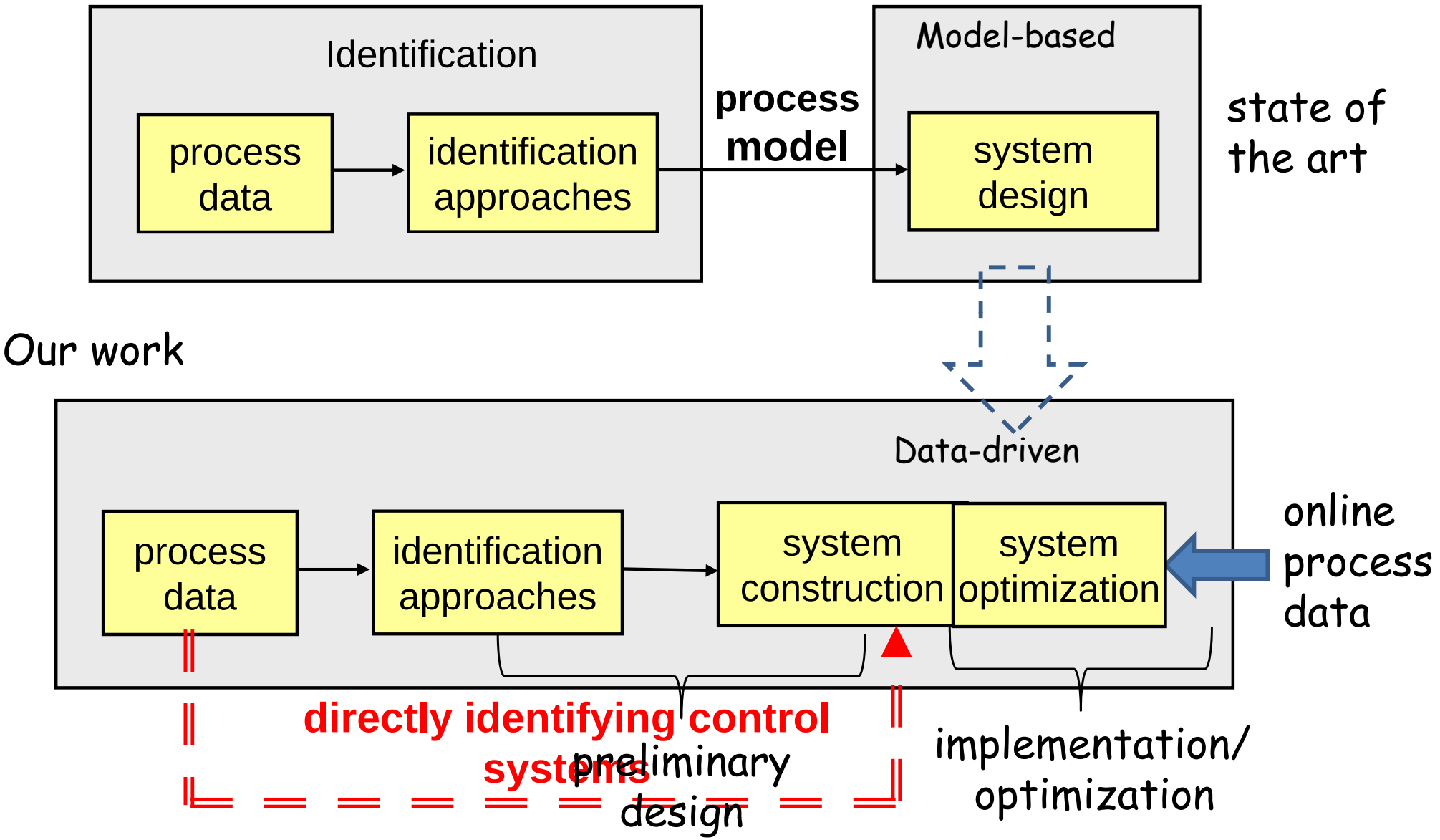
Data-driven fault diagnosis and fault-tolerant control

Steven X. Ding

Institute for automatic control and complex systems (AKS)
University of Duisburg-Essen

BIL Workshop in Leuven, 15/07/14

Background

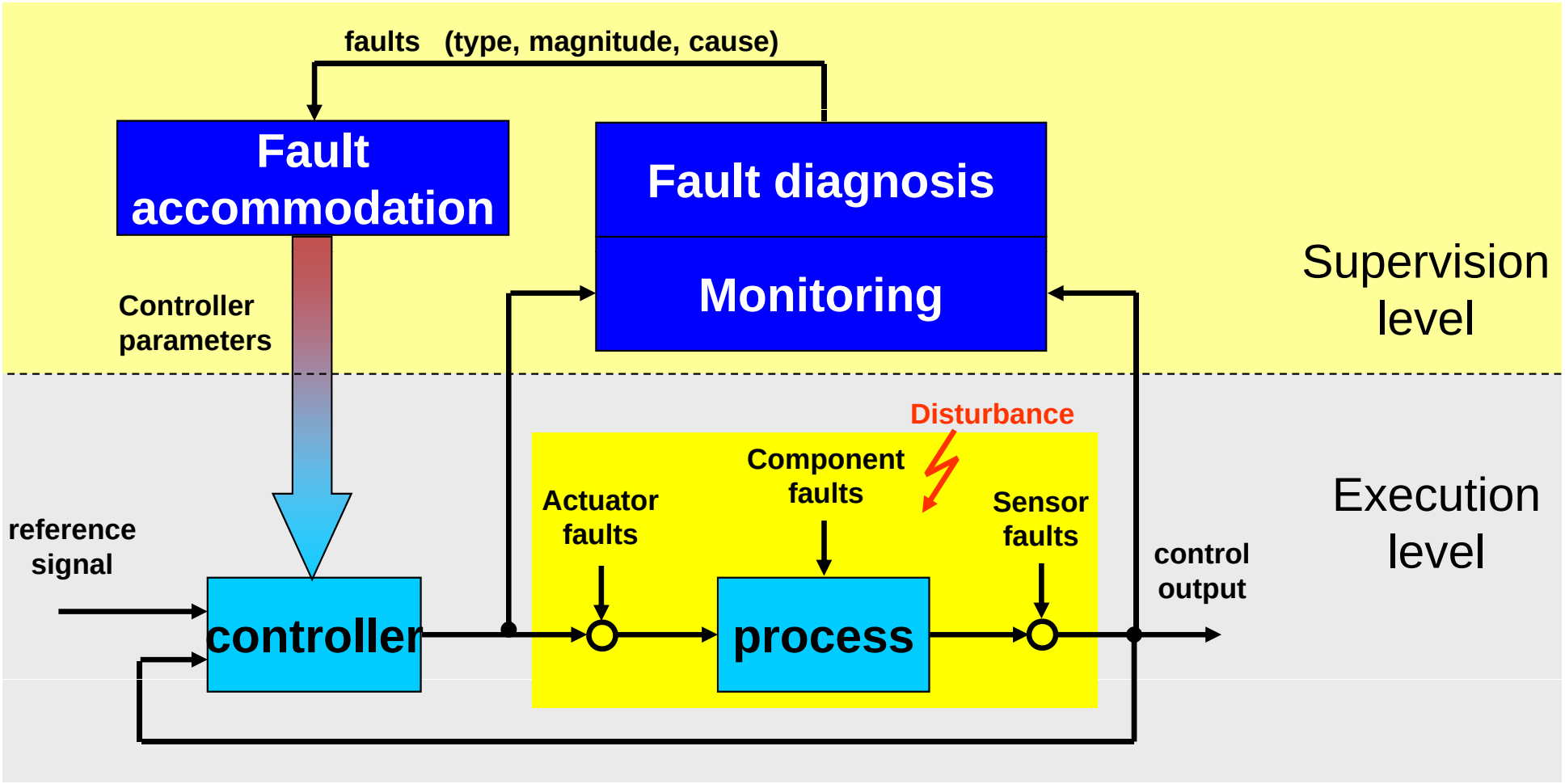

Identification
process data
identification approaches
process model
Model-based
system design
state of the art
Our work
Data-driven
process data
identification approaches
system construction
system optimization
online process data
directly identifying control systems
preliminary design
implementation/ optimization

Outline

- A model-based FD and FTC scheme
- Basic ideas and procedure
- Data-driven realization of kernel and image representations
- Data-driven construction and implementation of the control systems
- Concluding remarks

A model-based FD and FTC scheme

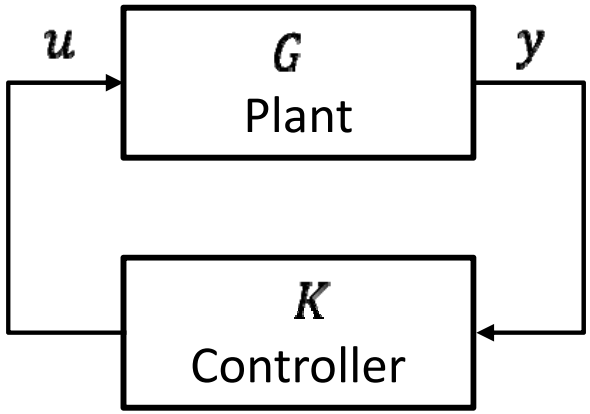
The state of the art of fault-tolerant control (FTC) systems with an embedded fault diagnosis (FD)



Controller is addressed as a system unit!

A model-based FD and FTC scheme

Stability: Fail-safe in the context of feedback control



Given plant model

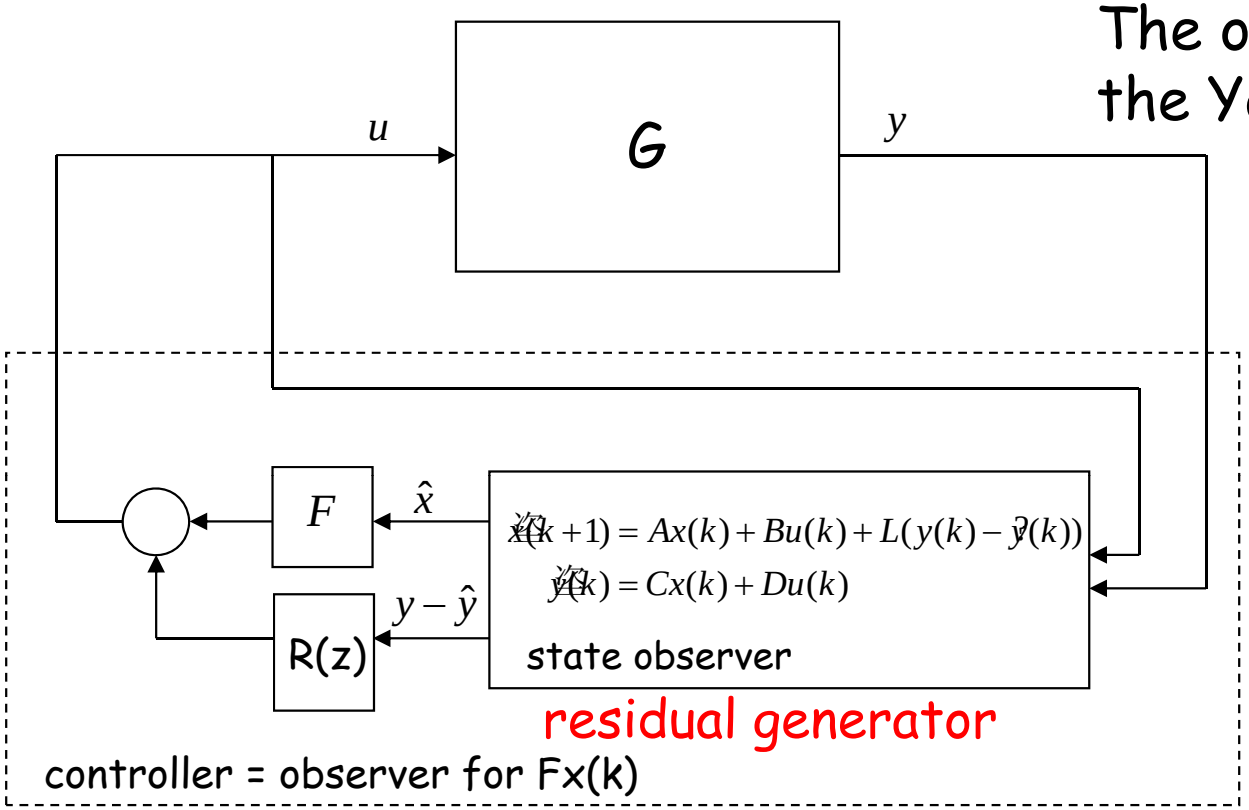
$$G = NM^{-1} = \tilde{M}^{-1}\tilde{N}$$

All stabilizing controllers are parameterized by

$$K = -(\tilde{Y} + MQ)(X - NQ)^{-1} = -(X - Q\tilde{N})^{-1}(Y + QM), \qquad Q \in RH_\infty$$

- The controller is realized in the "open-loop" form and can be unstable
- The controller is responsible for "everything" → trade-off is needed for the design →
- challenges with real-time issues (FTC, adaptive etc.)

A model-based FD and FTC scheme



The observer-based realization of the Youla parametrization:

$$u(z) = K(z)y(z)$$

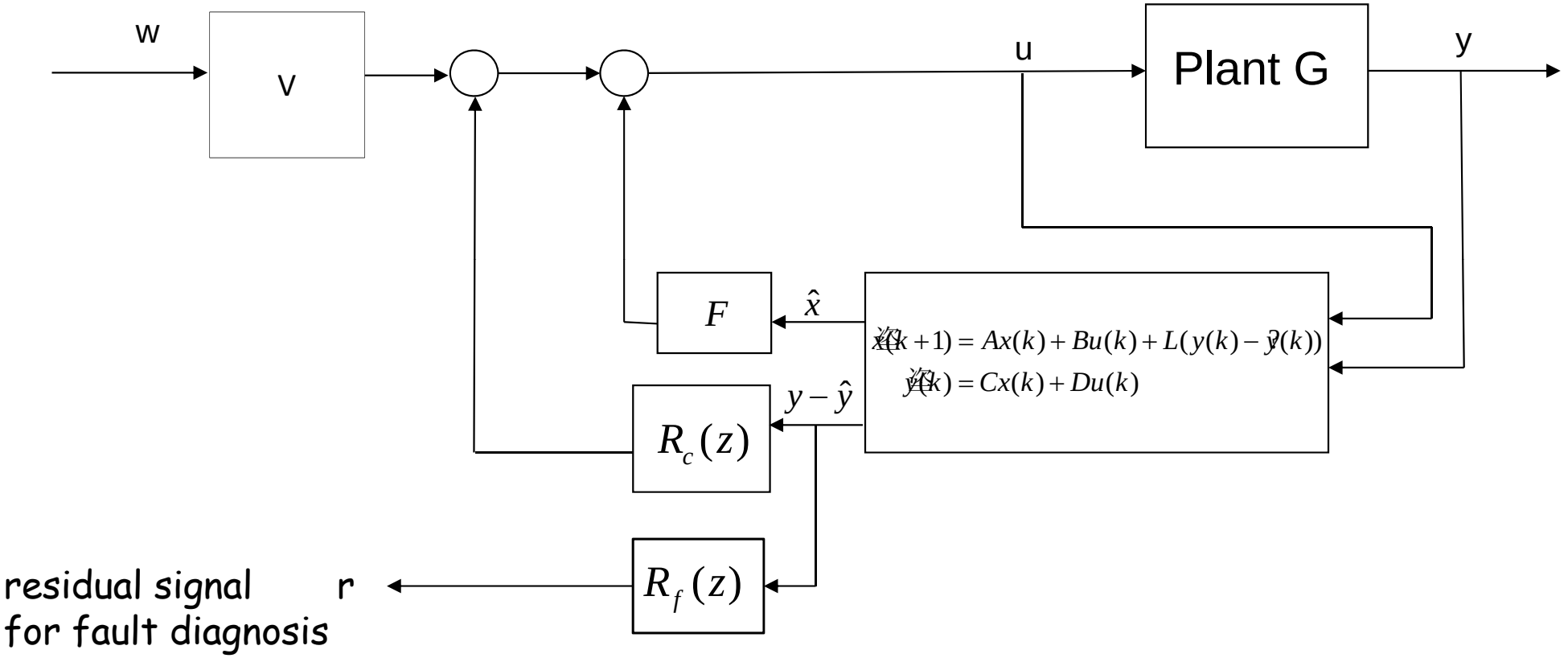
$$= \hat{Fx}(z) \text{ (an estimate for } Fx(k))$$

$$= \mathbf{F\hat{x}(z) + R(z)(y - y)}$$

- The controller in the closed-loop configuration → stable
- It enables a functionalization of the control parameters →
- It enables a prioritization of the controller parameters→ new schemes for FTC → LCM (life-cycle management)

A model-based FD and FTC scheme

A Youla parameterization based FTC configuration



$$u = F\dot{\hat{x}} + R_c(z)\left(y - \hat{y}\right) + V(z)w, r = R_f(z)\left(y - \hat{y}\right)$$

A model-based FD and FTC scheme

Functionalization of control parameters:

Robustness dist. rejection

disturbance estimator

$$u(z) = F\hat{x}(z) + R(z)r(z) + V(z)w_{ref}(z) + SK(z)r(z)$$

Stability eigen-dynanics

Feed-forward controller tracking performance

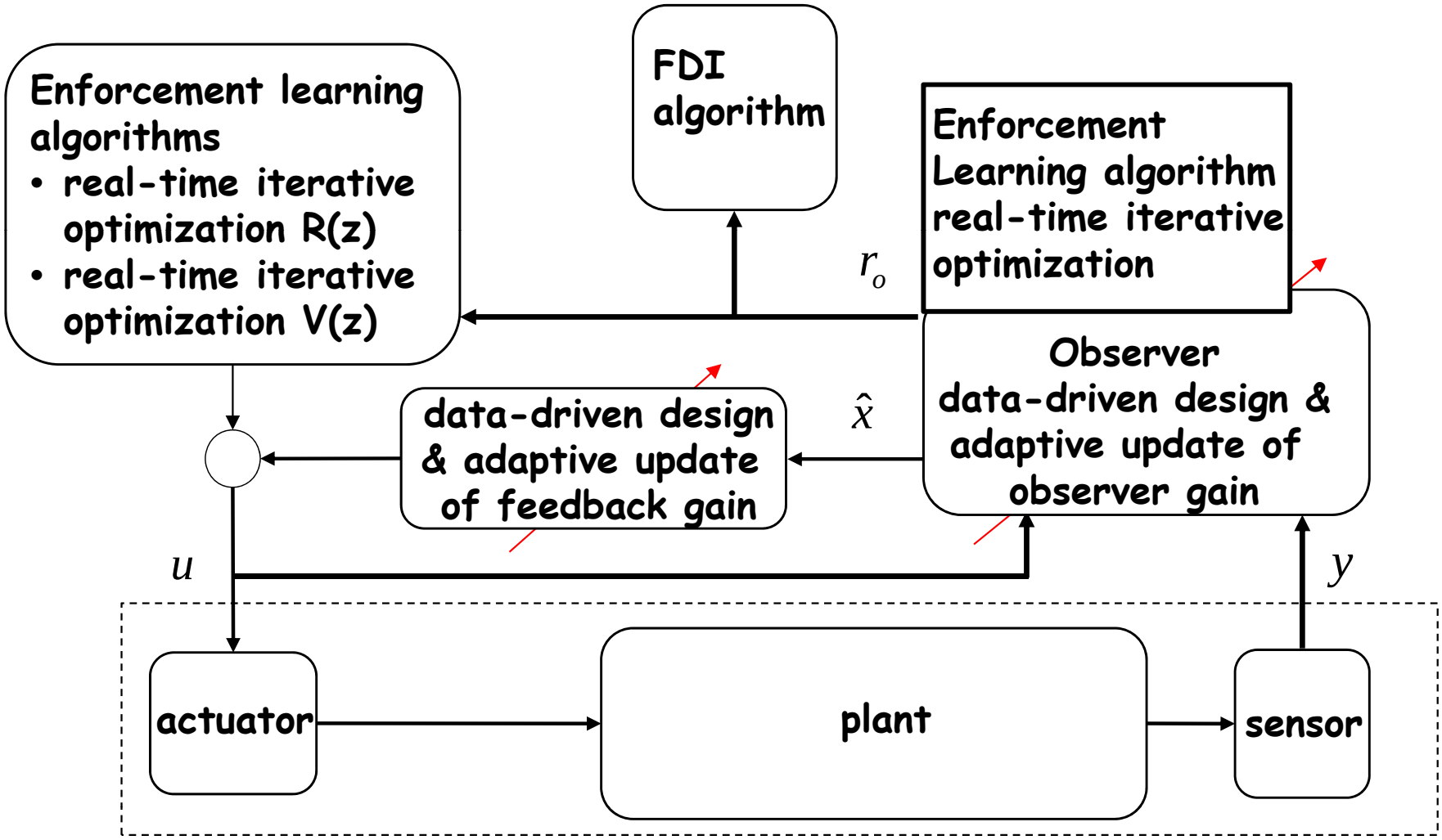
Prioritization of control parameters:

High-priori (H-PRIO) Parameters real-time update→Fail-safe operation

$$u(z) = F\hat{x}(z) + R(z)r(z) + V(z)w_{ref}(z) + SK(z)r(z)$$

Low-priori (L-PRIO) Parameters environment interactive update→ high performance operation

Life cycle management of automatic control systems



Stable kernel and image representations

$$G: x(k+1) = Ax(k) + Bu(k),\ y(k) = Cx(k) + Du(k) \Rightarrow$$

observer/residual generator
$$\begin{cases} \hat{x}(k+1) = Ax(k) + Bu(k) + L(y(k) - \hat{y}(k)) \\ r = y - \hat{y}, \quad \hat{y}(k) = C\hat{x}(k) + Du(k) \end{cases}$$

$$\Rightarrow \hat{M}(z)y(z) - N(z)u(z) = r(z) = \underbrace{\begin{bmatrix} -\hat{N}(z) & M(z) \end{bmatrix}}_{\text{SKR}} \begin{bmatrix} u(z) \\ y(z) \end{bmatrix} \equiv 0 \text{ for } \hat{x}(0) = x(0), \forall u$$

$$\hat{M} = \left(A - LC, L, -C, I\right), N = \left(A - LC, B - LD, C, D\right)$$

$$\xrightarrow{u:=Fx+v} \begin{cases} x(k+1) = \left(A + BF\right)x(k) + Bv(k) \\ u = Fx + v \\ y = Cx + Du = (C + DF)x + Dv \end{cases}$$

$$\Rightarrow \begin{cases} u = Mv, & M = (A + BF, B, F, I) \\ y = Nv, & N = (A + BF, B, C + DF, D) \end{cases} \Rightarrow \begin{bmatrix} y(z) \\ u(z) \end{bmatrix} = \underbrace{\begin{bmatrix} N(z) \\ M(z) \end{bmatrix}}_{\text{SIR}} v(z)$$

Basic ideas and procedure

I Construct Hankel data sets

II Perform QR

III Identify SKR, SIR

$$Z_f = \begin{bmatrix} Y_f \\ U_f \end{bmatrix}, Z_p = \begin{bmatrix} Y_p \\ U_p \end{bmatrix}$$

$$\begin{bmatrix} Z_p \\ U_f \\ Y_f \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 \\ R_{21} & R_{22} & 0 \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix}$$

$$K_{d,s} \begin{bmatrix} R_{21} & R_{22} \\ R_{31} & R_{32} \end{bmatrix} = 0$$

$$I_{d,s,s+n} : H_{u,s}, F$$

IV Construction of observer & control systems

V Adaptive H-priori observer & controller

VI iterative optimization of L-priori parametrization matrices

Definitions of data-driven realization of SKR and SIR

Given an (LTI) process. Matrix $K_{d,s}$ is called data-driven realization of SKR, if for some integer s, it holds

$$\forall x(0), u_s(k) = \begin{bmatrix} u(k-s) \\ \vdots \\ u(k) \end{bmatrix} \text{ and correspondingly } y_s(k) = \begin{bmatrix} y(k-s) \\ \vdots \\ y(k) \end{bmatrix},$$

$$K_{d,s} \begin{bmatrix} u_s(k) \\ y_s(k) \end{bmatrix} \equiv 0$$

And, matrix I_{d,s_1,s_2} is called data-driven realization of SIR, if for some integers s_1, s_2, it holds

$$\forall x(0), u_{s_1}(k) = \begin{bmatrix} u(k-s_1) \\ \vdots \\ u(k) \end{bmatrix} \text{ and correspondingly } y_{s_1}(k) = \begin{bmatrix} y(k-s_1) \\ \vdots \\ y(k) \end{bmatrix},$$

$$\exists v_{s_2}(k) = \begin{bmatrix} v(k-s_2) \\ \vdots \\ v(k) \end{bmatrix} \text{ s.t. } \begin{bmatrix} u_{s_1}(k) \\ y_{s_1}(k) \end{bmatrix} = I_{d,s_1,s_2} v_{s_2}(k)$$

Data-driven realization of SKR and SIR

Process (state space) model, unknown *a priori*

$$x(k+1) = Ax(k) + Bu(k) + w(k),\ y(k) = Cx(k) + Du(k) + v(k)$$

Process I/O model

$$y_s(k) = \Gamma_s x(k-s) + H_{u,s} u_s(k) + H_{w,s} w_s(k) + v_s(k)$$

$$y_s(k) = \begin{bmatrix} y(k-s) \\ \vdots \\ y(k) \end{bmatrix}, u_s(k) = \begin{bmatrix} u(k-s) \\ \vdots \\ u(k) \end{bmatrix}, w_s(k) = \begin{bmatrix} w(k-s) \\ \vdots \\ w(k) \end{bmatrix}, v_s(k) = \begin{bmatrix} v(k-s) \\ \vdots \\ v(k) \end{bmatrix}$$

$$\Gamma_s = \begin{bmatrix} C \\ \vdots \\ CA^s \end{bmatrix}, H_{u,s} = \begin{bmatrix} D & 0 & 0 \\ CB & D & \\ \vdots & & \ddots & 0 \\ CA^{s-1}B & \cdots & CB & D \end{bmatrix}, H_{w,s} = \begin{bmatrix} 0 & 0 & 0 \\ C & 0 & \\ \vdots & & \ddots & 0 \\ CA^{s-1} & \cdots & C & 0 \end{bmatrix}$$

Data-driven realizations of SKR and SIR

Data-driven realization form of the SKR:

$$\begin{bmatrix} u_s(k) \\ y_s(k) \end{bmatrix} = \begin{bmatrix} 0 & I \\ \Gamma_s & H_{u,s} \end{bmatrix} \begin{bmatrix} x(k-s) \\ u_s(k) \end{bmatrix} + \begin{bmatrix} 0 \\ H_{w,s} w_s(k) + v_s(k) \end{bmatrix} \Rightarrow$$

$$K_{d,s} \begin{bmatrix} u_s(k) \\ y_s(k) \end{bmatrix} = 0 \Leftrightarrow K_{d,s} \Psi_s = 0, \Psi_s = \begin{bmatrix} 0 & I \\ \Gamma_s & H_{u,s} \end{bmatrix} = 0$$

The data-driven realization of SKR is the null-matrix of Ψ_s.

Data-driven realizations of SKR and SIR

Data-driven realization form of the SIR:

$$u(k)=Fx(k)+v(k)\Rightarrow\begin{bmatrix}u_s(k)\\ y_s(k)\end{bmatrix}=\begin{bmatrix}\Gamma_{s,u} & H_{u,v,s}\\ \Gamma_{s,y} & H_{y,v,s}\end{bmatrix}\begin{bmatrix}x(k-s)\\ v_s(k)\end{bmatrix},A_F=A+BF,C_F=C+DF,$$

$$\Gamma_{s,u}=\begin{bmatrix}F\\ \vdots\\ FA_F^s\end{bmatrix},\Gamma_{s,y}=\begin{bmatrix}C_F\\ \vdots\\ C_FA_F^s\end{bmatrix},H_{u,v,s}=\begin{bmatrix}I & 0 & & 0\\ FB & I & & \\ \vdots & & \ddots & 0\\ FA_F^{s-1}B & \cdots & FB & I\end{bmatrix},H_{y,v,s}=\begin{bmatrix}D & 0 & & 0\\ C_FB & 0 & & \\ \vdots & & \ddots & 0\\ C_FA_F^{s-1}B & \cdots & C_FB & D\end{bmatrix}$$

By a deadbeat control gain:

$$x(k-s)=\underbrace{\begin{bmatrix}A_F^{n-1}B & \cdots & B\end{bmatrix}}_{\Sigma_{n\text{-}1,F}}\begin{bmatrix}v(k-s-n)\\ \vdots\\ v(k-s-1)\end{bmatrix}\Rightarrow\begin{bmatrix}u_s(k)\\ y_s(k)\end{bmatrix}=\begin{bmatrix}\Gamma_{s,u}\boldsymbol{\Sigma_{n\text{-}1,F}} & H_{u,v,s}\\ \Gamma_{s,y}\boldsymbol{\Sigma_{n\text{-}1,F}} & H_{y,v,s}\end{bmatrix}\begin{bmatrix}v_{n-1}(k-s-1)\\ v_s(k)\end{bmatrix}\Rightarrow$$

$$\begin{bmatrix}u_s(k)\\ y_s(k)\end{bmatrix}=\begin{bmatrix}\Gamma_{s,u}\Sigma_{n\text{-}1,F} & H_{u,v,s}\\ \Gamma_{s,y}\Sigma_{n\text{-}1,F} & H_{y,v,s}\end{bmatrix}v_{s+n}(k)\Rightarrow I_{d,s,s+n}=\begin{bmatrix}\Gamma_{s,u}\Sigma_{n\text{-}1,F} & H_{u,v,s}\\ \Gamma_{s,y}\Sigma_{n\text{-}1,F} & H_{y,v,s}\end{bmatrix}$$

The data-driven realization of SIR

Identification of the data-driven realization of SKR and SIR

I/O data model

$$y_s(k) = \Gamma_s x(k-s) + H_{u,s} u_s(k) + H_{w,s} w_s(k) + v_s(k)$$

$$\Rightarrow Y_{s,N} = \Gamma_s X_{s,N} + H_{s,u} U_{s,N} + \Theta_{s,N}$$

Divide the data sets into

$$Y_{s,N} = \begin{bmatrix} y(k-s) & \cdots & y(k-s+N) \\ \vdots & \vdots & \vdots \\ y(k-s_p) & \cdots & y(k-s_p+N) \end{bmatrix}, U_{s,N} = \begin{bmatrix} u(k-s) & \cdots & u(k-s+N) \\ \vdots & \vdots & \vdots \\ u(k-s_p) & \cdots & u(k-s_p+N) \end{bmatrix}$$

$$Y_p = \begin{bmatrix} y(k) & \cdots & y(k+N) \\ \vdots & \vdots & \vdots \\ y(k+s_p) & \cdots & y(k+s_p+N) \end{bmatrix}, U_p = \begin{bmatrix} u(k) & \cdots & u(k+N) \\ \vdots & \vdots & \vdots \\ u(k+s_p) & \cdots & u(k+s_p+N) \end{bmatrix}$$

$$X_{s,N} = \begin{bmatrix} x(k-s) & \cdots & x(k-s+N) \end{bmatrix}, \Theta_{s,N} = \begin{bmatrix} 0 & 0 & 0 \\ C & 0 & 0 \\ CA^{s-1} & \cdots & C & 0 \end{bmatrix} W_{s,N} + V_{s,N}$$

$$Y_f = \begin{bmatrix} y(k+1) & \cdots & y(k+1+N) \\ \vdots & \vdots & \vdots \\ y(k+s_f) & \cdots & y(k+s_f+N) \end{bmatrix}, U_f = \begin{bmatrix} u(k+1) & \cdots & u(k+1+N) \\ \vdots & \vdots & \vdots \\ u(k+s_f) & \cdots & u(k+s_f+N) \end{bmatrix}$$

$Y_f = \Gamma_s X_f + H_{s,u} U_f + \Theta_f, s_f = s,$ s is some integer (much) larger than n

QR decomposition of the data sets: the basic algorithm

$$\begin{bmatrix} Z_p \\ U_f \\ Y_f \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 \\ R_{21} & R_{22} & 0 \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix}$$

QR decomposition
(reliable numerical computation)

$$\begin{bmatrix} U_f \\ Y_f \end{bmatrix} = \begin{bmatrix} 0 & I \\ \Gamma_s & H_{u,s} \end{bmatrix} \begin{bmatrix} X_f \\ U_f \end{bmatrix} + \begin{bmatrix} 0 \\ \Theta_f \end{bmatrix} = \begin{bmatrix} R_{21} & R_{22} \\ R_{31} & R_{32} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} + \begin{bmatrix} 0 \\ R_{33}Q_3 \end{bmatrix}, R_{33}Q_3 = \Theta_f$$

$$Y_f = \Gamma_s L_p Z_p + H_{u,s} U_f + \Theta_f = \begin{bmatrix} \Gamma_s L_p & H_{u,s} & I \end{bmatrix} \begin{bmatrix} R_{11} & 0 & 0 \\ R_{21} & R_{22} & 0 \\ 0 & 0 & R_{33} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix}$$

Data-driven realizations of SKR and SIR

The algorithm for the SKR identification:

$$K_{d,s}: K_{d,s}\begin{bmatrix} R_{21} & R_{22} \\ R_{31} & R_{32} \end{bmatrix} = 0 \Leftrightarrow K_{d,s}\begin{bmatrix} R_{21} & R_{22} \\ R_{31} & R_{32} \end{bmatrix}\begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} = 0$$

$$\Leftrightarrow K_{d,s}\Psi_s = 0 \Leftrightarrow \Psi_s\begin{bmatrix} U_f \\ Y_f \end{bmatrix} = 0$$

The algorithm for the SIR identification consists of three steps:

- Identification of $H_{u,s}$
- Identification of F (deadbeat gain matrix)
- Construction of

$$I_{d,s,s+n} = \begin{bmatrix} \Gamma_{s,u}\Sigma_{n\text{-}1,F} & H_{u,v,s} \\ \Gamma_{s,y}\Sigma_{n\text{-}1,F} & H_{y,v,s} \end{bmatrix}$$

Data-driven realizations of SKR and SIR

Identification of $H_{u,s}$

$$Y_f Q_2^T = \begin{bmatrix} \Gamma_s L_p & H_{u,s} & I \end{bmatrix} \begin{bmatrix} R_{11} & 0 & 0 \\ R_{21} & R_{22} & 0 \\ 0 & 0 & R_{33} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix} Q_2^T \Leftrightarrow Y_f Q_2^T = H_{u,s} R_{22}$$

$$\Rightarrow H_{u,s} = \begin{bmatrix} D & 0 & & 0 \\ CB & D & & \\ \vdots & & \ddots & 0 \\ CA^{s-1}B & \cdots & CB & D \end{bmatrix} = Y_f Q_2^T R_{22}^?$$

$$\begin{bmatrix} CB & CAB & \cdots & CA^nB \\ CAB & CA^2B & \cdots & CA^{n+1}B \\ \vdots & & \ddots & 0 \\ CA^{n-1}B & \cdots & CA^{2n-2}B & CA^{2n-1}B \end{bmatrix} = \Gamma_{s-1}\Pi_n, \Pi_n = \begin{bmatrix} B & \cdots & A^nB \end{bmatrix}$$

Data-driven realizations of SKR and SIR

Identification of F

$$\Gamma_{s-1}\Pi_n w_n = 0 \Rightarrow \Pi_n w_n = 0, w_n = \begin{bmatrix} w_{n,0} \\ \vdots \\ w_{n,n} \end{bmatrix}$$

$$A_z = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \\ 0 & \cdots & 0 & 0 \end{bmatrix}, b_z = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, F_z = \begin{bmatrix} w_{n,0} & \cdots & w_{n,n-1} \end{bmatrix}, w_{n,i} \in \boldsymbol{R}^{k_u \times 1}$$

$$T = \begin{bmatrix} B & AB & \cdots & A^n B \end{bmatrix} \begin{bmatrix} w_{n,1} & w_{n,2} & \cdots & w_{n,n-1} & w_{n,n} \\ w_{n,2} & \cdots & w_{n,n-1} & w_{n,n} & 0 \\ \vdots & \iddots & \iddots & \iddots & \\ w_{n,n-1} & w_{n,n} & 0 & & \\ w_{n,n} & 0 & & & \end{bmatrix} \Rightarrow$$

$$AT + BF_z = TA_z, Tb_z = Bw_{n,n}, eig(A_z) = 0$$

Construction of SIR

$$I_{d,s,s+n}=\begin{bmatrix}\Gamma_{s,u}\Sigma_{n\text{-}1,F} & H_{u,v,s}\\ \Gamma_{s,y}\Sigma_{n\text{-}1,F} & H_{y,v,s}\end{bmatrix},H_{u,v,s}=\begin{bmatrix}I & 0 & & 0\\ FB & I & & \\ \vdots & & \ddots & 0\\ FA_F^{s-1}B & \cdots & FB & I\end{bmatrix}$$

$$H_{y,v,s}=H_{u,s}H_{u,v,s},\Gamma_{s,u}\Sigma_{n\text{-}1,F}=\left\{H_{u,v,s}\right\},\Gamma_{s,y}\Sigma_{n\text{-}1,F}=\left\{H_{u,v,s}\right\}$$

$$FA_F^iB=\omega_{n,n-1-i}/\omega_{n,n},i=1,\cdots,n-1$$

Data-driven construction and implementation

Realization of observer and residual generator

Luenberger observer scheme

$$z(k+1)=Gz(k)+Hu(k)+Ly(k), r(k)=Wz(k)+Vy(k)+Qu(k), z\in R^{l}, l\le n$$

$r(k)$: Residual (signal) vector, design parameters: G,H,L,W,V,Q and T

The key step: one-to-one mapping

$$\left[-v_s H_{u,s} \quad v_s\right]\in K_{d,s}, v_s\in\Gamma_s^{\perp}$$

$$v_s=\left[v_{s,0} \quad v_{s,1} \quad \cdots \quad v_{s,s-1}\right] \qquad v_s H_{u,s}=\left[v_s H_{s,1} \quad \cdots \quad v_s H_{s,s}\right]$$

Bridge

$$G=\begin{bmatrix}0 & 0 & \cdots & 0\\ 1 & 0 & \cdots & 0\\ \vdots & \ddots & \ddots & \vdots\\ 0 & \cdots & 1 & 0\end{bmatrix},\quad w=\begin{bmatrix}0 & \cdots & 0 & 1\end{bmatrix}$$

$$L=-\begin{bmatrix}v_{s,0}\\ \vdots\\ v_{s,s-1}\end{bmatrix},\quad V=-v_{s,s}$$

$$H=\begin{bmatrix}\alpha_s H_{s,1}\\ \vdots\\ \alpha_s H_{s,s-1}\end{bmatrix},\quad Q=v_s H_{s,s}$$

Data-driven construction and implementation

Realization of observer-based state feedback controller

$$\text{Given } \Pi_n = \begin{bmatrix} B & AB & \cdots & A^n B \end{bmatrix}, \text{ then } w_n = \begin{bmatrix} w_{n,0} \\ \vdots \\ w_{n,n} \end{bmatrix} \text{ satisfying } \Pi_n w_n = 0 \text{ results in}$$

$$F_z = \begin{bmatrix} w_{n,0} & \cdots & w_{n,n-1} \end{bmatrix}$$

$$u(k) = F_z z(k)$$

$$z(k+1) = Gz(k) + Hu(k) + Ly(k), r(k) = Wz(k) + Vy(k) + Qu(k), z \in R^l, l \le n$$

Data-driven construction and implementation

Design of feed forward controller

$$\begin{bmatrix} u_s(k) \\ y_s(k) \end{bmatrix} = I_{d,s,s+n} v_{s+n}(k), I_{d,s,s+n} = \begin{bmatrix} \Gamma_{s,u}\Sigma_{n\text{-}1,F} & H_{u,v,s} \\ \Gamma_{s,y}\Sigma_{n\text{-}1,F} & H_{y,v,s} \end{bmatrix} = \begin{bmatrix} M_{d,s,s+n} \\ N_{d,s,s+n} \end{bmatrix}$$

For instance, for good tracking behavior

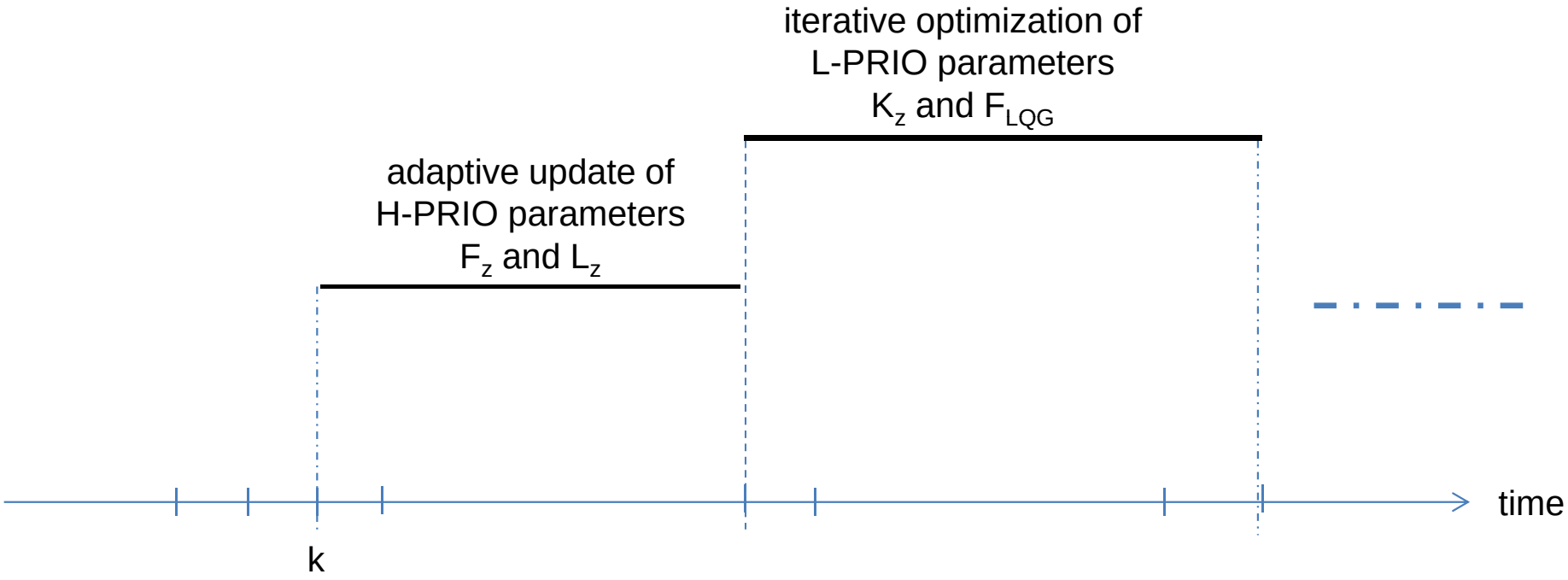
$$v_{s+n}(k) = N^{-}_{d,s,s+n} y_{ref,s+n}(k), N_{d,s,s+n} N^{-}_{d,s,s+n} = I$$

satisfying

$$\left\| u_s(k) \right\| \le \left\| M_{d,s,s+n} v_{s+n}(k) \right\|$$

Life cycle management of automatic control systems

Event-driven real-time adaptive and iterative optimization



Events: (i) Faults (ii) maintenance actions (iii) changes of operation conditions

Data-driven construction and implementation

Adaptive observer/residual generator, the basic idea:

- The possible changes in the process → changes in $K_{d,s}$
- based on the observer implementation form, application of an adaptive scheme becomes possible

A re-formulation of the observer implementation form

$$z(k+1) = \bar{A}_z z(k) + Q(u(k), y(k))\theta, \bar{A}_z = \begin{bmatrix} 0 & 0 & & -l_0 \\ 1 & \ddots & & -l_1 \\ & \ddots & 0 & \\ 0 & & 1 & -l_{s-1} \end{bmatrix}, \theta = \begin{bmatrix} col(B_z) \\ col(L) \\ g^T \end{bmatrix}$$

Constructed by input/output data

Independent of Process change

Data-driven realization of SKR

$$gy(k) = c_z z(k), c_z = [0 \quad \cdots \quad 0 \quad 1]$$

Data-driven construction and implementation

Application of the standard adaptive scheme to designing the adaptive observer/residual generator

Residual generator

$$\hat{z}(k+1) = \overline{A}_z \hat{z}(k) + Q\big(u(k), y(k)\big)\hat{\theta}(k) + V(k+1)\Big(\hat{\theta}(k+1) - \hat{\theta}(k)\Big)$$

Auxiliary filter

$$V(k+1) = \overline{A}_z V(k) + Q\big(u(k), y(k)\big), \varphi(k) = c_z V(k) - \begin{bmatrix} 0 & \cdots & 0 & y^T(k) \end{bmatrix}$$

Parameter estimator

$$\hat{\theta}(k+1) = \gamma(k)\varphi^T(k)r(r) + \hat{\theta}(k), \gamma(k) = \frac{\mu}{\delta + \varphi(k)\varphi^T(k)}, \delta > 0, 0 < \mu < 2$$

The stability has been proven.

Concluding remarks

A data-driven FTC (LCM) scheme is presented.

- Developed based on control theoretical study
  - functionalization and
  - prioritization

  of (feedback) fault-tolerant control systems
  - kernel and image representations

- Data-driven realization
  - by means of subspace technique
  - data-driven realization of kernel and image representations

- Adaptive and iterative optimization techniques are applied for fault-tolerant control, or more general life cycle management.

Thank you for your attention!