Performance Evaluation of Parallel-processing Networked System with Linear Time Delay

Sohag Kabir\(^1\) A S M Ashraful Alam\(^2\) Tanzima Azad\(^3\)

\(^1\) Department of Computer Science, University of Hull, Hull, United Kingdom
Email: s.kabir@hull.ac.uk

\(^2\) Department of Computer Science, University of Otago, Dunedin, New Zealand
Email: aalam@cs.otago.ac.nz

\(^3\) Department of CSE, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh
Email: azad.tanzima@dbbl.com.bd

Abstract

Time delays in system subcomponents or control may result into unacceptable system operation or uncertainty in specialised technical systems like aircraft control, plant control, robotics, etc. In parallel computing, different computing subunits share their tasks to balance loads to increase performance and throughput. To be able to do so, subsystems have to communicate among themselves, which adds further delay on top of the existing system delay. It is possible to improve performance and stability of the whole system, by designing observer for every subsystem in the system, overseeing the system-state and compensating for existing time-delay. This paper reviews the present literature to identify a linear time-delay system for load balancing and evaluates the stability and load-balancing performance of the system with and without an observer. Stability is analysed in terms of oscillation in the system responses and performance is evaluated as the speed of load-balancing operation.

Keywords: Networked System, Parallel Computing, Time-Delay Systems, Cluster Computing, Load Balancing

1 Introduction

Parallel computing is the dominant paradigm in large complex systems architecture. Simultaneous calculations are carried out in parallel-computing systems, following the principle that large problems can be split into smaller ones and can be solved concurrently. A busy computing unit may share some of its load with a relatively idle computing unit, resulting in a significant increase in system performance and throughput. However, in a network environment, information about the workload on subunits are not instantly available to other subunits due to time-delay experienced during communication, which affects system performance adversely.

Time-delay systems, generally known as systems with deceased time or after impact, transmissible systems, systems of equations with deviation in argument or systems of differential-difference equation. They are a constituent of infinite dimensional functional differential equations (FDEs) which are opposite of ordinary differential equations (ODEs) (Richard 2003). In the clustered machine environment, additional communication delay is inevitable. The needs for observing time-delay in systems comes from the real demand of system tracking, control and/or identification of system failure.

Feedbacks are important in mission critical systems. The difficulties concerning sensitivity and robustness of the feedback system with respect to time delays has attracted much attention as feedback control systems add additional delays. The adverse effect of time-delay can be minimised by observing the states of time-delay systems and then utilising the information to model a linear time-delay system aimed at designing observers for all subsystems to minimise delays and increase stability.

This paper briefly reviews the literature to identify a networked parallel processing system with linear time-delay. Then the system is modelled and simulated using SIMULINK to analyse the stability and performance of the system in performing load balancing operation. Stability is analysed in terms of oscillation in system responses and performance is measured as speed of operation, i.e. how quickly the system can balance loads among subsystems. We have designed the observers for all subsystems of the parallel processing linear time delay system based on the findings of these analyses. Observer of each subsystem estimates some necessary values for the subsystem to allow it to work independently without communicating with other subsystems thus minimising the effect of time-delay in the system performance. To evaluate the system performance the system with observers are modelled and simulated in SIMULINK.

The rest of the paper is organized as follows: Section 2 presents the background study and literature review on time-delay systems, observer design and load-balancing techniques. The parallel processing linear time-delay system is described, modelled, simulated, and results of the simulation are shown in section 3. In section 4, observers are designed, system with observers is modelled, simulated and the results of the simulation are presented. Section 5 presents the discussion of the results obtained during the analysis. Finally, concluding remarks are presented in section 6.

2 BACKGROUND STUDY AND LITERATURE REVIEW

2.1 Time-Delay Systems: Their Mathematical Models and Observers

Time-delay is experienced frequently in various control systems; either in the control input, the system states or processing (Fridman & Shaked 2002). Time-delay can be a source of instability and is very important for perfor-
mance and consistency of control systems. Researchers like Xia & Jia (2002), He et al. (2004, 2007), Lin et al. (2006) have contributed to stability analysis of time-delay systems. Robustness of time delay systems are studied by Kharitonov (1999). Improved time-delay systems and stabilised controllers are suggested by Fridman & Shaked (2002), Zhang et al. (2005), Hua et al. (2005). A network-based controller for time-delay system is studied by Gao et al. (2008) and stability of a networked control system is analysed by Ko et al. (2011). Abdallah et al. (2003) have studied a linear time delay system to see load balancing instabilities in parallel computing.

Time delay system can be modelled in different ways like the transfer function model, differential equation model and others. A simple linear time delay system can be represented as:

\[ x(t) = ax(t) + bu(t) \]  

where \( a \) and \( b \) are constants, \( x(t) \) and \( u(t) \) are functions that change with time. We can rewrite the equation for a time delay \( \tau \) in following way with another constant \( c \) as following:

\[ \dot{x}(t) = ax(t) + bu(t) + cx(t - \tau) \]  

We also can model a time-delay system using delay differential equation (DDE). In DDE, the derivative of an unknown function at a certain time is expressed in terms of the values of the function at previous time. DDE is similar to ODE in mathematical treatment, but their evolution involves previous values of the state variables. The solution of DDE therefore requires knowledge of not only the current state, but also the state knowledge at a certain historical time. A general form of the time delay differential equation for \( x(t) \in \mathbb{R}^n \) is:

\[ \frac{d}{dx}x(t) = f(t, x(t), x_\tau) \]  

where \( x_\tau = \{x(\tau) : \tau \leq t\} \) represents the trajectory of the solution in the past.

Different types of DDE can be used to address delay of various nature. Equation (4) expresses a system with continuous delay, and equation (5) with discrete delay:

\[ \frac{d}{dx}x(t) = f(t, x(t), \int_0^t x(t + \tau) d\mu(\tau)) \]  

\[ \frac{d}{dx}x(t) = f(t, x(t), x(t - \tau_1), \ldots, x(t - \tau_m)) \]  

for \( \tau_1 > \cdots > \tau_m \geq 0 \).

DDE for a linear system with discrete delay takes following form:

\[ \frac{d}{dx}x(t) = A_0x(t) + A_1x(t - \tau_1) + \cdots + A_mx(t - \tau_m) \]  

where, \( A_0, \ldots, A_m \in \mathbb{R}^{n\times n} \).

Different methods like algebraic, geometric, Kronecker canonical form, singular value decomposition, inversion approaches, generalized inverse techniques can be used to design observer for time delay systems. Presently, network-based observer design is an emerging topic because of the constraint that all the required system components cannot be accommodated at same place in many systems. These components add further transmission delay by using network medium for communicating.

### 2.2 Control Systems and Observer

Control systems use sensors to determine and control the state of the variables/quantities like movement, heat circulation, temperature, stress, fluid pressure etc. Sensors have limitations that may cause systems instability. Internal states of many systems cannot be observed directly. This raises problems in feedback systems and consequently rules out the possibility of state feedback. A separate system, termed as an observer or an estimator that attempts to duplicate the values of the state vector, can be designed.

Observers are methods that merge with sensed quantities alongside the other information of the control system to generate control signals. Observers can replace or supplement to the sensors in a control system. Observer generated control signals can be more precise, cheaper to generate, and more trustworthy than the control signals produced by sensors.

### 3 Linear Time-Delay System to Study Load Balancing Performance

#### 3.1 System Description and Mathematical Model

In this paper, a computer cluster consisting of \( n \) nodes is considered which was first used by Abdallah et al. (2003). In this model, all the nodes can communicate with each other. At the beginning, all nodes are provided with equal work load. With the passage of time, the loads on nodes become unequal due to the varied processing capability of the nodes. To balance loads among nodes, each node in the cluster sends its load information (queue size) \( q_j(t) \) to other nodes of the cluster. Due to time-delay node \( i \) receives information sent by node \( j \) delayed by a certain amount of time \( \tau_{ij} \); i.e., \( q_j(t) \) is received as \( q_j(t - \tau_{ij}) \).

After receiving information from all other nodes, each node locally calculates and determines the average amount of load using a simple estimator \( \frac{\sum_{j=1}^n q_j(t - \tau_{ij})}{n} \) where \( \tau_{ij} = 0 \). Then each node compares its own queue size against locally calculated queue size to make decision about task sharing. If the queue size of a node is greater than the average queue size then the node will send a portion of its tasks to other nodes. Again, the sending tasks also add further delay in the course of its execution. As a result, node \( j \) receives tasks sent by node \( i \) with a time delay of \( h_{ij} \). The load balancing algorithm decides on the frequency of performing load balancing operations considering that each node has delayed value of queue size and tasks send from one node to other is received with a certain time delay.

The mathematical model of the above system can be described as (Abdallah et al. 2003):

\[ \frac{dx_i(t)}{dt} = \lambda_i - \mu_i + u_i(t) - \sum_{j=1}^n p_{ij} f_{ij}(u_j(t - h_{ij})) \]  

\[ u_i(t) = -K_j y_j(t) \]  

\[ y_j(t) = x_j(t) - \frac{\sum_{j=1}^n x_j(t - \tau_{ij})}{n} \]  

where, \( n \) is the number of nodes in the cluster; \( x_j(t) \) is the expected waiting time experienced by a task inserted into the queue of the \( j^{th} \) node; \( \lambda_i \) is the rate of increase of waiting time and \( \mu_i \) is the rate of reduction in waiting time at node \( i \). \( u_i(t) \) is the rate of transfer of the tasks from node \( i \) at time \( t \) to other nodes; \( p_{ij}f_{ij}(t) \) is the rate that node \( j \) sends waiting time to node \( i \) at time \( t \) where \( p_{ij} \geq 0 \) and \( \sum_{j=1}^n p_{ij} = 1 \) and \( p_{jj} = 0 \); \( -p_{ij} f_{ij}(t - h_{ij}) \) is the rate
of increase of the expected waiting time (tasks) at time \( t \) from node \( j \) by (to) node \( i \) where \( h_{ij} \) is the time delay for the task transfer from node \( j \) to node \( i \) and \( h_{ii} = 0 \); and \( \tau_i \) is the time delay for communicating queue size from node \( j \) to node \( i \). In this paper, we have considered three computing nodes instead of \( n \) nodes generalised in the above equations.

3.2 SIMULINK Block Diagram and Simulation Results

As mentioned earlier the whole system consists of three subsystems with identical structure and characteristics. Therefore, block diagram of each subsystem is built first using Matlab SIMULINK. Then the subsystems are connected together to create the whole system.

In this paper, the system is simulated by setting \( \lambda_1 = 3 \mu_1, \lambda_2 = 0, \lambda_3 = 0, \mu_1 = \mu_2 = \mu_3 = 1, \tau_{ij} = \tau = 200 \mu s e c \) for \( i \neq j \), \( \tau_{ii} = 0 \), and \( h_{ij} = 2\tau = 400 \mu s e c \) for \( i \neq j \), \( h_{ii} = 0 \), and initial condition as \( x_1(0) = 0.85, x_2(0) = 0.63 \) and \( x_3(0) = 0.5 \). Processing gain for all the subsystems is thought as equal, i.e., \( K_1 = K_2 = K_3 = K \).

Figure 1a and 1b show the experimental responses with \( K_1 = K_2 = K_3 = 1000 \) and \( K_1 = K_2 = K_3 = 2000 \) respectively. It is observed from the simulation outcome that the oscillatory behaviour of system responses increased as the gain increased. At the same time we can see that, responses in Figure 1b with the higher gain oscillate more and die out slowly compared to responses in Figure 1a with less gain. This unexpected behaviour is due to the delay experienced by the subsystems during communication. When time delay is set to zero, then responses with higher gain dies out faster as expected. The oscillatory behaviour of the output response depends on the linear control law (load balancing algorithm). The system experiences time delay whenever any transmission takes place among nodes.

In the above system, transmission of task depends on the value of \( u_i(t) = -K\gamma_i(t) \) as decided by the linear control law. According to load balancing algorithm, if at the \( i \)-th node, \( u_i(t) = -K\gamma_i(t) < 0 \) i.e., \( \gamma_i(t) > 0 \), then \( i \)-th node sends some of its tasks to other nodes. On the other hand, if \( \gamma_i(t) < 0 \) i.e., \( -K\gamma_i(t) > 0 \), then the node will not send tasks to other nodes, rather the node instantly takes waiting time (tasks) from other nodes.

![Figure 1: System responses with different system gains](image)

4 Observers for linear time-delay systems

In this section, observers of the linear time-delay system is designed considering the effect of time-delay in the system performance. Combining a measured feedback signal with knowledge of the control system components, the behaviour of the plant can be known with greater precision than by using the feedback signal alone (Ellis & Ellis 2002).

4.1 Mathematical Model of Observer

Let a time delay system be defined as follows:

\[
\dot{X}(t) = A(d)X(t) + Bu, y(t) = C(d)X(t)
\]

where, \( X(t) \in \mathbb{R}^n \) is the system state and \( y(t) \in \mathbb{R}^m \) is the output of the system at time instant \( t \). \( A(d) \) and \( C(d) \) are polynomial matrices in the time-delay operators \( d = \{d_i\} \). For any \( i \), operator \( d_i \) is defined by \( d_iX(t) = X(t - \tau_i) \) with \( \tau_i > 0 \) which is the time-delay constant. When compared to the standard Luenberger observer, the following system might be considered as an observer for the system described by equation (10):

\[
\dot{\hat{X}}(t) = A(d)\hat{X}(t) + Bu + L(d)(y(t) - C(d)\hat{X}(t))
\]

where, \( \hat{X}(t) \rightarrow X(t) \) as \( t \rightarrow \infty \) for arbitrary initial values of \( \hat{X}(t) \) and \( L(d) \) is observer gain.

Now by using observer design concept described in equation (11), the observer for the linear-time delay system considered in this paper can be defined as:

\[
\dot{\hat{X}}(t) = A(d)(\hat{X}(t) + \lambda - \mu + L(d)(y - C\hat{X}(t)))
\]

4.2 SIMULINK Block Diagram and Simulation Results for the Systems with Observer

According to the observer equation and using SIMULINK built in blocks an observer is designed for subsystem 1. Figure 2a and 2b show the responses of the system in the presence of observer.

![Figure 2: System responses in the presence of observers with different system and observer gains](image)

Now, when we compare the responses produced from system with and without an observer, it is clearly evident that the responses of the system with observer oscillates less with higher gain and die out quickly, i.e. balances the load quickly with higher gain which is the expected behaviour. In this case, system with processing gain \( K_1 = K_2 = K_3 = 2000 \) and observer gain \( L = 2000 \) balances the load quickly with less oscillation in comparison to system with processing gain \( K_1 = K_2 = K_3 = 1000 \) and observer gain \( L = 1000 \).

5 Discussion

One of the important issues of linear time-delay system is whether or not the system model is stable in terms of oscillation in the responses. It is well established that the presence of delays in various states of the system has an enormous impact on the stability of the system. Even when the system is stable, system performance can be sluggish. For example, if the system is stable but oscillates much while transferring tasks back and forth among subsystems rather
than processing tasks, then system performance will degrade and the system will unnecessarily waste resources.

The system experiences delays during sending queue size and tasks from one subsystem to another. For simplicity, delays in sending queue size and tasks were considered as constant. Initially, all the subsystems were fed with an equal quantity of load, then subsystems experiencing creating and reducing waiting time, $x_i(t)$. Since the reduction rate of the waiting time for all subsystems is set as equal and generation of waiting time as unequal, therefore, loads become imbalanced within few moments. To balance loads among the nodes, each node shares load information with other nodes. As communications to share waiting time take place among nodes, all the nodes experience delays. As a result, subsystems receive queue size (expected waiting time) and tasks delayed by a certain amount of time.

The linear control law $u_i(t) = -K_i x_i(t)$ is used for load balancing, where $u_i(t)$ is the rate of reduction of waiting time $x_i(t)$ per unit time. According to the control law, within $\Delta t$ time interval between successive executions of the load balancing algorithm, a fraction of the queue is removed resulting into reduced waiting time. With higher system gains, the waiting time reduces quickly. In the simulation, it is observed that with higher gain the system responses oscillate more and die out slowly. This unexpected system behaviour is due to the delays experienced during communication. If the effect of delays on system performance can be compensated, then the system responses will die out fast as expected with higher gain. As delays are unavoidable, therefore the system gain is chosen in a way that the system responses oscillate moderately and system performance are mildly affected. To avoid the effect of time delay during the exchange of waiting time, observers are designed for all subsystems. Observer of each subsystem can estimate some important information, and thus enabling each subsystem to work independently. As a result, communication between nodes to exchange waiting time is no more necessary; therefore no delay is experienced by the system due to communication.

6 Conclusion

In this paper, a linear time-delay system is modelled to analyse stability and performance of the system in performing load balancing operation. It is assumed that the system consists of three symmetric subsystems and all the delays experienced by the subsystems are constant. In principle, system performance is considered as less oscillation in the responses and it increases as the system gain increases. Stability is another issue which requires to be considered along with the system performance. In the modelled system, to perform load balancing operation each subsystem communicates with other subsystems to get information about the amount of load on other subsystems. This results in the system experiencing delays due to mutual communications between subsystems and these delays limit the value of the system gain.

It is observed that, the system responses oscillate more with higher gain in comparison to lower the gain. More oscillation in system responses means the system waste resources by passing tasks back and forth among subsystems rather processing tasks which results in degraded system performance. This unexpected behaviour of the system is due to the delays experienced by the subsystems while communicating with each other. After analysing system responses, observer of the system is designed keeping the effect of time-delay on system performance in mind. As observers are designed considering all the issues that affect system performance, therefore expected system performance, i.e., the load is balanced quickly with higher gain, is obtained from the system with the observer.

In this experiment, all the subsystems were considered symmetric and delays as constant. But in real life, these assumptions are unlikely. Delays depend on different physical properties such as the availability and bandwidth of network, execution time of programs etc. Therefore, in future we plan to model linear time-delay system and design observers of the system considering variable delays and asymmetry of the subsystems. A processor of a subsystem uses a considerable amount of time to perform load balancing other than processing tasks which is not considered in this work. In the future, the time required by the processors of the subsystems to perform load balancing operation can be taken into account together with the time required to process tasks.

References


Hua, C., Guan, X. & Shi, P. (2005), 'Robust backstepping control for a class of time delayed systems', *IEEE Transactions on Automatic Control* 50(6), 894–899.


