UKACC International Conference on Control 2012
Cardiff, UK, 3-5 September 2012

Robust Stabilization of Networked Control Systems using the Markovian Jump System Approach

Ashraf Khalil
School of Electrical, Electronic and Computer Engineering
University of Birmingham
Birmingham, UK
ashrafkhalilg@gmail.com

Jihong Wang
School of Engineering
University of Warwick
Coventry, UK
jihong.wang@warwick.ac.uk

Abstract— **The key feature of Networked Control Systems (NCSs) is that the information is exchanged through a network among control system components. Transmitting control signals through shared networks induces time delays and data losses which may destabilize the system. This time delay may be constant periodic or random. The random time delay can be modeled using Markov Chains and the NCS can be modeled as Markovian jump system. The stochastic stability of the system has the form of Bilinear Matrix Inequality (BMI). The V-K iteration algorithm is used to solve the BMI and hence to design the stabilizing controller. A modified V-K iteration algorithm is presented in this paper where the decay rate is maximized in both the V- and K-loops. The V-K algorithm method is applied to the cart and inverted pendulum problem which shows that the decay rate is improved with the modified algorithm.**

Keywords-component; networked control system, time delay, Markov , random time delay, jump, stability

I. Introduction

The advances in communication and network technology, and the availability of high speed computers have resulted in an increasing interest in NCSs. This type of control systems can be defined as a control system where the control loop is closed through a real-time communication network [1]. In NCSs the reference input, plant output and control input are exchanged through a real-time communication network as shown in Figure 1. The main advantages of NCSs are modularity, simplified wiring, low cost, reduced weight, decentralization of control, integrated diagnosis, simple installation, quick and easy for maintenance [2], flexible expandability with low cost. NCSs are able to easily fuse global information to make intelligent decisions over large physical spaces.

As the control loop is closed through a communication network the time delay and data dropout are unavoidable. This may degrade the performance of NCSs or even destabilize the system. In general, the control systems with time delays can be classified into time delay independent where the stability is not affected by the time delay and time delay dependent where the time delay affects the stability [3]. Time delay, no doubt, increases the complexity in the analysis and the design of NCSs. There are many methods in the literature for studying the stability of NCS, see for example [4]-[5]. Among these methods is the Markovian jump system approach which is mostly used to study the stability and stabilization of system with abrupt changes due to the variations in the system structure or partly system failure. In this way the system will have a number of models or modes and jumps from one mode to another in a random fashion and in many cases the jump parameter can be modeled using Markov Chains. In NCS the time delay can be random and because there is a correlation between the previous, current and next time delay, the time delay can be modeled as a Markov Chain.

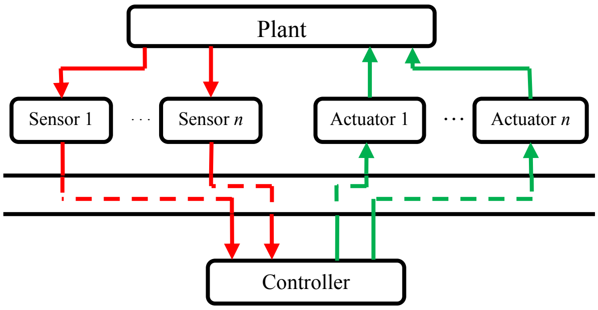

Figure 1. A Typical networked control system

The application of the discrete-time jump system in NCSs has been addressed in many papers, see for example [6-9]. In [6-8] the discrete-time model is augmented and the generated output feedback problem is formulated as BMI which is solved using the V-K iteration algorithm. In this paper we adopt the algorithm in [8] with some modification to the V-K iteration loop. The method in [8] is limited to time delays which are less than the sampling period and in [6][7] the method is extended to time delays larger than the sampling period. From control engineering point of view when the time delay larger than the sampling time the system performance is not acceptable. In [10] the authors use the discrete model for the plant and both the time delay between the sensor to controller and from the controller to the actuator are considered. The discrete mode dependent Lyapunov function has been used to derive a stabilizing switching controller. In [9] the authors concentrate on the problem of the random data drop outs and the sufficient conditions for the mean square stability are derived. The stability analysis and controller

978-1-4673-1558-6/12/$31.00 ©2012 IEEE

design with two random time delays are studied in [10-13]. In [12][13] the NCS is modeled where both the time delays are considered. The controller depends on the current sensor to controller time delay and the previous controller to actuator time delay and hence the controller depends on the three random variables, τ_k , d_k , $d_{k-\tau_{k+1}}$, which are interdependent. The resulting system cannot be regarded as the standard Discrete-Time Markovian Jump Linear System (DTMJLS). The derived theorem is in a set of LMI with nonlinear LMI constraint which are non-convex and can be solved by iterative algorithm such as the Cone Complementary Linearization (CCL). The optimal stochastic control is studied in [14] where the optimal stochastic controller is derived when the time delay is random. The use of the model predictive control in NCS has been studied in [15-17] where both the sensor to controller and controller to actuator time delays are considered.

The paper starts from the model of the NCS as DTMJLS is presented where the time delay is modeled as Markov Chain. Then the stability of the system is formulated as BMIs. The V-K algorithm is explained to solve the BMIs and it is tested on the cart and inverted pendulum problem.

II. Mathematical Modelling of NCS with Time Delay

A. NCS Systems with State Feedback Controller

The model of a single loop networked control system is shown in Figure 2. The measured plant signals are transmitted through the network and they will suffer random time delays and some of them may be lost. The random time delay makes the system to have the nature of a stochastic hybrid system. The discrete time-invariant plant model is given by:

$$\mathbf{x}(k+1) = \mathbf{A}_d\mathbf{x}(k) + \mathbf{B}_d\mathbf{u}(k) \qquad (1)$$

where $\mathbf{x}(k) \in \Re^n$ the system state vector, $\mathbf{u}(k) \in \Re^m$ the system control input, and the matrices $\mathbf{A}_d$ and $\mathbf{B}_d$ are given by:

$$\mathbf{A}_d = e^{\mathbf{A}h} \qquad \mathbf{B}_d = \int_0^h e^{\mathbf{A}(h-s)}\mathbf{B}ds \qquad (2)$$

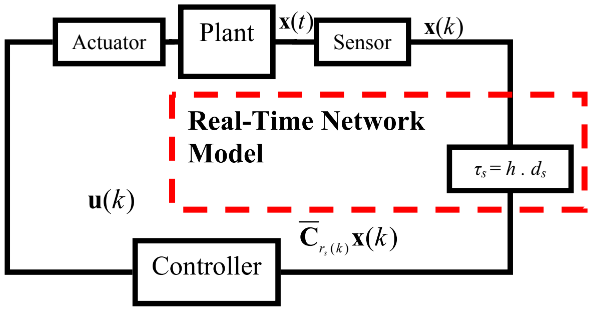

Figure 2. The networked control system.

In the model shown in Figure 2 the time delays are lumped together between the sensor and the controller. In many of the published work in the literature the time delay between the controller and the actuator is neglected. For the following analysis the following assumptions are required and are made:

Assumption 1:

- The sensors are clock driven. The actuator and the controller are event driven; which means that the sensors sample the plant states periodically and the actuators and the controllers use the data as soon as they arrive.
- The data are sent as a single packet.
- The data are received in chronological order which means that old data are disregarded.

The mode-dependent switching state feedback control law is given by:

$$\mathbf{u}(k) = \mathbf{K}(r_s(k))\mathbf{x}(k - r_s(k)) \qquad (3)$$

where $\tau(k) = r_s(k) \cdot h$, h is the sampling period and $r_s(k)$ is a bounded random integer sequence governed by Markov Chains with $0 \le r_s(k) \le d_s < \infty$, and d_s is the finite delay bound. By augmenting the state variable:

$$\overline{\mathbf{x}}(k) = \begin{bmatrix}\mathbf{x}(k)^T & \mathbf{x}(k-1)^T & \cdots & \mathbf{x}(k-d_s)^T\end{bmatrix}^T$$

where $\overline{\mathbf{x}}(k) \in R^{(d_s+1)n}$, applying the controller (3) into (1) the closed-loop system becomes:

$$\overline{\mathbf{x}}(k+1) = (\overline{\mathbf{A}} + \overline{\mathbf{B}}\mathbf{K}(r_s(k))\overline{\mathbf{C}}(r_s(k)))\overline{\mathbf{x}}(k) \qquad (4)$$

where;

$$\overline{\mathbf{A}} = \begin{bmatrix}\mathbf{A} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0}\\ \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{I} & \cdots & \mathbf{0} & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I} & \mathbf{0}\end{bmatrix} \quad \overline{\mathbf{B}} = \begin{bmatrix}\mathbf{B}\\ \mathbf{0}\\ \mathbf{0}\\ \vdots\\ \mathbf{0}\end{bmatrix}$$

$$\overline{\mathbf{C}}(r_s(k)) = \begin{bmatrix}\mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0}\end{bmatrix}$$

$\overline{\mathbf{C}}(r_s(k))$ incorporates the time delay into the model and has all elements being zero except for the $r_s(k)$th block being an identity matrix. The closed-loop system (4) can be rewritten as;

$$\overline{\mathbf{x}}(k+1) = \mathbf{A}_{cl}(r_s(k))\overline{\mathbf{x}}(k) \qquad (5)$$

B. NCS with Dynamic Output Feedback Controller

Stabilizing the plant (1) with a dynamic controller as shown in Figure 3, the dynamic controller model is given by:

$$\begin{aligned}\mathbf{z}(k+1) &= \mathbf{F}\mathbf{z}(k) + \mathbf{G}\mathbf{y}(k)\\ \mathbf{v}(k) &= \mathbf{H}\mathbf{z}(k) + \mathbf{J}\mathbf{y}(k) \qquad (6)\end{aligned}$$

In the case of the dynamic controller, both the time delay from the sensors to the controller and from the controller to the actuators are considered. Augmenting the controller states as;

$$\bar{\mathbf{z}}(k) = \left[\mathbf{z}(k)^T \quad \mathbf{v}(k)^T \quad \cdots \quad \mathbf{v}(k-d_{ca})^T\right]^T$$

The controller model with the augmenting states is then given by:

$$\begin{aligned}\bar{\mathbf{z}}(k+1) &= \bar{\mathbf{F}}\bar{\mathbf{z}}(k) + \bar{\mathbf{G}}\mathbf{y}(k)\\ \mathbf{u}(k) &= \bar{\mathbf{H}}(r_{ca}(k))\bar{\mathbf{z}}(k) + \bar{\mathbf{K}}(r_{ca}(k))\mathbf{y}(k)\end{aligned} \qquad (7)$$

where;

$$\bar{\mathbf{F}} = \begin{bmatrix} \mathbf{F} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{H} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I} & \mathbf{0} \end{bmatrix} \bar{\mathbf{G}} = \begin{bmatrix} \mathbf{G} \\ \mathbf{J} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}$$

$$\bar{\mathbf{H}}(r_{ca}(k)) = \begin{cases} [\mathbf{H} \quad \mathbf{0} \quad \cdots \quad \mathbf{0} \quad \mathbf{0} \quad \cdots \quad \mathbf{0}] & \text{if} \quad r_{ca}(k) = 0 \\ [\mathbf{0} \quad \cdots \quad \mathbf{0} \quad \mathbf{I} \quad \mathbf{0} \quad \cdots \quad \mathbf{0}] & \text{if} \quad r_{ca}(k) \neq 0 \end{cases}$$

$$\bar{\mathbf{K}}(r_{ca}(k)) = \begin{pmatrix} \mathbf{J} & \text{if} \quad r_{ca}(k) = 0 \\ \mathbf{0} & \text{if} \quad r_{ca}(k) \neq 0 \end{pmatrix}$$

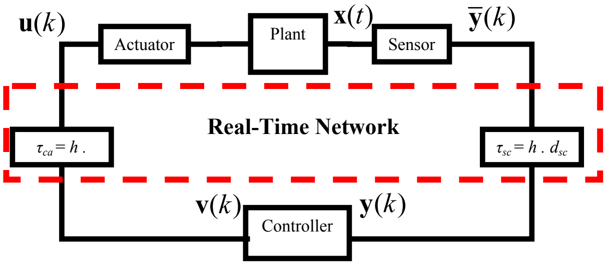

Figure 3. Networked control system with both time delays from sensor to controller and from controller to actuator are taking into account

When the time stamping is used, $\mathbf{F}$, $\mathbf{G}$, $\mathbf{H}$ and $\mathbf{J}$ are replaced by $\mathbf{F}(\tau_{sc})$, $\mathbf{G}(\tau_{sc})$, $\mathbf{H}(\tau_{sc})$, and $\mathbf{J}(\tau_{sc})$. The augmented plant model with output feedback can be described by:

$$\begin{aligned}\tilde{\mathbf{x}}(k+1) &= \tilde{\mathbf{A}}\tilde{\mathbf{x}}(k) + \tilde{\mathbf{B}}\mathbf{u}(k)\\ \mathbf{y}(k) &= \mathbf{C}\tilde{\mathbf{C}}(r_{sc}(k))\tilde{\mathbf{x}}(k)\end{aligned} \qquad (8)$$

Augmenting both the plant states and controller states as: $\bar{\mathbf{x}}(k) = \left[\tilde{\mathbf{x}}(k)^T \quad \tilde{\mathbf{z}}(k)^T\right]^T$. The closed loop system with the plant (8) and the controller (7) becomes;

$$\bar{\mathbf{x}}(k+1) = (\bar{\mathbf{A}} + \bar{\mathbf{B}}\mathbf{K}(r_{ca}(k))\bar{\mathbf{C}}(r_{sc}(k)))\bar{\mathbf{x}}(k) \qquad (9)$$

where;

$$\bar{\mathbf{A}} = \begin{bmatrix} \tilde{\mathbf{A}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \bar{\mathbf{B}} = \begin{bmatrix} \tilde{\mathbf{B}} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \bar{\mathbf{C}}(r_{sc}(k)) = \begin{bmatrix} \mathbf{0} & I \\ \mathbf{C}\tilde{\mathbf{C}}(r_{sc}(k)) & \mathbf{0} \end{bmatrix}$$

$$\mathbf{K}(r_{ca}(k)) = \begin{bmatrix} \tilde{\mathbf{F}} & \tilde{\mathbf{G}} \\ \tilde{\mathbf{H}}(r_{ca}(k)) & \tilde{\mathbf{K}}(r_{ca}(k)) \end{bmatrix}$$

Equation (9) can be written as;

$$\bar{\mathbf{x}}(k+1) = \mathbf{A}_{cl}(r_{sc}(k), r_{ca}(k))\bar{\mathbf{x}}(k) = \mathbf{A}_{cl}(r_s(k))\bar{\mathbf{x}}(k) \qquad (10)$$

The two time delays are random and bounded, $\tau_{\text{scm}} \geq \tau_{\text{sc}} \geq 0$ and $\tau_{\text{cam}} \geq \tau_{\text{ca}} \geq 0$. These can be modelled as two homogeneous Markov Chains and they jump from mode to mode according to their transition probabilities P_{sc} and P_{ca} respectively. The random variable τ_{sc} and τ_{ca} can be converted to single random variable, $r(k)$ where the transition probability, P, is given by Kronecker product of the P_{sc} and P_{ca} as;

$$P = P_{sc} \otimes P_{ca} \qquad (11)$$

For simplicity (10) can be written as:

$$\mathbf{x}(k+1) = \mathbf{A}(r(k))\mathbf{x}(k) \qquad (12)$$

Equations (5) and (10) are standard DTMJLS. Equation (5) is a jump system with one mode which is the sensor to the controller time delay while the system in (10) has two modes which are the sensor to the controller and the controller to the actuator time delays. The system matrix will be $\mathbf{A}_{cl}(r(k))$□$\{\mathbf{A}_{cl}(0),\ldots,\mathbf{A}_{cl}(d)\}$ according to the jump parameter $r(k)$□$\theta = \{0,\ldots,d\}$. In order to stabilize the system with mode-independent or mode-dependent controller the mean square stability must be established.

C. The model of the random time delay as Markov Chain

The random time delay is modelled as a finite state Markov process with the following properties:

$$\begin{aligned}&P\{r_s(k+1) = j \quad | \quad r_s(k) = i\} = p_{ij} && 0 \leq i, j \leq d_s \\ &0 \leq p_{ij} \leq 1 && \sum_{j=0}^{d} p_{ij} = 1\end{aligned} \qquad (13)$$

where d_s is the number of modes and $r_s(k)$ is the Markovian process. The general transition probability matrix is given by:

$$P = \begin{bmatrix} p_{00} & p_{01} & 0 & 0 & \cdots & 0 \\ p_{10} & p_{11} & p_{12} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & 0 \\ p_{d0} & p_{d1} & p_{d2} & p_{d3} & \cdots & p_{d_s d_s} \end{bmatrix} \qquad (14)$$

The constraint (13) means the summation of the probabilities in every row is one. The assumption made is that the old data are discarded. Suppose that at instant k we received $\mathbf{x}(k)$, at k+1 if there is no new data then the old data will be used by the controller, but if we receive $\mathbf{x}(k\text{ -}1)$ at $k+1$ then it will be older than **x**(*k*) and hence **x**(*k* -1) must be discarded, this can be interpreted as;

$$p\{r_s(k+1) > r_s(k)+1\} = 0 \qquad (15)$$

From (15) the time delay can increase only at one step but it can decrease as many steps as can be seen from (14). The diagonal elements in (14) represent the probability of successive equal time delays or in other words the probability that the network remain in the same state. The upper diagonal elements represent the possibility of receiving longer delays or increasing the network load. The zero elements represent the discard of the old data.

III. The Stability of the Discrete-Time Markovian Jump Linear System (DTMJLS)

The Mean Square stability of the Markovian Jump Systems is equivalent to the Asymptotic Wide Sense Stationary Stability (AWSS) [18]. For the jump system the stochastic stability, mean square stability and the exponential mean square stability are all equivalent and every condition implies the almost sure (asymptotic) stability.

Definition 1: [6]

The system (12) is mean square stable if for every initial condition state, $(\mathbf{x}_0, r_0)$,

$$\lim_{k\to\infty} \mathrm{E}\left[\|\mathbf{x}(k)\|^2\right] = 0 \qquad (16)$$

Definition 2: [6]

The system (12) is mean square stable with decay rate β [19] if for every initial condition state, $(\mathbf{x}_0, r_0)$,

$$\lim_{k\to\infty} \beta^k \mathrm{E}\left[\|\mathbf{x}(k)\|^2\right] = 0 \qquad \beta > 1 \qquad (17)$$

The necessary and sufficient conditions for mean square stability for jump system are given in the following theorem.

Theorem 1 [18]: The mean square stability of system (12) is equivalent to the existence of symmetric positive definite matrices **Q**$_0$, . . .,**Q**$_d$ satisfying any one of the following 4 conditions:

$$\mathbf{A}_i\left(\sum_{j=0}^{d} p_{ji}\mathbf{Q}_\mathbf{j}\right)\mathbf{A}_i^T < \mathbf{Q}_i, \qquad i = 0,\cdots,d$$

$$\mathbf{A}_j^T\left(\sum_{i=0}^{d} p_{ji}\mathbf{Q}_i\right)\mathbf{A}_j < \mathbf{Q}_j, \qquad j = 0,\cdots,d$$

$$\sum_{j=0}^{d} p_{ji}\mathbf{A}_j\mathbf{Q}_j\mathbf{A}_i^T < \mathbf{Q}_i, \qquad i = 0,\cdots,d$$

$$\sum_{j=0}^{d} p_{ji}\mathbf{A}_i^T\mathbf{Q}_j\mathbf{A}_i < \mathbf{Q}_j, \qquad i = 0,\cdots,d$$

Where *i*=0,…,*d* represents the number of the modes. The conditions 1-4 are equivalent for studying the stability of the DTMJLS but for the controller design each condition will lead to a different controller. Choosing condition (4) in Theorem 1 and replacing **Q**$_i$ by α**Q**$_i$ (where the decay rate or Lyapunov Exponent, $\beta = 1/\alpha$ and $\lim_{k\to\infty}\beta^k M(k) = 0$) on the right hand side, the closed-loop system becomes:

$$\sum_{j=0}^{d} p_{ji}(\mathbf{A}_i + \mathbf{B}_i\mathbf{K}_i\mathbf{C}_i)^T\mathbf{Q}_j(\mathbf{A}_i + \mathbf{B}_i\mathbf{K}_i\mathbf{C}_i) < \alpha\mathbf{Q}_j,\ i=0,\ldots,d \quad (18)$$

The coupled equations (18) are BMIS which are non convex and finding a global optimal solution is very difficult. However many control problems are formulated as BMIs, there are a few methods for solving the BMIs. For example the path-following linearization method reported in [20] can be used where each matrix is perturbed and the higher order terms are neglected. The most widely used techniques for the solution is by iteration methods such the D-K, G-K and V-K iteration algorithms [21]. If we fix **K**$_i$ (*i*=0,…,d) then we have a Generalized Eigenvalue Problem (GEVP) and if we fix **Q**$_i$ (i=0,…,d) then we have Eigenvalue Problem (EVP) [8]. Both of these problems can be solved very efficiently using the Matlab LMI toolbox. Using Schur complement to (18) then we have:

$$\begin{bmatrix} \alpha\mathbf{Q}_j & (\mathbf{A}_0 + \mathbf{B}_0\mathbf{K}_0\mathbf{C}_0)^T & \cdots & (\mathbf{A}_d + \mathbf{B}_d\mathbf{K}_d\mathbf{C}_d)^T \\ (\mathbf{A}_0 + \mathbf{B}_0\mathbf{K}_0\mathbf{C}_0) & p_{j0}^{-1}\mathbf{Q}_0^{-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ (\mathbf{A}_d + \mathbf{B}_d\mathbf{K}_d\mathbf{C}_d) & 0 & \cdots & p_{jd}^{-1}\mathbf{Q}_d^{-1} \end{bmatrix} > 0 \qquad (19)$$

IV. The V-K Iteration Algorithm

In the V-K algorithm the BMI is divided into two LMI's and by solving these two LMI's a local optimal solution can be found. The problem solution process is divided to three basic problems which are: FP (Feasibility Problem), EVP, and GEVP that can be solved using the Matlab LMI toolbox. In the V-K algorithm, the problem is iterated between the EVP and the GEVP. The proof of the algorithm convergence is given in [21]. The detailed algorithm is shown in the flowchart in Figure 4. The algorithm starts with the initialization, then if the solution is feasible the EVP and GEVP are iterated until the desired transition matrix is reached. In this improved algorithm the decay rate is maximized in both the EVP and GEVP. The initial transition probability matrix is:

$$P_0 = \begin{bmatrix} 1 & 0 & \ldots & 0 \\ 1 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \ldots & 0 \end{bmatrix} \approx \begin{bmatrix} 1-n\cdot\in & \in_1 & \ldots & \in_n \\ 1-n\cdot\in & \in_1 & \ldots & \in_n \\ \vdots & \vdots & \ddots & \vdots \\ 1-n\cdot\in & \in_1 & \ldots & \in_n \end{bmatrix}$$

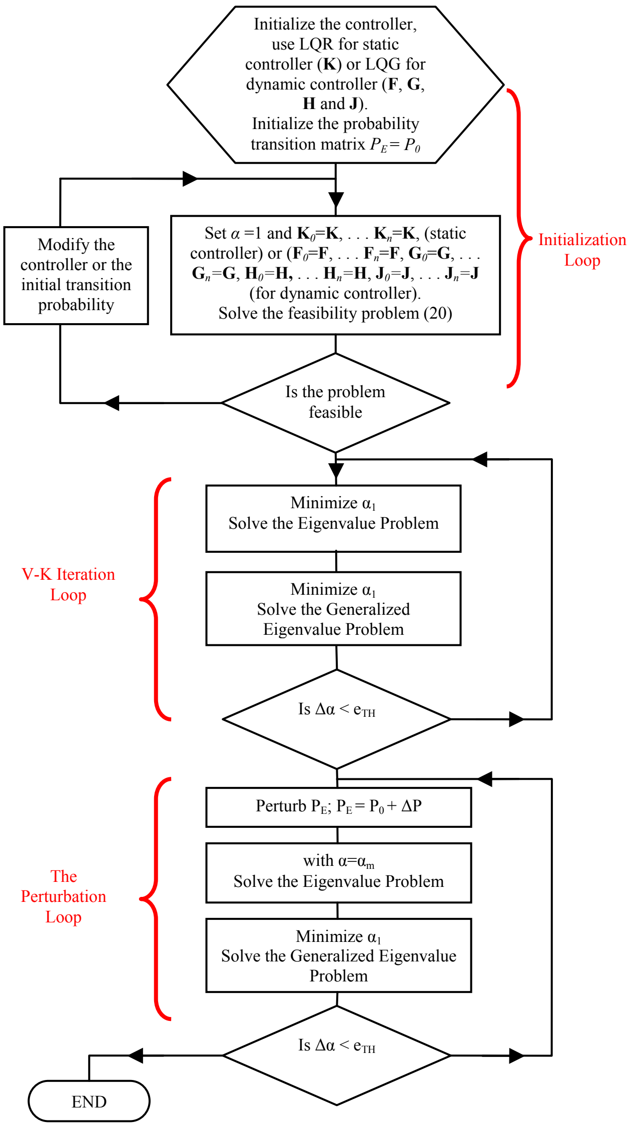
Initialize the controller, use LQR for static controller (K) or LQG for dynamic controller (F, G, H and J). Initialize the probability transition matrix $P_E = P_0$
Initialization Loop
Modify the controller or the initial transition probability
Set α =1 and K_0=K, . . . K_n=K, (static controller) or (F_0=F, . . . F_n=F, G_0=G, . . . G_n=G, H_0=H, . . . H_n=H, J_0=J, . . . J_n=J (for dynamic controller). Solve the feasibility problem (20)
Is the problem feasible
V-K Iteration Loop
Minimize α_1 Solve the Eigenvalue Problem
Minimize α_1 Solve the Generalized Eigenvalue Problem
Is $\Delta\alpha < e_{TH}$
The Perturbation Loop
Perturb P_E; $P_E = P_0 + \Delta P$
with $\alpha=\alpha_m$ Solve the Eigenvalue Problem
Minimize α_1 Solve the Generalized Eigenvalue Problem
Is $\Delta\alpha < e_{TH}$
END

Figure 4. The V-K iteration algorithm

It should be noted that the initial controller is designed for the free delay system. To get an initial feasible solution we have to start from small time delays and perturb the transition probability matrix toward higher time delays. The perturbation ∈ should be very small positive number in the order of 0.005. An example of the perturbation matrix is:

$$\Delta P_0 = \begin{bmatrix} -s & s & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \quad \Delta P_1 = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ -s & s & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}$$

As can be seen the sum of the perturbation through any row is zero. More aggressive initial transition probability matrix can be used. In [7][8] the perturbation is around 0.01 but even with this small perturbation sometimes the problem is divergent and we need to use smaller perturbation, for example around 0.005. Also for the two modes, the two probability matrices are perturbed at the same time while in our algorithm they are perturbed separately.

Example 1

The pendulum mass is denoted by m and the cart mass is M, the length of the pendulum rod is L. The open loop system is unstable. The states are defined as $x_1 = x$, $x_2 = \dot{x}$, $x_3 = \theta$, $x_4 = \dot{\theta}$. The linearized model can be given as:

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-mg}{M} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{(M+m)g}{ML} & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{-1}{ML} \end{bmatrix} u = \mathbf{Ax} + \mathbf{B}u$$

$$\mathbf{y} = \begin{bmatrix} x \\ \theta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{x} = h(x,u)$$

The parameters used are: M =1 kg, m = 0.4 kg, L = 0.7 m. The sampling time is $h = 0.1$ s. The time delay is bounded by 2: $r_s(k) \in \{0,1,2\}$. The initial condition is x = 0 and θ = 0.1. After sampling the system with 0.1 s sampling rate, the system matrices are given by:

$$\mathbf{A} = \begin{bmatrix} 1 & 0.1 & -0.0199 & -0.0007 \\ 0 & 1 & -0.4049 & -0.0199 \\ 0 & 0 & 1.0996 & 0.1033 \\ 0 & 0 & 2.0247 & 1.0996 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0.0050 \\ 0.1009 \\ -0.0073 \\ -0.1476 \end{bmatrix}$$

The required transition probability is given by:

$$P = \begin{bmatrix} 0.5 & 0.5 & 0 \\ 0.3 & 0.6 & 0.1 \\ 0.3 & 0.6 & 0.1 \end{bmatrix}$$

Using the LQR matlab function with **Q**=**I** and **R**=1. The controller is given by:

$$\mathbf{K}_{LQR} = \begin{bmatrix} 0.5943 & 1.4745 & 28.7321 & 6.7849 \end{bmatrix}$$

with the required transition probability and the LQR controller does not stabilize the system with the time delay because the solution is infeasible, the initial transition probability and the perturbation matrix are chosen as:

$$P_0 = \begin{bmatrix} 0.499 & 0.499 & 0.002 \\ 0.4 & 0.5 & 0.1 \\ 0.4 & 0.5 & 0.1 \end{bmatrix} \quad \Delta P_i = \begin{bmatrix} 0 & 0 & 0 \\ -0.005 & 0.005 & 0 \\ -0.005 & 0.005 & 0 \end{bmatrix}$$

After 20 iterations the desired transition matrix is reached and the stabilizing controller is given as:

$$\mathbf{K} = \begin{bmatrix} 0.3181 & 0.7972 & 21.2058 & 5.4654 \end{bmatrix}$$

Note that the process can be started with any P and $\mathbf{K}_{LQR}$ as long as they give feasible solution. Using Theorem 1 [22] the MADB using the LQR controller is 0.1210 s. With the stabilizing controller that takes the random time delay into consideration, Theorem 1 [22] gives 0.1420 s which shows an improvement in the stability margin with the new controller. The V-K iteration loop took 4-iterations and the perturbation loop took 20 iterations, the minimum decay rate is 0.8837. By changing the EVP loop by making an inner loop for minimizing α, the minimum attained decay rate is 0.8645 and the delay margin increased to 0.1563 s. The system response is shown in Figure 5. In the simulation the nonlinear dynamics is used. The stabilizing controller with the improved algorithm is:

$$\mathbf{K} = \begin{bmatrix} 0.2823 & 0.7050 & 20.5227 & 4.9714 \end{bmatrix}$$

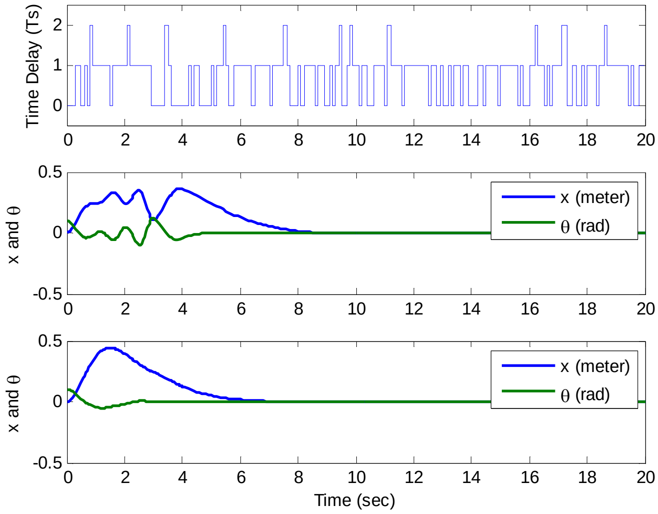

Figure 5. (a) The random time delay, (b) The response with the LQR controller, (c) The response with the controller generated by the improved V-K algorithm

V. Conclusion

In this paper, the NCS is modeled as discrete-time Markovian linear jump system where the time delay is modeled as Markov chain. Using the mean square stability the system stability as formulated as BMIs. The V-K iteration algorithm is used to solve the BMIs. We used an improved V-K iteration algorithm where the decay rate is improved in both the EVP and the GEVP loops. The method is tested on the cart and the inverted pendulum and we found that the decay rate is improved.

References

[1] Xiefu Jiang, Qing-Long Han, Shirong Liu, and Anke Xue. "A New *H∞* Stabilization Criterion for Networked Control Systems", *IEEE Transactions On Automatic Control*, Vol.53, No.4, 1025-1032. May 2008.

[2] Gregory C. Walsh, Hong Ye, and Linda G. Bushnell. “Stability analysis of networked control systems,” *IEEE Transactions on Control System Technology,* Vol. 10, No. 3, 438–446, May 2002.

[3] Magdi. S. Mahmoud, "Robust Control and Filtering For Time Delay systems", New York: Marcel Dekker, 2000.

[4] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," Proceedings of the IEEE, vol. 95, no. 1, pp. 138-172, 2007.

[5] C. Mo-Yuen and T. Yodyium, "Network-based control systems: a tutorial," in 27th Annual Conference of the IEEE Industrial Electronics Society, vol.3 ed Piscataway, NJ, USA, 2001, pp. 1593-1602.

[6] L. Lei-Ming, T. Chao-Nan, and Z. Hai-Jun, "Analysis and design of networked control systems with long delays based on Markovian jump model," in Proceedings of 2005 International Conference on Machine Learning and Cybernetics, Vol. 2 ed Piscataway, NJ, USA: IEEE, 2005, pp. 953-959.

[7] L. Lei-Ming and T. Chao-Nan, "Stabilization design of networked control systems," in 2008 International Conference on Machine Learning and Cybernetics, vol.4 ed Piscataway, NJ, USA: IEEE, 2008, pp. 2137-2142.

[8] X. Lin, A. Hassibi, and J. P. How, "Control with random communication delays via a discrete-time jump system approach," in Proceedings of the 2000 American Control Conference. ACC, vol.3 ed Danvers, MA, USA: American Autom. Control Council, 2000, pp. 2199-2204.

[9] M. Yu, L. Wang, T. Chu, and G. Xie, "Modelling and control of networked systems via jump system approach," IET Control Theory &amp; Applications, vol. 2, no. 6, pp. 535-541, June2008.

[10] Z. Liqian, S. Yang, C. Tongwen, and H. Biao, "A new method for stabilization of networked control systems with random delays," in Proceedings of the 2005 American Control Conference, vol. 1 ed Piscataway, NJ, USA: IEEE, 2005, pp. 633-637.

[11] Bo Yu and Yang Shi, "State Feedback Stabilization of Networked Control Systems With Random Time Delays and Packet Dropout," Michigan, USA: 2008, pp. 639-645.

[12] S. Yang and Y. Bo, "Output Feedback Stabilization of Networked Control Systems With Random Delays Modeled by Markov Chains," Automatic Control, IEEE Transactions on, vol. 54, no. 7, pp. 1668-1674, July2009.

[13] S. Yang, Y. Bo, and H. Ji, "Mixed H2/H∞ control of networked control systems with random delays modeled by Markov chains," 2009, pp. 4038-4043.

[14] Johan Nilsson, "Real-time Control Systems with Delays." PhD Lund Institute of Technology , 1998.

[15] Z. Guofeng, C. Xiang, and C. Tongwen, "A model predictive control approach to networked systems," in Proceedings of the 46th IEEE Conference on Decision and Control Piscataway, NJ, USA: IEEE, 2008, pp. 3339-3344.

[16] W. Jing, Z. Liqian, and C. Tongwen, "Model predictive control for networked control systems," International Journal of Robust and Nonlinear Control, vol. 19, no. 9, pp. 1016-1035, June2009.

[17] J. Wu, L. Zhang, and T. Chen, "An MPC Approach to Networked Control Design," 2007, pp. 10-14.

[18] Oswaldo L.V.Costa, "Stability Results for Discrete-Time Linear Systems with Markovian Jumping Parameters," Journal of Mathematical Analysis and Applications, vol. 197, pp. 154-178, 1993.

[19] L. El Ghaoui and M. A. Rami, "Robust state-feedback stabilization of jump linear systems via LMIs," International Journal of Robust and Nonlinear Control, vol. 6, no. 9-10, pp. 1015-1022, Nov.1996.

[20] A. Hassibi, J. How, and S. Boyd, "A path-following method for solving BMI problems in control," in Proceedings of the 1999 American Control Conference, vol.2 ed Piscataway, NJ, USA: IEEE, 1999, pp. 1385-1389.

[21] David Banjerdpongchai, "Parametric Robust Controller Synthesis Using Linear Matrix Inequalities." University of Stanford, 1997.

[22] A. F. Khalil and W. Jihong, "A new stability and time-delay tolerance analysis approach for Networked Control Systems," 2010, pp. 4753-4758.