the order of the closed-loop system is higher than the number of parameters to be estimated

$$\max(n_r - n_b, n_s + k - n_a) \geq 1 + n_p \tag{9}$$

n_p is the number of common factors in RC and $AR + q^{-k}BS$

Closed-Loop Identification - examples

System (revisited)

$$y_t + ay_{t-1} = bu_{t-1} + e_t, \quad e_t \sim N_{iid}(0, \sigma^2) \tag{10}$$

$$u_t = -fy_t \tag{11}$$

Polynomial orders

$$n_a = 1, \quad n_b = 0, \quad k = 1, \quad n_r = 0, \quad n_s = 0, \quad n_p = 0 \tag{12}$$

Evaluate controller

$$\max(n_r - n_b, n_s + k - n_a) \geq 1 + n_p \tag{13}$$

$$\max(0 - 0, 0 + 1 - 1) \geq 1 + 0 \tag{14}$$

$$0 \geq 1 \tag{15}$$

The order is too low to identify the system

Closed-Loop Identification - examples: Minimum Variance

ARMAX model

$$A(q^{-1})y_t = q^{-k}B(q^{-1})u_t + C(q^{-1})e_t, \quad e_t \sim N_{iid}(0,\sigma^2) \tag{16}$$

Minimum variance controller

$$J = \mathbb{E}[y_{t+k}^2], \qquad y_{t+k} = \frac{1}{C}(BGu_t + Sy_t) + Ge_{t+k} \tag{17}$$

Control law and closed-loop system

$$u_t = -\frac{S}{BG}y_t \qquad (AG + q^{-k}S)y_t = CGe_t \tag{18}$$

Polynomial orders

$$n_r = n_b + k - 1, \qquad n_s = n_a - 1, \qquad n_p = n_c \tag{19}$$

Evaluate polynomial orders

$$\max(n_r - n_b, n_s + k - n_a) = \max(k-1, k-1) \geq 1 + n_c, \tag{20}$$

$$k \geq n_c + 2 \tag{21}$$

Closed-Loop Identification - informative Data

- We say that the examples were not sufficiently informative (for $k < n_c + 2$)
- As in the open-loop case, the data must be sufficiently informative in order to identify the system
- A data set z_t is sufficiently informative if

$$\overline{\mathbb{E}}[||(\mathcal{M}_1 - \mathcal{M}_2)z_t||^2] = 0 \quad \Rightarrow \quad \mathcal{M}_1(w) \equiv \mathcal{M}_2(w) \tag{22}$$

  such that we can distinguish between two models, $\mathcal{M}_i \in \mathcal{M}$

- In general, the closed-loop experiment is informative if the reference w_t (or another probing signal) is persistently exciting
- Similarly, time-invariant, nonlinear or higher order feedback controllers should also provide informative experiments

Closed-Loop Identification - challenges

Potential pitfalls specific to closed-loop identification

1. The closed-loop experiment may be non-informative even if the input in itself is persistently exciting. Reason: the controller might be too simple
2. Spectral analysis applied in a straightforward manner will give erroneous results. The estimate of G will converge to

$$G^* = \frac{G_0 \phi_w - F \phi_v}{\phi_w + |F|^2 \phi_v}, \quad y = Gu + v \tag{23}$$

3. Correlation analysis will give a biased estimate of the impulse response because the assumption that $\mathbb{E}[u_t v_{t-k}] = 0$ is violated.
4. OE methods give unbiased estimates of G in open-loop experiments, even if the additive noise (v) is not white. This is not true in closed-loop.

Closed-Loop Identification - Direct approach

The system is identified in exactly the same way as in open-loop identification, using the data set $[y, u]$ and ignoring any information about the feedback structures

Strengths

1. It works regardless of the complexity of the controller and requires no knowledge about the character of the feedback
2. No special algorithms or software are required
3. Consistency and optimal accuracy are obtained if model structure contains the true system
4. Unstable systems can be handled without problems (as long as the closed-loop system and the predictor are stable)

Drawbacks

1. We need good noise models. (Not a problem if the true system (G, H) is contained in model structure.)
2. If noise model is incorrect (fixed incorrectly or not containing the true noise model), a bias in G will be introduced.

Closed-Loop Identification - Indirect Approach

Closed-loop transfer functions

$$y_t = Gu_t + He_t, \quad u_t = w_t - Fy_t \tag{24}$$

$$y_t = G_{cl}w_t + H_{cl}e_t, \quad G_{cl} = GS, \quad H_{cl} = HS, \quad S = \frac{1}{1+FG} \tag{25}$$

Derive the identified system (the control law is known)

$$\hat{G} = \frac{\hat{G}_{cl}}{1 - F\hat{G}_{cl}}, \quad \hat{H} = \hat{H}_{cl}(1 + F\hat{G}) \tag{26}$$

Properties of the indirect approach

1. $+$ Any (open-loop) method such as spectral analysis, instrumental variable, subspace and prediction error methods can be applied.
2. $-$ Any error in F will directly affect the estimate of the model (e.g., saturation, manual operation).

Closed-Loop Identification - joint input-output

Full closed-loop description

$$y_t = GSw_t + HSe_t + GSz_t = GSw_t + v_{1,t} \tag{27}$$

$$u_t = Sw_t - FSHe_t + Sz_t = Sw_t + v_{2,t} \tag{28}$$

where z_t is a partial unknown signal part of u_t

In this approach, we utilize the structure of both the input and output to estimate the closed-loop system and the sensitivity function S

Closed-Loop Identification - joint input-output V1

First version: Take correlation between v_1 and v_2 into account

$$\begin{bmatrix} y_t \\ u_t \end{bmatrix} = S \begin{bmatrix} G \\ 1 \end{bmatrix} w_t + S \begin{bmatrix} H & G \\ -FH & 1 \end{bmatrix} \begin{bmatrix} e_t \\ z_t \end{bmatrix} = \mathcal{G} w_t + \mathcal{H} \begin{bmatrix} e_t \\ z_t \end{bmatrix} \tag{29}$$

Using the variance of $[e_t, z_t]^T$, we can estimate the system parameters using, e.g., a ML or PEM method

$$J = \sum_{i=1}^{t} \epsilon_i^T R^{-1} \epsilon_i, \quad R = \mathrm{Var}\left(\begin{bmatrix} e_i \\ z_i \end{bmatrix} \right), \quad \epsilon_i = \mathcal{H}^{-1} \left(\begin{bmatrix} y_i \\ u_i \end{bmatrix} - \mathcal{G} w_i \right) \tag{30}$$

This is essentially the direct approach extended to controller estimation

Closed-Loop Identification - joint input-output V2

Second version: Disregard correlation

$$\begin{bmatrix} y_t \\ u_t \end{bmatrix} = \begin{bmatrix} G_{cl} \\ G_{uw} \end{bmatrix} w_t + \begin{bmatrix} v_{1,t} \\ v_{2,t} \end{bmatrix} \tag{31}$$

Cost function

$$J = \frac{1}{\sigma_1^2} \sum_{i=1}^{t} (y_i - G_{cl} w_i)^2 + \frac{1}{\sigma_2^2} \sum_{i=1}^{t} (u_i - G_{uw} w_i)^2, \tag{32}$$

The system can then be determined by

$$\hat{G} = \frac{\hat{G}_{cl}}{\hat{G}_{wu}} \approx \frac{\hat{G}\hat{S}}{\hat{S}} \tag{33}$$

Given these polynomials are estimated the cancellation is not perfect, making $\hat{G}$ higher order than it should be

A trick for mitigating this is to use independent parametrization of G and S: $G(\theta)$ and $S(\eta)$

Closed-Loop Identification - joint input-output V2 - 2 step

For the independently parametrized estimation, we consider the cost

$$J = \beta \frac{1}{\sigma_1^2} \sum_{i=1}^{t} (y_i - G(\theta)S(\eta)w_i)^2 + \sum_{i=1}^{t} (u_i - S(\eta)w_i)^2, \tag{34}$$

Estimation in two steps

1. first for $\beta = 0$, to estimate the parameters of $S(\eta)$
2. then use $\hat{u}_t = S(\hat{\eta})w_t$ to estimate $G(\theta)$ from $y_t = G(\theta)\hat{u}_t + v_{1,t}$

One previously suggested parametrization of S is the non-causal filter

$$S(\eta) = \sum_{i=-m}^{m} s_k q^{-k} \tag{35}$$

Adaptive control

Adaptive Control

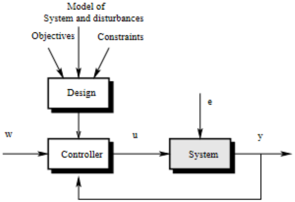
Stochastic control relies on a detailed model which might not be available

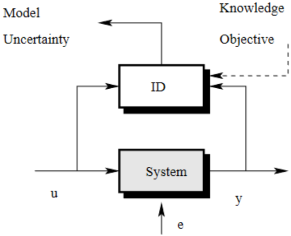
1. Parameter values cannot be measured
2. The underlying physics is not understood sufficiently well

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

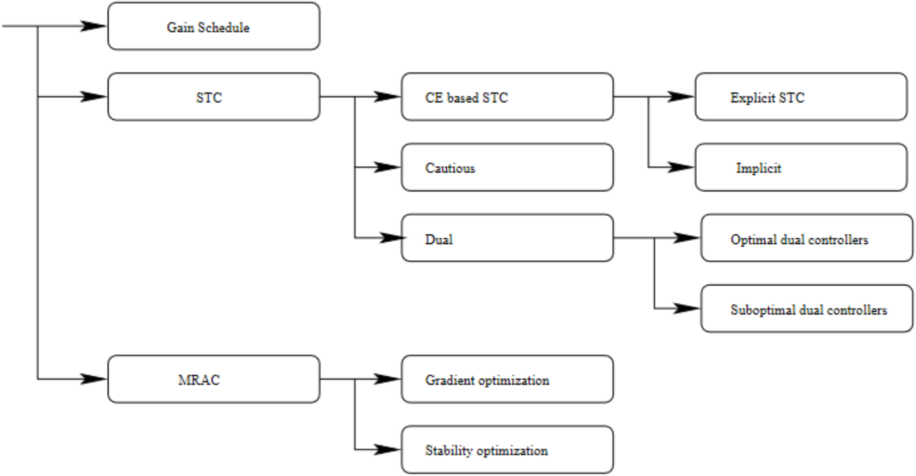
1. Robust control: Low sensitivity to the effect of uncertain system parts, a control that, in some sense, operates after worst-case scenario
2. Adaptive control: Monitors/estimates the uncertain parts, a control law that changes with the identified system.

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

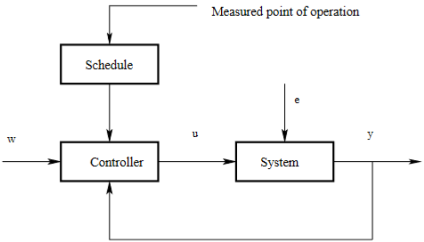


Adaptive control - Gain Scheduling Control

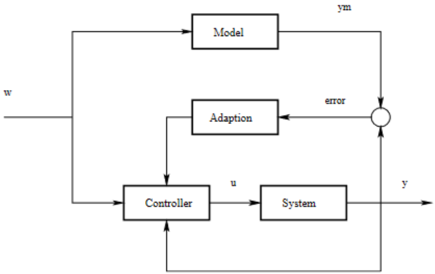
A simple approach is to manually change the model based on the operating point

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

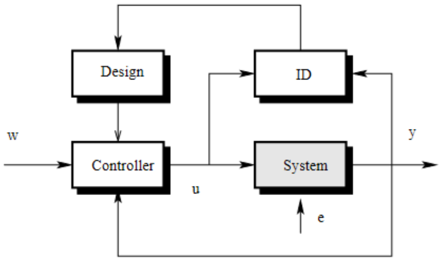
The focus is on the control problem and the adaptation is feedback on the model deviations

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

Certainty equivalence principle: Replace true parameters by an estimate

$$\theta \rightarrow \hat{\theta} \tag{36}$$

For linear systems with additive noise, the principle holds

$$u_t = -Lx_t \quad \rightarrow \quad u_t = -L\hat{x}_t \tag{37}$$

In adaptive control, the principle is an assumption (minimum variance control example)

$$C = AG + q^{-k}S \qquad \rightarrow \qquad \hat{C} = \hat{A}G + q^{-k}S \tag{38}$$

$$BGu_t = -Sy_t \qquad \rightarrow \qquad \hat{B}Gu_t = -Sy_t \tag{39}$$

The principle does not guarantee optimality – it is assumed for convenience

Explicit self-tuning controllers

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

$$A(q^{-1})y_t = q^{-k}B(q^{-1})u_t + C(q^{-1})e_t \tag{40}$$

$$B(q^{-1}) = b_0 + b_1 q^{-1} + \cdots + b_{n_b} q^{-n_b}, \quad b_0 \neq 0 \tag{41}$$

$$e_t \sim \mathcal{F}(0, \sigma^2) \text{ and white} \tag{42}$$

with the design of the minimum variance controller for the objective

$$J = \mathbb{E}[y_{t+k}^2] \tag{43}$$

$$u = \mathrm{func}(Y_t) \tag{44}$$

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 controller parameters + $C(q^{-1})$

The Basic Self tuner - Explicit

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

$$\hat{A}(q^{-1})y_t = q^{-k}\hat{B}(q^{-1})u_t + \hat{C}(q^{-1})\epsilon_t \tag{45}$$

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

$$y_t = \phi_t^T \theta_{t-1} + e_t \tag{46}$$

$$\hat{\theta}_t = \arg\min \sum_{i=1}^{t} \epsilon_i^2 \tag{47}$$

Using the estimate, we compute the control as

$$u_t = \arg\min \mathbb{E}[y_{t+k}^2] \tag{48}$$

and we repeat at the next sampling time.

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

$$J_e(t) = \sum_{i=1}^{t} \epsilon_i^2 \simeq t\sigma^2 \tag{49}$$

The Basic Self tuner - Explicit: example

First, we apply RML estimation:

$$\phi_t = (-y_{t-1}, ..., u_{t-k}, ..., \epsilon_{t-1}, ...)^T, \quad \psi_t = \frac{1}{\hat{C}(q^{-1})}\phi_t \tag{50}$$

$$\theta = (a_1, ..., b_0, ..., c_1, ...)^T \tag{51}$$

$$P_t^{-1} = P_{t-1}^{-1} + \psi_t \psi_t^T \tag{52}$$

$$\epsilon_t = y_t - \phi_t^T \theta_{t-1} \tag{53}$$

$$\hat{\theta}_t = \hat{\theta}_{t-1} + P_t \psi_t \epsilon_t \tag{54}$$

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

$$\hat{C}(q^{-1}) = \hat{A}(q^{-1})G(q^{-1}) + q^{-k}S(q^{-1}) \tag{55}$$

$$R(q^{-1}) = \hat{B}(q^{-1})G(q^{-1}) \tag{56}$$

with the control law given by

$$u_t = -\frac{S}{R}y_t = -\frac{1}{r_0}(s_0 y_t + s_1 y_{t-1} + \cdots - r_1 u_{t-1} - r_2 u_{t-2} - \cdots) \tag{57}$$

The Basic Self tuner - Explicit: ARX example

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

$$\hat{A}(q^{-1})y_t = q^{-1}\hat{B}(q^{-1})u_t + \epsilon_t \tag{58}$$

Our controller is then given by the Diophantine:

$$1 = \hat{A}(q^{-1}) + q^{-1}S(q^{-1}) \tag{59}$$

$$G(q^{-1}) = 1; \tag{60}$$

With the controller polynomials and law given as

$$S(q^{-1}) = q(1 - \hat{A}(q^{-1})) \tag{61}$$

$$R(q^{-1}) = B(q^{-1}) \tag{62}$$

$$u_t = -\frac{S}{R}y_t \tag{63}$$

The Basic Self tuner - Explicit: properties

The control loss function of the explicit self tuner:

$$J_r(t) = \sum_{i=1}^{t} y_t^2 \simeq \mathbb{E}[y_t^2]t \tag{64}$$

$$J_u(t) = \sum_{i=1}^{t} u_t^2 \simeq \mathbb{E}[u_t^2]t \tag{65}$$

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

$$J_e(t) = \sum_{i=1}^{t} \epsilon_t^2 \simeq \sigma^2 t \tag{66}$$

Explicit MV_0 Control

Identified system (general method):

$$\hat{A}(q^{-1})y_t = q^{-k}\hat{B}(q^{-1})u_t + \hat{C}(q^{-1})\epsilon_t + \hat{d} \tag{67}$$

Controller optimality criteria:

$$J = \mathbb{E}[(y_{t+k} - w_t)^2] \tag{68}$$

Controller design:

$$\hat{B}(q^{-1})G(q^{-1})u_t = \hat{C}(q^{-1})w_t - S(q^{-1})y_t - G(1)\hat{d} \tag{69}$$

$$\hat{C}(q^{-1}) = \hat{A}(q^{-1})G(q^{-1}) + q^{-k}S(q^{-1}) \tag{70}$$

QRS form:

$$Q = \hat{C}(q^{-1}), \quad R = \hat{B}(q^{-1})G(q^{-1}), \quad \xi_0 = G(1)\hat{d} \tag{71}$$

Explicit Pole Placement Control

Identified system (general method):

$$\hat{A}(q^{-1})y_t = q^{-k}\hat{B}(q^{-1})u_t + \hat{C}(q^{-1})\epsilon_t + \hat{d} \quad (72)$$

Controller optimality criteria:

$$J = \mathbb{E}[(A_m(q^{-1})y_{t+k} - B_{m1}(q^{-1})\hat{B}_-(q^{-1})w_t)^2] \quad (73)$$

$$\hat{B}(q^{-1}) = \hat{B}_+(q^{-1})\hat{B}_-(q^{-1}) \quad (74)$$

Controller and Design:

$$\hat{B}_+(q^{-1})G(q^{-1})u_t = B_{m1}(q^{-1})\hat{C}(q^{-1})w_t - S(q^{-1})y_t - \frac{G(1)}{B_-(1)}\hat{d} \quad (75)$$

$$A_m(q^{-1})\hat{C}(q^{-1}) = \hat{A}(q^{-1})G(q^{-1}) + q^{-k}\hat{B}_-(q^{-1})S(q^{-1}) \quad (76)$$

QRS form:

$$Q = B_{m1}(q^{-1})\hat{C}(q^{-1}), \quad R = \hat{B}_+(q^{-1})G(q^{-1}), \quad \xi_0 = \frac{G(1)}{B_-(1)}\hat{d} \quad (77)$$

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

Identified system (general method):

$$\hat{A}(q^{-1})y_t = q^{-k}\hat{B}(q^{-1})u_t + \hat{C}(q^{-1})\epsilon_t + \hat{d} \tag{78}$$

Controller optimality criteria:

$$J = \mathbb{E}[(A_m(q^{-1})y_{t+k} - B_m(q^{-1})w_t)^2] \tag{79}$$

Controller and Design:

$$\hat{B}(q^{-1})G(q^{-1})u_t = B_m(q^{-1})\hat{C}(q^{-1})w_t - S(q^{-1})y_t - G(1)\hat{d} \tag{80}$$

$$A_m(q^{-1})\hat{C}(q^{-1}) = \hat{A}(q^{-1})G(q^{-1}) + q^{-k}S(q^{-1}) \tag{81}$$

QRS form:

$$Q = B_m(q^{-1})\hat{C}(q^{-1}), \quad R = \hat{B}(q^{-1})G(q^{-1}), \quad \xi_0 = G(1)\hat{d} \tag{82}$$

Explicit GMV Control

Identified system (general method):

$$\hat{A}(q^{-1})y_t = q^{-k}\hat{B}(q^{-1})u_t + \hat{C}(q^{-1})\epsilon_t + \hat{d} \tag{83}$$

Controller optimality criteria (monic denominators):

$$J = \mathbb{E}\Bigg[\bigg(\frac{B_y(q^{-1})}{A_y(q^{-1})}y_{t+k} - \frac{B_w(q^{-1})}{A_w(q^{-1})}w_t\bigg)^2 + \rho\bigg(\frac{B_u(q^{-1})}{A_u(q^{-1})}u_t\bigg)^2\Bigg] \tag{84}$$

Controller and Design:

$$R(q^{-1})\check{u}_t = Q(q^{-1})\check{w}_t - S(q^{-1})\check{y}_t - \xi_0 \tag{85}$$

$$B_y(q^{-1})\hat{C}(q^{-1}) = A_y(q^{-1})\hat{A}G(q^{-1}) + q^{-k}S(q^{-1}) \tag{86}$$

QRS form:

$$Q = \hat{C}, \quad R = A_u\hat{B}G + \frac{\rho}{\hat{b}_0}B_u\hat{C}, \quad \xi_0 = G(1)\hat{d} \tag{87}$$

$$\check{u}_t = \frac{1}{A_u}u_t, \quad \check{y}_t = \frac{1}{A_y}y_t, \quad \check{w}_t = \frac{B_w}{A_w}w_t \tag{88}$$

Explicit LQG Control

Identified system (general method):

$$\hat{A}(q^{-1})y_t = q^{-k}\hat{B}(q^{-1})u_t + \hat{C}(q^{-1})\epsilon_t + \hat{d}, \quad \bar{B} = q^{-k}\hat{B} \tag{89}$$

Controller optimality criteria (monic denominators):

$$J = \lim_{N\to\infty} \mathbb{E}\Big[\frac{1}{N}\sum_{i=t}^{N}(y_i - w_i)^2 + \rho u_t^2\Big] \tag{90}$$

Controller and Design:

$$R(q^{-1})u_t = Q(q^{-1})w_t - S(q^{-1})y_t - \xi_0 \tag{91}$$

$$P(q^{-1})P(q) = \bar{B}(q^{-1})\bar{B}(q) + \rho\hat{A}(q^{-1})\hat{A}(q) \quad \text{(spectral factorization)} \tag{92}$$

$$P(q^{-1})\hat{C}(q^{-1}) = \hat{A}R(q^{-1}) + q^{-k}\hat{B}(q^{-1})S(q^{-1}) \quad \text{(Diophantine)} \tag{93}$$

QRS form:

$$Q = \frac{P(1)}{\hat{B}(1)}\hat{C}, \quad R = A_u\hat{B}G + \frac{\rho}{\hat{b}_0}B_u\hat{C}, \quad \xi_0 = G(1)\hat{d} \tag{94}$$

Matlab

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

Implicit self-tuning controllers

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:

$$Cy_{t+k} = [Ru_t + Sy_t] + CGe_{t+k} \tag{95}$$

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

$$y_{t+k} = [R(q^{-1})u_t + S(q^{-1})y_t] + G(q^{-1})e_{t+k} \tag{96}$$

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

$$R(q^{-1})u_t + S(q^{-1})y_t = 0 = \phi_{t+k}^T\theta \tag{97}$$

$$\theta = (s_0, s_1, ..., r_0, r_1, ...)^T \tag{98}$$

$$\phi_{t+k} = (y_t, y_{t-1}, ..., u_t, u_{t-1}, ...)^T \tag{99}$$

$$\phi_t = (y_{t-k}, y_{t-k-1}, ..., u_{t-k}, u_{t-k-1}, ...)^T \tag{100}$$

We can then apply an estimation method such as RLS:

$$\hat{\theta}_t : \quad y_t = \phi_t^T\hat{\theta} + \epsilon_t \tag{101}$$

$$u_t : \quad \phi_{t+k}^T\hat{\theta}_t = 0 \tag{102}$$

with the control being computed afterwards

Implicit Self tuning: ARMAX-model

For the ARMAX model, our control model is given by

$$y_{t+k} = \frac{1}{C(q^{-1})}[R(q^{-1})u_t + S(q^{-1})y_t] + G(q^{-1})e_{t+k} \tag{103}$$

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

$$R(q^{-1})u_t + S(q^{-1})y_t = \phi_t^T\theta = 0 \tag{104}$$

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

$$y_{t+k} = [R(q^{-1})u_t + S(q^{-1})y_t] + \epsilon_{t+k} = \phi_t^T\theta + \epsilon_{t+k} \tag{105}$$

$$\epsilon_t = G(q^{-1})e_t, \quad \epsilon_t \perp \phi_t \tag{106}$$

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

$$R(q^{-1})u_{t-k} + S(q^{-1})y_{t-k} = \phi_t^T\theta = 0 \tag{107}$$

$$\mathcal{S} : y_t = \frac{1}{C(q^{-1})}\phi_t^T\theta + G(q^{-1})e_t \tag{108}$$

$$\mathcal{M} : y_t = \phi_t^T\hat{\theta} + \epsilon_t \tag{109}$$

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

$$\epsilon_t = y_t - \phi_t^T\theta = \frac{1}{C(q^{-1})}\phi_t^T\theta + G(q^{-1})e_t - \phi_t^T\theta \tag{110}$$

$$= \frac{1 - C(q^{-1})}{C(q^{-1})}\phi_t^T\theta + G(q^{-1})e_t \tag{111}$$

Given the control design, the residuals become

$$\epsilon_t = G(q^{-1})e_t \tag{112}$$

$$E[\phi_t\epsilon_t] = 0 \tag{113}$$

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

Implicit MV_0 Control

System:

$$A(q^{-1})y_t = q^{-k}B(q^{-1})u_t + C(q^{-1})e_t + d \quad (114)$$

Controller optimality criteria:

$$J = \mathbb{E}[(y_{t+k} - w_t)^2] \quad (115)$$

Estimation and control:

$$\zeta_t = y_t - w_{t-k} = \phi_t^T \hat{\theta}_{t-1} + \epsilon_t \quad (116)$$

$$\phi_{t+k}^T \hat{\theta}_t = 0 \quad (117)$$

$$\hat{\theta}_t^T = (s_0, ..., r_0, ..., q_0, ..., \xi) \quad (118)$$

$$\phi_t^T = (y_{t-k}, ..., u_{t-k}, ..., -w_{t-k}, ..., 1) \quad (119)$$

$$\phi_{t+k}^T = (y_t, ..., u_t, ..., -w_t, ..., 1) \quad (120)$$

The theoretical control law:

$$R(q^{-1})u_t = Q(q^{-1})w_t - S(q^{-1})y_t - \xi \quad (121)$$

$$R = BG, \quad Q = C, \quad S = q^k(C - AG), \quad \xi = G(1)d \quad (122)$$

Implicit PZ Control

System:

$$A(q^{-1})y_t = q^{-k}B(q^{-1})u_t + C(q^{-1})e_t + d \tag{123}$$

Controller optimality criteria:

$$J = \mathbb{E}[(A_m(q^{-1})y_{t+k} - B_m(q^{-1})w_t)^2] \tag{124}$$

Estimation and control:

$$\zeta_t = A_m(q^{-1})y_t - B_m(q^{-1})w_{t-k} = \phi_t^T\hat{\theta}_{t-1} + \epsilon_t \tag{125}$$

$$\phi_{t+k}^T\hat{\theta}_t = 0 \tag{126}$$

$$\hat{\theta}_t^T = (s_0, ..., r_0, ..., q_0, ..., \xi) \tag{127}$$

$$\phi_t^T = (y_{t-k}, ..., u_{t-k}, ..., -w_{t-k}, ..., 1) \tag{128}$$

The theoretical control law:

$$R(q^{-1})u_t = Q(q^{-1})w_t - S(q^{-1})y_t - \xi \tag{129}$$

$$R = BG, \quad Q = B_mC, \quad S = q^k(A_mC - AG), \quad \xi = G(1)d \tag{130}$$

Implicit GMV Control

System:

$$A(q^{-1})y_t = q^{-k}B(q^{-1})u_t + C(q^{-1})e_t + d \tag{131}$$

Controller optimality criteria (monic denominators):

$$J = \mathbb{E}\Big[\Big(\frac{B_y(q^{-1})}{A_y(q^{-1})}y_{t+k} - \frac{B_w(q^{-1})}{A_w(q^{-1})}w_t\Big)^2 + \rho\Big(\frac{B_u(q^{-1})}{A_u(q^{-1})}u_t\Big)^2\Big] \tag{132}$$

Filtered variables

$$\tilde{y}_t = \frac{B_y(q^{-1})}{A_y(q^{-1})}y_t, \quad \tilde{w}_t = \frac{B_w(q^{-1})}{A_w(q^{-1})}w_t, \quad \tilde{u}_t = \frac{\rho}{b_0}\frac{B_u(q^{-1})}{A_u(q^{-1})}u_t \tag{133}$$

Implicit GMV Control

Estimation and control:

$$\zeta_{t+k} = \tilde{y}_{t+k} - \tilde{w}_t + \frac{\rho}{\hat{b}_0}\tilde{u}_t = \phi_t^T \hat{\theta}_{t-1} + \epsilon_t \tag{134}$$

$$\phi_{t+k}^T \hat{\theta}_t = 0 \tag{135}$$

$$u_t = A_u(q^{-1})\check{u}_t \tag{136}$$

$$\hat{\theta}_t^T = (s_0, ..., r_0, ..., q_0, ..., \xi) \tag{137}$$

$$\phi_t^T = (\check{y}_{t-k}, ..., \check{u}_{t-k}, ..., -\check{w}_{t-k}, ..., 1) \tag{138}$$

Theoretical control law:

$$R(q^{-1})\check{u}_t = Q(q^{-1})\check{w}_t - S(q^{-1})\check{y}_t - \xi \tag{139}$$

$$R = A_u BG + \frac{\rho}{b_0} B_u C, \quad Q = C, \quad S = q^k (B_y C - A_y AG), \tag{140}$$

$$\xi = G(1)d \tag{141}$$

$$\check{u}_t = \frac{1}{A_u} u_t, \quad \check{y}_t = \frac{1}{A_y} y_t, \quad \check{w}_t = \frac{B_w}{A_w} w_t \tag{142}$$

Implicit GMV Control

1. Measure y_t
2. Create $\zeta_t = \tilde{y}_t - \tilde{w}_{t-k} + \alpha \tilde{u}_{t-k}$
3. Create $\check{y}_t$, $\check{u}_t$, and $\check{w}_t$
4. Create $\phi_t = (\check{y}_{t-k}, \ldots, \check{u}_{t-k}, \ldots, -\check{w}_{t-k}, \ldots, 1)^T$ and ϕ_{t+k}
5. Update the parameter estimate:

$$\epsilon_t = \zeta_t - \phi_t^T \hat{\theta}_{t-1}, \tag{143}$$

$$P_t^{-1} = P_{t-1}^{-1} + \phi_t \phi_t^T, \tag{144}$$

$$\hat{\theta}_t = \hat{\theta}_{t-1} + P_t \phi_t \epsilon_t \tag{145}$$

6. Determine $\check{u}_t$ such that $\phi_{t+k}^T \hat{\theta}_t = 0$
7. Determine $u_t = A_u(q^{-1}) \check{u}_t$

Examples

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

Questions

Questions?