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

Seyed Mohammad Hashemi[1], Mohammad Bagher Menhaj[2] and Ali Moradi Amani[3]
[1]Department of Electrical Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran
[2]Department of Electrical Engineering, Amirkabir University of Technology, Tehran, Iran
[3]Department 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.

VOL. 11, NO. 3, FEBRUARY 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

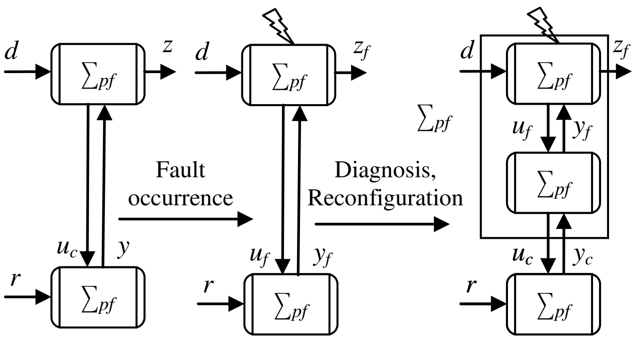

Figure-1. Fault hiding and reconfiguration steps.

Here, reconfiguration with the method of fault hiding from the controller has been used, so according to figure-1, the reconfiguring block $\sum_R$ is placed between the faulty system $\sum_{Pf}$ and the nominal controller $\sum_C$ to conceal the fault from the controller [10].

Virtual actuator structure

Consider a nominal dynamic system with state space equations as below.

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

$$y(k) = Cx(k) \tag{2}$$

In which, $x(k) \in R^n$, $u(k) \in R^r$ and $y(k) \in R^m$ are nominal system's state, input and output vectors respectively. Also $A \in R^{n\times n}$, $B \in R^{n\times r}$ and $C \in R^{m\times n}$ are system state matrices. A fault occurs in the system in step N, which destabilizes the system, therefore the equations governing the system are [2, 3]:

$$x_f(k+1) = Ax_f(k) + B_f u_f(k) \tag{3}$$

$$y_f(k) = Cx_f(k) \tag{4}$$

In these equations, $x_f(k) \in R^n$, $u_f(k) \in R^r$ and $y_f(k) \in R^m$ are the faulty system's state, input and output vectors, respectively, Also $A \in R^{n\times n}$, $B_f \in R^{n\times r}$ and $C \in R^{m\times n}$ are the faulty system's matrices. B_f is obtained by zeroing the column corresponding to the faulty actuator in B.

When nominal system stabilized with state feedback, the controller state equations will be written as [4, 5]:

$$u_c(k) = K\big(-y_c(k)\big) = K\big(-Cx(k)\big) \tag{5}$$

And the equations governing the closed-loop system will be written as:

$$x(k+1) = (A - BKC)x(k) \tag{6}$$

In these equations, K is the state-feedback gain.

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 ate considered as the reference states, and the control rule will be written as below [10, 11]:

$$u_f(k) = M(x(k) - x_f(k)) \tag{7}$$

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

$$x_f(k+1) = Ax_f(k) + B_f u_f(k) \tag{8}$$

$$x(k+1) = Ax(k) + Bu_c(k) \tag{9}$$

$$y_c(k) = Cx(k) \tag{10}$$

$$u_f(k) = M\left(x(k) - x_f(k)\right) \tag{11}$$

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:

$$x_f = x - x_f \tag{12}$$

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

$$x_f(k+1) = Ax_f(k) + B_f u_f(k) \tag{13}$$

$$x_\Delta(k+1) = Ax_\Delta(k) + Bu_c(k) - B_f u_f(k) \tag{14}$$

$$y_c(k) = C\left(x_f(k) + x_\Delta(k)\right) \tag{15}$$

$$u_f(k) = M\big(x_\Delta(k)\big) \tag{16}$$

$$x_f(0) = x \, , x_\Delta(0) = 0 \tag{17}$$

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

VOL. 11, NO. 3, FEBRUARY 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

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].

Constrained linear quadratic optimal control

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

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

$$Ex(k) + Fu(k) \le \varphi \tag{19}$$

Which, the vectors $x(k) \in R^n$ and $u(k) \in R^m$ introduce system state and input respectively. $\varphi \in R^p$ is the system constraint vector. $E \in R^{p\times n}$ and $F \in R^{p\times m}$ 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].

$$J_N(x_0) = \begin{matrix} inf \\ u(0) \end{matrix} \left[x^T(N)p_0x(N) + \sum_{K=0}^{N-1} \left(x^T(K)Qx(K) + u^T(K)Ru(K) \right) \right], subject\ to: \quad Ex(k) + Fu(k) \le \varphi \tag{20}$$

$$J_{(p,m)}(x_0) = inf \left[x^T(p)p_0x(p) \ + \sum_{i=0}^{p-1} x^T(i)Qx(i) + \sum_{i=0}^{m-1} u^T(i)Ru(i)) \right], subject\ to: \quad Ex(k) + Fu(k) \le \varphi \tag{21}$$

p≥m: In which, *p* is the predictive horizon and *m* is the control horizon. Assume that *i=0 ,..., m-1* and $u^*_{(p,m)}(i)$ aim for a control minimum for the cost function $J^*_{(p,m)}(x(k))$. The processing policy of implementation receding horizon, is only started with the first control action $\hat{u}_{(p,m)}(x(k)) = u^*_{(p,m)}(0)$ (we remove the rest of the control sequence) and with the obtaining of $x(k+1) = A\,x(k) + B\,u^*_{(p,m)}(0)$, and now $x(k+1)$ is used as the primary condition to update the (21) optimization problem. This process is repeated each time only the first control action is used to check the primary conditions then the cost function, takes a step forward.

Specifically, if $p = m = n$ then $J_{(p,m)} = J_N$ and will be defined according to (22). This finite horizon can be simplified as below based on receding horizon policy [17, 18, 19].

$$\hat{u}_N\big(x(k)\big) = \arg\ min \{ \, x^T(k)Qx(k) + u^TR\,u \ + J_{N-1}(Ax(k) + Bu(k)\},\ \ subject\ to: \ Ex(k) + Fu(k) \le \varphi \tag{22}$$

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 $\left[Q^{1/2}, A\right]$ pair.
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_∞) 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].

$$let\ I_0 = \{0\}$$

VOL. 11, NO. 3, FEBRUARY 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

$Take\ \ to\ be = \{x\colon \exists u, Ex + Fu \le \varphi, Ax + Bu \in I_k\}$
$Defin\colon\ \ I_\infty = \cup_{k=0}^{\infty} I_k$
If the $[A,B]$ be controllable, then $x \in I_\infty \leftrightarrow J(x) < \infty$.

Assume that $x \in I_\infty$, 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 $\hat{u}(0), \dots, \hat{u}(k)$, and this sequence, takes the system state to zero in $x(k+1) = 0$ therefore by minimizing a cost function (23).

The finite horizon linear quadratic optimal control approach with the consideration of the constraint on the control signal is $u(k) \le \varphi$ and the condition for stability establishment is:
$con(A, B_f)$ 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].

$$J(u) = \sum_{k=0}^{N-1} \big(x(k)'Qx(k) + u(k)'Qu(k)\big) +\ x(N)'Q_f x(N)\ , subject\ to\ u(k) \le \varphi \tag{23}$$

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

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

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 $\tilde{x} = x_f + x_\Delta$.

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

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].

$$\begin{bmatrix} \tilde{x} \\ x_\Delta \end{bmatrix}(k+1) = \begin{bmatrix} A & 0 \\ 0 & A - B_f M \end{bmatrix}\begin{bmatrix} x \\ x_\Delta \end{bmatrix}(k) + \begin{bmatrix} B \\ B \end{bmatrix} u_C(k) \tag{26}$$

$$\begin{bmatrix} x_f \\ x_\Delta \\ x_C \end{bmatrix}(k+1) = \begin{bmatrix} A & B_f M & 0 \\ -BD_C C & A - BD_C C - B_f M & BC_C \\ -BC_C & -B_C C & A_C \end{bmatrix}\begin{bmatrix} x_f \\ x_\Delta \\ x_C \end{bmatrix}(k) \tag{27}$$

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

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 $\sigma\begin{pmatrix} A - BD_C C & BC_C \\ -BC_C & A_C \end{pmatrix}$ and the poles of the virtual actuator are $\sigma(A - B_f M)$. 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).

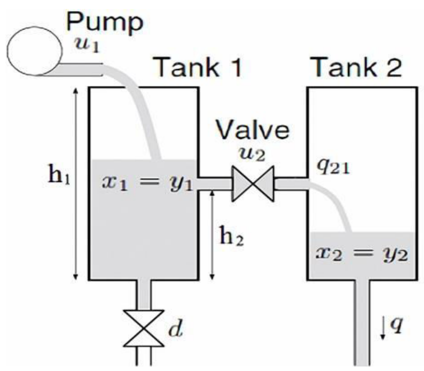

Figure-2. Tow Tanks model.

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

$A = \begin{bmatrix} 4/3 & -2/3 \\ 1 & 0 \end{bmatrix}, B = \begin{bmatrix} 1 & -0.5 \\ 0 & 0.5 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, D = 0$
$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, R = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}, x(0) = [1 \quad 2].$

VOL. 11, NO. 3, FEBRUARY 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

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).

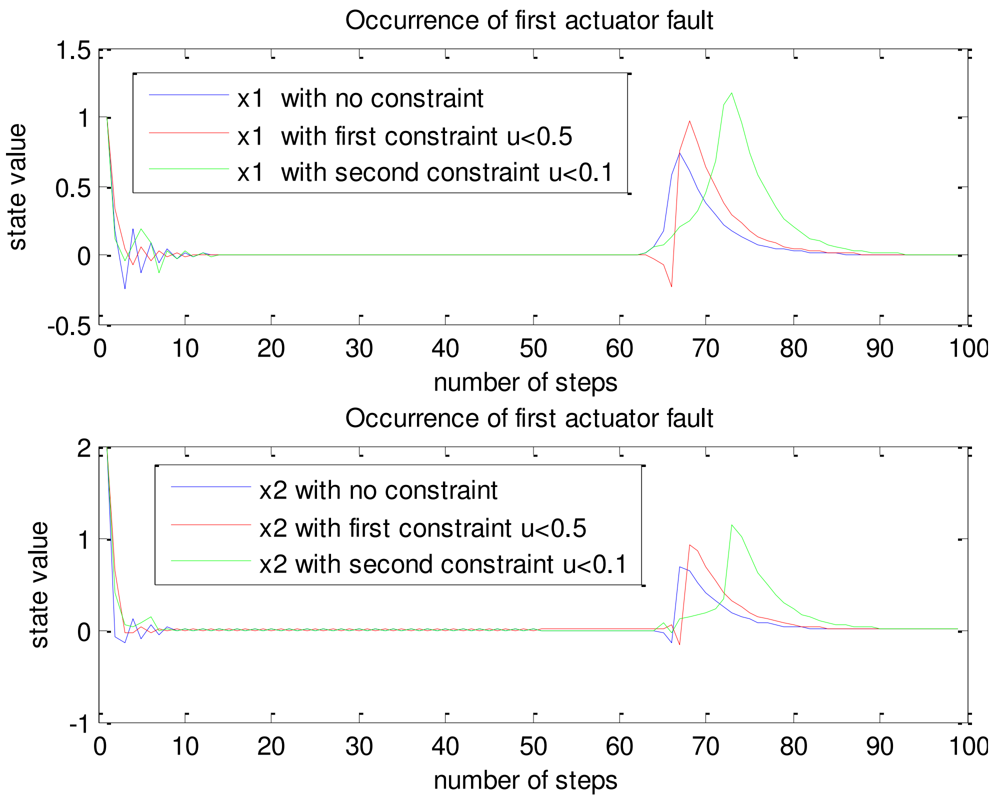

Figure-3. State variables x_1, x_2 – constrained and unconstrained – fault on u_1 at N=50.

ARPN Journal of Engineering and Applied Sciences

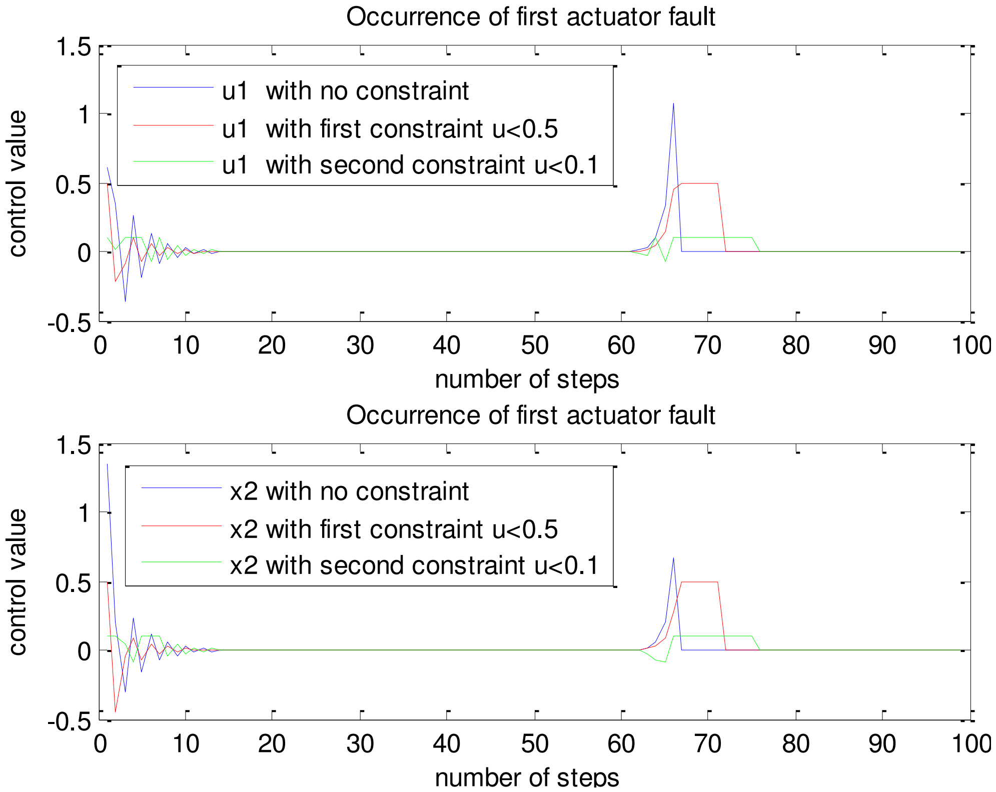

Figure-4. Control variables u_1, u_2 – constrained and unconstrained – fault on u_1 at N=50.

REFERENCES

[1] Zhang. Y, Jiang. J. 2008. Bibiographical review on reconfigurable fault-tolerant control system. Annual Reviews in Control. 32, 229-252.

[2] I.K. Peddle. 2007. Discrete State Space Control. Control Systems 414.

[3] Bernard Friedland. 2005. Control System Design an Introduction to State-Space Methods. Dover Publications.

[4] John C. Doyle, Bruce A. Francis, Allen R. Tannenbaum. 2009. Feedback Control Theory. Macmillan Publications.

[5] Srinivas Akella, Nancy M Amato, Wesley H Huang. 2008. Algorithmic Foundation of Robotics. Springer.

[6] Inseok Hwang, Sungwan Kim, Youdan Kim. 2010. A Survey of Fault Detection and Isolation and Reconfiguration Methods Control Systems Technology. IEEE Transactions on Springer. 18(3): 636-653.

[7] Nader Meskin, Khashayar Khorasani. 2011. Fault Detection and Isolation Multi-Vehicle Unmanned Systems. Springer.

[8] Magdi S. Mahmoud, Yuanqing Xia 2014. Analysis and Synthesis of Fault-Tolerant Control System. Wiley.

[9] Ducard Guillaume J. J. 2009. Fault-tolerant Flight Control and Guidance Systems. Springer.

[10] Looze, Douglas P., Weiss Jerold L., Eterno John S. 2003. An automatic redesign approach for restructurable control systems. Control Systems Magazine, IEEE. 5(2): 16-22.

[11] Moradi Amani. A, Afshar. A, Menhaj. M.B. 2012. Reconfiguration-based fault tolerant control systems: A control reconfiguration approach. IEICE Transactions on Information and Systems. E95.D(4): 1074-1083.

[12] Camacho EF., Bordons C. 2003. Model Predictive Control. Springer.

[13] Wenyu Sun , Ya-Xiang Yuan. 2010. Optimization Theory and Methods. Springer.

[14] Sauter. D, Hamelin. F, Noura. H. 1998. Fault tolerant control in dynamic systems using convex optimization. IEEE International Symposium Intelligent Control. pp. 187-192.

[15] Mare J. B., Ftese U. 2007. Solution Of The Input-Constrained LQR problem Using Dynamic Programming. Systems and Control Letters. pp. 342-348.

[16] Esna Ashari A., Khaki Sedigh A., Yazdanpanah M. J. 2005. Reconfigurable control system design using eigenstructure assignment: static, dynamic and robust approaches. International Journal of Control. 78(13): 1005-1016.

[17] J.S. Shamma, D.Xiong. 1997. Linear Non-Quadratic Optimal Control. IEEE Trans. Auto. Control. 42(6): 875-879.

[18] J.B.R awiings, K.R. Iviuske. 1993. The Stability of Constrained Receding Horizon Control. In IEEE Trans. Auto. Control. 38(10): 1512-1516.

[19] Rina Dechter. 2003. Constraint Processing. Morgan Kaufmann Publishers.

[20] Donald E. Kirk. 2004. Optimal Control Theory. Dover Publications.

[21] J. H. Richter, W. P. M. H. Heemels, N. van de Wouw and J. Lunze. 2008. Reconfigurable control of PWA systems with actuator and sensor faults stability. IEEE Conference on Decision and Control. pp. 1060-1065.

[22] Thomas Steffen. 2005. Control Reconfiguration of Dynamical Systems Linear Approaches and Structuraltests. Springer.

[23] YM Zhang J Jiang. 2002. Active fault-tolerant control system against partial actuator failures. IEEE proceedings-Control Theory and applications. 149(1): 95-104.

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 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].

$$q_{12} = ku_2\sqrt{x_1 - h} \tag{a.1}$$

$$q_0 = k_0\sqrt{x_2} \tag{a.2}$$

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.

$$x_1^a(k+1) = q_{max}u_1^a(k) - ku_2^a(k)\sqrt{x_1^a(k) - h} \quad -k\sqrt{x_1^a(k)d} \tag{a.3}$$

$$x_2^a(k+1) = ku_2^a(k)\sqrt{x_1^a(k) - h} - k\sqrt{x_2^a(k)} \tag{a.4}$$

$$z^a(k) = x_2^a(k) \tag{a.5}$$

Note that the index $\dot{a}$ is introduced here to denote an absolute value.
After linearization will have state space as:

$$x(k+1) = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} x(k) + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} u(k) \tag{a.6}$$

$$y(k) = [c_{11} \quad c_{12}]x(k) \tag{a.7}$$