International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 1 Issue 6, August - 2012

Implementation & Performance Analysis of Real Time Kernel for design of distributed control network using True Time Simulator

Mr. Vishal S. Vora[1], Dr. Ajay K. Somkuwar[2]
Dr. K.N. Modi University Rajasthan, India[1], M.A.N.I.T. Bhopal, India[2]

Abstract

Designing of Real Time Kernel for particular embedded applications is essential. Out of various applications here we design it for distributed control network which in this case consists of four computer nodes, time driven sensor node, actuator node & wired network. The designing of network is taken care by various interferences sources. We also disturb traffic parameters over the network to measure the control performance. Our kernel design will promise to have better performance against known disturbing and interference sources compare to current kernel design. TrueTime simulator is used along with Matlab software for realization of the various design issue.

1. Introduction

Embedded systems are considered to be used for major applications like Automation, Automobile, Distributed Networks, Consumer Electronics, and Home Appliances, etc. As we know Major components of any embedded system is Software, Hardware & RTOS (Real Time Operating System). Real time Kernel is the crucial component any RTOS. The real-time implementation of a control law is sometimes a difficult task because of the real-time environment that was not taken into account. The design of the control law is the control engineer task and his real time implementation is the job of the software engineer. Real-time control systems are becoming increasingly complex systems, from both control and computer science perspectives. Today, even a seemingly simple embedded control system often contains a multitasking real-time kernel and supports networking. The TrueTime Library is a Simulink library extension that gives us the possibility to simulate the behaviour of a control law in the presence of the dynamic model of the plant by taking into account the physical delays induced by the computational time of the effective command and by the network that implement the distributed control. Using TrueTime extension, we can simulate more realistically the real time control system behaviour. The TrueTime MATLAB/Simulink real-time extension gives the possibility to a system engineer to deal with both design and real time implementation of the control law. Also, using TrueTime toolbox we are able to simulate distributed control systems by using a general (Ethernet) or an industrial (CAN or FlexRay) network strategy. The TrueTime library contains two objects that allow us to create in a very convenient way our S-functions for using in real-time simulation. We shall emphasize the importance of a new strategy that considers both classical control design and the real time implementation of a networked distributed control.

2. Our distributed network model

Our model consists of distributed network which connect to four computer nodes. This Design contains four computer nodes, each represented by a TrueTime kernel block. A time-driven sensor node samples the process periodically and sends the samples over the network to the controller node. The control task in this node calculates the control signal and sends the result to the actuator node, where it is subsequently actuated. The simulation also involves an interfering node sending disturbing traffic over the network, and a disturbing high-priority task executing in the controller node. The basic structure is shown in below figure 1. As shown in below diagram the main fixed network is connected to various four compute nodes thorough DC Servo as an intermediate between sensor and actuator

This design considers simple PID control of a DC-servo process, and is intended to give a basic introduction to the TrueTime simulation environment. The process is controlled by a controller task implemented in a TrueTime kernel block. Four different implementations of the controller task are provided to show different ways to implement periodic activities. The DC-servo is described by the continuous-time transfer function can be written as:

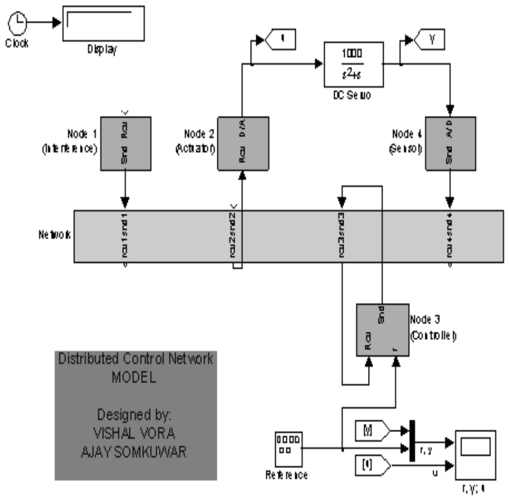


Figure 1. Basic Structure of Model

$$G(s) = 1000/s(s+1) \tag{1}$$

Network is considered as a real time kernel whose structure is shown as below:

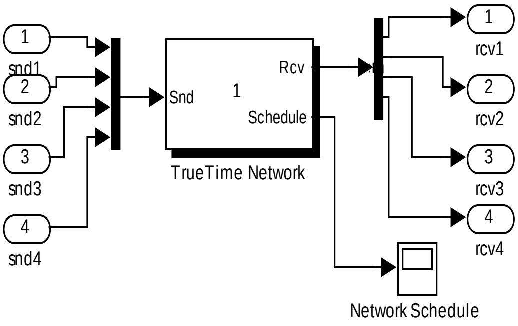
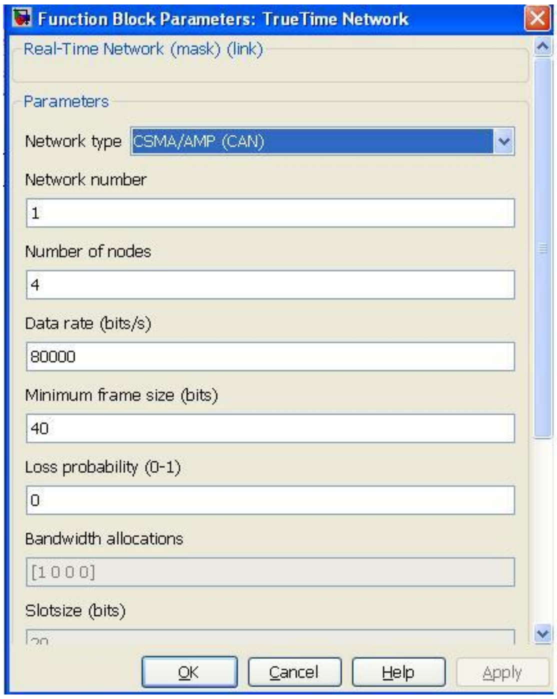


Figure 2(a). True Time Network

Such a network has true time kernel as its prime network and network scheduler in association with it organize & schedule the parameters of the network. Parameters of true time network is shown in fig. 2(b) which allow user to decides the no. of nodes, data rate, frame size, bandwidth allocation with slot & frame size.

Node-1 is considered as an interference node which taken digital signal along with interrupt as an input. Node-2 consider as actuator node which is connected to true time kernel for digital to analogue conversion and with scheduler for real time scheduling. Node-3 consider as Controller which is programming & designing tool for real time kernel which decides characteristics of the network. While Node-4 is consider as Sensor node for true time kernel design.

A time-driven sensor node samples the process periodically and sends the samples over the network to the controller node. The control task in this node calculates the control signal and sends the result to the actuator node, where it is subsequently actuated. The simulation also involves an interfering node sending disturbing traffic over the network, and a disturbing high-priority task executing in the controller node


Function Block Parameters: TrueTime Network

Real-Time Network (mask) (link)

Parameters

Network type: CSMA/AMP (CAN)

Network number: 1

Number of nodes: 4

Data rate (bits/s): 80000

Minimum frame size (bits): 40

Loss probability (0-1): 0

Bandwidth allocations: [1 0 0 0]

Slotsize (bits):

Figure 2(b). Parameters of True Time Network

Basic structures of all four nodes are shown in fig 3, 4, 5 & 6 respectively

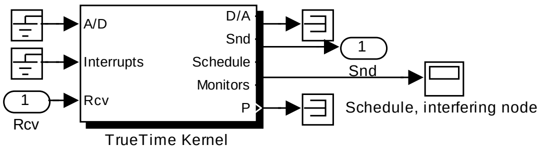


Figure 3. Node 1- Interference

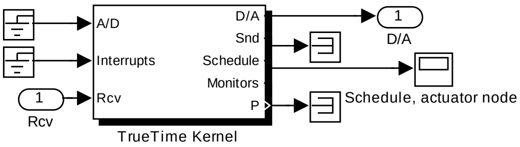


Figure 4. Node 2-Actuator

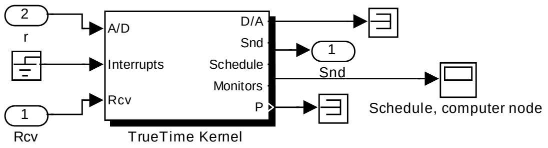


Figure 5. Node 3-Controller

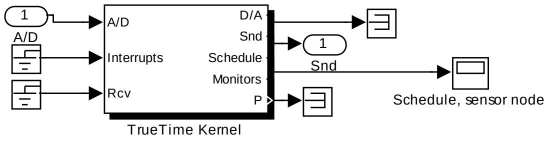


Figure 6. Node 4-Sensor

The DC servo is designed as per its transfer function shown in equation 1. Parameters like No. of coefficients in numerator and denominator with its absolute tolerance are important for design of the system. The basic window for configure its parameters are shown in figure 7.

3. Performance & Output analysis

Three different output terminals are come from basic true time kernel design block. From node-2 i.e. Actuator's output terminal is x, while node-3 i.e. Controller's output terminal is r while node-4 i.e. sensor's output terminal is designated as y. The output of all these there terminals is shown in figure 8.

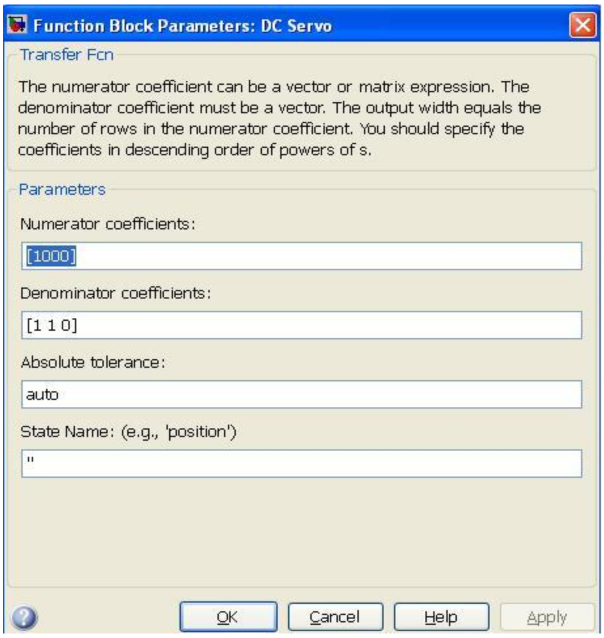
3.1 Procedure for analysis

The procedure using Matlab simulink along with true time toolbox as per below will achieve the desired result. It can be summarized as:

Study the initialization scripts and code functions for the different nodes.

The event-driven nodes contain interrupt handlers, which are activated as messages arrive over the network. The handler then triggers the task that will read and process the message Run a first simulation without disturbing traffic and without interference in the controller node. This is obtained by setting the variable BW share in the code function of the interfering node. In this case we will get a constant round-trip delay and satisfactory control performance. Study the network schedule (high=sending, medium=waiting, low=idle) and the resulting control performance.

Switch on the disturbing node and the interfering task in the controller node. Set the variable BW share to the percentage of the network bandwidth to be used by the disturbing node. Again study the network schedule and the resulting control performance. Experiment with different network protocols and different scheduling policies in the controller node.



Transfer Fcn

The numerator coefficient can be a vector or matrix expression. The denominator coefficient must be a vector. The output width equals the number of rows in the numerator coefficient. You should specify the coefficients in descending order of powers of s.

Parameters

Numerator coefficients:

[1000]

Denominator coefficients:

[1 1 0]

Absolute tolerance:

auto

State Name: (e.g., 'position')

''

Figure 7. DC Servo Parameters

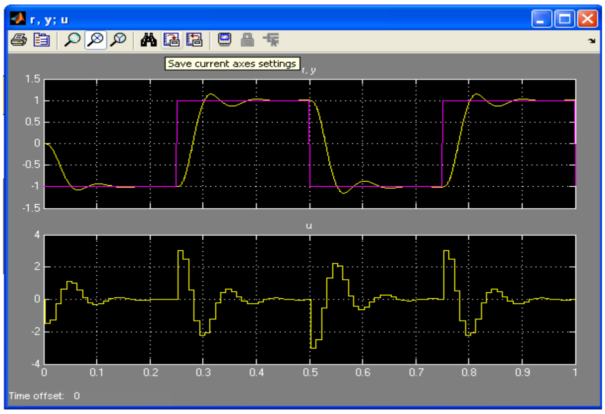


Figure 8. Output of r, x & y nodes

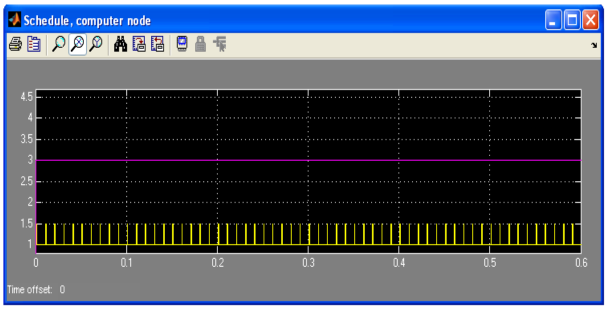


Figure 9. Computer/Controller node

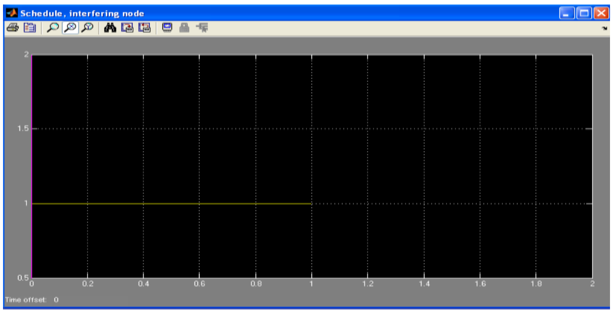


Figure 10. Interfering node

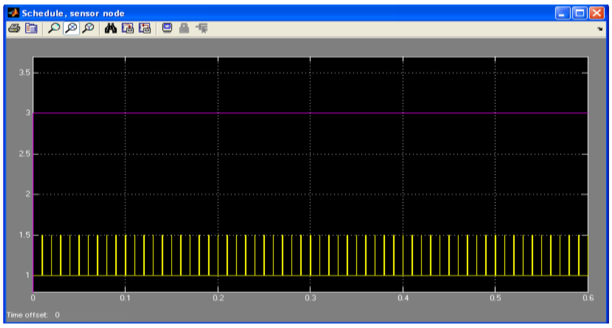


Figure 11. Actuator/Sensor node

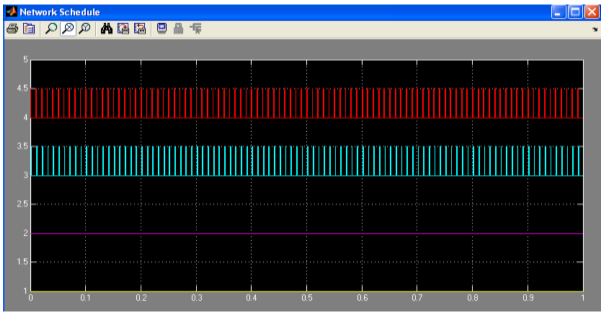


Figure 12. Network Schedule

Above figures shows the output at various nodes of the distributed network. Computer or Controller is the heart of this distributed network whose pattern shown in figure 9. While interfering and Actuator/Sensor node spectrum shown in figure 10 & 11 respectively. The network schedule is shows its scheduling pattern depending on this terminals and it is shown in figure 12.

4. Conclusion

Designing a real-time control system is essentially a co design problem. Choices made in the real-time design will affect the control design and vice versa. For instance, deciding on a particular network protocol will give rise to certain delay distributions which must be taken into account in the controller design. On the other hand, bandwidth requirements in the control loops will influence the choice of CPU and network speed.

Using an analysis tool such as TRUETIME it is possible to quickly assert how sensitive the control loop is to slow sampling rates, delay, jitter, and other timing problems. Aided by this information, it is then possible to proceed with more detailed, system wide real-time and control design using a simulation tool such as TRUETIME. It allows the user to compute a quadratic performance criterion for a linear control system under various timing conditions. The control system is described using a number of continuous time and discrete time linear systems.

A stochastic timing model with random delays is used to describe the execution of the system. The tool can also be used to investigate for instance aperiodic controllers, multi-rate controllers, and jitter compensating controllers. TRUETIME facilitates event based co simulation of a multitasking

Real-time kernel containing controller tasks and the continuous dynamics of controlled plants. The simulations capture the true, timely behaviour of real-time controller tasks and communication networks, and dynamic control and scheduling strategies can be evaluated from a control performance

5. References

[1] Martin Ohin, Dan Henriksson and Anton Cervin, True Time 1.5 -Reference Manual, Lund University, Jan 2007.

[2] Andersson, M., D. Henriksson, A. Cervin, and K.-E. Årzén, Simulation of wireless networked control systems, In Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference ECC 2005. Seville, Spain.

[3] Åström, K. J. and T. Hägglund (1995): PID Controllers: Theory, Design, and Tuning. Instrument Society of America, Research Triangle Park, North Carolina.

[4] Bollella, G., B. Brosgol, P. Dibble, S. Furr, J. Gosling, D. Hardin, and M. Turnbull ,The Real-Time Specification for Java, Addison-Wesley.

[5] Cervin, A., D. Henriksson, B. Lincoln, J. Eker, and K.-E. Årzén, how does control timing affect performance? IEEE Control Systems Magazine, 23:3, pp. 16–30.