Performance Evaluation of Co-Design of Discrete Event Networked Controlled DC Motor System

Ankur Jain\textsuperscript{1}\textsuperscript{*}, B. K. Roy\textsuperscript{2}

\textsuperscript{1,2}National Institute of Technology, Silchar, India
*Corresponding Author Email: ‘ankurjainjob@gmail.com

Abstract

In this paper, we have designed a control technique for a networked DC motor in the presence of networked delay, packet loss, and jitter. We have used the predicted states for the controller design to achieve the transient and steady-state objectives. A networked compensator is designed to overcome the network constraints. The network link is modelled using the queue server mechanism which can assimilate a lot of features of the network. The proposed technique can also be applied to various other applications. The analysis of the networked control system is done in frequency and time domains. The simulation results are presented to test the performance of the proposed control technique.

Keywords: Controller design; Internet of Things (IoT); Networked control system (NCS); DC motor speed control.

1. Introduction

The networked control system (NCS) is an interdisciplinary subject which consists of control engineering, communication engineering \textsuperscript{[Error! Reference source not found.]}\textsuperscript{[Error! Reference source not found.].} It can be seen as a distributed system \textsuperscript{[Error! Reference source not found.]}\textsuperscript{[Error! Reference source not found.],} which composed of computation in cyber-space, sensing in physical space and communication functionality to link cyber-space with physical space \textsuperscript{[Error! Reference source not found.. Error! Reference source not found.. Error! Reference source not found.].} Hence, it can also be called as one type of cyber-physical system (CPS) \textsuperscript{[Error! Reference source not found.].} The very fast growth in VLSI and ubiquitous computing \textsuperscript{[Error! Reference source not found.], cloud computing \textsuperscript{[Error! Reference source not found.. big data analysis\textsuperscript{[Error! Reference source not found.]. has been shifting the control engineering into a new paradigm and leads to significant number of research publications \textsuperscript{[Error! Reference source not found.]. In NCS, control loop(feedback, forward, or both) is closed by a network\textsuperscript{[Error! Reference source not found. Error! Reference source not found. Error! Reference source not found.]. The introduction of a network gives some advantages like ease of maintenance, low power consumption in signal transmission, better resource sharing, better data handling, low-cost control, less wiring, etc. At the same time, a network introduces delay (D), dropout (DP), packet disorder (DO), limited bandwidth (LBW), less computational power at plant side and high computational power on cyber side, etc. Various network constraints are reported in the literature and are listed Table 1.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
\textbf{Time delay} & \textbf{[Error! Reference source not found. Error! Reference source not found. Error! Reference source not found.] [Error! Reference source not found.]} \\
\hline
\textbf{Packet dropout} & \textbf{[Error! Reference source not found. Error! Reference source not found.]} \\
\hline
\textbf{Packet disorder} & \textbf{[Error! Reference source not found. Error! Reference source not found.]} \\
\hline
\textbf{Jitter} & \textbf{[Error! Reference source not found.]} \\
\hline
\textbf{Limited bandwidth} & \textbf{[Error! Reference source not found. Error! Reference source not found.]} \\
\hline
\textbf{Scheduling} & \textbf{[Error! Reference source not found. Error! Reference source not found.]} \\
\hline
\textbf{Security} & \textbf{[Error! Reference source not found.]} \\
\hline
\end{tabular}
\caption{The literature survey on network constraints}
\end{table}

Among all the mentioned constraints, delay, dropout and jitter have the major influences on the performance. These constraints may even cause instability. These constraints are modelled in various ways in the literature such as Markov chain \textsuperscript{[Error! Reference source not found.],} transportation delay \textsuperscript{[Error! Reference source not found.],} virtual switch among others.
Reference source not found]. a periodic sampling, etc. In this paper, we have formulated a discrete event system in which delay is modelled by service time of the server, and packet loss is modelled by the switch.

Nowadays, networked control system is used in variety of fields like surveillance system, remote surgery, robotics, automated manufacturing, ball maglev system, DC motor, automated highway, cooperative adaptive cruise control, inverted pendulum, telemedicine and many more. A survey on the applications of NCS is shown in Table 2

| Table 2: The literature survey on the applications of NCS |
|-------------|-----------------------------|
| Ball-maglev system | [Error! Reference source not found. ] |
| Robotics | [Error! Reference source not found. ] |
| Surveillance system | [Error! Reference source not found. ] |
| Automated Highway | [Error! Reference source not found. Error! Reference source not found. ] |
| Automated Manufacturing | [Error! Reference source not found. Error! Reference source not found. ] |
| TeleMedicine | [Error! Reference source not found. ] |

In this paper, we have designed a NCS for a DC motor. The networked control has been studied using various control approaches such as robust control, adaptive control, optimal control, stochastic control, predictive control. A survey on the use of various control approaches is listed in Table 3. Here, we have modelled our system as a discrete event system and for the purpose of analysis, we use time delay framework. Here, one of the objectives of the paper is to reduce the usage of network without compromising the performance of the system. This objective demands transmitting of information as and when required. Thus, an event-based design is an obvious solution.

| Table 3: The literature survey on approach for controller design |
|-------------|-----------------------------|

We introduce the condition for sampling (when to take a sample) for energy efficiency, the event threshold condition (when to transmit a sample) for bandwidth reduction and predictive control model (future control input) for networked delay compensation. We propose a networked predictive technique with an LQR control which is able to compensate the delay introduced by the network. For a DC motor, various control problems such as speed regulation, speed tracking, position tracking have been studied in the literature. The main idea behind an event triggered control is to sample a signal as and when needed.

In event-driven control system input is updated on the occurrence of a particular event rather than specified time [Error! Reference source not found.]. For NCS point of view, the sample is sent to the controller when the error (Tracking or stabilization) is large from the specified limit [Error! Reference source not found.]. We can classified event-triggered the system into continuous time event triggered system (CTETS)[Error! Reference source not found.], the periodic event-triggered system (PETS) [Error! Reference source not found.], and self-triggered system (STS) [Error! Reference source not found.]. In CTETS systems, plant states or output monitored continuously with dedicated hardware which obviously consumes more battery, while in PETS plant’s states or outputs are monitored periodically. In STS, event instant is predicted when to take a sample by the detector. From communication design perspective, researchers choose control data as priority data and try to execute as early as possible to reduce delay in communication, while control system designer, treat communication network as a source of delay and measurement noise or disturbance. So Robust controllers are designed which can tolerate these network-induced effects [Error! Reference source not found.]. Here we will focus co-design of Networked effects with the control system and evaluate the quality of performance from a control perspective and with less requirement of network services which motivates this paper.

In this paper, we designed a discrete event system for the speed control problem of networked DC motor with Delay, Drop out, Packet disorder, Jitter. Secondly, we designed predictive control strategy to predict the states as well as a control input. Third, we will design event triggered condition to reduce network and controller utilization. Fourth, we analyzed the system in the frequency domain. Fifth, we designed network compensator to choose correct control input in real time. Sixth, we evaluated the performance for control as well as $x(t) = Ax(t) + Bu(t), \forall t \geq 0$, communication perspective.

The organization of the paper is as follows: In Section 2, we described our system model, network model, assumptions and objectives. Our proposed methodology, event generation mechanism and network compensator design are given in Section 3. Thereafter, we have provided our simulation results in section 4. In section 5, conclusion and future research scope is presented.

2. Problem Formulation

2.1 System Description

The motor is a most common actuator for the purpose of automation in Industries, Smart homes, Electrical vehicles, Robots etc. A motor could be used to provide translational and rotational motion. Here for the purpose of the study, we are considering an armature controlled DC motor. A mathematical model of the motor could be derived by an electrical equivalent circuit (Fig. 1) follows:
Fig. 1: Electrical equivalent circuit of the DC motor

By torque v/s armature current characteristics of the DC motor, we know that torque generated is directly proportional to the armature current \( T \).

\[ T = K_i i \]  

(1)

where \( K_i \) is a torque constant. The back emf, \( E_a \), is proportional to the angular velocity of the motor shaft.

\[ E_a = K_c \theta \]  

(2)

where \( K_c \) is a back emf constant. In SI units, the motor torque and back emf constants are equal, that is, \( K_c = K_v \). So from there onwards we can use \( K_c \) for the sake of notation simplicity.

Above equations can be transformed into a state-space form which is shown in Fig. 1.

\[ J \dot{\theta} + b \dot{\theta} = K_v i \]  

(3)

\[ L \frac{di}{dt} + Ri = V - K_c \theta \]  

(4)

Here we assumed that magnetic field is constant. Motor shaft and rotor are rigid in nature. Let maximum and the minimum voltage applied to DC motor are \( V_{\text{max}} \) and \( V_{\text{min}} \).

For the purpose of analysis and controller design, we can transform Eq. 3 and Eq. 4 into a state-space form which is shown in Eq. 5 and Eq. 6, here we have chosen the rotational speed \( \dot{\theta} \) and electric current \( i \) as the state variables. In this problem, armature voltage and rotational speed will be input and output respectively.

\[ \frac{d}{dt} \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} = \begin{bmatrix} \frac{-b}{J} & \frac{K_v}{J} \\ \frac{-K_v}{L} & \frac{-R}{L} \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} V \]  

(5)

\[ y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} \]  

(6)

Above equations describe the model of single DC motor. This plant could be written in the following manner:

\[ u(t) = K_x(t_k), \forall t \in [t_k, t_{k+1}), \forall k \in \mathbb{N} \]  

(7)

Here, the plant is modelled as linear time-invariant system and controller is modelled as linear state feedback controller, where \( K \) is controller gain. Let’s suppose sampler is being activated at the time instances \( t_k = \{1, 2, 3, \ldots \} \), and in between time instances sample value is piece-wise constant. Whenever the controller is activated, it takes the latest packet available at the buffer and calculates control input. The above system Eq. 7 & 8 could be written in discrete form as

\[ x_{k+1} = A_j (h) x_k, \forall k \in \mathbb{N} \]

\[ A_j = e^{Ah} + \int_0^h e^{v^i} dsBK \]

where,

\[ x_k = x(t_k) \]  

and \( h \) is the constant sampling time. In case of time-varying sampling, the analysis of the above system is very complex. The networked control system operates in presence of communication delay which is shown in Fig. 2. We can modify the system Eq. 7 & 8 as

\[ u(t) = K_x(t_k) = K_x(t - \tau(t)) \]  

(10)

where, delay \( \tau(t) = t - t_k , \forall t \in [t_k, t_{k+1}) \), is a piecewise constant. So a time-varying system could be written as below,

\[ \dot{x}(t) = Ax(t) + BKx(t - \tau(t)), \forall t \geq 0 \]  

(11)

3. Modelling of the Network

Here, we have modeled the network with the help of queuing theory. It is an arrangement of entities in a sequential manner. For this particular paper, we are considering a packet as an entity. Here packet is used for information transformation from the sensor to the controller (samples) and from the controller to an actuator (controller input values). The server is a physical or a logical node which can provide a particular type of computation service.

Here we would like to study effects of the network on the performance of the control system. The network characteristics like delay, packet loss, network Bandwidth, scheduling policy, etc. could be modelled by a logical queue and server arrangement. In this paper we have modelled networked delay as service time of the server, packet dropout could be modelled by a switch with a terminator block. Terminator block shows the lost packet. Scheduling policy could be created by the priority queue. Higher priority could be given to concerned packet from a particular plant or from a particular time.

Here we have assumed round-robin scheduling policy. Only one packet could be transmitted at a time. The network has a minimum \( T_{\text{min}} \) and maximum \( T_{\text{max}} \) transmission time. The network has maximum packet dropout probability \( P_{\text{drop}} \). Here, our objectives are to design controller gain \( K \) while ensuring stability under the networked constraints. Meet performance objective (maximum overshoot should be less than 3%, settling time should be less than 2 seconds, steady state error should be less than 2%).

4. Proposed Approach

Here we are proposing a discrete event full state feedback networked control system architecture which is shown in Fig. 2.
4.1 Design of Discrete Event Generation

A continuous event (packet) generation [Error! Reference source not found.] could block network which results in packet loss or information delay. Here our objective is to reduce packet generation as much as possible until control performance is compromised. Here we are going to present a state differential triggering method for event generation. Here we can observe the local state in continuous, periodic or predicted instants, then compare the threshold condition, if the sample violates the threshold condition then the particular sample is encapsulated into a packet and sent to the controller otherwise the packet is dropped locally. Here, Event is defined when a current sample of observing state is more than a certain threshold value of the previously sent packet. Mathematically it could be defined as

$$x(k_i - 1) - x(t_i) > \delta$$

where, $t_i$ is the time instant at which state is being observed, $k_i$ is the time instant at which packet is being sent, $x(k_i) = \text{value of the sample at } k_i \text{ instant}$, $x(t_i) = \text{value of the sample at } t_i \text{ instant}$, $\delta = \text{value of the threshold}$.

4.2 Packet Binding

We can send measurements and control sequence in one packet. Here we considered an entity as a packet which is used for the information exchange among nodes. Here, plant and controller are considered as different nodes which are connected by a packet-based network. Presently we have two existing packet type which is universal datagram protocol (UDP) and transmission control protocol (TCP). A packet consists of a header (overhead) and payload (Data).

4.3 Controller Design

Prediction of states: Let’s suppose there is a bounded maximum delay $\tau$ for their physical significance. We can express this delay in multiple of sampling time. Assume that $x(k_i)$ and $u(k_i)$ are the current sampled states and current control input respectively. Then from Eq. 7, we can derive that,

$$x(k_i + 1 | k_i) = px(k_i) + qu(k_i)$$
$$x(k_i + 2 | k_i) = p^2x(k_i) + pqu(k_i)$$
$$x(k_i + \tau | k_i) = p^\tau x(k_i) + p^\tau q u(k_i) + p^{\tau-1} q u(k_i) + \ldots + q u(k_i)$$

The above equations could be represented as an augmented model below

$$X = P x(k_i) + Qu(k_i)$$

where,

$$P = \begin{bmatrix} p & 0 & 0 & \ldots & 0 \\ p^2 & pq & q & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots \\ p^\tau & p^{\tau-1} q & p^{\tau-2} q & \ldots & q & \ldots & \ldots \end{bmatrix}$$

let’s suppose we would like to regularize our output at $Ref(k_i)$. So here we have the objective to design a controller which bring predicted output near to the set point as soon as possible. Assume that a reference data vector $Ref(k_i)$ which have the set point information,

$$Ref(k_i)^T = \begin{bmatrix} 1 & 1 & \ldots & 1 \end{bmatrix} x(k_i)$$

here, we can achieve our control objective by minimizing a cost function $J$,

$$J = (Ref(k_i) - X)^T (Ref(k_i) - X) + U^T R U$$

we know then, for minimum $J$

$$\frac{\partial J}{\partial U} = 0$$

Then,

$$U = (Q^T Q + \bar{R})^{-1} Q^T (Ref(k_i) - P x(k_i))$$
4.4 Design of Network Effects Compensator

Introduction of network introduces random network delay which should be taken care to improve the performance of control design. We know that in a network transmission we can pack multiple data in a single packet. So, with the controller design in the previous section, we will predict multiple control input depend on the delay bound, and send to the network compensator. Network compensator chooses right control input for the real-time delay and sends the actuator. Mathematically it should be written as

\[ u_t = c(\tau) * U(17) \]

5. Simulation Results

For verification of the compensation techniques and the controller performance, we have used armature controlled DC motor as a plant. Physical parameters like Moment of inertia (J), Viscous friction constant (b), force constant (f), torque constant (k), motor resistance (R), inductance (L) are taken as 0.01, 0.1 N.m.s, 0.01, 0.01, 1 Ohm, 0.5 respectively.

Fig. 3: Inter-generations time for checking states
Here to avoid complex continuous measurement and for saving energy we have used periodic event monitoring scheme which can be seen in Fig. 3, periodic check of state for event recognition is done with 0.1s inter-generations time (IGT).

Fig. 4: Random round trip delay for each packet

At each inter-generations time we check the threshold violation, if it occurs then the particular sample value is packed and sent to the controller, else it is discarded locally. In Fig. 4, random network delay is seen during transmission. Step response of DC motor for speed regulation could be seen in Fig. 5, all the transient and steady-state performance, we have decided is being full-filled by the proposed design.

6. Conclusion

Here, we have solved speed regulation Problem under random networked delay. In this paper, we have modelled NCS as queue server model which could easily capture the effects of network delay and packet dropout and jitter. Here we have considered round trip delay which is a summation of delay occurred in feedback channel, forward channel, computation delay, and delay due to packet loss. A predictive control scheme is used to generate future control input according to delay bound. A network effects compensator is designed. A simulation has been performed to show the suitability of the method. A controller is designed which is capable to meet the desired performance under networked scenario.

7. Acknowledgement

The authors acknowledge the financial support given by TEQIP-III, NIT Silchar, Silchar - 788010, Assam, India.

References

[7] Zoleikha Abdollahi Biron and Pierluigi Pisu. Distributed fault detection and estimation for cooperative adaptive cruise control sys-


