

Controller Architectures for Switching

Henrik Niemann

Dept. of Electrical Engineering, Automation
 Technical University of Denmark
 DK-2800 Kgs. Lyngby, Denmark
 hhn@elektro.dtu.dk

Niels Kjølstad Poulsen

Dept. of Informatics and Math. Modeling
 Technical University of Denmark
 DK-2800 Kgs. Lyngby, Denmark
 nkp@imm.dtu.dk

Abstract—This paper investigate different controller architectures in connection with controller switching. The controller switching is derived by using the Youla-Jabr-Bongiorno-Kucera (YJBK) parameterization. A number of different architectures for the implementation of the YJBK parameterization are described and applied in connection with controller switching. An architecture that does not include inversion of the coprime factors is introduced. This architecture will make controller switching particular simple.

Keyword:

Controller architecture, system theory, YJBK parameterization, controller switching.

I. INTRODUCTION

The Youla-Jabr-Bongiorno-Kucera (YJBK) parameterization, [6], [20], [21] has been investigated in a number of books, [4], [7], [18], [22] and in papers [1], [8], [9], [10], [14], [19] for mention a few references. These books and papers deals with both theoretical results as well as with applications.

One of the later applications of the YJBK parameterization is in connection with Active Fault Diagnosis (AFD), [11], [12], [16], [17] and Fault Tolerant Control (FTC), [14].

Both the AFD and the FTC approaches make directly use of the YJBK architecture. In the FTC architecture described in [14] use the YJBK transfer function directly to change the controller when faults has been detected. The involved fault diagnosis in the FTC architecture is based on the same input and output signals that are connected through the YJBK transfer function.

A central element in an FTC architecture is to be able to change controller in a suitable way. One possibility is to use the YJBK architecture as the basic for an FTC architecture as described in [14]. A systematic way to switch between different controllers through the YJBK transfer function is needed. Controller switching through the YJBK parameterization was first shortly considered in [8]. Later, these results has been extended in [15]. Here, both controller switching and controller optimization using the YJBK transfer function has been investigated.

A drawback with using the YJBK architecture directly in connection with controller switching is the complexity of the YJBK transfer function. The transfer function include the coprime matrices from both controllers and from the system. This will e.g. give a YJBK transfer function 3 times the

order of the system when switching between two full order observer based feedback controllers.

The main focus in this paper is firstly to complete the results from [15] by considering an alternative implementation of the YJBK parameterization. Secondly, a new architecture for implementation of nominal controllers as well as for YJBK parameterized controllers is introduced. The new architecture give both a simple and direct implementation of feedback controllers as well as of the YJBK parameterization. It is shown that this new structure will result in a more simple YJBK transfer function for controller switching than using the standard implementation. It is shown that it is possible to implement all feedback controllers as a feedback from an output estimation error from an arbitrary observer. At last in this paper, the new controller structure is considered in connection with extension of the system. The system is extended by adding extra sensors and actuators. In this paper, only systems with extra actuators are considered.

The rest of this paper is organized as follows. In Section II, the system set-up is given together with some preliminary results for coprime factorization and the YJBK parameterization. Controller switching is introduced in Section III. A new architecture is introduced in Section IV following of Section V where the systems is extended with additional actuators. The paper is closed with a conclusion in Section VI.

II. SYSTEM SET-UP

Let a general system be given by:

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

where $d \in \mathcal{R}^r$ is a disturbance signal 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 system be controlled by a stabilizing feedback controller given by:

$$\Sigma_C : \{ u = Ky \quad (2)$$

A. Coprime factorization

Let a coprime factorization of the system G_{yu} from (1) and the stabilizing controller K from (2) be given by:

$$\begin{aligned} G_{yu} &= NM^{-1} = \tilde{M}^{-1}\tilde{N}, & N, M, \tilde{N}, \tilde{M} &\in \mathcal{R}_{\mathcal{H}_\infty} \\ K &= UV^{-1} = \tilde{V}^{-1}\tilde{U}, & U, V, \tilde{U}, \tilde{V} &\in \mathcal{R}_{\mathcal{H}_\infty} \end{aligned} \quad (3)$$

by using a more suitable selection of the implementation of the controller. The critical point in the implementation is the inversion of matrices. Using the Bezout equation, it is possible to remove \tilde{V}^{-1} or V^{-1} in Fig. 1 or in 2, respectively, by using a feedback loop instead. The above two implementations take then the following forms as shown in Fig. 3 and 4, respectively.

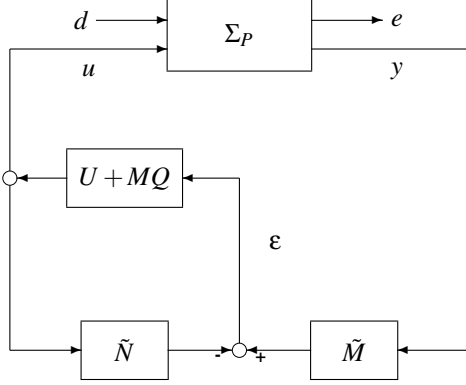


Fig. 3. A new implementation of the YJBK parameterization shown in Fig. 1.

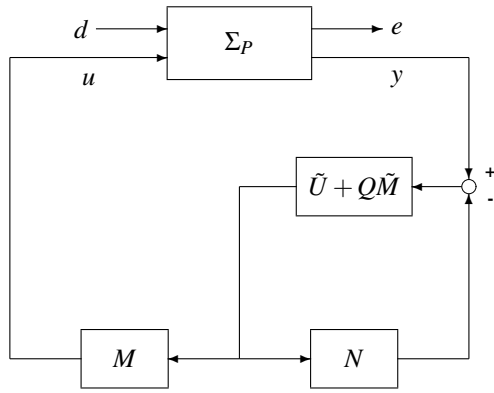


Fig. 4. A new implementation of the YJBK parameterization shown in Fig. 2.

A direct calculation of the control vector u of the YJBK parameterization in Fig. 3 gives:

$$\begin{aligned} u &= (U + MQ)(\tilde{M}y - \tilde{N}u) \\ &= (I + U\tilde{N} + MQ\tilde{N})^{-1}(U + MQ)\tilde{M}y \\ &= (M\tilde{V} + MQ\tilde{N})^{-1}(U\tilde{M} + MQ\tilde{M})y \\ &= (\tilde{V} + Q\tilde{N})^{-1}(\tilde{U} + Q\tilde{M})y \end{aligned}$$

which coincides with Fig. 1.

Alternatively, the right factored form given by (5) can also be derived directly from Fig. 3 in the following way:

$$\begin{aligned} u &= (U + MQ)(\tilde{M}y - \tilde{N}u) \\ &= (I + (U + MQ)\tilde{N})^{-1}(U + MQ)\tilde{M}y \\ &= (U + MQ)(I + \tilde{N}(U + MQ))^{-1}\tilde{M}y \\ &= (U + MQ)(\tilde{M}V + \tilde{M}NQ)^{-1}\tilde{M}y \\ &= (U + MQ)(V + NQ)^{-1}y \end{aligned}$$

which coincides with Fig. 2.

The controller structure in the Fig. 3 and 4 is very simple and it does not involve any inversions of matrices/transfer functions. As it will be shown below, it will also simplify the controller switching.

It should be mentioned that another controller architecture has been considered in connection with Loop Transfer Recovery (LTR), [13]. This LTR controller architecture is similar to the architectures shown in Fig. 3 and Fig. 4.

A. Controller Switching

Using the new implementation of the YJBK parameterization shown in Fig. 3 or 4, it is possible to reduce the implementation complexity of the implementation of Q_i significantly. The feed forward part of the controller in Fig. 3 (i.e. $U(Q_i(\alpha)) = U + MQ_i(\alpha)$) can now be rewritten into:

$$\begin{aligned} U(Q_i(\alpha)) &= U + \alpha MQ_i \\ &= U + \alpha M_i(\tilde{U}_i V - \tilde{V}_i U) \\ &= (1 - \alpha)U + \alpha((I - M_i \tilde{V}_i)U + M_i \tilde{U}_i V) \\ &= (1 - \alpha)U + \alpha(-U_i \tilde{N}_i U + U_i \tilde{M}_i V) \\ &= (1 - \alpha)U + \alpha U_i \tilde{M}_i (-G_{yu} U + V) \\ &= (1 - \alpha)U + \alpha U_i \tilde{M}_i \tilde{M}^{-1} \end{aligned} \quad (12)$$

In the special case where $\tilde{M} = \tilde{M}_i$, $U(Q_i(\alpha))$ take the following simple form:

$$U(Q_i(\alpha)) = (1 - \alpha)U + \alpha U_i \quad (13)$$

Using either (12) or (13), the controller switching get very simple. It will only require a calculation of the U_i for the controller K_i and possibly also $\tilde{M}_i^{-1} \tilde{M}$. The implementation of the controller switching using $U(Q_i)$ given by (12) will give a lower order of the controller than using the general equation for Q_i given by (9).

Equivalent, using the set-up in Fig. 4, the feed forward term (i.e. $\tilde{U}(Q_i(\alpha)) = \tilde{U} + Q_i(\alpha)\tilde{M}$) take the following form:

$$\begin{aligned} \tilde{U}(Q_i(\alpha)) &= \tilde{U} + \alpha Q_i \tilde{M} \\ &= (1 - \alpha)\tilde{U} + \alpha M^{-1} M_i \tilde{U}_i \end{aligned} \quad (14)$$

or

$$\tilde{U}(Q_i(\alpha)) = (1 - \alpha)\tilde{U} + \alpha \tilde{U}_i \quad (15)$$

in the simple case where $M = M_i$.

It is possible to obtain controllers as a combination of a number of controllers by using the YJBK parameterization, [15]. This can be done by using a YJBK transfer function given by:

$$Q = \sum_{i=1}^s \alpha_i Q_i \quad (16)$$

Note that there is no condition that $\alpha = \sum_{i=1}^s \alpha_i$ should be equal to 1. Using Q given by (16), it will be possible to optimize the controller, based on a number of predesigned

controllers. The feed forward part of the controller in Fig. 3 $U(Q)$ can now be rewritten into for $s = 2$:

$$\begin{aligned}
U(Q) &= U + \alpha_1 M Q_1 + \alpha_1 M Q_2 \\
&= U + \alpha_1 M_1 (\tilde{U}_1 V - \tilde{V}_1 U) + \alpha_2 M_2 (\tilde{U}_2 V - \tilde{V}_2 U) \\
&= (1 - \alpha_1 - \alpha_2) U + \alpha_1 ((I - M_1 \tilde{V}_1) U + M_1 \tilde{U}_1 V) \\
&\quad + \alpha_2 ((I - M_2 \tilde{V}_2) U + M_2 \tilde{U}_2 V) \\
&= (1 - \alpha_1 - \alpha_2) U + \alpha_1 (-U_1 \tilde{N}_1 U + U_1 \tilde{M}_1 V) \\
&\quad + \alpha_2 (-U_2 \tilde{N}_2 U + U_2 \tilde{M}_2 V) \\
&= (1 - \alpha_1 - \alpha_2) U + \alpha_1 U_1 \tilde{M}_1 (-G_{yu} U + V) \\
&\quad + \alpha_2 U_2 \tilde{M}_2 (-G_{yu} U + V) \\
&= (1 - \alpha_1 - \alpha_2) U + \alpha_1 U_1 \tilde{M}_1 \tilde{M}^{-1} + \alpha_2 U_2 \tilde{M}_2 \tilde{M}^{-1}
\end{aligned} \tag{17}$$

or in a more compact notation

$$U(Q) = (1 - \alpha) U + \left(\sum_{i=1}^s \alpha_i U_i \tilde{M}_i \right) \tilde{M}^{-1} \quad \alpha = \sum_{i=1}^s \alpha_i \tag{18}$$

Again, if $\tilde{M}_i = \tilde{M}$, then (18) take the following simple form:

$$U(Q) = (1 - \alpha) U + \sum_{i=1}^s \alpha_i U_i \tag{19}$$

The equivalent equations can also be derived when for YJBK implementation shown in Fig. 4. $\tilde{U}(Q)$ is then given by:

$$\tilde{U}(Q) = (1 - \alpha) \tilde{U} + M^{-1} \left(\sum_{i=1}^s \alpha_i M_i \tilde{U}_i \right) \tag{20}$$

or

$$\tilde{U}(Q) = (1 - \alpha) \tilde{U} + \sum_{i=1}^s \alpha_i \tilde{U}_i \tag{21}$$

when $M_i = M$ can be applied.

The controller switching based on the architecture in Fig. 3 can also be used in connection with implementation of arbitrary controllers. A controller $K_i = U_i V_i^{-1}$ can be implemented by using the control input given by:

$$u = U_i \tilde{M}_i \tilde{M}^{-1} \varepsilon \tag{22}$$

This is a direct consequence of (12) (for $\alpha = 1$). The ε vector given by

$$\varepsilon = \begin{pmatrix} \tilde{M} & -\tilde{N} \\ y \\ u \end{pmatrix} \tag{23}$$

can always be implemented as an output estimation error vector, i.e. $\begin{pmatrix} \tilde{M} & -\tilde{N} \end{pmatrix}$ describe an observer. ε will be the innovation vector when a Kalman filter is applied. It will always be possible to obtain a coprime factorization that will include an observer. This has been shown in [8]. Based on the controller architecture in Fig. 3, it is possible to implement feedback controller using the general structure shown in Fig 5.

The controller architecture shown in Fig. 5 has also a relation with fault diagnosis and active fault diagnosis. The ε vector is here used as the residual vector for detection and diagnosis of additive and parametric faults in the system, [2], [3], [5], [11], [16].

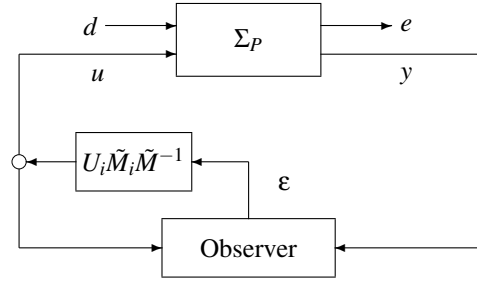


Fig. 5. An observer based implementation of the feedback controller K_i .

It will also be possible to reformulate the controller architecture shown in Fig. 4 in a similar way.

B. Input-Output Implementation

Another issue in connection with the implementation of the controller switching is the accessibility of internal signal vectors in the nominal controller, special the input point for the η vector in between \tilde{U} and \tilde{V}^{-1} in Fig. 1 (or the ε vector in Fig. 2). It is therefore interesting to based the parameterization part of the controller only on the measurement vector y and control vector u . The original formulation of the YJBK parameterization given by (7) can be rewritten into, [7], [18]:

$$\begin{aligned}
K(Q) &= K(0) + \mathcal{F}_l \left(\begin{pmatrix} 0 & \tilde{V}^{-1} \\ V^{-1} & -V^{-1} N \end{pmatrix}, Q \right) \\
&= K(0) + \tilde{V}^{-1} Q (I + V^{-1} N Q)^{-1} V^{-1} \\
&= K(0) + \tilde{V}^{-1} Q (V + N Q)^{-1} \\
&= K(0) + \tilde{K}(Q)
\end{aligned} \tag{24}$$

As it can be seen from (24), this realization will require that \tilde{V}^{-1} need to be implemented separately in $\tilde{K}(Q)$. This will give an unstable transfer function for nominal unstable controllers. It is not impossible to use this realization for an unstable nominal controller, but it is not to prefer.

To remove the inversion of \tilde{V} , another implementation has been considered in e.g. [7]. The implementation is shown in Fig. 6.

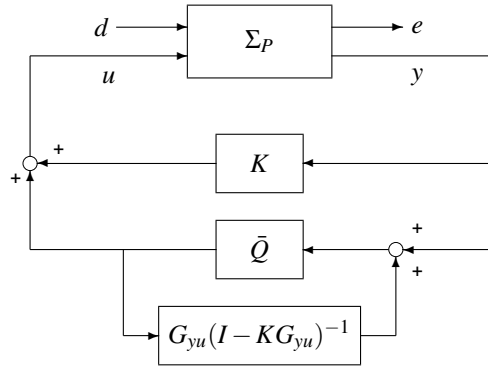


Fig. 6. An implementation of the YJBK parameterization when K is a stable controller, [7].

The controller in Fig. 6 is given by:

$$K(\bar{Q}) = K(0) + \bar{Q} (I + N \bar{V} \bar{Q})^{-1} \tag{25}$$

A simple calculation show that the connection between Q in (7) and \bar{Q} in (25) is, ([7]):

$$\bar{Q} = \tilde{V}^{-1} Q V^{-1} \quad (26)$$

or

$$Q = \tilde{V} \bar{Q} V$$

It is possible to remove either V^{-1} or \tilde{V}^{-1} from (26) by using the controller implementation shown in Fig. 3 or in Fig. 4. From (4) we have directly that V^{-1} can be written as:

$$V^{-1} = U(I + \tilde{N}U)^{-1} \tilde{M} \quad (27)$$

From Fig. 3 we also have that V^{-1} is the transfer function from y to ε given by ($r=0$)

$$\varepsilon = U(I + \tilde{N}U)^{-1} \tilde{M} y = V^{-1} y$$

Further, ε can also be written directly as function of the measurement output and the control input given by:

$$\varepsilon = \tilde{M} y - \tilde{N} u$$

Equivalent, \tilde{V}^{-1} can also be substituted by using the double Bezout equation and the controller set-up shown in Fig. 4. It should be pointed out that it is only possible to remove either V^{-1} or \tilde{V}^{-1} , not both at the same time without affecting the general YJBK parameterization.

V. SYSTEM EXTENSION

In some cases, it is relevant to consider the possibility to extend the system with extra sensors and/or actuators. This will give an additional freedom in the YJBK parameterization. Some preliminary results has been given in [14] and more detailed results has been given in [10]. These results are all derived with respect to the controller implementation shown in Fig. 1. Instead of using the implementation in Fig. 1, it is possible to use the two implementations shown in Fig. 3 and 4. It will here be possible to simplify the implementation of the YJBK parameterization.

Without loss of generality, let's only consider the case when additional actuators is added to the system. The case where additional sensors or actuators and sensors are added to the system can be derived in the same way by using the general results from [10].

Let the system Σ_P given by (1) be extended with some extra actuators resulting in the following system:

$$\Sigma_{P,\text{ext}} : \begin{cases} e(t) = G_{ed}d(t) + G_{eu,\text{ext}}u_{\text{ext}}(t) \\ y(t) = G_{yd}d(t) + G_{yu,\text{ext}}u_{\text{ext}}(t) \end{cases} \quad (28)$$

with

$$\begin{aligned} u_{\text{ext}}(t) &= \begin{pmatrix} u(t) \\ u_a(t) \end{pmatrix} \\ G_{eu,\text{ext}} &= \begin{pmatrix} G_{eu} & G_{eu,a} \end{pmatrix} \\ G_{yu,\text{ext}} &= \begin{pmatrix} G_{yu} & G_{yu,a} \end{pmatrix} \end{aligned}$$

where u_a is the additional actuator inputs.

It has been shown in [10] that the coprime matrices has the following structure:

The above eight matrices has the following form:

$$\begin{aligned} N_{\text{ext}} &= \begin{pmatrix} N & N_1 \end{pmatrix} & M_{\text{ext}} &= \begin{pmatrix} M & M_1 \\ 0 & I \end{pmatrix} \\ U_{\text{ext}} &= \begin{pmatrix} U \\ 0 \end{pmatrix} & V_{\text{ext}} &= V \\ \tilde{M}_{\text{ext}} &= \tilde{M} & \tilde{N}_{\text{ext}} &= \begin{pmatrix} \tilde{N} & \tilde{N}_1 \end{pmatrix} \\ \tilde{V}_{\text{ext}} &= \begin{pmatrix} \tilde{V} & \tilde{V}_1 \\ 0 & I \end{pmatrix} & \tilde{U}_{\text{ext}} &= \begin{pmatrix} \tilde{U} \\ 0 \end{pmatrix} \end{aligned} \quad (29)$$

This structure can be obtained when an observer based feedback controller is applied as the nominal controller or when the general state space description of the coprime matrices given in [18] is applied. This structure might not be obtained in other cases.

Based on the coprime matrices for the extended system given in (29), the standard implementation of the YJBK parameterization can be derived. However, using instead the implementation shown in Fig. 3 will give a much more simple structure of the complete controller. Using the coprime matrices directly in Fig. 3 given the block diagram shown in Fig. 7 where the YJBK transfer function for the extended system is given by:

$$Q_{\text{ext}} = \begin{pmatrix} Q \\ Q_1 \end{pmatrix}$$

where Q_1 is the transfer function due to the extension.

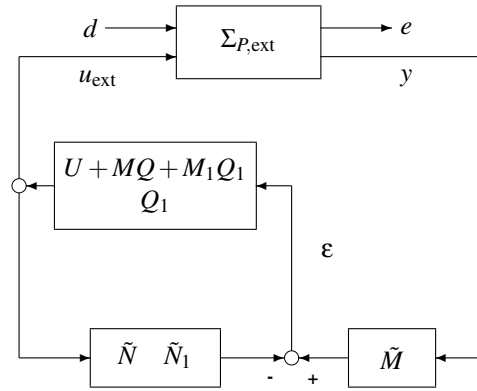


Fig. 7. Implementation of the YJBK parameterization when additional actuators has been included.

A little manipulation of the block diagram will give a much more clear structure of the controller. This has been done in Fig. 8.

The transfer function from ε_1 to \tilde{u}_{ext} in Fig. 8 is the additional term that will occur when extra actuators are added to the system.

Equivalent, including extra sensors or both extra sensors and actuators can be handled in the same way.

Another important issue in connection with this is the closed-loop stability aspect when the applied sensors and/or actuators are changed. In this case a stability analysis of the closed-loop system is needed. Such an analysis is independent of the controller implementation. This stability analysis can be found in [10].

