Proceedings of the 13th International Conference on Automation and Computing, 15 September 2007, Staffordshire University, Stafford, UK

Interactive co-simulation of MATLAB and OPNET for Networked Control Systems

M Shahidul Hasan (PhD student), Hongnian Yu, Alison Griffiths and T C Yang*
Faculty of Computing, Engineering and Technology, Staffordshire University, UK
*Department of Engineering and Design, University of Sussex, UK
{m.s.hasan, h.yu, a.l.griffiths}@staffs.ac.uk, *t.c.yang@sussex.ac.uk

***Abstract*— Matlab and OPNET are widely used by researchers to simulate the mathematical models of systems and computer networks, respectively. This paper presents the interface mechanism between these two simulation packages for networked control systems where Matlab and OPNET model the plant/controller and communication network, respectively. The paper applies the interactive co-simulation to a double pendulum with two sensors and actuators plant. Mobile ad-hoc network (MANET) has been considered as the communication network between the plant and the PID controller.**

Keywords- Interactive co-simulation, Networked Control Systems, OPNET.

I. INTRODUCTION

Matlab/Simulink is a very powerful tool for modelling systems behaviour. However, it has limitations in simulating computer networks. On the other hand, OPtimised Network Engineering Tool (OPNET) allows more detailed communication network simulation. Many aspects of the network such as the number of nodes, network data rate, node movement etc. can be specified in OPNET. But, OPNET uses Proto-C language which is similar to the C programming language. It is a tedious and time consuming task to implement the systems behaviour using Proto-C language. Therefore, combining the strengths of Matlab and OPNET will unleash more realistic simulation results in the research field of networked controlled systems (NCS). This paper investigates the interactive co-simulation of Matlab and OPNET where these two simulators exchange data interactively as the simulation progresses.

TrueTime [1], [2] is a Matlab-Simulink based toolbox that allows performance evaluation of multitasking real time kernel executing various tasks with network support. TrueTime includes support for both wired and wireless network protocols. However, the network blocks have limited support for detailed network simulation. Simulation of wireless NCS using TrueTime is paper [3]. The interface between Matlab and OPNET has been considered in [4]. A co-simulation of control and network, implemented in Matlab-Simulink, is presented in [5]. The authors investigated NCS performance for various data rate, traffic, load etc.

Many research works consider offline co-simulation where the output data from one simulation package is stored in a file. Then the other simulation package reads the data file and generates the final results. Unlike those papers, under this interactive co-simulation, both Matlab and OPNET execute in parallel interactively in a synchronised fashion. The major contributions of this paper are:

- Using Mobile Ad-hoc Network (MANET) to carry real time data.
- Applying the distributed NCS architecture using multiple sensors and a single controller.
- Implementing a realistic wireless communication model using OPNET simulation.
- Using interactive co-simulation of SIMULINK and OPNET to simulate the plant model and MANET model, respectively.

This paper is organised as follows. Section II discusses simulation model and Matlab-OPNET interface mechanism, section III presents the results, and finally section IV draws some conclusions.

II. SIMULATION MODEL

A. Plant/Controller model: Double pendulum coupled by a spring

In many research works numeric examples are used to evaluate performance of the system. Unlike those works, this paper considers a benchmark case plant model [6] that implements the distributed nature of NCS as shown in Figure 1. The system constants and variables are given in Table I. It is assumed that the mass of each pendulum is uniformly distributed and the mass of the spring is zero. The length of the spring is chosen so that $F = 0$ when $\theta_1 = \theta_2$ which implies $(\theta_1\ \dot{\theta}_1\ \theta_2\ \dot{\theta}_2)^T = 0$ is an equilibrium of the system if $\tau_1 = \tau_2 = 0$. The model is based on the mathematical equations (1-6). The initial conditions of the two pendulums are noted as $\theta_1(0) = x01 \quad \theta_2(0) = x02$. In this model, if any angle of the pendulums exceeds 60 degrees (1.04 radians) from their central positions, the simulation will stop and the system is considered as unstable.

The states of the pendulums are sent at different sampling rates to the controller through two different wireless channels. The control objective of the system is to keep both the pendulums upright or to follow a particular reference/trajectory by applying the controls to the both actuators separately as depicted in Figure 1.

This project has been funded by the Faculty of Computing, Engineering and Technology, Staffordshire University

This architecture guarantees that the system has a distributed structure with two sensors and two actuators. The challenging issues are to maintain suitable communication network packet delays, packet losses etc. so that the system does not become unstable.

B. Mobile ad-hoc network (MANET) model

Mobile Ad-hoc NETworks (MANET) offer very dynamic and flexible wireless networks; these are self-organising and can be easily deployed without any infrastructure [7]. An NCS over MANET is a comparatively new research area and is still thriving.

The radio propagation model used in this paper considers both *path loss* and *fading* effects to achieve realistic wireless signal propagation. The model is expressed in (7) where P_r is the received power, β is path loss exponent, d is the distance between the transmitter and the receiver, d_0 is the reference distance and X_{dB} is a Gaussian random variable with zero mean and standard deviation σ_{dB}. Here σ_{dB} is called fading deviation that can be obtained by measurement. This model extends the *ideal circular model* to a sensible statistic model in which nodes communicate probabilistically at the edge of the communication range [8]. The simulation model implements IEEE 802.11b technology that can support up to 11 Mbps data rate. An open field with MANET nodes equipped with Lucent Orinoco wireless network cards are taken into consideration [9]. To reflect the open field environment, path loss exponent, $\beta=2.8$ and $X_{dB}=6dB$ fading effect have been implemented in the OPNET simulation as suggested in [9], [10], [11].

The Lucent Orinoco wireless network card specification [12] and corresponding transmission ranges obtained from the implemented OPNET simulation are given in Table II. The WNCS area has been chosen as a square based on the transmission range, r and is shown in Figure 2. An open field of size of 174m × 174m and 13 mobile nodes are considered in this paper using the same node density [9], [10], [11].

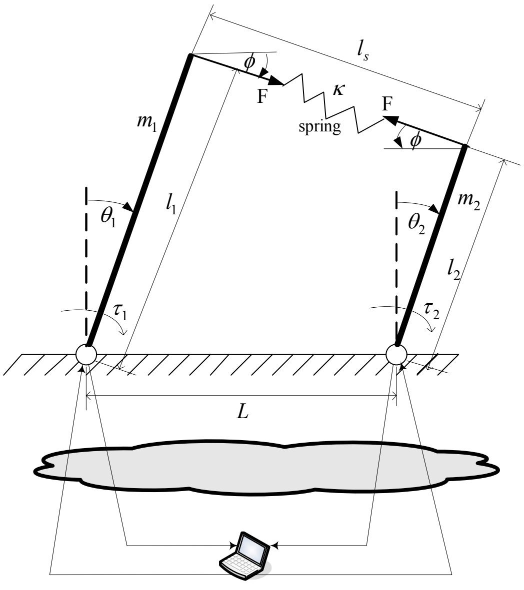

Figure 1: Double pendulum coupled by a spring.

TABLE I: MODEL VARIABLES AND CONSTANTS.

Symbol	Description
θ_i	Angular displacement of pendulum i (i=1, 2)
τ_i	torque input generated by the actuator for pendulum i (i=1, 2)
F	Spring force
l_s	Spring length (i=1, 2)
ϕ	Slope of the spring to the earth
l_i	Length of pendulum i (i=1, 2)
m_i	Mass of pendulum i (i=1, 2)
L	Distance of two pendulums
κ	Spring constant

$$[m_1(l_1)^2/3]\ddot{\theta}_1 = \tau_1 + m_1 g(l_1/2)\sin\theta_1 + l_1 F\cos(\theta_1-\phi) \quad \text{...(1)}$$

$$[m_2(l_2)^2/3]\ddot{\theta}_2 = \tau_2 + m_2 g(l_2/2)\sin\theta_2 - l_2 F\cos(\theta_2-\phi) \quad \text{...(2)}$$

$$F = \kappa\,(l_s - [L^2 + (l_1 - l_2)^2]^{1/2}) \quad \text{... (3)}$$

$$l_s = [(L + l_2\sin\theta_2 - l_1\sin\theta_1)^2 + (l_2\cos\theta_2 - l_1\cos\theta_1)^2]^{1/2} \quad \text{...(4)}$$

$$\phi = \tan^{-1}\frac{l_1\cos\theta_1 - l_2\cos\theta_2}{L + l_2\sin\theta_2 - l_1\sin\theta_1} \quad \text{... (5)}$$

$$\begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix} = -\begin{bmatrix}11.327 & 3.001 & 0 & 0\\ 0 & 0 & 7.594 & 1.861\end{bmatrix}\begin{bmatrix}\theta_1\\ \dot{\theta}_1\\ \theta_2\\ \dot{\theta}_2\end{bmatrix} \quad \text{...(6)}$$

$$\left[\frac{P_r(d)}{P_r(d_0)}\right]_{dB} = -10\beta\log\left(\frac{d}{d_0}\right) + X_{dB} \quad \text{...(7)}$$

For movement, *random way-point model* has been implemented. In this model, nodes move from one point to another random point at a constant speed chosen from a specified range. It then waits at the new point for some time and then another random destination point is chosen. This movement model provides continuous node movement so that MANET routing algorithms can be evaluated [9], [11]. In this simulation, node speeds are chosen uniformly between 1 and 10 m/s. Nodes wait for 60s before moving to a new destination point [9].

TABLE II: LUCENT ORINOCO PC CARD SPECIFICATION AND TRANSMISSION RANGE

	1 Mbps	**2 Mbps**	**5.5 Mbps**	**11 Mbps**
Output power (dBm)	15			
Receiver sensitivity (dBm)	-94	-91	-87	-82
OPNET maximum transmission range (m)	245	195	140	90

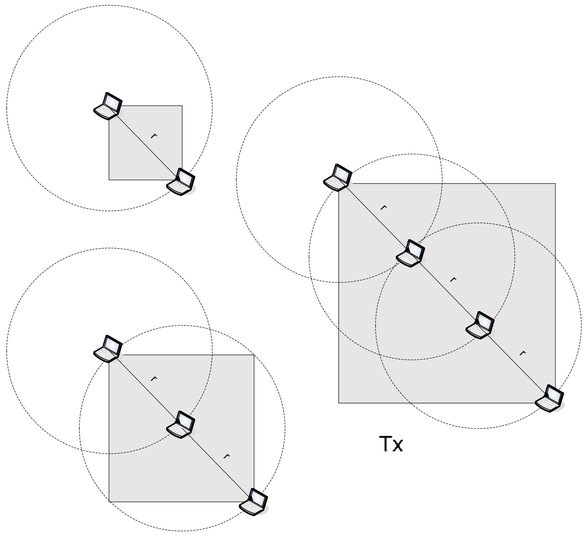

Figure 2: Simulation area based on transmission range

C. *SIMULINK-OPNET interactive co-simulation*

In the interactive co-simulation environment, OPNET executes as master simulator and maintains the simulation time. OPNET plant and controller nodes invoke two Matlab engine servers to execute the plant and controller SIMULINK models, respectively as shown in Figure 3. During sampling task, the state of a particular sensor is read from the SIMULINK model and passed to OPNET to generate the state packet. When a control packet arrives at the plant, it is passed to the corresponding actuator. As OPNET and SIMULINK maintain independent simulation time concepts, their simulation time have to be synchronised to run them interactively. The synchronisation mechanism between OPNET and SIMULINK plant is explained in Figure 4. OPNET starts execution and pauses at simulation time 0. The OPNET plant node invokes the corresponding SIMULINK model. After initialisation the SIMULINK plant model pauses at time 0. OPNET resumes execution and pauses at sampling time T1. It passes a command to SIMULINK to execute until SIMULINK time T1. When SIMULINK pauses at time T1, OPNET plant node reads the plant state from SIMULINK and generates a sample packet.

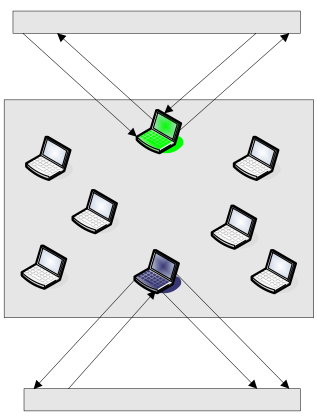

Figure 3: Interactive OPNET and Matlab SIMULINK co-simulation.

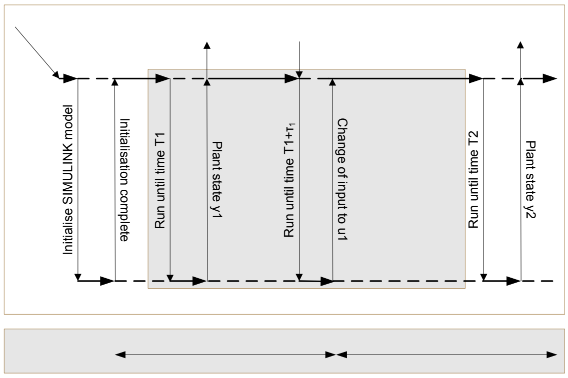

Figure 4: Synchronisation mechanism of OPNET and SIMULINK time for interactive co-simulation.

Upon receiving a control packet at time T1+τ1, OPNET issues a command to SIMULINK to execute until SIMULINK time T1+τ1 with previous input u0 and then change input to u1. When SIMULINK finishes execution, OPNET continue to run the simulation in this fashion.

D. *Comparison scenarios*

Figure 5 depicts the scenarios that are considered in this paper. Figure 5A shows the direct control configuration implemented in Matlab. Figure 5B shows the mechanism of sharing the MANET among two sensors-actuators for the distributed control using OPNET and Matlab. In the result section, performances of scenarios B have been compared with the scenario A (matlab).

E. *Assumptions*

The following assumptions have been made in this paper.

- States and control information can be placed in a single packet. As the pendulum angle ranges from 0 to 1.04 radians, the angle and control values can be carried by a 512 byte or bigger packet.
- Total control loop delay should not exceed the sampling period and no delay compensation mechanism is implemented. The sampling period is kept long enough so that the state and the control packets can reach their destinations within this period.
- Sampling, actuation and control computation times are negligible compared to network delays. These computation tasks can be executed in the order of microsecond with modern computers. On the other hand, wireless communication delays fall in the order of millisecond range.
- Node movement does not have any effect on the double pendulum control.

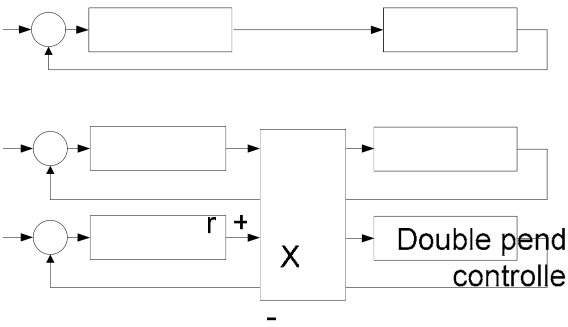

Figure 5: Comparison scenarios for system performance.

$$r = 8\sin\left(\frac{2\pi}{1.5}t\right) \quad ...(8)$$

III. Results

The DSR ad-hoc network routing algorithm [13] is used to investigate the performance for various sampling periods, network data rate and node movements.

A. *Effect of sampling period*

To explore the impact of sampling period, the lowest network data rate of 1 Mbps has been implemented without any node movement under the DSR routing protocol. The controller with has been placed at one corner of the field and the double pendulum is located at the opposite corner. A sinusoidal wave has been taken as the reference signals. The largest acceptable sampling periods without significant output distortions is 0.03s. For the rest of the paper, different sampling periods (0.02s for pendulum 1) and (0.03s for pendulum 2) have been considered for investigation.

B. *Effect of network data rate*

Lucent Orinoco wireless network card exhibits different receiver sensitivities at different data rates. As the data rate increases, the receiver needs higher signal power to receive packets properly thus reducing successful transmission range. Table III summarises the plant stability for different network data transmission rates under the DSR routing strategy with different sampling rates.

TABLE III: NETWORK DATA RATE AND PLANT STABILITY.

	1 Mbps	2 Mbps	5.5 Mbps	11 Mbps
Stationary nodes	☑	☑		
Mobile nodes	☑	☑		

As data rate increases, transmission range becomes smaller and packets need to travel via more intermediate nodes to reach the destination. Therefore, the overall packet delay increases. The control packet and state packet delays for the double inverted pendulum control system are shown in Figure 6 and Figure 7, respectively. Both 1 Mbps and 2 Mbps data rates produced acceptable packet delays. It is noted that 2 Mbps data rate showed higher and more fluctuating delays than 1 Mbps data rate. This is because the plant and the controller can communicate directly at 1 Mbps data rate. On the other hand, one or more intermediate nodes are required to maintain the communication between the plant and the controller under 2 Mbps data rate. Hence packet delays are irregular.

The outputs of pendulum 1 and pendulum 2 in WNCS (Figure 5B) have been compared with the outputs of Matlab (Figure 5A). The absolute angle errors for pendulums 1 and 2 are depicted in Figure 8 and Figure 9, respectively. It is clearly noted from these figures that the absolute error is much higher at data rate of 2 Mbps. Under 2 Mbps data rate, packets experience higher delays and delay jitter that caused worse absolute error for the double pendulum plant.

C. *Effect of node movement*

Figure 10 and Figure 11 show the control packet and state packet delays for node movements, respectively. As the packet delays are very close for 1Mbps and 2 Mbps data rates, the performance produced is almost the same. It is noted that the delays are much higher at the beginning of the simulation and then gradually become stable. At the starting of the simulation, as the MANET establishes the routes, packets experience higher delays. Once the route discovery procedure is finished, the network exhibits more stable delays. However, unlike the stationary nodes, 2 Mbps data rate showed lower delays than 1 Mbps. This might be the case that node movements caused shorter routes that produced lower delays for 2 Mbps data rate.

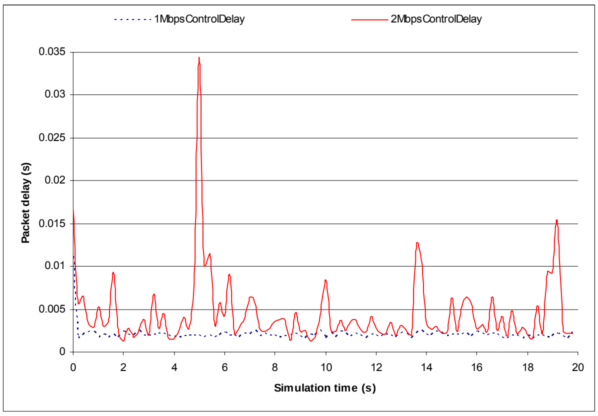

Figure 6: Control packet delays (stationary nodes).

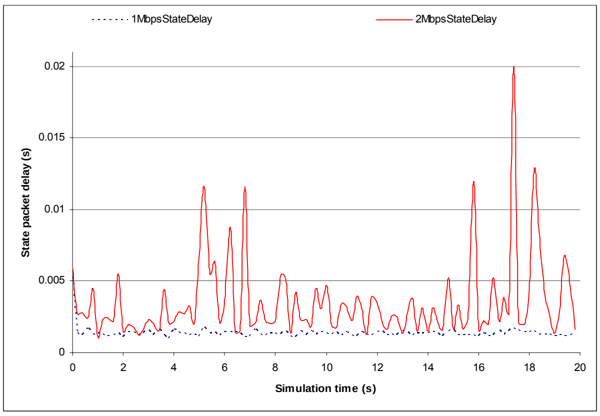

Figure 7: State packet delays (stationary nodes).

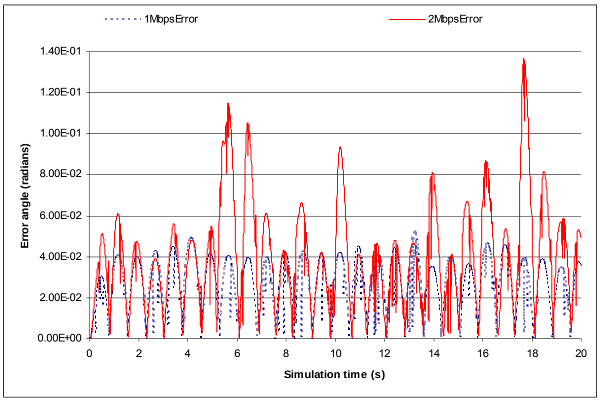

Figure 8: Absolute angle error comparison for pendulum 1 (stationary nodes).

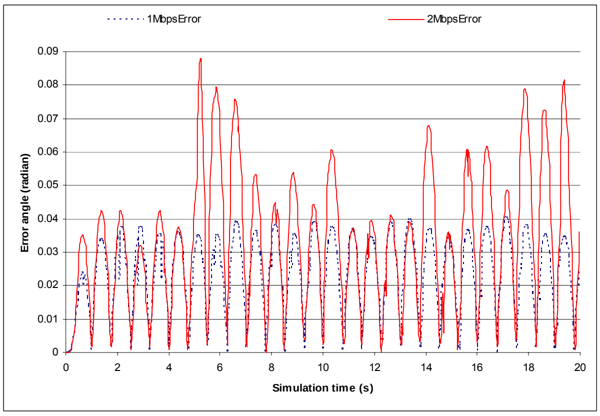

Figure 9: Absolute angle error comparison for pendulum 2 (stationary nodes).

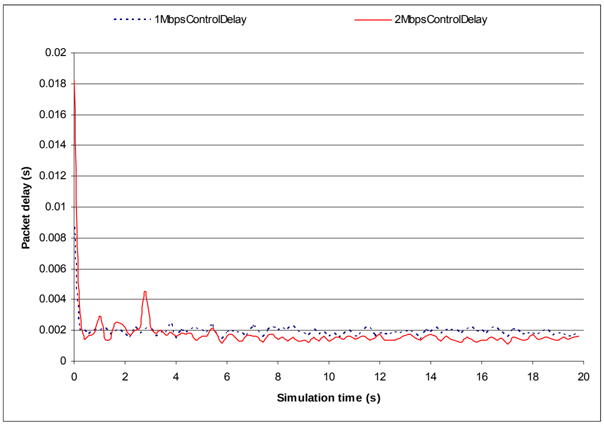

Figure 10: Control packet delays (mobile nodes).

The comparison of the absolute angle error of the pendulum 1 and 2 are depicted in Figure 12 and Figure 13, respectively. As the delays of 2 Mbps data rate were lower and more stable, 2 Mbps data rate produced lower absolute error than 1 Mbps data rate. The maximum absolute errors for 1 and 2 Mbps data rate under stationary and mobile nodes are shown in Table IV.

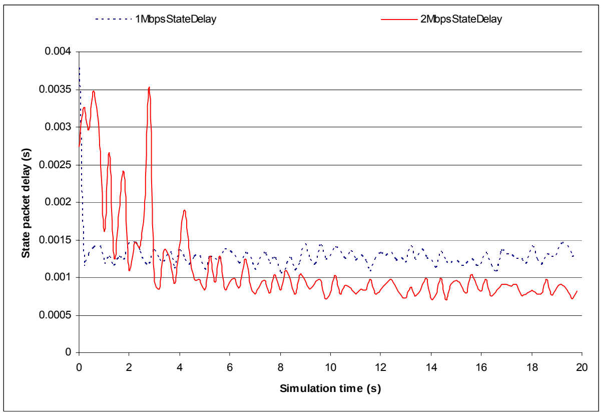

Figure 11: State packet delays (mobile nodes).

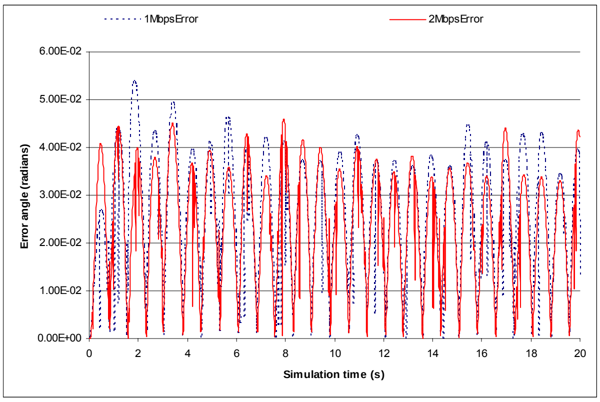

Figure 12: Absolute angle error comparison for pendulum 1 (mobile nodes).

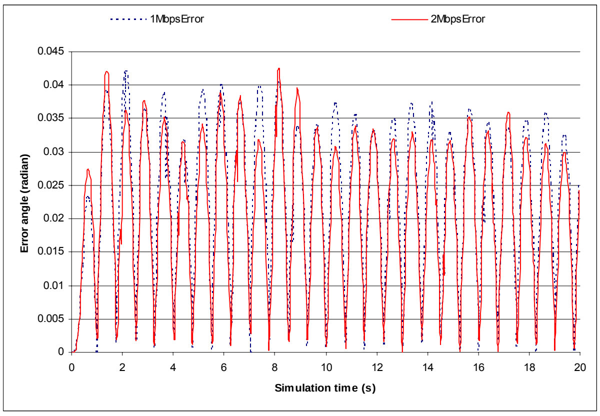

Figure 13: Absolute angle error comparison for pendulum 2 (mobile nodes).

TABLE IV: MAXIMUM ABSOLUTE ERROR

	1 Mbps, pendulum 1	**2 Mbps, pendulum 1**	**1 Mbps, pendulum 2**	**2 Mbps, pendulum 2**
Stationary nodes	5.23×10^{-2}	1.37×10^{-1}	0.04	0.087
Mobile nodes	5.4×10^{-2}	4.59×10^{-2}	0.042	0.042

IV. CONCLUSION

The main focus of this paper is to implement an interactive co-simulation environment for Matlab and

OPNET. To synchronise the simulation times from both packages, OPNET is executed as the master process and Matlab is as child process. Each plant is modelled by one Matlab engine server and the controller is simulated by one more Matlab engine server. The double pendulum coupled by a spring plant could maintain stability for 1 and 2 Mbps data rates for both stationary and mobile nodes. However, for the 5.5 and 11 Mbps data rates transmission ranges were smaller and more intermediate nodes were required to maintain the plant-controller communication. This made the double pendulum plant unstable.

REFERENCES

[1] A. Cervin, D. Hanriksson, B. Lincoln, J. Eker, and K. E. ArZen, "How does control timing affect performance? Analysis and Simulation of Timing using Jitterbug and TrueTime," IEEE Control Systems Magazine, vol. 23, no. 3, pp. 16-30, Jun. 2003.

[2] A. Cervin, D. Henriksson, B. Lincoln, J. Eker, and K. E. Arzen, "Jitterbug and TrueTime: Analysis Tools for Real-Time Control Systems," presented at the 2nd Workshop on Real-Time Tools, Copenhagen, Denmark, Aug. 2002.

[3] M. Andersson, D. Henriksson, A. Cervin, and K. E. Arzen, "Simulation of Wireless Networked Control Systems," presented at the 44th IEEE Conference on Decision and Control and European Control Conference (ECC), pp. 476-481, Dec. 2005.

[4] V. Dham, "Link Establishment in Ad Hoc Networks Using Smart Antennas," in Electrical and Computer Engineering, Master's thesis, Virginia Polytechnic Institute and State University, 2003, http://scholar.lib.vt.edu/theses/available/etd-05072003-180228/unrestricted/etd.pdf.

[5] J. Colandairaj, G. W. Irwin, and W. G. Scanlon, "Analysis and Co-Simulation of an IEEE 802.11B Wireless Networked Control System," presented at the 16th IFAC World Congress, Prague, Czech Republic, Jul. 2005.

[6] T. C. Yang, "Double pendulums coupled by a spring (point-to-point connection)," http://www.sussex.ac.uk/Users/taiyang/software/pend_spring.zip , Jun 2007.

[7] E. M. Royer and C. K. Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks," IEEE Personal Communications, vol. 6, no. 2, pp. 46-55, 1999.

[8] K. Fall and K. Varadhan, "The Ns Manual (formerly ns Notes and Documentation), The VINT project," 2006, http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf, Mar 2006.

[9] J. Liu, Y. Yuan, D. M. Nicol, R. S. Gray, C. C. Newport, D. F. Kotz, and L. F. Perrone, "Simulation validation using direct execution of wireless ad-hoc routing protocols," presented at the 18th Workshop on Parallel and Distributed Simulation (PADS'04), May. 2004.

[10] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott, "Experimental Evaluation of Wireless Simulation Assumptions," presented at the ACM/IEEE International Symposium on Modelling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), pp. 78-82, Oct. 2004.

[11] C. Newport, "Simulating mobile ad hoc networks: a quantitative evaluation of common MANET simulation models," Dartmouth College Science Technical Report TR2004-504, Jun. 2004, http://cmc.cs.dartmouth.edu/cmc/papers/newport:thesis.pdf.

[12] "ORiNOCO PC Card, Home, office, and Public Mobile Broadband Internet Access," 2000, http://www.ictp.trieste.it/~radionet/2001_school/docs/specs/orinoco/datasheet_pc_card.pdf, May 2007.

[13] D. B. Johnson, D. A. Maltz, and J. Broch, Ad Hoc Networking. Boston, USA: Addison-Wesley, 2001.