RECONFIGURABLE FAULT-TOLERANT CONTROL BY LINEAR QUADRATIC VIRTUAL ACTUATOR UNDER CONTROL SIGNAL CONSTRAINT

Seyed Mohammad Hashemi¹, Mohammad Bagher Menhaj² and Ali Moradi Amani³

¹Department of Electrical Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran
²Department of Electrical Engineering, Amirkabir University of Technology, Tehran, Iran
³Department of Electrical Engineering, Islamic Azad University, Kangan Branch, Kangan, Iran
E-Mail: hashemi.ecep@gmail.com

ABSTRACT

When a fault occurs in a system, after the fault detection and isolation (FDI), the system applies fault-tolerant control, reconfiguration and control restructuring in the best way possible to achieve stability. The Fault-Tolerant Control (FTC) fixes the system's control problems by creating the capabilities of self-repair and fault resilience. In fault-tolerant control literature, the major interests are centered in safety-critical systems. In this paper, the fault considered, is actuator failure which is one of the most severe conditions of failure for a system. The constraint in applying the actuator control signal as a physical constraint has overshadowed the problem. The purpose is to design a reconfiguration block for systems to recover control purposes against negative impacts resulting from the failure, which makes doing this possible without requiring to identification the parameters of faulty system and create changes in the nominal controller. The approach is to distribute the role of the faulty actuator among other working system actuators by adding a virtual actuator between the faulty system and the nominal control as the reconfiguration block, so that it hides the fault from the nominal controller and establishes stability in applying the actuator control signal, in the presence of the constraint.

Keywords: fault-tolerance, reconfiguration, virtual actuator, actuator constraint.

INTRODUCTION

Nowadays, fault adverse effects in system are still discussed in control engineering field. The analysis of fault fault adverse effects are especially important as it's always involved in control systems. When a fault occurs in a system, the most critically vulnerable issue is the stability. Increasing Artificial Intelligence (AI) in safety-critical control system and creating self-repair capability in the event of fault occurrence in the control system also increasing system reliability and protecting the survivability of the safety-critical systems in the event of a fault, are of the highly important purposes in this field. Therefore, it’s necessary to provide an operational approach to prevent purposes above [1, 6, 7].

Fault tolerance is the property that enables a system to continue operating properly in the event of the failure of (or one or more faults within) some of its components. If its operating quality decreases at all, the decrease is proportional to the severity of the failure, as compared to a naively designed system in which even a small failure can cause total breakdown. Fault tolerance is particularly sought after in high-availability or life-critical systems.

A fault-tolerant design enables a system to continue its intended operation, possibly at a reduced level, rather than failing completely, when some part of the system fails. The term is most commonly used to describe computer systems designed to continue more or less fully operational with, perhaps, a reduction in throughput or an increase in response time in the event of some partial failure. That is, the system as a whole is not stopped due to problems either in the hardware or the software [8]. Control reconfiguration is an active approach in control theory to achieve fault-tolerant control for dynamic systems. It is used when severe faults, such as actuator or sensor failure, cause a break-up of the control-loop which must be restructured to prevent failure at the system level. In addition to loop restructuring, the controller parameters must be adjusted to accommodate changed plant dynamics. Control reconfiguration is a building block toward increasing the dependability of systems under feedback control. Fault hiding approaches paradigm aims at keeping the nominal controller in the loop. To this end, a reconfiguration block can be placed between the faulty plant and the nominal controller. Together with the faulty plant, it forms the reconfigured plant. The reconfiguration block has to perform the requirement that the behaviour of the reconfigured plant matches the behaviour of the nominal that is fault-free plant [9] (Figure-1).

RECONFIGURATION BY VIRTUAL ACTUATOR

The purpose is to achieve a reconfiguration block which first, hides the fault from the nominal controller to prevent any changes in the controller that may be caused by the reconfiguration then the closed-loop stability of the faulty system is established.
The condition for the solution of the problem is the stability of the unobserved poles of the faulty system. The idea for configuring by the virtual actuator is the behavioral similarity between the faulty and nominal plant. Therefore, the nominal plant states are considered as the reference states, and the control rule will be written as below [10, 11]:

\[ u_f(k) = M(x(k) - x_f(k)) \]  

(7)

\[ M \] is the stabilizing matrix of the \((A, B_f)\) pair. Therefore, the state space is written as below:

\[ x_f(k + 1) = A x_f(k) + B_f u_f(k) \]  

(8)

\[ x(k + 1) = A x(k) + B u_c(k) \]  

(9)

\[ y_c(k) = C x(k) \]  

(10)

\[ u_f(k) = M \left( x(k) - x_f(k) \right) \]  

(11)

Since the controller rule is dependent on \(x\) and \(x_f\), also since \(x_f\) cannot be measured, then to prevent the requirement to detect the faulty system, the following equation will be obtained by a single state transition:

\[ x_f = x - x_f \]  

(12)

And the state equations governing the reconfigured system are written as below [10][11]:

\[ x_f(k + 1) = A x_f(k) + B_f u_f(k) \]  

(13)

\[ x_{\Delta}(k + 1) = (A - B K C) x(\Delta) + B u_c(k) - B_f u_f(k) \]  

(14)

\[ y_c(k) = C \left( x_f(k) + x_{\Delta}(k) \right) \]  

(15)

\[ u_f(k) = M \left( x_{\Delta}(k) \right) \]  

(16)

\[ x_f(0) = x, x_{\Delta}(0) = 0 \]  

(17)

**VIRTUAL ACTUATOR UNDER CONSTRAINT**

The \(M\) block in the actuator which is the stabilizer of the \((A, B_f)\) pair, is the actuator control signal generator and should be designed in a way that it addresses the constraint in the actuator control signal [12, 16].

**Constrained Linear Quadratic (CLQ) control with finite receding horizon**

In this section, topics related to feasibility, stability and efficiency for finite horizon control formulation with receding horizon for linear systems that are a combination of constraint on control signal and system states will be considered. We will show that for an appropriate horizon length, receding horizon policy will be
feasible and the results will remain stable even if several constraints are applied to a system [15, 20].

Receding Horizon Control (RHC)

Receding Horizon Control (RHC) also known as Model Predictive Control (MPC), is a discrete time method in which, control is frequently obtained any time by the online solving system as an optimization problem which can combine constraints on system input that is one of the most important issues in control. Normally, constraints appear in two ways. Hard constraints (control input saturation) which is one of the most common constraints for physical system and situations which are created due to changes in domain (derivative) and control level (domain) and soft constraints. Scientifically, the appeal of RHC is directly related to optimization for its ability to handle multi-input multi-output system constraint. When constraint is considered in a problem, not only does it make system stability analysis more complex [12, 13].

\[
J_m(x_0) = \inf_{u(0)} \left\{ x^T(N)p_0x(N) + \sum_{K=0}^{N-1} x^T(K)Qx(K) + u^T(K)Ru(K) \right\}, \text{subject to: } Ex(k) + Fu(k) \leq \varphi 
\]

\[
J_{(p,m)}(x_0) = \inf_{x(p)} \left\{ x^T(p)p_0x(p) + \sum_{i=0}^{p-1} x^T(i)Qx(i) + \sum_{i=0}^{m-1} u^T(i)Ru(i) \right\}, \text{subject to: } Ex(k) + Fu(k) \leq \varphi 
\]

Constrained linear quadratic optimal control

Consider a discrete-time linear system with a combination of state and control constraints [2] [15].

\[
x(k + 1) = Ax(k) + Bu(k), \quad x(0) = x_0 \tag{18}
\]

\[
Ex(k) + Fu(k) \leq \varphi \tag{19}
\]

Which, the vectors \(x(k) \in \mathbb{R}^n\) and \(u(k) \in \mathbb{R}^m\) introduce system state and input respectively. \(\varphi \in \mathbb{R}^p\) is the system constraint vector. \(E \in \mathbb{R}^{p \times n}\) and \(F \in \mathbb{R}^{p \times m}\) are real matrices of full ranks. A popular design environment for non-variable systems is with the use of linear time of Linear Quadratic (LQ) optimization method. The formulation of the LQ problem related to finite horizon is like the cost function below, which should be minimized (20) and Receding horizon implementation is usually formulated by introducing optimization problem (21) subject to constraints [17, 18].

\[
\bar{u}_n(x(k)) = \arg \min \{ x^T(k)Qx(k) + u^T R u + J_{N-1}(Ax(k) + Bu(k)) \}, \text{subject to: } Ex(k) + Fu(k) \leq \varphi \tag{22}
\]

From now on, the problem assumptions will be \(p = m = N\) [19, 20]. The below assumptions are considered for the finite optimal control problem.

\(Q > 0, R > 0\) and these two suggest the visibility of the pair \([Q^{1/2}, A]\).

The \([A, B]\) pair is controllable

\(P_0 = Q\) which means that \(J_N\) is monotonically non-decreasing.

A neighboring to the center that is useless for optimal control and is possible without reaction exists.

Feasibility and constraints

The feasibility of finite horizon problem is a serious problem in the implementation of RH policy. Finite receding horizon may take a system state to an area of state space in which the optimal control problem becomes infinite and insolvable, the latter problem may occur for two main reasons, first it's possible that the control of feasibility and state transmission path cannot overcome system constraints in the finite horizon, and the second state is that it's possible that the control action is feasible at all times, but cannot meet the system stability and this makes the control cost become infinite. Then it's possible to consider an appropriate area in alignment with the RH policy, which provides the feasibility problem for the N horizon [17, 18].

The feasibility area \(I_0\) can be introduced with the recursive pattern below. The set of points that can reach a destination with a move (within one stage) while satisfying some classified constraints. Then we consider a set of points that reach the previous points (mentioned above) within one stage. And we will continue this process to infinite [18].

\[\text{let } I_0 = \{0\}\]
Take to be \( \{x; 3u, Ex + Fu \leq \varphi, Ax + Bu \in I_k\} \)

Definition: \( I_k = \bigcup_{k=0}^{\infty} I_k \)

If the \([A, B]\) be controllable, then \( x \in I_m \leftrightarrow f(x) < \infty \).

Assume that \( x \in I_m \), then there is a \( k \) that \( x \in I_k \), therefore by contradiction in the \( I_k \) sets, this means that there is a control sequence that has \( k \) control actions \( \hat{u}(0), \ldots, \hat{u}(k) \), and this sequence, takes the system state to zero in \( x(k + 1) = 0 \) therefore by minimizing a cost function (23).

\[
J(u) = \sum_{k=0}^{N-1} [(x(k)'Qx(k) + u(k)'Qu(k)) + x(N)'Q_f x(N)] , \text{subject to } u(k) \leq \varphi
\]  \( \tag{23} \)

\[
u(k) = (-R + B'P(k + 1)B)^{-1}B'P(k + 1)A)x(k)
\]  \( \tag{24} \)

\[
P(k) = Q + A'P(k + 1)A - A'P(k + 1)B(R + B'P(k + 1)B)^{-1}B'P(k + 1)A
\]  \( \tag{25} \)

**Analysis of the reconfigured closed-loop system**

A state transformation is used to separate the observable part of the state space from the unobservable part. The new state is \( \tilde{x} = x_f + x_A \).

The reconfigured plant can be constructed from the faulty plant and the virtual actuator as follow (26).

\[
\begin{bmatrix}
\tilde{x} \\
x_A
\end{bmatrix}(k + 1) =
\begin{bmatrix}
A & 0 \\
0 & A - B_f M
\end{bmatrix}
\begin{bmatrix}
x \\
x_A
\end{bmatrix}(k) +
\begin{bmatrix}
B \\
B_f M
\end{bmatrix} u_c(k)
\]  \( \tag{26} \)

\[
\begin{bmatrix}
x_f \\
x_A
\end{bmatrix}(k + 1) =
\begin{bmatrix}
A & B_f M \\
-B_D C & A - B_D C - B_f M & B_C
\end{bmatrix}
\begin{bmatrix}
x_f \\
x_A \\
\tilde{x}
\end{bmatrix}(k)
\]  \( \tag{27} \)

\[
\begin{bmatrix}
x_f \\
x_C
\end{bmatrix}(k + 1) =
\begin{bmatrix}
A - B_f M & 0 & 0 \\
0 & A - B_D C & B_C \\
0 & -B_C & A_C
\end{bmatrix}
\begin{bmatrix}
x_f \\
x_C \\
\tilde{x}
\end{bmatrix}(k)
\]  \( \tag{28} \)

It is clear from the structure of the system matrix that it has two separate sets of poles. The poles of the nominal loop are \( \sigma \left( A - B_D C A_C \right) \) and the poles of the virtual actuator are \( \sigma \left( A - B_f M \right) \). This confirms that the stabilisation goal has been reached.

**NUMERICAL SIMULATION RESULTS**

The system contains two tanks connected by a valve and filled by a pump. The goal is to maintain a constant outflow of the system [9] (Figure-2).

The finite horizon linear quadratic optimal control approach with the consideration of the constraint on the control signal is \( u(k) \leq \varphi \) and the condition for stability establishment is:

\[
\text{con}(A, B_f) \text{ be full rank}
\]

After addressing all the above conditions, the control signal for applying to the actuator, with the consideration of the constraint is obtained from (24) which \( P(k) \) is obtained from solving the discrete-time Riccati equation (25) in finite horizon [13, 14, 15].

Combining the faulty plant, the virtual actuator and a dynamical controller leads to the model of the reconfigured loop as follow (27). By applying the same transformation (21), the relevant subspaces can be separated as follow (28), [21, 22, 23].

**Figure-2. Tow Tanks model.**

Consider numberal example with state space matrices the system is as below [17] (see appendix-A):

\[
A = \begin{bmatrix}
4/3 & -2/3 \\
1 & 0
\end{bmatrix}, B = \begin{bmatrix}
1 & -0.5 \\
0 & 0.5
\end{bmatrix}, C = \begin{bmatrix}
1 & 0 \\
0 & 1
\end{bmatrix}, D = 0,
\]

\[
Q = \begin{bmatrix}
1 & 0 \\
0 & 1
\end{bmatrix}, R = \begin{bmatrix}
10 & 0 \\
0 & 10
\end{bmatrix}, x(0) = [1 \ 2].
\]
Numerical simulation results are shown in (Figure-3) and (Figure-4).

CONCLUSIONS

The constraint in the actuator control signal is practically applied to the design of the reconfiguration block with the use of the virtual actuator which controls the most severe states of fault. The role of the faulty actuator among the other ones is divided such that the system stability is established without requiring to identify the faulty system parameters and to cause any changes in the system nominal controller and also without requiring add any physical additives in the finite horizon. Also as is observed in the simulations the increase in the intensity of the constraint, extends the time the system will reach zero and stability states and control signal (Figure-3) and (Figure-4).

**Figure-3.** State variables $x_1$, $x_2$ – constrained and unconstrained – fault on $u_1$ at $N=50$. 
REFERENCES


relevant measured, the level out of it. The control right \( q \) continuously controlled valve pump \( y \) and close to a set-point \( x_2 \). Since \( q \) itself cannot be measured, the level \( x_2 \) of the right tank is used as the relevant value, and the level necessary to reach the desired outflow \( q_0 \) is used as a reference \( w_2 \). The left tank has an outlet via a valve \( d \), which can be used to simulate a leak in this tank. The law defines the flow through a valve as proportional to the square root of the pressure [23].

\[
q_{12} = k u_2 \sqrt{x_1 - h} \\
q_0 = k_0 \sqrt{x_2}
\]  

This equation applies to both valves and to the outflow. The flow through the pump is assumed to be proportional to its control input. The law of massconservation leads to the following non-linear model of the plant.

\[
x^a(k + 1) = q_{\text{max}} u_1^a(k) - k u_2^a(k) \sqrt{x_1^a(k) - h} - k \sqrt{x_2^a(k)} \\
x^a_2(k + 1) = k u_2^a(k) \sqrt{x_1^a(k) - h} - k \sqrt{x_2^a(k)}
\]  

Note that the index \( a \) is introduced here to denote an absolute value.

After linearization will have state space as:

\[
x(k + 1) = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} x(k) + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} u(k)
\]

\[
y(k) = \begin{bmatrix} c_{11} & c_{12} \end{bmatrix} x(k)
\]  

Appendix-A

The system consists of two tanks with the levels \( x_1 \) and \( x_2 \). Both levels can be measured by \( y_1 = x_1 \) and \( y_2 = x_2 \). Water can be brought into the left tank using the pump \( u_1 \), and it can be let into the right tank using the continuously controlled valve \( u_2 \) at the height \( h \). The right tank has an uncontrolled outlet, so that water flows out of it. The control objective is to maintain the outflow \( q \) close to a set-point \( q_0 \). Since \( q \) itself cannot be measured, the level \( x_2 \) of the right tank is used as the relevant value, and the level necessary to reach the...