Communication Delay Modelling and its Impact on Real-Time Distributed Control Systems

Rachna Dhand
School of Engineering and Energy
Murdoch University
Murdoch, WA, Australia
R.Dhand@murdoch.edu.au

Gareth Lee
School of Engineering and Energy
Murdoch University
Murdoch, WA, Australia
Gareth.Lee@murdoch.edu.au

Graeme Cole
School of Engineering and Energy
Murdoch University
Murdoch, WA, Australia
G.Cole@murdoch.edu.au

Abstract—**Communication delays are random in nature. A distributed real-time control system linked through a communication network is bound to be affected by the randomness of communication delay patterns. Statistical modelling techniques, like Auto-regressive Integrated Moving Average (ARIMA), may be used to model the network traffic. This paper provides a comprehensive coverage of network traffic modelling through the stochastic approach ARIMA with a case study of National Instruments (NI) DataSocket Transport Protocol (DSTP) based on high bandwidth Ethernet. In real-time control systems the controller optimization requires accurate temporal specification of sensitive controller tasks. Logical computation languages such as Time Definition Language (TDL) have successfully eliminated the temporal inaccuracies in designing control software. The paper provides an analytical and programmatic view on the impact and compensation of unpredictable network delays through discrete-time control algorithms, that are designed in Time Definition Language (TDL). The results validate that the discrete implementations are able to compensate for the delay, thus guaranteeing the stability of the control loop in the presence of unpredictable delays.**

Keywords-Real-Time Control Systems; Time Definition Language (TDL); Auto-Regressive Integrated Moving Average (ARIMA; Network Delay, Smith Predictor; Buffered Time-stamped Dahlin Algorithm

I. INTRODUCTION

A complex industrial control system is designed in a hierarchy as:

1) Supervisory and Control Software at the top layer.
2) Control equipment and PLC's linked together to form a middle layer.
3) Control devices like sensors and actuators at the bottom layer.

The system is finally connected via a communication network. Thus a network control system requires at least one link to be carried by a real-time network [11][12]. The most preferred network protocols for control systems are Ethernet-based MODBUS, PROFIBUS, or Controller Area Network (CAN). The time delays are not always local to the controller tasks. They can occur as transmission delays from a sensors to a controller and from controller to an actuator [9][10] because control equipment is connected via network.

The aim of this paper is to model network induced delays and analyse their impact on control operations through a posterior analysis. The paper provides comprehensive coverage of the statistical modelling technique ARIMA, used to model network traffic with a case study exploring National Instruments' (NI) DataSocket Transport Protocol (DSTP) based on Ethernet. The last section analyses the impact of delays on control loop operation and compares several delay compensation algorithms for control system design. A second order interacting system for liquid level within two tanks is considered. The control data is managed through National Instruments' DataSocket Server Manager. The delay modelling is consequently done for Gigabit Ethernet traffic based on DataSocket Transport Protocol (DSTP).

The paper is organized in four sections. The following two sections explain the role of delays and their modelling for Industrial Control Systems. Section four presents an overview of the statistical delay modelling technique ARIMA for network traffic. The case study for Ethernet traffic is analysed . The last section provides a brief overview of the Smith Predictor and Dahlin technique, and their implementation in Time Definition Language (TDL) for delay compensation of control loop operation. The aim is to analyse a typical control problem

Copyright (c) IARIA, 2010 ISBN: 978-1-61208-101-4

for real-time requirements, precise modelling and compensation of network delays to optimize the performance of the system.

II. INDUSTRIAL CONTROL SYSTEM ARCHITECTURE

Communication delays are random in nature, hence are difficult to model. Moreover they are affected by the type of network protocols in use. In an industrial control system, the role of the network becomes more crucial if the plant operates in real-time mode. The network layer forms the central data highway with which display and control equipment are exchanged. A challenging problem in control of an industrial plant is minimization of delay within a control loop. The time delay in executing the control algorithm originates from:

1) Control operation;
2) Sampling time chosen if a discrete-time controller is used;
3) Transmission delays due to network characteristics like network protocol in use, the network topology, or the type of physical network hardware used.

Real-time programming methodologies like RT-Java or Timing Definition Language (TDL) have evolved in the last decade to address the increasing complexity of control systems and scalability of equipment in the industrial automation domain. As explained by C.M. Kirsch, and R. Sengupta in their work in [1], the programming abstractions for real-time systems can be classified according to the processor execution time cycle for a given control task. The developments achieved by the computing community through RT-Languages like TDL conform to compensation of one-unit delay in control algorithm execution. TDL achieves timing predictability through the time-triggered architecture [2]. It executes the real-time code by separating the platform dependant issues like schedulability from platform independent issues like generating code from a given SIMULINK model of the system. TDL integrates well with simulation and modelling environments such as SIMULINK [3]. TDL guarantees control stability once the computational delays within the control loop operation are known. Transmission delays, on the other hand also play a vital role in control system’s stability. The transmission delays are tough to model and estimate because of the stochastic nature of network traffic. Many statistical models are available these days to model the behaviour of network traffic and hence estimate the average network delay. The focus in this paper is on ARIMA.

III. MATHEMATICAL MODELLING FOR TIME-DELAYED CONTROL SYSTEM

The Laplace transform models delays in transfer function using the exponential term $e^{-\alpha s}$, where α is the associated time delay that can either represent an input delay or output delay [1]. Thus mathematically the transfer function can be defined as:

$$g(s) = \frac{K}{\tau^2 S^2 + 2\zeta\tau s + 1} u(s)\, e^{-\alpha s} \quad (1)$$

where g*(s) is the non-delayed transfer function for the system. Here α signifies the delay that must be taken into account while designing continuous or discrete controllers. The stability of an overall system depends on all elements that make up the control system architecture, including the communication network. So long as real-time operating system and software development methods are employed, the computational delays can be assumed to be fixed. A well-designed discrete model of the system makes critical assumptions about the controller gain and sampling time. The critical assumptions for a closed loop system are given with characteristic equation specified as [4]:

$$1 + g_p(z)*g_c(z) = 0 \quad (2)$$

where $g_p(z)$ is the z-transformation of a continuous plant transfer function and $g_c(z)$ is the discrete controller transfer function. For a discrete-time closed loop system to be stable, all the roots of equation 1, must lie within the unit circle [4].

One important factor that is still not taken into consideration is the effect of random delays that come from the communication medium in the control loop operation. The stability margins specified above lay stress on retuning the controller parameters for a discrete-time system to take into consideration the impact of the A/D and D/A converters introduced within the control loop. In order to analyse the impact of delays that originate from a communication medium, the first step is to mathematically model the control system for a communication delay.

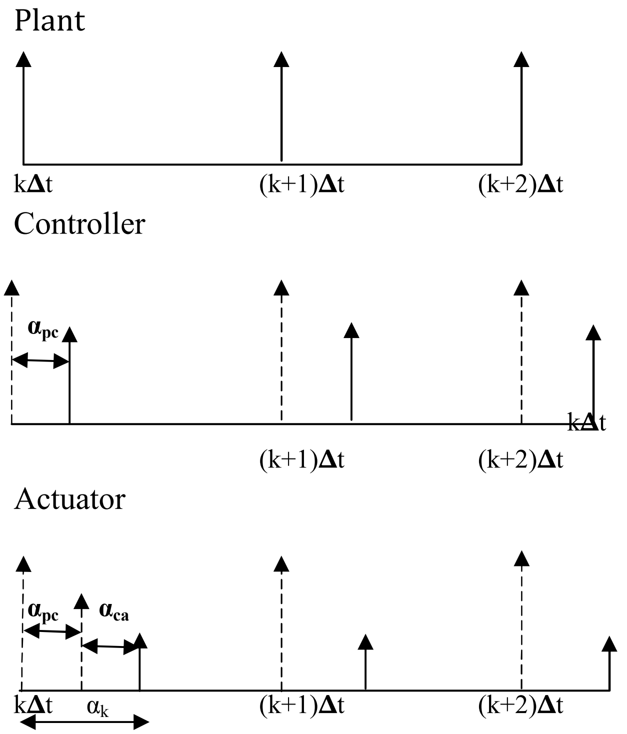


Figure 1. Delay effect for a particular time instant

The delayed response is characterized in two cases for a discretized system for any time instant k and for **Δ**t sampling period. The first case is when the delay magnitude is smaller than the sampling period. The control loop operation is delayed, but tolerable by the system. As is evident from Fig. 1, the total delay is specified as

$$\alpha_k = \alpha_{pc} + \alpha_{ca} \tag{3}$$

where α_{pc} is the network delay from Plant to Controller; α_{ca} is the network delay from the Controller to an Actuator. The maximum delay tolerable by the system, guaranteeing stability, is specified as [5],

$$\frac{d}{dx}(x_p(k+1)\Delta t) = \phi_P x_P[k\Delta t] + \phi_P \alpha_k u_P[k\Delta t] + \gamma_p[\alpha_k]u_p[(k-1)\Delta t] \tag{4}$$

where $\phi_P = \int_0^{\Delta t - \alpha_k} e^{As}\,ds$ and $\gamma_p = \int_{\Delta t - \alpha k}^{\Delta t} e^{As}\,ds$

The maximum tolerable delay can be estimated from the relative magnitudes of ϕ_P and γ_p[5].

The second case arises when the delay magnitude becomes larger than the sampling period. The main consequence for such a case is loss of information or jitter. It is very important to compensate for the delay by controller optimization to stabilize the control loop behaviour. One novel way to handle the communication delay is to estimate the delay magnitude by appropriate stochastic models like ARIMA and then generate forecasts that can be used for controller modelling and delay compensation. The delay magnitude can be compensated by retuning the discrete models with appropriate predictive control techniques such as the Smith Predictor.

IV. Statistical Modelling of Network Delay for Control Networks

Communication network traffic is stochastic in nature, since it arises from multiple independent sources, and is therefore often modelled using statistical approaches. The statistical approaches, specifically time-series models help to generate forecasts that can be used effectively. ARIMA (Autoregressive Integrated Moving Average) [6] is a time series model to capture the behaviour of the network traffic. An ARIMA (p, d, q) is a process where p is the autoregressive order, q is the moving average order and d is the differencing order [6]. Since network traffic always predicts non-stationary behaviour, the most effective modelling technique must consider both autoregressive and moving average terms. The importance of an ARMA processes lies in the fact that a stationary time series may often be adequately modelled by an ARMA model involving fewer parameters than a pure AR or MA processes alone [6]. The general ARIMA (p, d, q) process is of the form [6]

$$W_t = \alpha_1 W_{t-1} + \ldots + \alpha_p W_{t-p} + Z_t + \ldots + \beta_q W_{t-q} \tag{5}$$

or such that $\Phi(B)W_t = \theta(B)Z_t$

where B is the backward shift operator specified as $BX_t = X_{t-1}$, Wt is $(1-B)^d X_t$, ϕ and θ are polynomials of order p and q respectively, and X_t represents the original non-stationary time series with mean zero and variance σ^2.

The non-stationary series is often dealt with using the Box-Jenkins [6] approach. It assumes the parameter d is used for differencing the series to induce stationary behaviour. This approach deals with the periodic component of the time series. Box-Jenkins generalized the ARIMA(p,d,q) model to deal with seasonality, and defined the general multiplicative seasonal ARIMA model as [6][7]

$$\Phi_p(B)\Phi_P(B^s)W_t = \theta_q\Theta_Q(B^s)Z_t \quad (6)$$

where Φ_P and Θ_Q are autoregressive and moving average polynomials of the order P and Q , B^s is specified as $B^sX_t = X_{t-s}$ and s is the seasonal span. The model in equation (6) is a multiplicative seasonal ARIMA model of the order $(p,d,q)*(P,D,Q)_s$ [6][7].

When fitting a seasonal model, the preliminary observation regarding the non-stationary series is to identify the difference (d) and seasonal difference (D) order. Then the values of p, q, P, and Q are determined from the autocorrelation and partial autocorrelation functions of the differenced series. The differencing parameters yield the W_t series which can be modelled using equation (6), to finally produce a fit of a SARIMA (Seasonal ARIMA) model. The fitted model is finally checked for its preciseness i.e. how adequately it represents the original time series data. This step is called the residual analysis. The residuals are often obtained as a difference of actual and fitted values.

For a good model, the residuals are random and have small variance. The most common test used for residual analysis is the Portmanteau lack-of-fit test [6] that uses the chi-square statistic. The test statistic is given as [1]

$$Q = N\sum_{k=1}^{K} r_{z,k}^2 \quad (7)$$

where N is the number of terms in the differenced series, and K lies in the range 15 to 30.

Q is the chi square statistic with (K-p-q) degrees of freedom. One useful application of this model is for making predictions based on the past data values. The time series that shows trend and seasonality in its behaviour is often forecasted using the *model equation [6]*.

Consequently the ARIMA algorithm for traffic prediction can be summarized by the following steps:

1) Model Specification which involves identification of trend and seasonality within the original series (Random Walk Model), finally identifying the differencing order for the trend and seasonal components.
2) Model Building, which involves identifying the Autoregressive (p,P), and Moving Average (q,Q) orders for general and seasonal patterns from the autocorrelation and partial autocorrelation patterns of Original Series.
3) Model Validation, which involves a validation check for residuals obtained after fitting the model with appropriate orders for AR and MA terms.
4) Forecasting, this involves generating the forecasts if the model is validated and the residuals are random and have small variance.

Experiments were performed to capture real-traffic traces to enable prediction of Ethernet traffic carried using the DataSocket Transport Protocol (DSTP) as application layer protocol. DSTP is the TCP based protocol that manages the National Instruments' (NI) DataSocket Server Manager. The DataSocket Server Manager is a data repository for control data. The delays induced by the network were modelled and forecasted using SARIMA model with (p,d,q)*(P,D,Q) order. DSTP uses a publish-subscribe pattern for data exchange. A system participating in a DSTP data exchange usually consists of three components – a publisher, the DataSocket Server and one or more subscribers [8]. A publisher acquires data from a data acquisition device and sends it to the server [8]. The server may be located on the same machine or, remotely on a local network. Subscribers can subscribe to receive the data from the server [8]. Complex applications may have decentralized setup for subscribers and publishers [8]. The interesting factor that determines the data transfer from publisher to subscriber is that publisher broadcasts the entire active control data to a subscriber thus creating lot of network overhead. This can lead to poor scalability of the network as more and more subscribers become involved.

ARIMA modelling was undertaken for one and three subscribers. The response time behaviour induces a seasonal impact in the inter-arrival time gaps. The seasonal component is more obvious in the case of a single subscriber and dies out as more subscribers come into existence. Since the entire control data is broadcasted to the subscribers, the transmission involves fifteen data packets of equal size. The fifteen data packets are transferred within the time range of 10 milliseconds to 150 milliseconds, thus causing seasonal impact on the autocorrelation patterns. The traffic patterns for a period of ten seconds are recorded and autocorrelation graphs for Random Walk models are shown in Fig. 2. As is evident from Fig. 2, the seasonality of the time gaps is obvious in case of one subscriber but is less clearly defined as the number of subscribers increases. The seasonal component is identified at the multiples of time-lag 16, as there are fifteen response data packets transferred to the subscriber

Copyright (c) IARIA, 2010 ISBN: 978-1-61208-101-4

for every request. The time gap for the first response packet is high and decreases for subsequent response packets, thus causing the seasonal impact.

The autocorrelation patterns and partial autocorrelation patterns specify the order required by the SARIMA model. The best fits for both traffic patterns were obtained using Minitab 15 for orders as specified in Table 1. The residual autocorrelations for both models are shown in Fig. 3. As evident from the Fig. 3, the residuals are random and close to zero. Consequently the fitted models were deemed adequate. Forecasts obtained through ARIMA are based on specific assumptions that are assumed to be fixed for a particular set of experimental data.

TABLE 1. FITTED MODELS FOR DSTP TRAFFIC PATTERNS

Subscribers	Model

1 Subscriber	**SARIMA(0,0,1)*(1,1,0)$_{16}$**
	`Type Coef SE Coef`
	`SAR 16 -0.6317 0.0403`
	`MA 1 0.3603 0.0487`
	`Constant 0.0001944 0.0007260`
	`Differencing: 0 regular, 1 seasonal of order 16`
	`Number of observations: Original series 443, after differencing 427`

3 Subscribers	**SARIMA(5,0,0)*(0,1,1)$_{56}$**
	`Type Coef SE Coef`
	`AR 1 0.0147 0.0327`
	`AR 2 -0.0394 0.0327`
	`AR 3 -0.0578 0.0327`
	`AR 4 -0.0379 0.0327`
	`AR 5 -0.0556 0.0328`
	`SMA 56 0.9484 0.0139`
	`Constant 0.00014991 0.00004716`
	`70.90 0.000`
	`Differencing: 0 regular, 1 seasonal of order 56`
	`Number of observations: Original series 994, after differencing 938`

V. DELAY COMPENSATION ALGORITHMS: A-POSTERIOR ANALYSIS

The ARIMA technique helps in modelling and forecasting network delays. The forecasted delay pattern can be used by discrete control algorithms for delay compensation. The control algorithm for the second order water tanks problem was optimized to compensate for delays induced by the network. The control data was managed by National Instruments' DataSocket Server Manager. The delay forecasts generated through ARIMA were utilised to predict communication delays within the control loop. An analytical view of control loop operation was generated in the presence of these forecasted delays using MATLAB/SIMULINK. The control algorithms were implemented using the Time Definition Language (TDL) to guarantee the computation time stability.

The two most common techniques for delay compensation in discrete control design are [4]:

1) The Smith Predictor
2) The Buffered Dahlin Algorithm

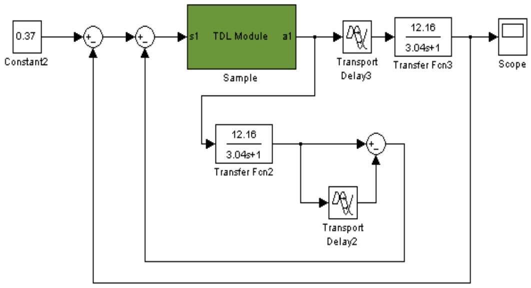
In a closed loop system, the conventional control design strategies allow delay compensation through reduction of controller gain [4]. Consequently the control systems will display sluggish response when compared to a non-delayed system. A better approach is to deal with the delay explicitly and introduce compensation techniques to the allow system to behave like a non-delayed system [4]. The Smith Predictor is a delay compensation algorithm. It assumes that the system is non-delayed and compensates the delay value by introducing a minor loop modelling the non-delayed control loop operation. The Smith Predictor introduces a minor loop that deals with the model process for delay compensation. Thus the error signal after compensation of delay α is

$$e_c = y_c - y^*(s) \tag{8}$$

where y*(s) is process output for the non-delayed control loop operation. The Smith predictor algorithm allows the controller gains for undelayed process to be used without instability arising within the control loop. The closed loop transfer function for the delayed process using a Smith predictor is defined as

$$Y = \left(\frac{g * g_c}{1 + g * g_c}\right) e^{-\alpha s} y_c \tag{9}$$

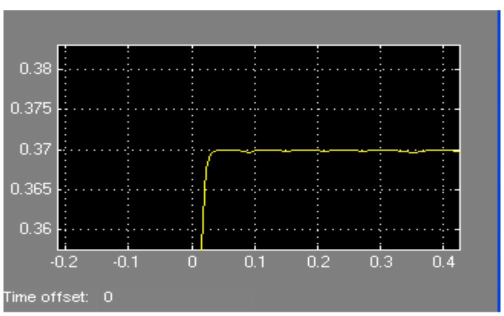
The SIMULINK model for a delayed process using Smith Predictor designed in TDL is shown in Figure 4.



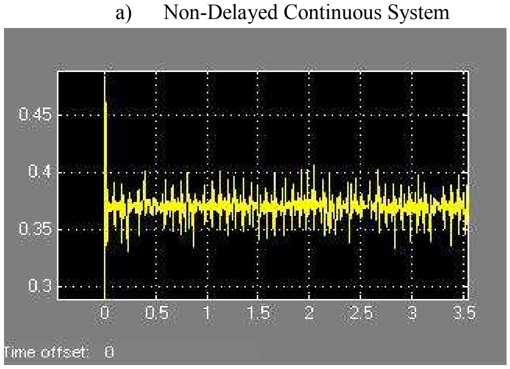
(K_c=0.0833, τ_i=0.0433, $\Delta t_{critical}$=0.009)

Figure 4. Smith Predictor with TDL Controller Optimization

Now the non-delayed process is delayed by the certain amount of time, α, which is assumed to be derived from the ARIMA generated forecasts. The delay value is assumed to have average, best and the worst case values of 60ms, 10ms, and 255ms respectively.



a) Non-Delayed Continuous System



b) Delay Magnitude is 255ms

Figure 5. Control Loop Stability in the Presence of Delay

As is evident from Fig. 5, the communication delays, if greater than the sampling period, can destabilize the system. The discretized system may be compensated for delay using a Smith Predictor. The Smith Predictor is able to compensate the delay and induce stability. But it has certain disadvantages. It assumes the magnitude of the delay is constant, which is not with network communication delays. Network delays are more random and non-stationary over time.

The application of this algorithm must be restricted to accommodate the worst-case delay value which has a rare probability of occurrence. This conventional design strategy does not provide the controller type and parameters. Thus it can act as a major loophole for stability in the discrete-time domain. A realistic system with time delay is inherently unable to respond instantaneously to a control event. The high controller gains fail to provide a realistic image of the process. The remedy to this problem is to design the controller with parameters found using the direct synthesis control strategy with a reference trajectory that can provide desired control loop behaviour. Fig. 6 shows the delayed system behaviour using a Smith Predictor with Direct Synthesis Controller Parameters. The PI Controller used for the α delay in the system is realizable as

$$g_c = \frac{\tau s+1}{K}\left(\frac{1}{\tau_r s+1-e^{-\alpha s}}\right) \quad [4] \qquad (10)$$

The choice of τ_r allows the system to have a more realistic view of the process behaviour in the presence of time delay.

Discrete Model (K_c=0.25 τ_i=0.0822 τ_r=1)

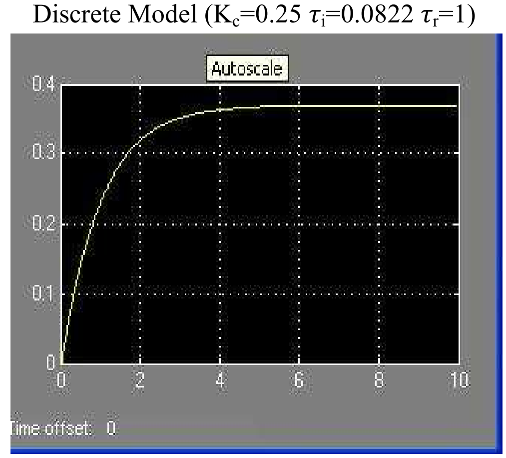


Figure 6. Smith Predictor with Direct Synthesis Control

Another useful strategy for controlling time-delayed systems is the Dahlin Algorithm. It is a digital control design technique that assumes the controller realization including the ZOH (zero order hold) and explicit time delay is [4]:

Copyright (c) IARIA, 2010

$$g_c(z) = \frac{1}{g(z)}\left[\frac{(1-e^{-\varphi_r})z^{-m-1}}{1-e^{-\varphi_r}-(1-e^{-\varphi_r})z^{-m-1}}\right] \qquad (11)$$

where φ_r is the assumed reference trajectory for the desired control loop behaviour. G(z) is the delayed plant model with **Δ**t sampling period such that the time delay α = m **Δ**t. The Dahlin algorithm is a digital control algorithm having two tuning parameters:

1) M, chosen such that the controller is realizable in the presence of delay
2) φ_r determines the speed of the closed loop response.

The forecasted delay pattern follows an ARIMA based model. The model equation can be used to predict the delay for control loop. The scheme uses buffered approach to save the preceding timestamps for the next prediction. The Dahlin strategy is simulated with buffered timestamps for indicating the magnitude of delays using TDL/SIMULINK with sampling time **Δ**t ranging from 0.004 to 0.09 minutes. The plant model is assumed to have the delay specified as ranging from α. = 100 to 255 milliseconds.

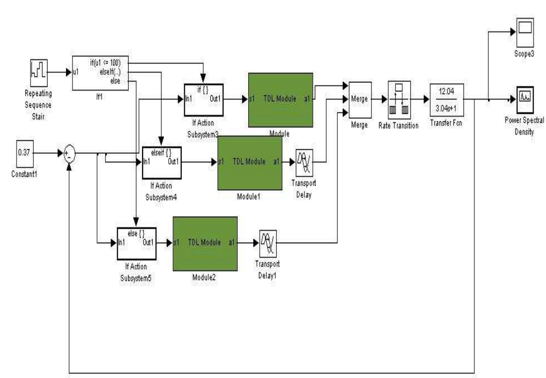
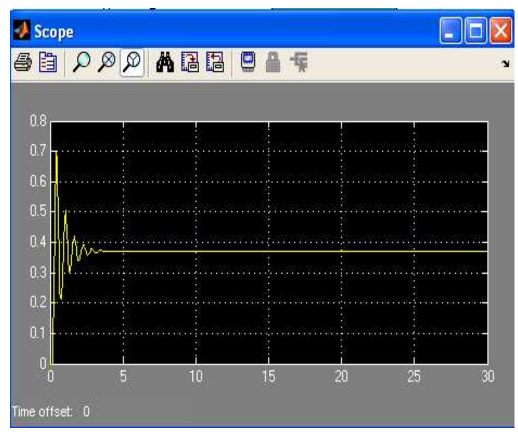


Figure 7. TDL/Simulink Model for Delay Compensation

Listing 1. Buffered Dahlin Algorithm

Repeat
// For any time instant k
τ_{k-1} = Buffer(k), τ_{k-2} = Buffer(k-1) // Get the timestamp for the //preceding delay
τ_k = ARIMA fun(τ_{k-1}, τ_{k-2}) // Next Prediction
Read h(k) // Current Process Variable
e (k) = sp – h(k)
//Random delay block
// Introduce the Computational Delay
//g_c(e(k) optimized for delay τ_k with m units
d = Δt/ τ_k
compute f(k) = g_c(e(k), τ_k , d)
//Manipulated Variable
write f(k)
compute h(k+1) = G_p(f(k))
// New Error Signal
Until (Set point)

The TDL/SIMULINK model behaviour is shown in Fig. 7 and the response is analysed in Fig. 8.



Kc = 53.933, I = 18.6125 for Delays less than 100ms
Kc = 5.7421, I = 1.8869 for 100< Delays < 200ms
Kc = 2.2734, I = 0.9243 for 200<Delays<300

Figure 8. Dahlin Strategy for Variable Network Delay Compensation

As evident from the discrete implementations of the Smith Predictor and Dahlin Algorithm from Fig. 7 and Fig. 8, stability is achieved but the response time is high in comparison to the continuous system response time. These algorithms are implemented in TDL to guarantee computation time stability, thus providing the precise analytical impact of stochastic network delays on control loop stability.

VI. CONCLUSION

The response times and TDL computation times are summarized in Table 2. These comparative estimates of response times are for the second order system with time constant of 3 minutes. In the first case, the network delays are modelled using the stochastic approach, ARIMA. The forecasts are then utilized to view their impact on control loop operation. The discrete implementation of control algorithm uses TDL to guarantee the computation time stability. RT-Languages such as TDL can help in analysing and guaranteeing the computational stability.

Finally, the unpredictability of communication delays can be dealt with using appropriate discrete algorithm implementations. As is evident from the Table 2, the discrete implementations are able to compensate for the delay, thus guaranteeing the stability of the control loop in the presence of unpredictable delays. However, in practice, the response time maybe high. Secondly, in both strategies the delays are treated as a constant and varied through average, best and worst case values.

TABLE 2. RESPONSE TIME ESTIMATES IN TDL/SIMULINK

Non-Delayed Continuous System		**Smith Predictor**		**Dahlin Algorithm**	
TDL Computation Time(ms)	*Response Time(sec)*	*TDL Computation Time(ms)*	*Response Time(sec)*	*TDL Computation Time(ms)*	*Response Time(sec)*
20	2.4	40	5	40	240
Reference Trajectory (3 Minutes) without Computation time stability		-	230	-	480
Controller Optimization through TDL		40	150	40	350

REFERENCES

[1] C.M. Kirsch and R. Sengupta, “The Evolution of Real-Time Programming”, in Handbook of Real-Time Systems and Embedded Systems by I. Lee, J. Y-T. Leung, S.H. Son, 2007.

[2] T.A. Henzinger, C.M. Kirsch, and B. Horowitz, “GIOTTO : A Time-Triggered Language for Embedded Programming”, in T.A. Henzinger, C.M. Kirsch, Editors, Proc. of the 1st International Workshop on Embedded Software (EMSOFT), number. 2211 in LNCS, Springer, Berlin, Oct 2001

[3] T.A. Henzinger, C.M. Kirsch, M.A.A. Sanvido, and W. Pree, “From Control Models to Real-Time Code using GIOTTO”, IEEE Control Systems Magazine, Feb 2003.

[4] B. A. Ogunnaike and W. H. Ray, “Process Dynamics, Modelling, and Control”, Oxford University Press, 1994.

[5] J. Nilsson, B. Bernhardsson, and B. Wittenmark, “Stochastic Analysis and Control of Real-Time Systems with Random Time Delays”, Automatica, Vol. 34 Issue 1, Publication Year 1998, pp. 57-64.

[6] C. Chatfield, “The Analysis of Time Series An Introduction”, Chapman and Hall/CRC, 2003.

[7] W. Vandaele, “Applied Time Series and Box-Jenkins Models”, Academic Pree, INC, 1983.

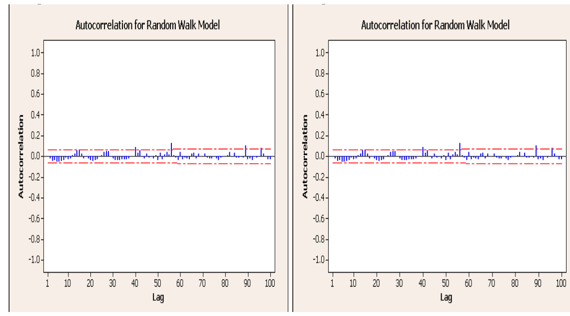
[8] “DataSocket Transport Protocol- Overview”, NI Developer Zone. URL- http://zone.ni.com/devzone/cda/tut/p/id/3223 accessed date: 21st May 2009.

[9] A. Ray, “Distributed Data Communication Networks for Real-Time Process Control”, Chem. Eng. Commun., vol 65, pp. 139-154,1998.

[10] M. S. Mahmoud and A. Ismail, “Role of Delays in Networked Control Systems”, In Proc. of IEEE ICECS-2003.

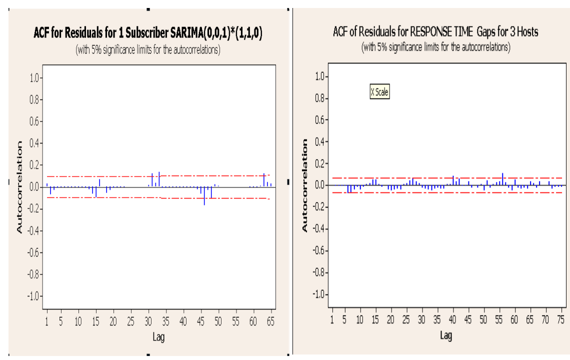
[11] Z. Wei, M.S. Branicky, and S.M. Phillips, “Stability of Networked Control Systems”, IEEE Control System Magazine, vol. 21, no. 1, pp. 84-99, 2001.

[12] Ray, Y and Halevi, “Integrated Communication and Control Systems: Part II Design Considerations”, ASME Journal of Dynamic Systems, Measurement and Control, vol. 110, pp. 374-381, 1988.



1 Subscriber 3 Subscribers

Figure 2. Autocorrelation for Original Series



1 Subscriber 3 Subscribers

Figure 3. Residual Autocorrelations for Fitted Models