Fault Tolerant Control for Unstable Systems: A Linear Time Varying Approach

Jakob Stoustrup

Department of Control Engineering
Aalborg University
Fr. Bajers Vej 7C
DK-9220 Aalborg
Denmark
E-mail: jakob@control.aau.dk
www.control.aau.dk/~jakob/

Henrik Niemann

Ørsted•DTU, Automation
Technical University of Denmark
Building 326
DK-2800 Lyngby
Denmark
E-mail: hhn@oersted.dtu.dk
www.oersted.dtu.dk/staff/all/hhn/

***Abstract*— In (passive) fault tolerant control design, the objective is to find a fixed compensator, which will maintain a suitable performance - or at least stability - in the event that a fault should occur. A major theoretical obstacle to obtain this objective, is that even if the system models corresponding to the occurrence of various faults are simultaneously stabilizable by a linear, time-invariant compensator, this compensator might have to be of very high order, as shown in a recent publication. In this paper, we propose a design procedure for a time-varying compensator, which overcomes the obstacle for any finite number of faults with a controller order of no more than the plant order. The performance of this compensator might be poor, but a heuristic procedure for improving the performance is also shown, and an example demonstrates that this improvement can be truly significant.**

I. Introduction

The interest for using fault tolerant controllers is increasing. A number of theoretical results as well as application examples has now been described in the literature, see e.g. [1], [2], [3], [4], [5], [6], [7], [8], [9], [10] to mention some of the relevant references in this area.

The approaches to fault tolerant control can be divided into two main classes: *Active* fault tolerant control and *passive* fault tolerant control. In active fault tolerant control, the idea is to introduce a fault detection and isolation block in the control system. Whenever a fault is detected and isolated, a supervisory system takes action, and modifies the structure and/or the parameters of the feedback control system. In contrast, in the passive fault tolerant control approach, a fixed compensator is designed, that will maintain (at least) stability if a fault occurs in the system in consideration.

This paper will only discuss the passive fault tolerant control approach, also sometimes referred to as *reliable* control. This approach has mainly two motivations. First, designing a fixed compensator can be made in much simpler hardware and software, and might thus be admissible in more applications. Second, classical reliability theory states that the reliability of a system decreases rapidly with the complexity of the system. Hence, although an active fault tolerant control system might in principle accomodate specific faults very efficiently, the added complexity of the overall system by the fault detection system and the supervisory system itself, might in fact sometimes deteriorate plant reliability.

In [11], a fault tolerant control problem has been addressed for systems, where specific sensors could potentially fail such that the corresponding outputs were unavailable for feedback, whereas other outputs were assumed to be available at all times.

In [12, Sec. 5.5], the question of fault tolerant parallel compensation has been discussed, i.e. whether it is possible to design two compensators such that any of them alone or both in parallel will internally stabilize the closed loop system.

In a recent paper [13], it was shown that on one hand, under mild conditions (stabilizability, detectability), a linear time-invariant (LTI) finite-dimensional controller always exists which stabilizes the system both in the nominal situation, as well as in case any one sensor should fail (a dual result is given for actuator faults). On the other hand, it was also shown in [13] that even for a second order system, the required controller order to achieve this simultaneous stabilization can be unbounded.

Since very high controllers orders are often unacceptable for a number of good reasons, we propose in this paper instead to use a linear *time-varying* (LTV) fault tolerant compensator, and show that this type of compensator - in contrast to the time-invariant case - can be designed with a controller order of at most the same as the plant. The approach is based on the ability of LTV compensators to achieve simultaneous stabilization of several systems. A seminal paper in this context was [14], where the authors showed that for every finite set of plants, a linear time-varying controller can be designed which provides closed loop stability. For further literature regarding LTV controllers - see the references in [14].

0-7803-8335-4/04/$17.00 ©2004 AACC

II. Problem Formulation

In the sequel, we shall consider systems of the form:

$$\Sigma_0 \; : \; \left\{ \begin{array}{rcl} x(k+1) & = & A_{\mathrm{p},0}x(k) + B_{\mathrm{p},0}u(k) \\ y(k) & = & C_{\mathrm{p},0}x(k) \end{array} \right. \tag{1}$$

Thus, we shall restrict attention to discrete time systems. The proposed methodology, however, carries over to continuous time systems with only minor modifications.

It is assumed that the system (1) might fail in one of several a priori known ways. The faults could be actuator faults, sensor faults, or internal faults, which change the dynamics of the system. In either case, the faulty system will be described by a model of the form:

$$\Sigma_i \; : \; \left\{ \begin{array}{rcl} x(k+1) & = & A_{\mathrm{p},i}x(k) + B_{\mathrm{p},i}u(k) \\ y(k) & = & C_{\mathrm{p},i}x(k) \end{array} \right. , \quad i = 1,\ldots,q \tag{2}$$

The fault tolerant problem considered in this paper is to find a linear (but not necessary time-invariant) feedback compensator, which stabilizes the nominal system (1), but which also preserves stability if any one of the faults described by the models of faulty system (2) occurs.

III. Main Result

Theorem 1: Consider an n_{p}th order faulty system Σ with a nominal model of the form (1). Assume that the system can fail in one among q possible ways, each giving rise to a model $\Sigma_i, i = 1,\ldots,q$ of the form (2). Further, we assume that the $q+1$ models $\Sigma_i, i = 0,\ldots,q$ are all stabilizable and detectable. Then, there exists a time-varying compensator of order at most n_{p}, such that the closed loop system remains stable even if one of the q faults should occur at one time instance.

Proof: Since there is no infinite switching between models going on, the stability condition reduces to asymptotic stability for each of the $q+1$ individual closed loop systems formed by the compensator and either the nominal or one of the faulty systems. Hence, Theorem 1 can be reduced to a simultaneous (or multi-model) stabilization problem, for which it is actually well-known in the control community, that an nth order linear, time-varying compensator always suffices. We shall, however, repeat the proof here since it is simple, and more importantly, since it is constructive and will be used in the sequel.

To that end, we shall without loss of generality assume that the models both of the nominal and of the faulty systems are minimal. Let $K_0, K_1, \ldots, K_q$ be linear, time-invariant compensators with a minimal model of the form:

$$\Sigma_{K_i} \; : \; \left\{ \begin{array}{rcl} \xi(k+1) & = & A_{\mathrm{c},i}\xi(k) + B_{\mathrm{c},i}y(k) \\ u(k) & = & C_{\mathrm{c},i}\xi(k) \end{array} \right. \tag{3}$$

such that K_0 is a dead-beat controller (can be achieved due to minimality) for the nominal system,

$$\begin{array}{rcl} x(k+1) & = & A_{\mathrm{p},0}x(k) + B_{\mathrm{p},0}u(k) \\ y(k+1) & = & C_{\mathrm{p},0}x(k) \end{array}$$

and such that for $i = 1,\ldots,q$, K_i is a dead-beat controller for Σ_i

$$\begin{array}{rcl} x(k+1) & = & A_{\mathrm{p},i}x(k) + B_{\mathrm{p},i}u(k) \\ y(k+1) & = & C_{\mathrm{p},i}x(k) \end{array}$$

Thus, each of the $q+1$ models and corresponding controllers generates a closed loop system of the form

$$\left(\begin{array}{c} x(k+1) \\ \xi(k+1) \end{array} \right) = A_{\mathrm{cl},i} \left(\begin{array}{c} x(k) \\ \xi(k) \end{array} \right)$$

where

$$A_{\mathrm{cl},i} = \left(\begin{array}{cc} A_{\mathrm{p},i} & B_{\mathrm{p},i}C_{\mathrm{c},i} \\ B_{\mathrm{c},i}C_{\mathrm{p},i} & A_{\mathrm{c},i} \end{array} \right)$$

has the property:

$$A_{\mathrm{cl},i}^{n_{\mathrm{p}}+n_{\mathrm{c}}} = 0, \; i = 0,\ldots,q \tag{4}$$

where n_{c} is the (largest) order of the $q+1$ time-invariant compensators.

We now introduce the following time-varying compensator:

$$\Sigma_{K_{\mathrm{TV}}} \; : \; \left\{ \begin{array}{rcl} \xi(k+1) & = & A_{\mathrm{c}}(k)\xi(k) + B_{\mathrm{c}}(k)y(k) \\ u(k) & = & C_{\mathrm{c}}(k)\xi(k) \end{array} \right. \tag{5}$$

where

$$\begin{array}{rcl} A_{\mathrm{c}}(k) & = & A_{\mathrm{c},j(k)} \\ B_{\mathrm{c}}(k) & = & B_{\mathrm{c},j(k)} \\ C_{\mathrm{c}}(k) & = & C_{\mathrm{c},j(k)} \\ j(k) & = & \left(\frac{k - \left(k \mod \left(n_{\mathrm{p}}+n_{\mathrm{c}}\right)\right)}{n_{\mathrm{p}}+n_{\mathrm{c}}} \mod q+1 \right) \end{array} \tag{6}$$

Thus, as described in (6), each controller is repeated $n_{\mathrm{p}}+n_{\mathrm{c}}$ times in a cycle through all $q+1$ controllers.

Now, it is easy to see, that the closed loop system resulting from joining K_{TV} and the nominal system or either of the faulty systems:

$$\left(\begin{array}{c} x(k+1) \\ \xi(k+1) \end{array} \right) = \left(\begin{array}{cc} A_{\mathrm{p},i} & B_{\mathrm{p},i}C_{\mathrm{c}}(k) \\ B_{\mathrm{c}}(k)C_{\mathrm{p},i} & A_{\mathrm{c}}(k) \end{array} \right) \left(\begin{array}{c} x(k) \\ \xi(k) \end{array} \right)$$

must converge to the origin in finite time from any initial state, since

$$\begin{array}{l} \left(\begin{array}{c} x(q(n_{\mathrm{p}}+n_{\mathrm{c}})+1) \\ \xi(q(n_{\mathrm{p}}+n_{\mathrm{c}})+1) \end{array} \right) \\ \qquad = \prod_{k=0}^{q(n_{\mathrm{p}}+n_{\mathrm{c}})} \left(\begin{array}{cc} A_{\mathrm{p},i} & B_{\mathrm{p},i}C_{\mathrm{c}}(k) \\ B_{\mathrm{c}}(k)C_{\mathrm{p},i} & A_{\mathrm{c}}(k) \end{array} \right) \left(\begin{array}{c} x(0) \\ \xi(0) \end{array} \right) \end{array}$$

will contain at least one of the sequences (4), and therefore

$$\left(\begin{array}{c} x(q(n_{\mathrm{p}}+n_{\mathrm{c}})+1) \\ \xi(q(n_{\mathrm{p}}+n_{\mathrm{c}})+1) \end{array} \right) = 0$$

Thus, K_{TV} is fault tolerant compensator, which stabilizes the system in the nominal situation, and preserves stability in the event that any of the q faults occur. ■

We shall use this constructive proof in the design procedure described below. As a consequence, the design will be based on a number of LTI dead-beat controllers, each

acting in a certain time interval. However, it is well-known that dead-beat controllers are not always robust, and that they may shown violent transients if the plant is not very well known. It is therefore worthwhile observing that the dead-beat property is convenient in the proof above but not strictly necessary. In practice, it is only necessary to make the poles 'sufficiently small'. To be more specific, stability can be guaranteed, if the product of the largest singular values of possible sequences all are bounded by 1.

The continuous time case can be handled in much the same manner, although an equivalent to the dead-beat controller is lacking. Instead, the stability argument can be established by considering state transition matrices of the form

$$\Phi_i = \exp\left(A_{\mathrm{cl},i} T_i\right)$$

where $A_{\mathrm{cl},i}$ is the closed loop system matrix in the ith time interval, and T_i is the duration of that time interval. Just like in the discrete time case, the product of the largest singular values of the Φ_i's for all possible sequences then must be bounded by 1.

IV. Design Procedure

The specific LTV compensator proposed in the proof of Theorem 1, which is described by (5) and (6), is only guaranteed to stabilize the faulty system. It might, however, be rather poor in terms of performance. The reason for this is, that the nominal compensator is only in operation in a fraction of the time which is $\frac{1}{q}$. Thus, if the LTI compensators designed for the faulty situations are highly suboptimal in the nominal situation, then also the performance of the LTV compensator will be rather suboptimal. In this section, we shall approach the performance problem by means of time-scheduling.

The LTV fault tolerant scheme suggested in this paper will inevitably introduce a trade-off between performances in the nominal and the faulty situations. One way to overcome part of this dilemma is to make a multi-model design in the first case, rather than designing the $q+1$ LTI controllers entirely independent. A description of the large number of methods for multi-model design falls outside the scope of this paper, and in the sequel we shall assume that the $q+1$ LTI compensators are given and fixed.

An entirely different handle, however, to improve performance of the LTV fault tolerant compensator suggested above, is to modify the number of samples, each of the individual LTI compensators is applied in each cycle.

To that end, we propose a heuristic (re-)design procedure, which is an iterative scheme, based on a quantification on how poorly each LTI compensator is performing in the loops, for which it was *not* designed.

In particular, we shall study the following square matrix:

$$T = \begin{pmatrix} \left\| A_{\mathrm{cl},00}^{r_0} \right\| & \cdots & \left\| A_{\mathrm{cl},0q}^{r_q} \right\| \\ \vdots & \ddots & \vdots \\ \left\| A_{\mathrm{cl},q0}^{r_0} \right\| & \cdots & \left\| A_{\mathrm{cl},qq}^{r_q} \right\| \end{pmatrix} \tag{7}$$

where $A_{\mathrm{cl},ij}$ denotes the state transition matrix for the closed loop system achieved by joining the ith system (see (1) and (2)) with the jth LTI compensator (see (3)), and the design parameters $r_0, \ldots, r_q$ are integer powers. Each r_i indicates the number of repetitions of controller K_i in each cycle. To be more specific, the controller proposed has the form:

$$\Sigma_{K_{\mathrm{TV}}} \; : \; \left\{ \begin{array}{rcl} \xi(k+1) & = & A_{\mathrm{c}}(k)\xi(k) \; + \; B_{\mathrm{c}}(k)y(k) \\ u(k) & = & C_{\mathrm{c}}(k)\xi(k) \end{array} \right. \tag{8}$$

where

$$\left\{ \begin{array}{rcl} A_{\mathrm{c}}(k) & = & A_{\mathrm{c},j(k)} \\ B_{\mathrm{c}}(k) & = & B_{\mathrm{c},j(k)} \\ C_{\mathrm{c}}(k) & = & C_{\mathrm{c},j(k)} \end{array} \right\} \tag{9}$$

and

$$j(\cdot) = \overbrace{\underbrace{0,\ldots,0}_{r_0 \text{ times}}, \underbrace{1,\ldots,1}_{r_1 \text{ times}}, \ldots, \underbrace{q,\ldots,q}_{r_q \text{ times}}}^{\text{first cycle}}, \underbrace{0,\ldots,0}_{r_0 \text{ times}}, \ldots$$

and where $A_{\mathrm{c},j}$, $B_{\mathrm{c},j}$, and $C_{\mathrm{c},j}$ are the controller parameters introduced in (3).

The following algorithm iterates on the duration of each LTI controller by evaluating T as defined in (7).

Algorithm 1 (Time-scheduling):

1) Choose a minimal and a maximal duration, $r_{\min}$ and $r_{\max}$, resp., for the LTI controllers
2) Choose a set of initial values for the durations $r_0, \ldots, r_q$
3) Compute T as defined in (7)
4) Find the largest value in T and note the corresponding indices $(i_{\max}, j_{\max})$
5) Find the column in T with the property that its largest value is smallest among all the columns and note the corresponding index $j_{\min}$
6) Decrease $r_{j_{\max}}$ and/or increase $r_{j_{\min}}$, if $r_{j_{\max}} > r_{\min}$ and $r_{j_{\min}} < r_{\max}$. If the latter is not the case, Steps 4 and 5 are repeated with the corresponding columns removed
7) Repeat from Step 3 unless indices did not change in past iteration

It should be noted that the algorithm is heuristic, based on the assumption, that it will help to reduce the influence of the LTI controllers that performs poorer on other systems than they were designed for. The algorithm does not guarantee optimality and in some cases, the resulting controller might not even be stabilizing. Stability, however, can be guaranteed if the following two measures are taken:

1) Each LTI controller is choosen as a dead-beat controller
2) $r_{\min}$ is chosen at least as large as $n_{\mathrm{p}} + n_{\mathrm{c}}$

With these two precautions, stability can be proved among the same lines as Theorem 1.

V. Example

The example in this section is chosen to illustrate the following points:

- The LTV scheme in this paper can be used for stabilizing a system for which the alternative LTI compensator might be of high order
- Performance might not be excellent, if the individual LTI controllers are not chosen to satisfy some reasonable cross-performance
- Performance can be improved by the algorithm described in Section IV

Thus, the example should not be seen as an ideal and realistic design study.

The system in consideration is the following:

$$\begin{aligned} x(k+1) &= \begin{pmatrix} 2 & 0 \\ -1 & \frac{1}{2} \end{pmatrix} x(k) + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u(k) \\ \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} &= \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} x(k) \end{aligned}$$

with the following transfer matrix:

$$\frac{1}{z^2-2.5z+1} \begin{pmatrix} z-\frac{3}{2} \\ z-\frac{5}{2} \end{pmatrix}$$

For this system, it is assumed that the two sensors corresponding to the two outputs can both fail. Notice, that the system degrades to an unstable, non-minimum phase system in either of the two faulty situations. In [13] it is shown that systems of this type might require LTI controllers of arbitrarily high order just to achieve (fault-tolerant) stability. In the sequel, we shall demonstrate a systematic design of an LTV controller of order at most n_p.

First, we design three LTI compensators for each of the three situations:

$$\begin{aligned} K_0 &= \frac{1}{z+2.5} \begin{pmatrix} -10.63 \\ 5.375 \end{pmatrix} \\ K_1 &= \frac{1}{z^2+2.5z-16.13} \begin{pmatrix} -21.37z+10.75 \\ 0 \end{pmatrix} \\ K_2 &= \frac{1}{z^2+2.5z+26.56} \begin{pmatrix} 0 \\ 21.31z-10.62 \end{pmatrix} \end{aligned}$$

which are dead-beat compensators for the nominal situation, the situation where only the sensor corresponding to y_1 functions, and the situation where only the sensor corresponding to y_2 functions, respectively.

Starting Algorithm 1 with the following values:

$$r_0 = r_1 = r_2 = 6$$

gives the following matrix of norms of powers:

$$T = \begin{pmatrix} 6.2373e-015 & 0 & 0 \\ 1.4326e+003 & 0 & 8.1391e+004 \\ 4.0792e+003 & 4.8727e+004 & 0 \end{pmatrix}$$

A simulation with 180.000 samples and gaussian noise on both plant states shows the following variances on the first output variable (its 'real' value - whether the sensor shows it or not):

Variance for nominal system:	$4.2537e-007$
Variance if only first sensor functions:	$1.3058e+007$
Variance if only second sensor functions:	$1.1956e+010$

It can be seen that although stability in theory is obtained, the variances are so large that the system in practice most likely will be unstable.

After some iterations, the algorithm stops at the following values:

$$r_0 = 0, \quad r_1 = 4, \quad r_2 = 4$$

giving the following matrix of norms of powers

$$T = \begin{pmatrix} 1.0000e+000 & 0 & 0 \\ 1.0000e+000 & 0 & 2.7330e+003 \\ 1.0000e+000 & 1.6108e+003 & 0 \end{pmatrix}$$

A simulation with 80.000 samples and state noise as above gives the following variances:

Variance for nominal system:	$8.9325e-007$
Variance if only first sensor functions:	$1.9303e+000$
Variance if only second sensor functions:	$2.4521e+000$

It can be seen that now that the output variances are $7-10$ orders of magnitude smaller. The price, however, is a doubling of the output variance in the nominal case.

Figure 1 shows a simulation for the same system and with the same controllers, but with the time scheduling parameters set as: $r_0 = 2$, $r_1 = 4$, $r_2 = 4$. The excitation is sinusoidal

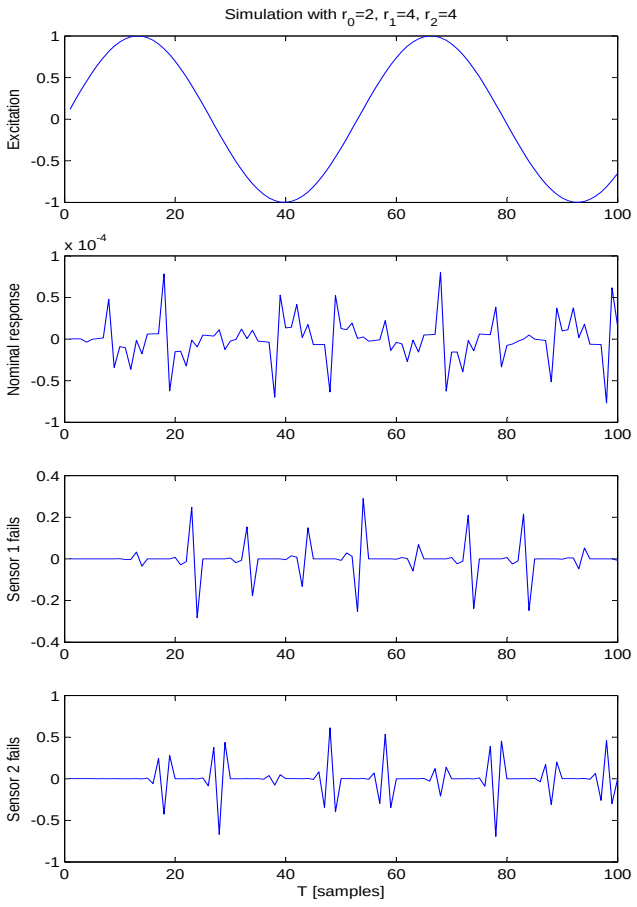

Fig. 1. Simulation with sinusoidal excitation. Note the highly different scales in the three cases. The significant transients are caused by the aggressive nature of the dead-beat controllers.

and thus smooth. Nevertheless, significant transients occur which are due to switching between dead-beat controllers. A less aggressive design would have given slower responses, but less over-shoot. The overall conclusion, however, is that the fault tolerant control scheme functions as predicted.

VI. Conclusions

In this paper, we have demonstrated the existence of fault tolerant LTV controllers, which can be designed to be of low order. The LTV controller is a periodic compensator, which cycles between a number of LTI controllers. Each LTI controller is designed for one situation - either the nominal situation or one of the faulty situations. To obtain a good overall performance, however, these controllers should be designed to give a reasonable cross-performance.

A time-scheduling algorithm was proposed, that usually is able to improve the performance significantly, as compared to a parsimonious LTV controller, which allocates equal time slots to each LTI controller. It should be emphasized, though, that the proposed time-scheduling optimization is entirely heuristic, and will not lead to optimality in all circumstances.

To improve the time-scheduling, a cumbersome approach is to go through the process of comparing all possible combinations by applying a lifting technique to compute the norms of the corresponding periodic systems. An alternative, kindly suggested by one of the anonymous reviewers of this paper, would be to use genetic algorithms (GA) to improve the performance, where the GA could be set up to find the number of samples. The matrix (7) could then be used as a performance index.

References

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

[2] M. Blanke, M. Staroswiecki, and E. Wu, "Concepts and methods in fault-tolerant control," in *Proceedings of American Control Conference, ACC-2001*, Washington DC, USA, 2001, pp. 2606–2620.

[3] H. Niemann and J. Stoustrup, "An architecture for fault tolerant controllers," January 2003, submitted for journal publication.

[4] ——, "Passitive fault tolerant control of an inverted double pendulum - A case study example," in *Proceedings of IFAC SAFEPROCESS 2003*, Washington, DC, USA, 2003, pp. 1029–1034.

[5] ——, "Reliable control using the primary and dual Youla parameterization," in *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, NV, USA, 2002, pp. 4353–4358.

[6] R. Patton, "Fault tolerant control: The 1997 situation," in *Proceedings of the IFAC Symposium SAFEPROCESS'97*, Hull, England, 1997, pp. 1033–1055.

[7] J. Stoustrup and H. Niemann, "Fault tolerant feedback control using the youla parameterization," in *Proceedings of the 6th European Control Conference*, Porto, Portugal, Sept. 2001.

[8] N. Wu, Y. Zhang, and K. Zhou, "Detection, estimation and accommodation of loss of control effectiveness," *International Journal of Adaptive Control and Signal Processing*, vol. 14, pp. 775–795, 2000.

[9] N. Wu, K. Zhou, and G. Salomon, "Control reconfigurability of linear time-invariant systems," *Automatica*, vol. 36, pp. 1767–1771, 2000.

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

[11] A. Alos, "Stabilization of a class of plants with possible loss of outputs or actuator failures," *IEEE Transactions on Automatic Control*, vol. 28, no. 2, pp. 231–233, Feb. 1983.

[12] M. Vidyasagar, *Control System Synthesis: A Factorization Approach*. Cambridge, Massachusetts: The MIT Press, 1987.

[13] J. Stoustrup and V. Blondel, "Fault tolerant control: A simultaneous stabilization result," *IEEE Transactions on Automatic Control*, vol. 49, no. 2, pp. 305–310, Feb. 2004.

[14] D. Miller and M. Rossi, "Simultaneous stabilization with near optimal LQR performance," *IEEE Transactions on Automatic Control*, vol. AC-46, pp. 1543 – 1555, 2001.