

REAL TIME COMPUTER CONTROL OF DISTRIBUTED CONTROL SYSTEMS USING TRUE TIME TOOLBOX IN MATLAB

Kondabolu Nishanth and Pendyala sai sundara sashank*

Department of ECE, GITAM University, Visakhapatnam, India

*Corresponding author: Email: sai.sashank93@gmail.com, Tel: +91-8885243731

[Received-14/09/2013, Published-15/02/2014]

ABSTRACT:

This paper proposes an application based on the architecture targeting Distributed embedded control systems (DECS) with a single CPU and limited computational sources. The application provides an approach for integration of hard and critically hard control mechanisms with truetime toolbox in matlab. Our control task consists of target task to be controlled and adaptation task that implement's control algorithms. This paper compares the time sharing concurrency and efficiency of real time operating systems like rate monolithic and earliest deadline first process. The solution for the process is simulated in matlab/Simulink

Keywords: distributed control, embedded systems, true time toolbox, RTOS, Rate monolithic(R M) scheduling, Earliest deadline first (EDF) scheduling.

[I] NTRODUCTION

Real time computer control is an implementation of a controller using a digital computer in real time [1]. A real time computer control is necessary in order to bridge the gap between controller and external OS environment. Currently there is a trend to incorporate distributed embedded system technology in order to reduce the cost and development time [2]

1.1. Distributed Embedded System

A Distributed Embedded System consists of several control tasks which are either related or mutually

exclusive and are implemented using a single CPU and an RTOS [3]

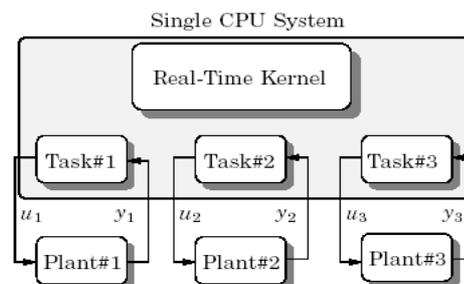


Fig 1:Layout of distributed embedded system

These systems are often designed with ad hoc resource allocation strategies

1.2. Challenges Involved

One of the basic challenges involved in the embedded system design is develop a software for systems used in different fields of application such as control system, chemical, biological, automation field. The second most important factor is determination of Quality of service (QOS) with limited amount of memory [4]

1.3. Previous Work

Problem of distributed embedded control systems can be easily understood by considering synchronous and predictable environments. The problem arises when hard and soft control applications are in the same environment [4]

Feasibility applications must ensure sound scheduling and analysis techniques (Audley et al., 1993) allows checking if the task will meet deadlines, a required condition for real time applications

Fault tolerances can be reduced by coordinating a group of processes replicating at different nodes. It explains two types of processes primary backup or (Passive), active process (Powell,1991).[5]

The present work involves real time implementation of embedded control systems .It includes

- Mathematical modeling of physical systems
 - Implementing a single control law in real time
 - Implementing distributed control laws in RTOS using existing scheduling techniques
- The three tasks involved in the system design are
- D.C Motor (Linear and Hard system)
 - Simple Inverted Pendulum (Nonlinear and Hard system)
 - Inverted pendulum on cart (Highly unstable and critically hard system)

[II] MATHAMATICAL MODELLING OF CONTROL SYSTEMS

2.1. DC Motor

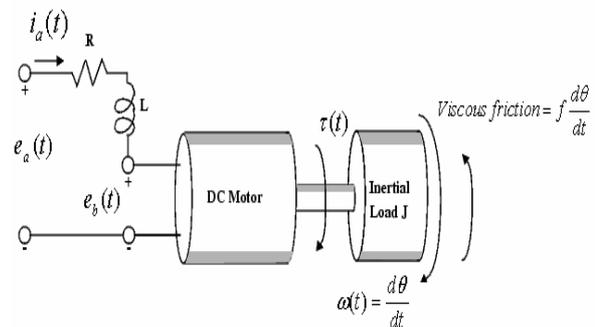


Fig2: DC Motor with Inertial Load

A dc motor consists of two sub processes

- (1) Electrical
- (2) Mechanical

The electrical sub process consists of armature resistance, armature inductance and magnetic flux of the stator. The second sub process is a mechanical and mainly consists of an inertia (J) .[6]

The Mathematical equations can be derived as

$$di_a(t)/dt = (e_a(t)/L) - (R/L i_a(t)) - K_b/L d\theta/dt \tag{2.1}$$

$$d^2\theta/dt^2 = KT/J i_a(t) - f/J d\theta/dt \tag{2.2}$$

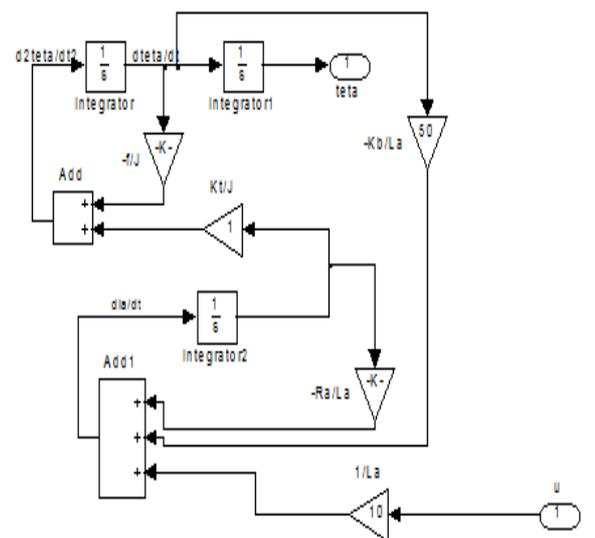


Fig3 : Simulink model for a DC motor

2.2.Simple Inverted Pendulum

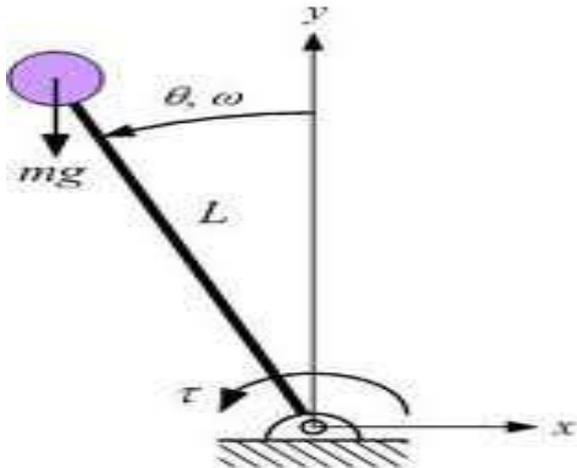


Fig 4: simple inverted pendulum

An inverted pendulum is a pendulum whose center of mass is above the pivot element. Here the pivot point is fixed and is similar to that of an inverted pendulum. [7]

The free body equations can be formulated as

$$\dot{\theta}' = -\frac{g}{l} \sin(\theta) + \frac{u}{ml^2} \tag{2.3}$$

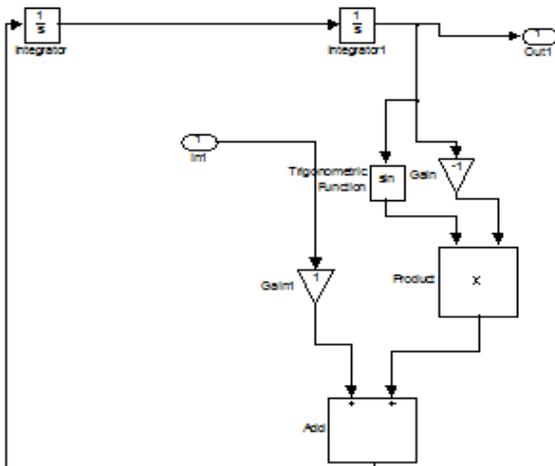


Fig5: Simulink model for simple inverted pendulum

2.3.Simple Inverted Pendulum on Cart

Here the cart is restricted to linear motion and is subjected to forces of motion. The free body

diagram of the system is used to represent the state equations [8]

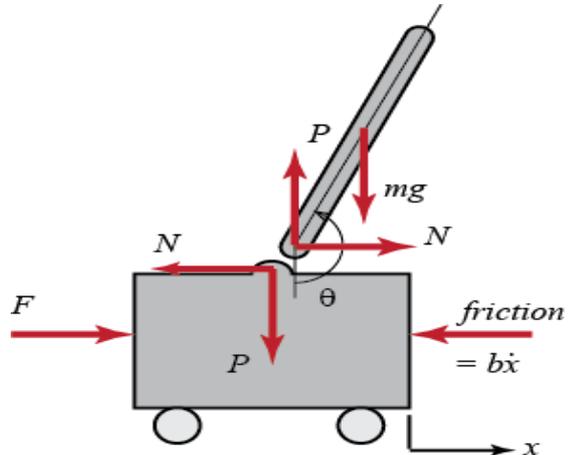


Fig6: Simple pendulum on cart

The state equations can be derived as

$$(M + m)x + bx + ml\theta \cos\theta - ml\theta \sin\theta = F \tag{2.4}$$

$$(I + ml) \ddot{\theta} + mgl \sin\theta = ml\ddot{x} \cos\theta \tag{2.5}$$

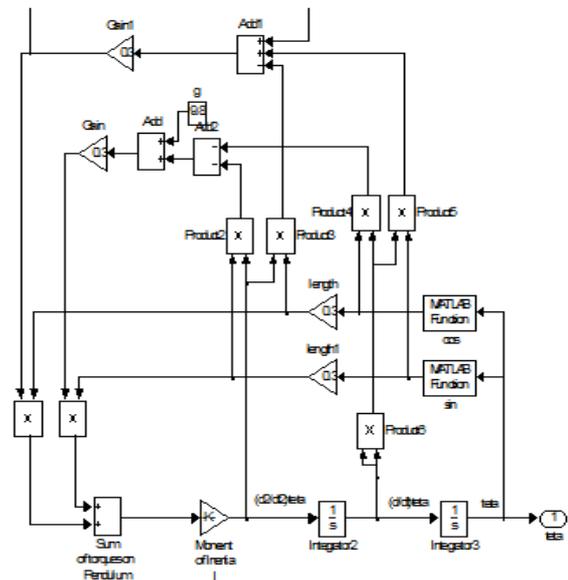


Fig7: Simulink model for simple pendulum on cart

[III] REAL TIME SCHEDULING METHODS FOR CONTROL SYSTEMS

The time sharing concurrency of real time operating system schedules tasks for CPU at a

time priority. Scheduling is a task or a dataflow for effective utilization of resources .A real time scheduler mainly concerns with 1) Throughput
2)Latency

3.1. Types Of scheduling

There are many types of scheduling techniques[9] .important of them are

- Rate- Monolithic (RM) scheduling
- Earliest Deadline first (EDF) scheduling.

3.2. (RM) Rate Monolithic scheduling

In Rate-Monolithic (RM) type of scheduling tasks have a fixed priority and high priority in a given task of performance. This is a static process and the priority cannot be changed. It assigns priority according to the request rates, depending upon the runtimes [10]

3.2.2. Processor Utilization Factor(U)

A fraction of processor time used for execution of a task set is called as processor utilization factor .

For ‘n’ tasks to be performed the utilization factor can be given by

$$U = \sum Ci / Ti \quad (i=1 \text{ to } n)$$

3.3. (EDF) Earliest Dead line scheduling

In earliest deadline first(EDF) algorithm the processor runs the process which is closest to its deadline. It has a property of dynamic priority scheduling [11]

3.2.3. Processor Utilization Factor(U)

For ‘n’ number of tasks to be performed the processor utilization factor can be given by

$$U = \sum Cn / Tn \leq 1$$

If this condition is satisfied then the event has an highest priority among the given set of tasks

3.4. Comparison Between (EDF) Earliest Dead line scheduling and (RM) Rate monolithic scheduling

An EDF system can use higher processor utilization when compared to RM. EDF has higher, simple and exact analysis when compared to Rate monotonic. The speed of the processor can

be defined by type of operating system used, priorities etc..

[IV] TRUE TIME TOOLBOX

True time toolbox is a matlab toolbox which creates a RTOS environment. The form of coding involved in this toolbox is by C++ platform. Thus it is very useful to use this toolbox for real-time applications[12] The main idea of true time is

- Co-Simulation of controller task execution
- Network Transmissions
- Continuous plant Dynamics

Here the task is accomplished by providing models of real-time networks as Simulink blocks, here the user code is in the form of tasks and interrupt handlers is modeled by Matlab or C code [13]

4.1. True Time toolbox main Idea

The main ideas of true time toolbox are

- Co-Simulation of controller task execution, network transmissions and continuous plant dynamics
- Accomplished by providing models of real-time kernels’ and networks as Simulink blocks
- User code is in the form of tasks and interrupt handlers are modeled by matlab code or C code

4.2. Main Components of True Time toolbox

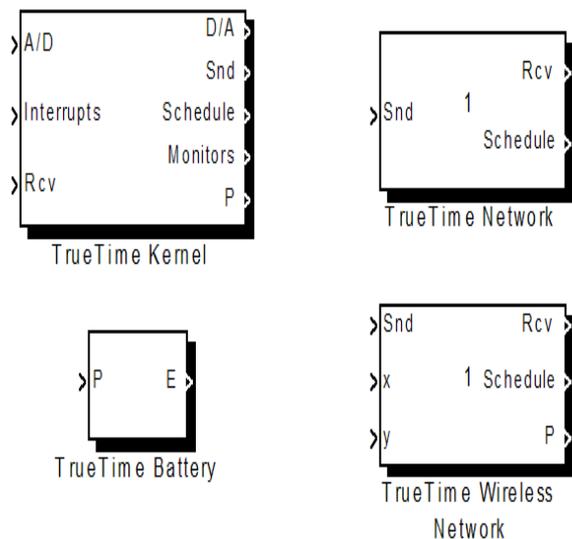


Fig8: Components of True time toolbox

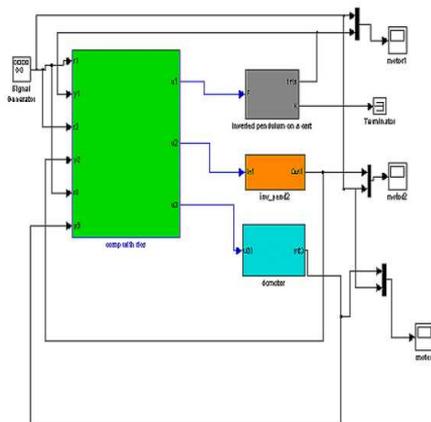


Fig9: Distributed control system simulation

[IV] SIMULATION RESULTS

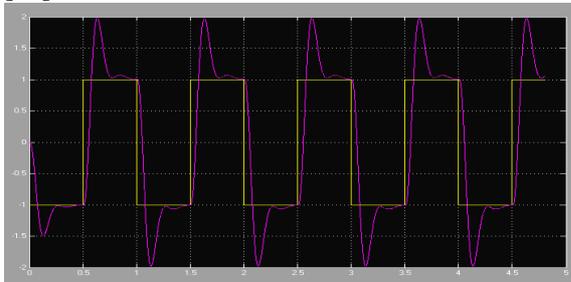


Fig10 : Scope of a DC motor Simulated in Simulink

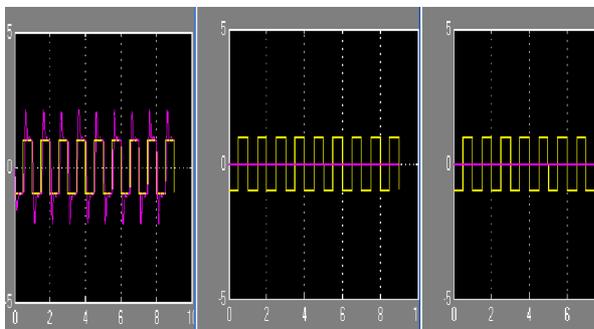


Fig11: Three DC motors Scheduled in Sche (RM)

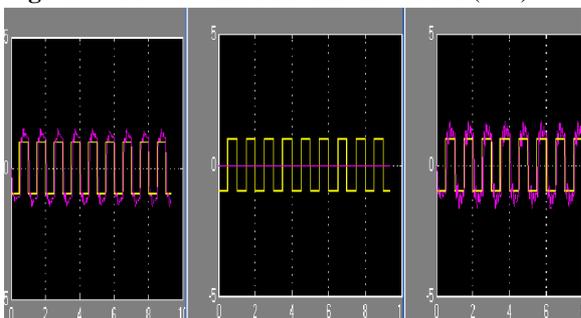


Fig12: Three DC motors Scheduled in EDF

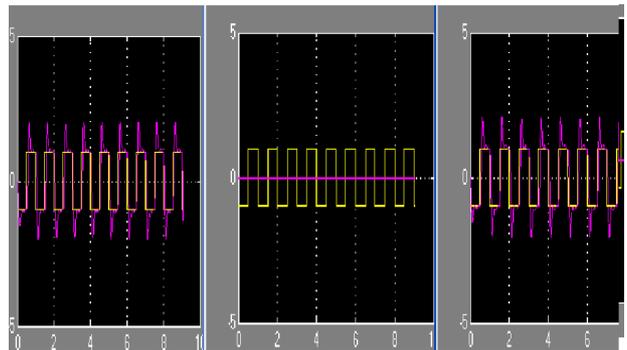


Fig13 :Three DC tasks Scheduled inRM

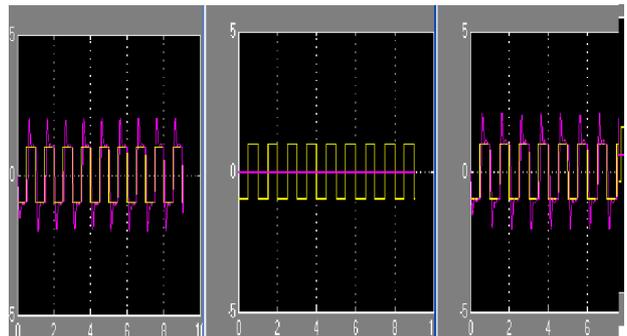


Fig14 : Three DC Tasks scheduled in EDF

[IV] CONCLUSIONS

A real time computer controlled control system makes use of the efficient utilization of processor . It has been proved that out of the comparison the task done in rate monolithic scheduling has better performance when compared to earliest deadline first scheduling. It has been proved that multiplexing of tasks in distributed embedded control system yields better results than a processor with a single task

REFERENCES

- [1] Nilsson, Johan. *Real-time control systems with delays*. Diss. Ph. D. dissertation, Department of Automatic Control, Lund Institute of Technology, 1998.
- [2] Khatib, Oussama. "Real-time obstacle avoidance for manipulators and mobile robots." *The international journal of robotics research* 5.1 (1986): 90-98.
- [3] Yen, Ti-Yen, and Wayne Wolf. "Communication synthesis for distributed embedded systems." *Computer-Aided Design, 1995. ICCAD-95. Digest of Technical Papers., 1995 IEEE/ACM International Conference on*. IEEE, 1995.

- [4] Ka, Aloysius, and Lau Mok. *Fundamental design problems of distributed systems for the hard-real-time environment*. Vol. 1. No. 3. MIT Thesis, May, 1983.
- [5] Kopetz, Hermann. *Real-time systems: design principles for distributed embedded applications*. Springer, 2011.
- [6] Kuo, Benjamin C., and M. Farid Golnaraghi. *Automatic control systems*. Vol. 4. New York: John Wiley & Sons, 2003.
- [7] Wang, Hua O., Kazuo Tanaka, and Michael F. Griffin. "An approach to fuzzy control of nonlinear systems: stability and design issues." *Fuzzy Systems, IEEE Transactions on* 4.1 (1996): 14-23.
- [8] Pathak, Kaustubh, Jaume Franch, and Sunil Kumar Agrawal. "Velocity and position control of a wheeled inverted pendulum by partial feedback linearization." *Robotics, IEEE Transactions on* 21.3 (2005): 505-513.
- [9] Audsley, Neil C., et al. "Real-Time scheduling: the deadline-monotonic approach." in *Proc. IEEE Workshop on Real-Time Operating Systems and Software*. 1991.
- [10] Sarkar, Vivek. *Partitioning and scheduling parallel programs for multiprocessors*. MIT press, 1989.
- [11] Audsley, Neil C., et al. "Real-Time scheduling: the deadline-monotonic approach." in *Proc. IEEE Workshop on Real-Time Operating Systems and Software*. 1991.
- [12] Anton, Cervin. "TrueTime: Real-time control system simulation with MATLAB/Simulink." *Proceedings of the Nordic Matlab conference*. 2003.
- [13] Kim, Jun H., et al. "Tracking real-time user experience (TRUE): a comprehensive instrumentation solution for complex systems." *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, 2008.