Proceeding of the 2004 American Control Conference
Boston, Massachusetts June 30 - July 2, 2004

Fault Tolerant Controllers for Sampled-data Systems

Henrik Niemann and Jakob Stoustrup

***Abstract*— A general compensator architecture for fault tolerant control (FTC) for sampled-data systems is proposed. The architecture is based on the YJBK parameterization of all stabilizing controllers, and uses the dual YJBK parameterization to quantify the performance of the fault tolerant system. The FTC architecture is based on a discrete-time nominal feedback controller and with the FTC part also in discrete-time.**

Further, a number of problems for the design of the controller reconfiguration part in the FTC architecture is considered. It is shown how these design problems can be transformed into standard design problems for feedback controllers.

I. Introduction

In the past years, the area of fault tolerant control has received an increasing interest. The reason is the increasing use of more and more complex systems and control systems. The research in this area has been derived both for general dynamical systems, [2], [3], [12], [13], [14], [15], [18], [20], [24].

The focus in this paper is on an architecture for fault tolerant controllers for sampled-data systems. Fault tolerant controllers for sampled-data systems is important, because fault tolerant controllers can be quite complicated and therefore in general a digital implementation will be required. Further, digital control is standard today in many complicated control systems. It is therefore also natural to use digital implementation of the FTC part of feedback controller.

The proposed FTC architecture is based on the Youla-Jabr-Bongiorno-Kucera (YJBK) parameterization of all stabilizing controllers. This architecture was first introduced in connection with continuous-time system, see e.g. [12], [13], [15] for a description of the continuous-time architecture. The architecture has a number of advantages that makes it very useful in connection with fault tolerant control. First of all, both the fault diagnosis part as well as the controller reconfiguration part are derived based on the same setup. Further, it is possible to change/modify the nominal controller without breaking the feedback loop. The modification is derived by using the parameterization in the architecture to add a modification signal into the nominal controller. This will make controller changes due to faults much more smooth.

II. System Setup

Consider the following generalized nominal 2×2 system,

$$\Sigma : \begin{cases} e(t) = G_{ed}d(t) + G_{eu}u(t) \\ y(t) = G_{yd}d(t) + G_{yu}u(t) \end{cases} \qquad (1)$$

where $d \in \mathcal{R}^r$ is a disturbance input vector, $u \in \mathcal{R}^m$ the control input signal vector, $e \in \mathcal{R}^q$ is the external output signal vector to be controlled, and $y \in \mathcal{R}^p$ is the measurement vector.

Further, let the dynamical system in (1) be controlled by the following stabilizing sampled-data feedback controller

$$u_k = K(z)y_k \qquad (2)$$

where the connections between the continuous-time signals $u(t), y(t)$ and the discrete-time signals u_k, y_k are given by

$$\begin{aligned} y_k &= \mathcal{S}y(t) \\ u(t) &= \mathcal{H}u_k \end{aligned}$$

where $\mathcal{S}$ is a sampler and $\mathcal{H}$ is a zero order hold. A block diagram of the system is shown in Figure 1.

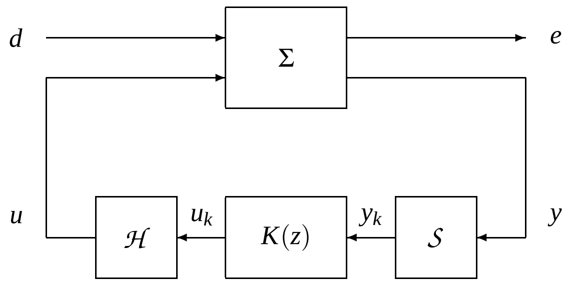

Fig. 1. Standard sampled-data system setup.

Let the system given in (1) include a number of parametric/multiplicative faults, described by the vector θ, $\theta^T = [\theta_1, \cdots, \theta_l]^T$. All four transfer functions in (1) will be a function of the fault vector, i.e.

$$\Sigma_\theta : \begin{cases} e(t) = G_{ed}(\theta)d(t) + G_{eu}(\theta)u(t) \\ y(t) = G_{yd}(\theta)d(t) + G_{yu}(\theta)u(t) \end{cases}$$

It is in general possible to give a more explicit description of the system setup for systems with parameter faults by including an extra input and output vector. The above system is then given by

$$\Sigma_\theta : \begin{cases} z = G_{zw}w + G_{zd}d + G_{zu}u \\ e = G_{ew}w + G_{ed}d + G_{eu}u \\ y = G_{yw}w + G_{yd}d + G_{yu}u \end{cases} \qquad (3)$$

H. Niemann is with Ørsted•DTU, Automation, Technical University of Denmark, Building 326, DK-2800 Lyngby, Denmark. Email: hhn@oersted.dtu.dk

J. Stoustrup is with the Dept. of Control Engineering, Institute of Electronic Systems, Aalborg University, Fr. Bajers Vej 7C, DK-9220 Aalborg, Denmark. Email: jakob@control.aau.dk

0-7803-8335-4/04/$17.00 ©2004 AACC

where the connection between the two external vectors w and z is given by

$$w = \theta z \tag{4}$$

This description is equivalent with the general description of system with model uncertainties, see e.g. [23].

For simplifying the following analysis, we have the following assumption.

Assumption II.1 *It is assumed that each parametric fault* θ_i *occurs by itself, i.e. two parameter faults never occur simultaneously. Moreover, faults occur sufficiently apart in time so that at any given time at most one fault affects the measurement signal.*

The general result without this assumption can be obtained quite easily from the results stated in this paper. The above assumption is often satisfied and makes the presentation of our results more transparent without the need to cloud the presentation with more technically complex methods.

III. The YJBK Parameterization

The YJBK and the dual YJBK parameterization are shortly introduced in this section. The YJBK parameterization was first derived by Youla et al. [21], [22] and independently by Kucera [11]. It has later been applied in numerous cases in connection with feedback control, see e.g. [1], [4], [5], [8], [19], [23].

A. The YJBK Parameterization

Consider a generalized nominal 2×2 system given by (1) controlled by a sampled-data controller $K(z)$ given by (2). Further, let the discrete-time transfer function from u_k to y_k be defined by

$$G_{yu}(z) = \mathcal{S} G_{yu}(s) \mathcal{H}$$

i.e. the transfer function that the controller look into. A coprime factorization of the system $G_{yu}(z)$ and the controller $K(z)$ is given by:

$$\begin{aligned} G_{yu}(z) &= NM^{-1} = \tilde{M}^{-1}\tilde{N}, \quad N, M, \tilde{N}, \tilde{M} \in \mathcal{RH}_\infty \\ K(z) &= UV^{-1} = \tilde{V}^{-1}\tilde{U}, \quad U, V, \tilde{U}, \tilde{V} \in \mathcal{RH}_\infty \end{aligned} \tag{5}$$

where the eight matrices in (5) must satisfy the double Bezout equation given by, see [23]:

$$\begin{aligned} \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix} &= \begin{pmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{pmatrix} \begin{pmatrix} M & U \\ N & V \end{pmatrix} \\ &= \begin{pmatrix} M & U \\ N & V \end{pmatrix} \begin{pmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{pmatrix} \end{aligned} \tag{6}$$

Based on the above coprime factorization of the system $G_{yu}(z)$ and the controller $K(z)$, we can give a parameterization of all controllers that stabilize the system in terms of a stable parameter $Q(s)$, i.e. all stabilizing controllers are given by [19]:

$$K(Q) = U(Q)V(Q)^{-1} \tag{7}$$

where

$$U(Q) = U + MQ, \; V(Q) = V + NQ, \; Q \in \mathcal{RH}_\infty$$

or by using a left factored form:

$$K(Q) = \tilde{V}(Q)^{-1}\tilde{U}(Q) \tag{8}$$

where

$$\tilde{U}(Q) = \tilde{U} + Q\tilde{M}, \; \tilde{V}(Q) = \tilde{V} + Q\tilde{N}, \; Q \in \mathcal{RH}_\infty$$

Using the Bezout equation, the controller given either by (7) or by (8) can be realized as an LFT in the parameter Q,

$$K(Q) = \mathcal{F}_l(J_K, Q) \tag{9}$$

where J_K is given by

$$J_K = \begin{pmatrix} UV^{-1} & \tilde{V}^{-1} \\ V^{-1} & -V^{-1}N \end{pmatrix} \tag{10}$$

The controller $K(Q)$ with the structure given by (9) can be implemented with a resulting the closed loop system as depicted in Figure 2, [19].

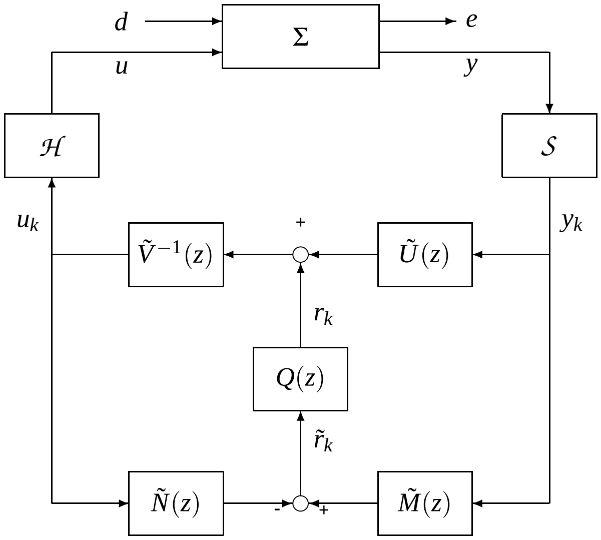

Fig. 2. Controller structure with parameterization for a sampled data system with a discrete time controller

The main observation which shall be exploited in the solution to the fault tolerant control problem, is the following simple expression for the transfer function from faults to the external output e terms of the parameter Q:

$$\begin{aligned} e &= G_{ed}(s)d + G_{eu}(s)\mathcal{H}K(Q)(z) \\ &\quad \times (I - G_{yu}(z)K(Q)(z))^{-1}\mathcal{S}G_{yd}(s)d \\ &= G_{ed}(s)d + G_{eu}(s)\mathcal{H} \\ &\quad \times (U(z)\tilde{M}(z) + M(z)Q(z)\tilde{M}(z))\mathcal{S}G_{yd}(s)d \end{aligned} \tag{11}$$

where (6) has been exploited. Note that it is not possible to describe the connection between external input d and external output e directly by a transfer function.

From (11), it is clear that the closed loop operator between the external input d and the external output e is an affine operator in Q. Therefore, the closed loop stability will not be affected as long as Q is selected as a stable transfer function. Further, as a direct consequence of the closed loop operator being an affine function of Q, Q cannot be applied for stabilizing the nominal closed loop system.

B. The Dual YJBK Parameterization

In the same way, it is possible to derive a parameterization in terms of a stable parameter S of all systems that are stabilized by one controller, i.e. the dual YJBK parameterization. The parameterization is given by [19]:

$$G_{yu}(S) = N(S)M(S)^{-1} \tag{12}$$

where

$$N(S) = N + VS,\ M(S) = M + US,\ S \in \mathcal{RH}_\infty$$

or by using a left factored form:

$$G_{yu}(S) = \tilde{M}(S)^{-1}\tilde{N}(S) \tag{13}$$

where

$$\tilde{N}(S) = \tilde{N} + S\tilde{V},\ \tilde{M}(S) = \tilde{M} + S\tilde{U},\ S \in \mathcal{RH}_\infty$$

An LFT representation of (12) or (13) is given by:

$$G_{yu}(S) = \mathcal{F}_l(J_G, S) \tag{14}$$

where J_G is given by

$$J_G = \begin{pmatrix} NM^{-1} & \tilde{M}^{-1} \\ M^{-1} & -M^{-1}U \end{pmatrix} \tag{15}$$

It has been shown in e.g. [19], that the dual YJBK parameter S transfer function can be calculated by using the primary YJBK parameterization. It turns out that S is given by:

$$S = \mathcal{F}_u(J_K, G_{yu}(S)) \tag{16}$$

This simple relationship between the YJBK parameterization and the dual YJBK parameter S can be used in connection with calculation/estimation of S. A general description of estimation of S will not be given in this paper, see instead [6], [19].

IV. Fault Tolerant Controller Architecture for Sampled-data Systems

In the sequel, an architecture for fault tolerant controllers for sampled-data systems will be proposed, based on the YJBK parameterization shown in the block diagram in Figure 2. There is a number of reasons for using the architecture from the YJBK parameterization in connection with FTC. First of all, a fault tolerant controller consists of two parts, a fault diagnosis part (the FDI part) and a controller reconfiguration part (the CR part). Both parts can be based on the YJBK parameterization. From Section III, we have that the Q parameter will be the CR part of the FTC controller. This means that the CR part of the feedback controller is a modification of the existing controller. Thus, the required change of controller when a fault appears in the system does not imply a complete shift to another controller, but only a modification of the existing controller by adding a correction signal in the nominal controller, the r signal in Figure 2. However, it should be pointed out that it is possible to modify the controller arbitrarily by designing the YJBK parameter Q, see e.g. [17], [19].

The other part in the general FTC architecture is the fault diagnosis part. This part can also be derived by using the same signals as used in connection with the YJBK parameterization. Equivalent with the YJBK parameterization of all stabilizing controllers, a parameterization of all residual generators is given by, [9], [10]

$$r = Q_{\mathrm{FDI}}\tilde{r}_k = Q_{\mathrm{FDI}}(\tilde{M}y - \tilde{N}u) \tag{17}$$

This is just the YJBK parameterization of all filters, i.e. there is no feedback. This means that it is possible to combine both fault diagnosis and controller reconfiguration in the same architecture without any problems. A block diagram for this combined FDI and CR architecture based on the YJBK parameterization for sampled-data systems is shown in Figure 3 for three potential multiplicative faults - the generalization to any number of faults should be obvious.

The above controller architecture applied for FTC shown in Figures 2 and 3 has a fixed structure with respect to the number of measurement signals and control signals. This will not in general be the case in real applications. Here, faults in e.g. sensors can be handled by applying other sensors in the system, i.e. the measurement output from the system is changed. An equivalent approach applies to faults in connection with the actuators in the system. This type of system change has not directly been included in the system description given by (3). However, it is possible to include change of sensors and/or actuators in the FTC architecture given above. This aspect will not be considered here, see instead [13] for a discussion of how to change the number of sensors/actuators in connection with the FTC architecture shown in Figure 3.

V. Design of the Controller Reconfiguration Part

In the remaining part of this paper, design of the controller reconfiguration part, Q_{CR} in the FTC architecture will be considered. The design of the reconfiguration part can be derived with respect to closed loop stability only or with respect to both closed loop stability and performance of the faulty sampled-data system. Both these two design problems will be considered in the following.

First, let us consider the closed-loop stability problem when the system is affected by multiplicative faults, i.e. if G_{yu} depends on some multiplicative faults. Let a YJBK parameterized controller $K(Q_{\mathrm{CR}})$ be applied, where the nominal controller $K(0) = K_0$ is designed for the nominal

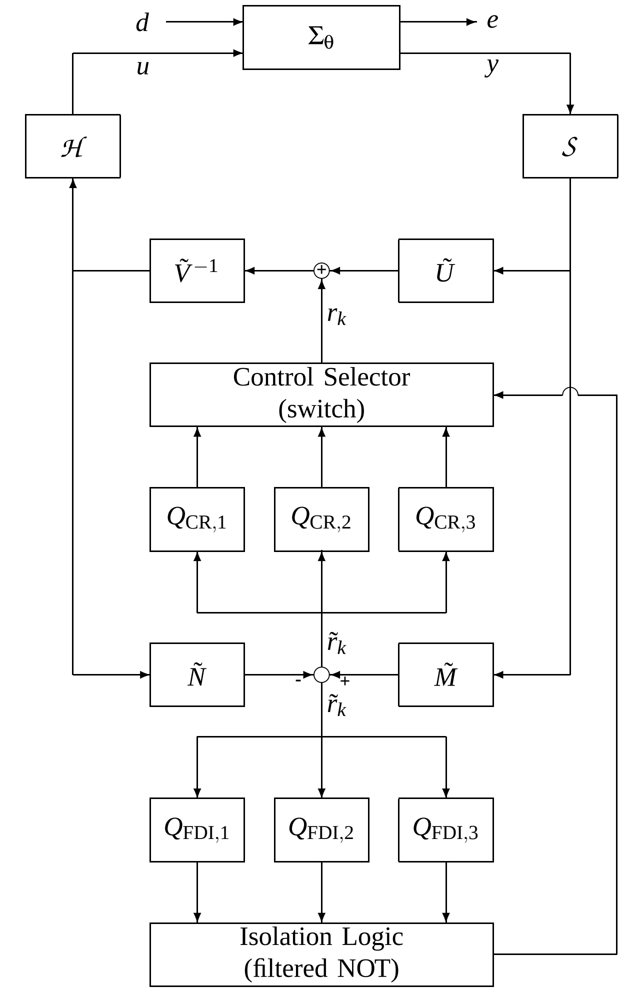

Fig. 3. Fault tolerant scheme for sampled-data systems with three potential parameter faults. The residual signal is used both for isolation and for feedforward in the fault handling.

system. The YJBK parameter is then applied for the controller reconfiguration in the faulty case, i.e. $Q_{\mathrm{CR}}(z)$ needs to stabilize the sampled-data closed-loop system when a fault has appeared in the system. The stability of the sampled-data closed loop system requires stability of the nominal sampled-data closed-loop system and closed-loop stability of a certain loop where both $Q_{\mathrm{CR}}(z)$ and the multiplicative faults θ are included, [19]. The stability of the nominal sampled-data closed-loop system is satisfied by the design of the nominal feedback controller $K(0)$. Using (16), we have that the two YJBK parameters $(Q_{\mathrm{CR}}, S(\theta))$ will give a closed loop system that is decoupled from the nominal closed-loop system. It is therefore quite easy to show that the closed loop system is stable if and only if the nominal system is closed-loop stable and the closed-loop performed by the YJBK parameters is stable, [19], given by

$$\tilde{S}(Q_{\mathrm{CR}})(z) = (I - Q_{\mathrm{CR}}S(\theta))^{-1}(z) \tag{18}$$

where $S(\theta)(z)$ is the dual YJBK parameter, depending on the multiplicative faults θ.

To guarantee closed-loop stability, it is required that S is stable. Combining the YJBK parameterization with the dual YJBK parameterization, it is not a condition that Q_{CR} and S themselves need to be stable. Q_{CR} and S just need to satisfy that the closed-loop system given by (18) is stable, [19]. From the general explicit description of S given by (16), we have that S can be derived as the open loop transfer function between the input signal r_k and the output signal $\tilde{r}_k$ in Figure 3. Based on this, $S(\theta)(z)$ then has the following form:

$$\begin{aligned} S(\theta)(z) \quad = \quad & \tilde{M}(z)\mathcal{S}G_{yw}(s)\theta(I - G_{zw}(s)\theta)^{-1}G_{zu}(s)\mathcal{H} \\ & \times M(z)(I - \tilde{U}\mathcal{S}G_{yw}(s)\theta(I - G_{zw}(s)\theta)^{-1} \\ & G_{zu}(s)\mathcal{H}M(z))^{-1} \end{aligned} \tag{19}$$

In connection with (19), it is important to note that the stability condition of S and/or of $\tilde{S}(Q_{\mathrm{CR}})$ in (18) for satisfying that the faulty closed loop system is stable, is valid only if the faulty system is still detectable and stabilizable from the specified input signals u and output signals y. This is a standard condition in connection with FTC systems. If the faulty system is not detectable and/or stabilizable, additional actuators and/or sensors need to be included in the system to satisfy these two conditions. It should be pointed out that the FTC setup considered in this paper, does not restrict the possibility to include more general controller architectures, where the number of actuators and/or sensors can be changed/modified in connection with faults. This subject is discussed in [13].

It is important to note that if S is stable, we do not need a Q_{CR}-parameter to stabilize the system. In this way, S can also be used for analyzing which faults are admissible and how large they can be before the closed-loop system will become unstable.

Based on the general equation for $S(\theta)$ given by (19), we have the following controller reconfiguration design problem.

Problem 1 ***The controller reconfiguration design problem*** *is defined as the problem of designing, if existent, a feedback controller $Q_{CR}(z)$, such that $\tilde{S}(Q_{CR})(z)$ given by*

$$\tilde{S}(Q_{CR})(z) = (I - Q_{CR}(z)S(\theta)(z))^{-1}$$

is stable, where $S(z)$ is given by (19).

This design problem is a pure discrete-time design problem and stabilizing $Q_{\mathrm{CR}}(z)$ controllers can be found by using standard discrete-time design methods.

So far, only the stability part with respect to multiplicative faults has been treated. This is the most important part of the CR. However, it will also in some cases be possible to design $Q_{\mathrm{CR}}(z)$ controller with respect to both closed-loop stability as well as closed-loop performance. Closing the loop of in Σ_θ in (3) with the feedback controller $K(Q_{\mathrm{CR}})$, we get the following closed loop transfer operators between the inputs d, r_k and the outputs e, $\tilde{r}_k$:

$$\Sigma_{SD,\theta,K} : \begin{cases} e = T_{ed}(\theta,s)d + T_{er}(\theta,s)r_k \\ \tilde{r}_k = T_{\tilde{r}d}(\theta,s)d + T_{\tilde{r}r}(\theta,s)r_k \end{cases} \tag{20}$$

where

$$\begin{aligned} T_{ed}(\theta,s) &= G_{ed}(\theta) + G_{eu}(\theta)\mathcal{H}M(z)(I - \tilde{U}(z)\mathcal{S} \\ &\quad \times G_{yw}(s)\theta(I - G_{zw}(s)\theta)^{-1}G_{zu}(s)\mathcal{H} \\ &\quad \times M(z))^{-1}\tilde{U}(z)\mathcal{S}G_{yd}(\theta) \\ T_{er}(\theta,s) &= G_{eu}(\theta)\mathcal{H}M(z)(I - \tilde{U}(z)\mathcal{S}G_{yw}(s) \\ &\quad \times \theta(I - G_{zw}(s)\theta)^{-1}G_{zu}(s)\mathcal{H}M(z))^{-1} \\ T_{\tilde{r}d}(\theta,s) &= \tilde{M}(z)(I - \mathcal{S}G_{yw}(s)\theta(I - G_{zw}(s)\theta)^{-1} \\ &\quad \times G_{zu}(s)\mathcal{H}U(z)\tilde{M}(z))^{-1}\mathcal{S}G_{yd}(\theta) \\ T_{\tilde{r}r}(\theta,s) &= S(\theta)(z) \end{aligned} \tag{21}$$

when a discrete-time nominal feedback controller is applied. Closing the loop from $\tilde{r}_k$ to r_k by Q_{CR}, the connection between the external input d and the external output e is then given by:

$$e = T_{SD,\mathrm{cl}}(\theta,s)d = \mathcal{F}_l(\Sigma_{SD,\theta,K}, Q_{\mathrm{CR}})d \tag{22}$$

Note that the closed loop operator is a periodic system with the sampling time as the period. Based on this, we have the following $\mathcal{H}_2$ and $\mathcal{H}_\infty$ controller reconfiguration design problems.

Problem 2 ***The optimal $\mathcal{H}_2$ controller reconfiguration design problem with performance recovery*** *is defined as the problem of designing, if existent, a feedback controller Q_{CR}, such that the closed loop operator $T_{SD,cl}(\theta,s)$ is stable and the $\mathcal{H}_2$ norm of $T_{SD,cl}(\theta,s)$ is minimized.*

Problem 3 *For a given number $\gamma > 0$,* ***The suboptimal $\mathcal{H}_\infty$ controller reconfiguration design problem with performance recovery*** *is defined as the problem of designing, if existent, a feedback controller Q_{CR}, such that the closed loop transfer function $T_{SD,cl}(\theta,s)$ is stable and the $\mathcal{H}_\infty$ norm of $T_{SD,cl}(\theta,s)$ is less than or equal to γ.*

Standard sampled-data design methods cannot be applied directly in connection with Problems 2 and 3. The reason is that the discrete-time nominal feedback controller is mixed with the continuous-time system. However, it is still possible to find a (sub)optimal controller Q_{CR} by using the lifting technique followed by standard discrete-time design. The lifting needs only to be applied on the continuous-time part of the closed loop system.

Let us consider a faulty closed loop system where a discrete-time nominal feedback controller is applied shown in Figure 4.

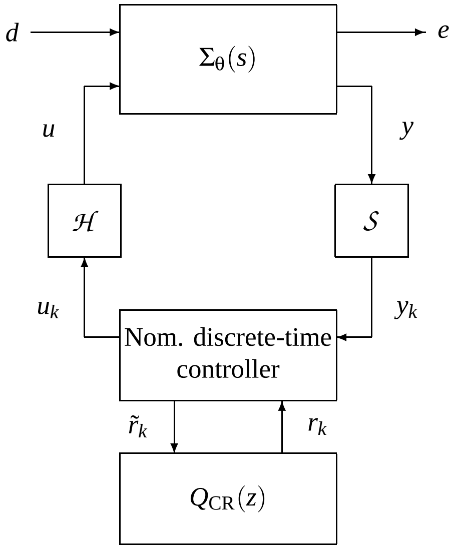

Fig. 4. Controller architecture with a discrete-time nominal feedback controller and a discrete-time reconfiguration block.

Using the lifting on the open loop system between the inputs d, u_k and the outputs e, y_k, we get the following equivalent discrete-time system:

$$\Sigma_{L,\theta}(z) : \begin{cases} e_k = LG_{ed}(\theta)L^{-1}d_k \;+\; LG_{eu}(\theta)\mathcal{H}u_k \\ \quad\; = \;G_{L,ed}(\theta)(z)d_k \;+\; G_{L,eu}(\theta)(z)u_k \\ y_k = \mathcal{S}G_{yd}(\theta)L^{-1}d_k \;+\; \mathcal{S}G_{yu}(\theta)\mathcal{H}u_k \\ \quad\; = \;G_{L,yd}(\theta)(z)d_k \;+\; G_{yu}(\theta)(z)u_k \end{cases} \tag{23}$$

where L is the lifting operator. It is important to note that the equivalent finite dimensional discrete-time system will depend on the subsequent controller design, see e.g. [7]. Introducing the lifted system in (23) in Figure 4, results in the equivalent discrete-time feedback system shown in Figure 5.

The open loop discrete-time transfer functions from the inputs d_k and r_k to the outputs e_k, $\tilde{r}_k$, in Figure 5 are then given by:

$$\Sigma_{L,\theta,K}(z) : \begin{cases} e_k = T_{\mathrm{L},ed}(z)d_k + T_{\mathrm{L},er}(z)r_k \\ \tilde{r}_k = T_{\mathrm{L},\tilde{r}d}(z)d_k + S(\theta)(z)r_k \end{cases} \tag{24}$$

where $T_{\mathrm{L},ed}(z)$, $T_{\mathrm{L},er}(z)$ and $T_{\mathrm{L},\tilde{r}d}(z)$ are functions of the lifted system given by (23). Let the closed loop transfer function from d_k to e_k be given by

$$T_{L,\mathrm{cl}}(\theta,z) = \mathcal{F}_l(\Sigma_{L,\theta,K}(z), Q_{\mathrm{CR}}) \tag{25}$$

The controller reconfiguration design problem is now transformed into a standard design problem. Design Problems 2 and 3 are then given by:

Problem 4 ***The optimal $\mathcal{H}_2$ controller reconfiguration design problem with performance recovery*** *is defined as*

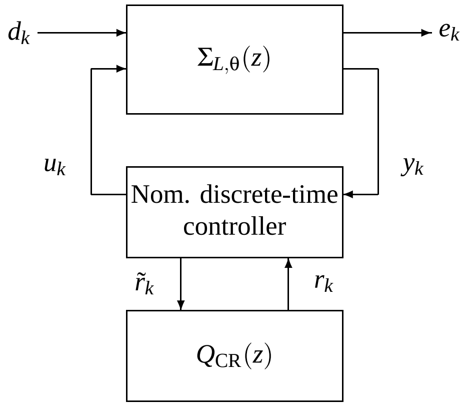

Fig. 5. Equivalent discrete-time controller architecture.

the problem of designing, if existent, a feedback controller $Q_{CR}(z)$, such that the closed loop operator $T_{L,cl}(\theta,z)$ is stable and the $\mathcal{H}_2$ norm of $T_{L,cl}(\theta,z)$ is minimized, where $T_{L,cl}(\theta,z)$ is given by (25).

Problem 5 *For a given number $\gamma > 0$, **The suboptimal $\mathcal{H}_\infty$ controller reconfiguration design problem with performance recovery** is defined as the problem of designing, if existent, a feedback controller $Q_{CR}(z)$, such that the closed loop transfer function $T_{L,cl}(\theta,z)$ is stable and the $\mathcal{H}_\infty$ norm of $T_{L,cl}(\theta,z)$ is less than or equal to γ, where $T_{L,cl}(\theta,z)$ is given by (25).*

VI. Conclusion

An architecture for fault tolerant controllers for sampled data systems has been considered in this paper. The architecture is based on the YJBK parameterization, which facilitates both the inclusion of a fault diagnosis part as well as a controller reconfiguration part.

A number of design problems for the controller reconfiguration part in the FTC architecture has been considered in this paper. It turns out that the controller design problem is a discrete-time design problem if stability of the faulty system is considered only. When performance is included in the design problem, the CR design problem is a sampled-data design problem. Here, lifting can be applied with advantage to transform the sampled-data design problem into an equivalent discrete-time design problem.

The FTC architecture can been derived in two versions, one version based on a nominal discrete-time feedback controller as described in this paper and one version based on a nominal continuous-time feedback controller. The second version of the architecture allows it to be used directly on a continuous-time feedback systems without any modifications. This second architecture is considered in details in the journal version of this paper, [16].

References

[1] B.D.O. Anderson. From Youla-Kucera to identification, adaptive and nonlinear control. *Automatica*, 34(12):1485–1506, 1998.

[2] M. Blanke, C.W. Frei, F. Kraus, R.J. Patton, and M. Staroswiecki. What is fault-tolerant control? In *Preprints of 4th IFAC Symposium on Fault Detection Supervision ans Safety for Technical Processes, SAFEPROCESS'2000*, pages 40–51, Budapest, Hungary, 2000.

[3] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault-Tolerant Control*. Springer, 2003.

[4] S. Boyd, V. Balakrishnan, C.H. Barratt, N.M. Khraishi, X. Li, D.G. Meyer, and S.A. Norman. A new CAD method and associated architectures for linear controllers. *IEEE Transactions on Automatic Control*, 33(3):268–283, 1988.

[5] S.P. Boyd and C.H. Barratt. *Linear controller design - limits of performance*. Prentice Hall, 1991.

[6] F. De Bruyne, B.D.O. Anderson, and N. Linard. The Hansen scheme revisited. In *Proceedings of the 37th Conference on Decision and Control*, pages 706–711, Tampa, FL, USA, 1998.

[7] T. Chen and B. Francis. *Optimal sampled-data control systems*. Springer-Verlag, 1995.

[8] M.A. Dahleh and I.J. Diaz-Bobillo. *Control of Uncertain systems*. Prentice Hall, 1995.

[9] P.M. Frank and X. Ding. Frequency domain approach to optimally robust residual generation and evaluation for model-based fault diagnosis. *Automatica*, 30:789–804, 1994.

[10] J. Gertler. *Fault detection and diagnosis in engineering systems*. Marcel Dekker, 1998.

[11] V. Kucera. Stability of discrete linear feedback systems. In *Proceedings of the 6th IFAC World Congress*, Boston, MA, 1975. Paper 44.1.

[12] H.H. Niemann and J. Stoustrup. Reliable control using the primary and dual Youla parameterization. In *Proceedings of the 41st IEEE Conference on Decision and Control*, page 6p, Las Vegas, NV, USA, 2002.

[13] H.H. Niemann and J. Stoustrup. An architecture for fault tolerant controllers. Submitted for publication, September 2003.

[14] H.H. Niemann and J. Stoustrup. Controller reconfiguration based on LTR design. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, Hawaii, USA, December 2003.

[15] H.H. Niemann and J. Stoustrup. Passive fault tolerant control of a double inverted pendulum - A case study example. In *Proceedings of IFAC SAFEPROCESS 2003*, pages 1029–1034, Washington, DC, USA, 2003.

[16] H.H. Niemann and J. Stoustrup. An architecture for sampled-data fault tolerant controllers. Submitted for publication, January 2004.

[17] H.H. Niemann, J. Stoustrup, and R.B. Abrahamsen. Switching between multivariable controllers. *Optimal Control - Application and Methods*, pages –, 2004.

[18] R. Patton. Fault tolerant control: The 1997 situation. In *Proceedings of the IFAC Symposium SAFEPROCESS'97*, pages 1033–1055, Hull, England, 1997.

[19] T.T. Tay, I.M.Y. Mareels, and J.B. Moore. *High performance control*. Birkhäuser, 1997.

[20] N.E. Wu, K. Zhou, and G. Salomon. Control reconfigurability of linear time-invariant systems. *Automatica*, 36:1767–1771, 2000.

[21] D.C. Youla, J.J. Bongiorno, and H.A. Jabr. Modern Wiener-Hopf design of optimal controllers - Part I: The single-input-output case. *IEEE Transactions on Automatic Control*, 21(1):3–13, 1976.

[22] D.C. Youla, H.A. Jabr, and J.J. Bongiorno. Modern Wiener-Hopf design of optimal controllers - Part II: The multivariable case. *IEEE Transactions on Automatic Control*, 21(3):319–338, 1976.

[23] K. Zhou, J.C. Doyle, and K. Glover. *Robust and optimal control*. Prentice Hall, 1995.

[24] K. Zhou and Z. Ren. A new controller architecture for high performance robust, and fault-tolerant control. *IEEE Transactions on Automatic Control*, 46(10):1613–1618, 2001.