Co-simulation of Wireless Networked Control Systems over Mobile Ad-hoc Network using SIMULINK and OPNET

Mohammad Shahidul Hasan, Hongnian Yu, Alison Carrington 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.carrington}@staffs.ac.uk, *t.c.yang@sussex.ac.uk

Abstract—**Wireless Networked Control Systems (WNCS) over Mobile Ad-hoc Network (MANET) is a new area of research and has many potential applications, for instance, military or rescue missions, exploring hazardous environments etc. For performance evaluation, researchers mostly rely on computer simulations as WNCS experiments are expensive to execute. It will generate a significant benefit to conduct performance analysis of WNCS over MANET using co-simulation that utilises SIMULINK and OPNET to simulate plant/controller behaviour and the MANET respectively. Previous conference papers by the authors reported the initial SIMULINK-OPNET co-simulation for only one network size. This paper extends our previous works and presents the SIMULINK-OPNET co-simulation, methodology and comprehensive simulation results which have not been reported previously. It also considers the impact of five network sizes with stationary and mobile nodes. The proposed SIMULINK-OPNET co-simulation is applied to WNCS over MANET using a realistic wireless communication model. It investigates the impact of network data rates, node mobility, the packet delay, packet drop on the system stability and performance.**

Keywords- Wireless Networked Control Systems (WNCS), Mobile Ad-hoc NETwork (MANET), OPtimised Network Engineering Tool (OPNET), Network Simulator version 2 (NS2), Wireless Signal propagation model, DSR policy.

I. INTRODUCTION

Networked Control Systems (NCS) are now being implemented over wireless networks because of the latest development of high speed reliable wireless communication technologies and the need for node mobility in many applications. These systems are known as Wireless Networked Control Systems (WNCS). The simplest WNCS includes a plant and a controller with point-to-point wireless communication between them. An advanced version of WNCS applies the control mechanism over a multi-hop Mobile Ad-hoc Network (MANET) as it offers a dynamic, self-organising wireless network and can be easily deployed without any infrastructure [1]. However, WNCS over MANET has brought many challenges to researchers, such as unpredictable network packet delay and dropouts, random node movements etc.

Research on WNCS mostly relies on simulation studies since launching real experiments is expensive and time consuming [2], [3]. The motive of WNCS co-simulation is to simultaneously simulate both the system dynamics and the network events [4]. A brief description of co-simulation tools for NCS, e.g., MATLAB, *Jitterbug*, *TrueTime* etc. can be found in [4], [5]. For WNCS simulations, a network simulator has been implemented as C MEX S-functions to execute simultaneously with the SIMULINK control system [6]. Co-simulation of control and network has been implemented by MATLAB/SIMULINK in [7], [8], [9], [10] that investigated NCS performance for various data rates, traffic, loads, network delays, networked predictive control, compensation of transmission delay etc. Paper [11] uses MATLAB to simulate the MAC (Medium Access Control) sub-layer protocols of control networks where network parameters such as the number of nodes, the message periods, and message sizes can be specified in the simulation model. Control experiments over a physical network were performed between two computers in [12] where the Visual C++ program provides the user interface to the network and the simulation of the plant and controller is carried out using MATLAB. However, the MATLAB/SIMULINK environment does not provide sufficient support for simulation of real time implementation issues. Real Time Workshop (The Mathworks, 2001a) allows prototyping and implementation of real

time control systems, but has very limited support for simulation of shared CPU resources and no support for simulation of networks [13]. MATLAB also has limitations in simulating the process models running and many aspects of wireless networks, e.g., node movement model, wireless signal propagation model etc.

Jitterbug and TrueTime have been used to investigate the effects of the sampling period, communication delay, jitter, control-task scheduling, blocking of real time tasks etc. on the system performance in [14], [15] and [16]. However, these network modelling toolboxes for MATLAB have some limitations. For instance, TrueTime does not support wireless networks and uses simplified network models [7]. It is not possible to use Jitterbug to evaluate the performance of a feedback scheduling system where the CPU loads change and where the sampling periods of the controllers are changing over time. Another limitation of Jitterbug is that only linear systems can be analysed [17].

Network Simulator version 2 (NS2) [18], [19] and *OPtimised Network Engineering Tool (OPNET)* [20] are well-used simulation packages in the computer network research community that allow detailed simulation of communication networks. Network simulators have also been used to model both the system dynamics and the network. For instance, NS2 is used to simulate the entire NCS in [21], [22]. An investigation of network delay for WNCS over MANET has been carried out using only OPNET in [23]. However, it is a tedious task to implement dynamic models and control algorithms in systems using network simulator languages such as *Proto-C* or *Tool Command Language (TCL).*

Research works e.g. [24], [25], [26], [27] etc. combined two simulation packages to achieve a more efficient co-simulation approach. OPNET and MATLAB have been integrated to evaluate the performance of smart antennas using the MX interface provided by MATLAB, which allows C programs to call functions developed in MATLAB [24]. A co-simulation platform that combines the NS2 network simulator with the *Modelica* framework has been presented in [25] where NS2 models the communication network and Modelica simulates the system, sensor, actuator etc. SIMULINK and OPNET co-simulation for WNCS over MANET has been considered in [26], [27]. Paper [26] presented the initial simulation results and analysis for thirteen stationary and mobile nodes. Paper [27] investigated the situation where the C/C++ controller communicates with the simulated stationary MANET and plant nodes over a real wireless link using two computers. This paper extends the works conducted in [26], [27] and presents the SIMULINK-OPNET co-simulation, methodology and comprehensive simulation results which have not been reported previously. It also considers the impact of five network sizes with stationary and mobile nodes.

This paper implements a co-simulation approach that integrates the strengths of SIMULINK and OPNET to produce more realistic simulation results. Both simulators execute in parallel interactively in a synchronised fashion. However, as SIMULINK is time driven and OPNET is event driven, the challenge for the co-simulation approach is to synchronise the time concepts that have been implemented and discussed in this research work. Though both system and network can be simulated by a general purpose programming language such as '*C*', researchers and engineers can save a lot of time and effort by using the pre-built customisable system blocks in SIMULINK and network models in OPNET. Once the plant and controller models are built in SIMULINK, various network models and scenarios can be chosen in OPNET to optimise the overall system performance. On the other hand, if the network model and plant models are known in advance, different controllers can be applied without affecting the network or the plant model. This gives a strong modularity feature to the co-simulation. Furthermore, the proposed SIMULINK-OPNET co-simulation is a generic approach and can be applied to any NCS simulation.

Section I has reviewed the challenging issues and presented current related research works on co-simulation of WNCS. Section II introduces the *double inverted pendulum coupled by a spring plant* model that has been taken as the

case study and proposes the novel co-simulation approach. Section III presents the validation of the co-simulation approach using the case study. Finally, section IV draws some conclusions and points to future works.
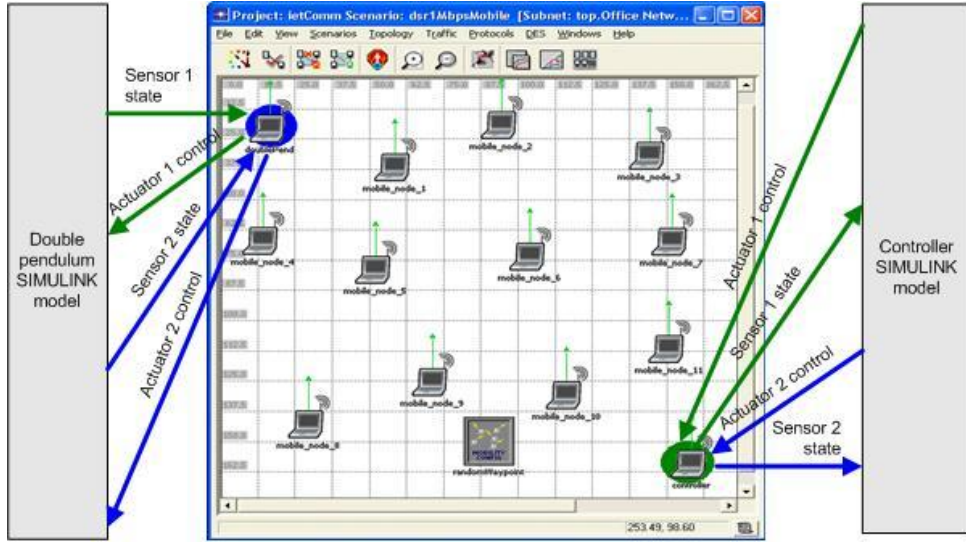
II. SIMULATION MODEL

A. *Plant/controller model*

This paper considers a benchmark case plant model: a double inverted pendulum coupled by a spring. A detailed development of the model and design of the linear control law can be found in [28], [29]. In this study, we implement the distributed nature of NCS using four sensors and two actuators as shown in Figure 1(a) where θ_i is angular displacement of pendulum i (i=1, 2), τ_i is torque input generated by the actuator for pendulum i (i=1, 2), F is spring force, l_s is spring length, ϕ is slope of the spring relative to earth, l_i is length of pendulum i (i=1, 2), m_i is mass of pendulum i (i=1, 2), L is distance between two pendulums and κ is the spring constant. The plant model is based on the mathematical equations (1) to (5) and the decentralised PID control law is given in (6). 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 θ_1= θ_2 which implies $(\theta_1 \ \dot{\theta}_1 \ \theta_2 \ \dot{\theta}_2)^T = 0$ is an equilibrium of the system if τ_1= τ_2=0. The initial conditions of the two pendulums are noted as $\theta_1(0)$=x01, $\theta_2(0)$=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 to the controller at different sampling rates through two different wireless channels. The control objective of the system is to keep both the pendulums upright or to make them follow a particular reference/trajectory such as a sinusoidal or pulse signal by applying the controls to both actuators separately as depicted in Figure 1(a). In this paper, the pendulums are required to follow pulse signals and to maintain the stability condition mentioned previously. The challenging issue is to maintain suitable communication network packet delays, packet losses etc. so that the system remains stable.
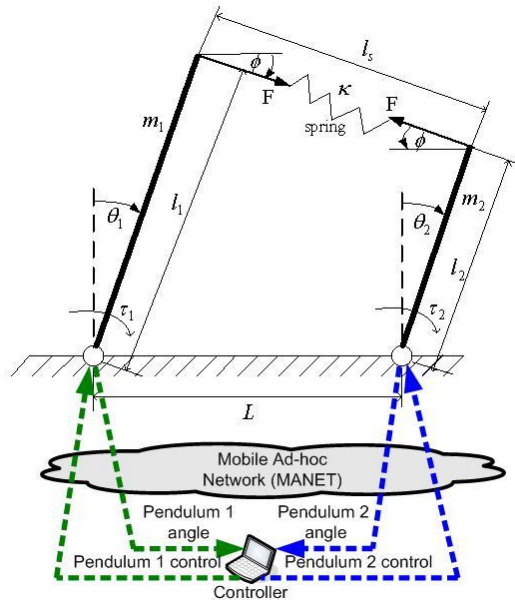
(a)

(b)

Figure 1: (a) The plant model and WNCS communication mechanism, (b) Interactive SIMULINK-OPNET co-simulation.

The system model is described as

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

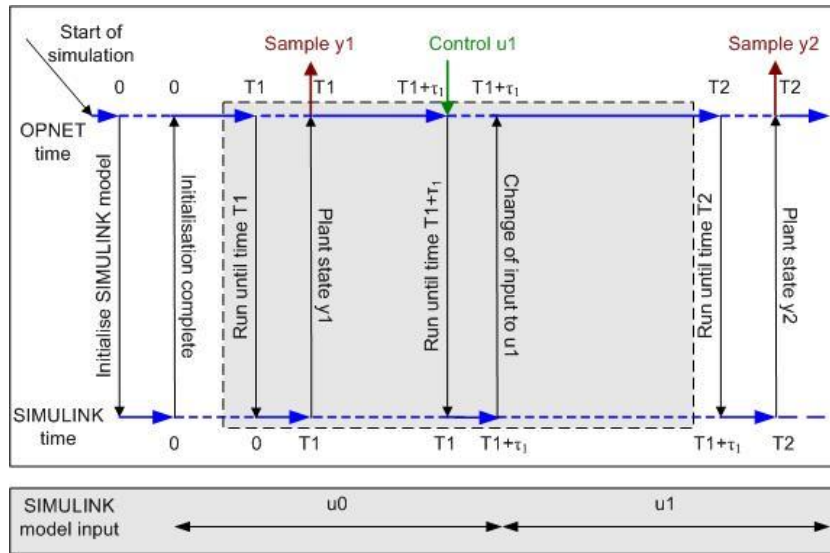$$[m_2(l_2)^2/3]\ddot{\theta}_2 = \tau_2 + m_2 g(l_2/2)\sin\theta_2 - l_2 F\cos(\theta_2 - \phi) \tag{2}$$

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

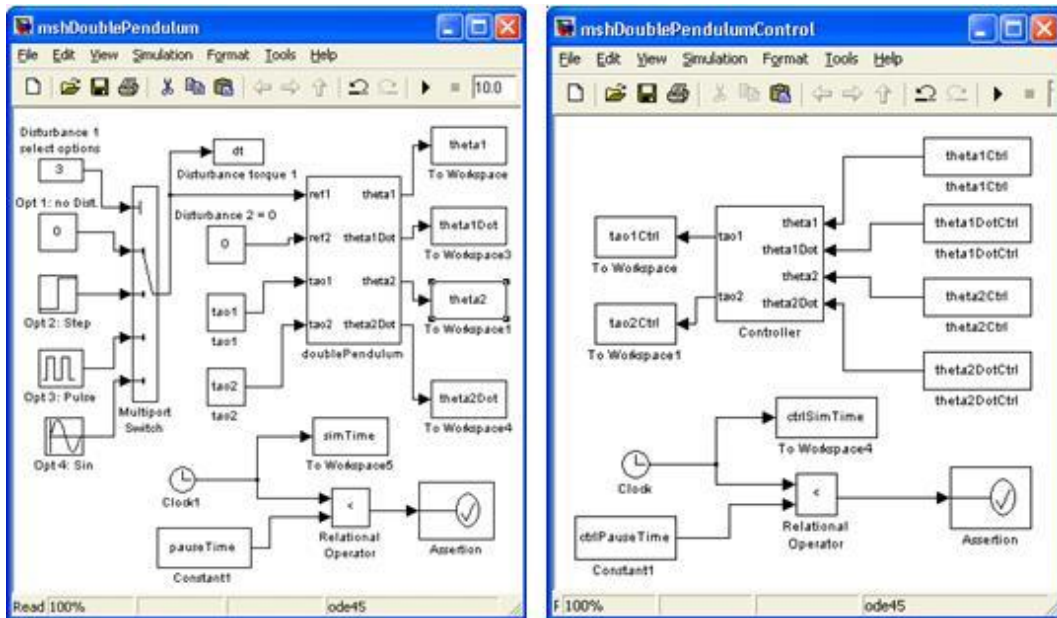$$l_s = [(L + l_2\sin\theta_2 - l_1\sin\theta_1)^2 + (l_2\cos\theta_2 - l_1\cos\theta_1)^2]^{1/2} \tag{4}$$

$$\phi = \tan^{-1}\frac{l_1\cos\theta_1 - l_2\cos\theta_2}{L + l_2\sin\theta_2 - l_1\sin\theta_1} \tag{5}$$

The control law is

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

(a)

(b)

Figure 2: (a) SIMULINK-OPNET synchronisation, (b) SIMULINK plant and controller models.

B. SIMULINK-OPNET interactive co-simulation

In the interactive co-simulation environment, OPNET executes as the master simulator and maintains the co-simulation time. The OPNET plant and controller nodes invoke two MATLAB engine servers to execute the plant and the controller SIMULINK models respectively as shown in Figure 1(b). The state of the particular sensor is read from the SIMULINK model by the OPNET plant model before generating the state packet. When a control packet arrives at the plant, the control information is passed to the corresponding actuator of the SIMULINK model.

The synchronisation mechanism between the SIMULINK and the OPNET models is explained in Figure 2(a). OPNET begins execution and pauses at simulation time 0. The OPNET node models invoke the corresponding SIMULINK models. After initialisation, the SIMULINK plant model pauses at time 0. OPNET resumes execution and pauses at sampling time $T1$. It passes a command to the SIMULINK plant model to execute until time $T1$. When SIMULINK pauses at time $T1$, the OPNET plant node model reads the plant state from the SIMULINK model and generates a sample packet.

The total closed loop delay is denoted by *t1* which is measured as sensor to controller delay plus controller to actuator delay. Upon receiving a control packet at time *T1+t1*, OPNET issues a command to the SIMULINK plant model to execute until time *T1+t1* with the previous input *u0*. At time *T1+t1*, the SIMULINK plant model pauses and the input is changed to the new control *u1*. When SIMULINK finishes execution, OPNET continues to run the simulation in this fashion. The same synchronisation mechanism has been used for co-simulation of NS2 and Modelica in [25].

The left part of Figure 2(b) shows the SIMULINK double pendulum model. The multi-port switch is used to select the type of *reference*, e.g., step, pulse, sinusoidal etc. The *theta1*, *theta1Dot*, *theta2*, *theta2Dot* are the angles and angle derivative outputs. The *tao1*, *tao2* are the control inputs for the corresponding pendulums. The bottom part of the plant model implements the pausing and resuming of the SIMULINK model from the OPNET. The constant *pauseTime* is set by OPNET and it stores the next pause time when the SIMULINK model is to be paused. As soon as the *simTime* becomes equal to pauseTime, the assertion executes the command *set_param('mshDoublePendulum', 'SimulationCommand', 'pause')*. Once the SIMULINK model is paused OPNET can read the values for the outputs of the pendulums. Then OPNET can set the next pause time and issue the command *set_param('mshDoublePendulum', 'SimulationCommand', 'update')* to reflect the change of pause time. OPNET can execute the command *set_param('mshDoublePendulum', 'SimulationCommand', 'continue')* to resume the SIMULINK model. When a control packet arrives, OPNET updates the *tao1* or *tao2* based on the control packet information and issues the command *set_param('mshDoublePendulum', 'SimulationCommand', 'update')* to update all the variables.

The right part of Figure 2(b) shows the SIMULINK controller model where *theta1Ctrl*, *theta1DotCtrl*, *theta2Ctrl*, *theta2DotCtrl* are inputs and *tao1Ctrl*, *tao2Ctrl* are outputs. The *ctrlPauseTime*, *ctrlSimTime*, assertion etc. implement the synchronisation with the OPNET as explained previously.

C. Sampling, actuation and control tasks invocation model

Table I summarises different combinations of task invocation schemes. The WNCS task invocation model of this paper applies clock-driven sampling. Control and actuation tasks are invoked when an event occurs, for instance, when it receives an information packet from another node through the network [5], [10]. Clock-driven sampling and an event-driven control-actuation approach has several advantages. For instance, it does not require plant-controller synchronisation, supports multi-rate sampling [30], [31] and is power efficient [32].

Table I: Control task invocation schemes and characteristics [30], [31].

Sensing	Control	Actuation	Clock synchronisation	Delay nature
Clock	Event	Clock	Not required	Fixed
Clock	Event	Event	Not required	Varying
Clock	Clock	Event	Required	Varying
Clock	Clock	Clock	Required	Fixed

D. MANET model

MANET simulation study presents mainly two challenges: radio channel characteristics and node mobility models [2].

1) Radio signal propagation model

Figure 3 shows the comparison of computer simulation models and real world experiments [3], [33], [34]. *Model 1* involves two components: *path loss exponent* and *fading*. *Model 2* is the *two-ray ground-reflection* model that uses only

the path loss component. Finally, *model 3* represents the *ideal propagation* model. The comparison revealed that *model 1* exhibits the behaviour closest to the real world experiment [3].
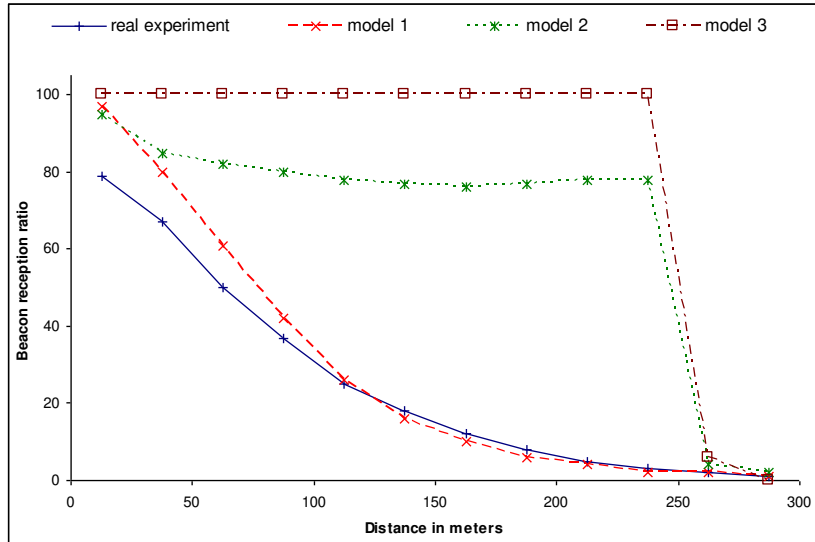

Figure 3: Comparison of three simulation models with real world experiment [3].

The radio channel used in this paper implements *model 1* expressed in (7) where P_r is the received wireless power, d is the distance between the transmitter and the receiver, d_0 is the reference distance, β is path loss exponent 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 [19].

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

The simulation model implements IEEE802.11b technology using MANET nodes equipped with Lucent Orinoco wireless network cards [33]. Table II gives the transmission ranges obtained from the OPNET simulation based on the Lucent Orinoco card specification [35]. It is noted that as the data rate increases, the receiver needs higher signal power to receive packets properly thus reducing the successful transmission range. Hence, under IEEE802.11 technologies, a multi-hop ad-hoc network exists at two-three hops and ten-twenty nodes [2], [36]. The WNCS area has been chosen as a square open field of size 174m × 174m based on the transmission range of 1Mbps data rate. To reflect the open field environment, path loss exponent, β=2.8 and fading effect, X_{dB}=6dB have been implemented. Thirteen MANET nodes produce the same node density as presented in [3], [33], [34]. To observe the effect of the number of network nodes, five network sizes, i.e., six, ten, thirteen, twenty and twenty six nodes have been considered in section III.

Table II: Lucent Orinoco wireless network card specification and transmission ranges obtained from OPNET simulation.

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

2) Node mobility model

For movement, a *random way-point model* has been implemented where nodes move from one point to another random point at a constant speed chosen from a specified range. They wait 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 [33], [34]. 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 [33].

E. MANET routing protocols

MANET routing protocols are categorised into two major classes: proactive and reactive (on-demand). A *proactive protocol* attempts to keep an up-to-date routing table by constantly requesting update information and sharing routing tables [37]. In contrast, *reactive routing protocols* establish a route when requested and routes are maintained until the destination becomes unreachable or the route is no longer required. To date, no clear indication was found regarding the best routing policy. The performance depends significantly on the scenario under consideration [2].

Dynamic Source Routing (DSR) [38] and Ad-hoc On-demand Distance Vector (AODV) [39] are widely used reactive routing protocols. The OPNET simulations of WNCS presented in [22], [23] revealed that DSR exhibited better performance than AODV. Moreover, it should be noted that AODV is unable to handle unidirectional links whereas DSR can [40]. Therefore, DSR has been investigated in this paper.

F. Communication mechanism

Transmission Control Protocol (TCP) is not suitable for MANET as it uses connection oriented packet transfer [2]. On the other hand, as *User Datagram Protocol (UDP)* offers low overheads and discards obsolete or lost packets, it is preferable for networked control applications [31]. UDP is also chosen to validate the simulation results for wireless networks [33]. Therefore, UDP has been used in this simulation investigation.

In IEEE802.11 protocols each packet has a 34 byte MAC and 24 byte PHY header [6]. Each pendulum state can be included in an 8-byte field. The plant id, sensor id and sequence number can be transmitted using three separate 4-byte fields. Therefore the total information for a state or control packet can be contained in an 86-byte packet.

G. Assumptions

The following assumptions have been made in this paper.

- Complete state and control information can be carried by an 86-byte packet. Small packet size guarantees short packet transmission time and produces low network traffic.

- The current control packet must arrive at the plant before the next sampling as the control does not implement any delay compensation mechanism. Otherwise the control packet is treated as obsolete and therefore, discarded.

- Sampling, actuation and control computation task times are negligible compared to network delays as these tasks can be executed in the order of microseconds on most modern computers.

III. RESULTS AND ANALYSIS

A pulse signal of amplitude 5 radians, period 1.5s and 10% pulse width has been applied as the reference signal. The controller has been placed at the south-east corner of the field and the double pendulum plant is located at the opposite (north-west) corner as shown in Figure 1(b). The DSR policy is used to investigate the WNCS performance in terms of pendulum stability and tracking capability for various sampling periods, network data rates, node movements and network sizes, i.e., number of nodes. First the results with thirteen network nodes have been presented. Then the networks with halved (six), doubled (twenty six) and two intermediate sizes (ten, twenty) nodes are explored.

A. *Effect of sampling period*

Typical values of sampling periods can range from hundreds of microseconds to hundreds of milliseconds [41]. One of the rules for choosing sampling period *T* is given in (8) where ω is the natural frequency of the closed loop system [42]. A higher sampling rate improves the performance of an NCS [43]. But it increases computational overheads and generates excessive traffic into the network [44].

$$0.2 \le \omega * T \le 0.6 \tag{8}$$

Five sampling periods (0.005s, 0.006s), (0.02s, 0.03s), (0.05s, 0.06s), (0.10s, 0.11s), (0.15s, 0.16s) and the lowest data rate of 1 Mbps have been applied to the thirteen stationary DSR nodes first. The pendulum tracking performance for the sampling periods is shown in Figure 4. The *direct control* that is implemented without the network using only SIMULINK is shown as *matlabPend1* and *matlabPend2*. The SIMULINK-OPNET co-simulation control performance is compared with the direct control. It is noted that for the sampling period of (0.005s, 0.006s), the plant became unstable as defined in section II.A. Again, though the sampling period of (0.15s, 0.16s) kept the plant model within the stable conditions, it showed occasional spikes in the pendulum angles. This is because the sampling period was too large for the double pendulum plant. Therefore, for the rest of the investigation, sampling periods (0.02s, 0.03s), (0.05s, 0.06s), (0.10s, 0.11s) are considered.
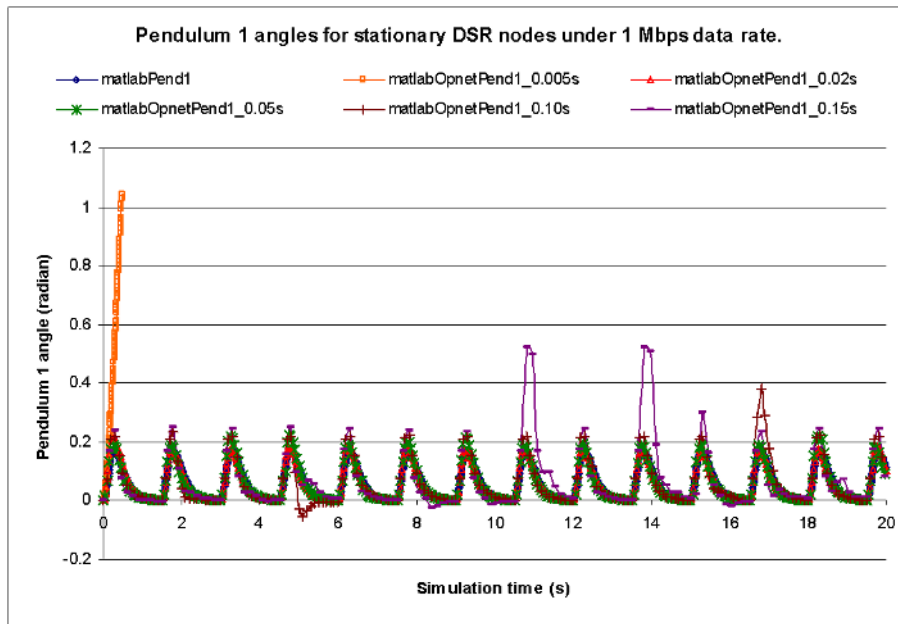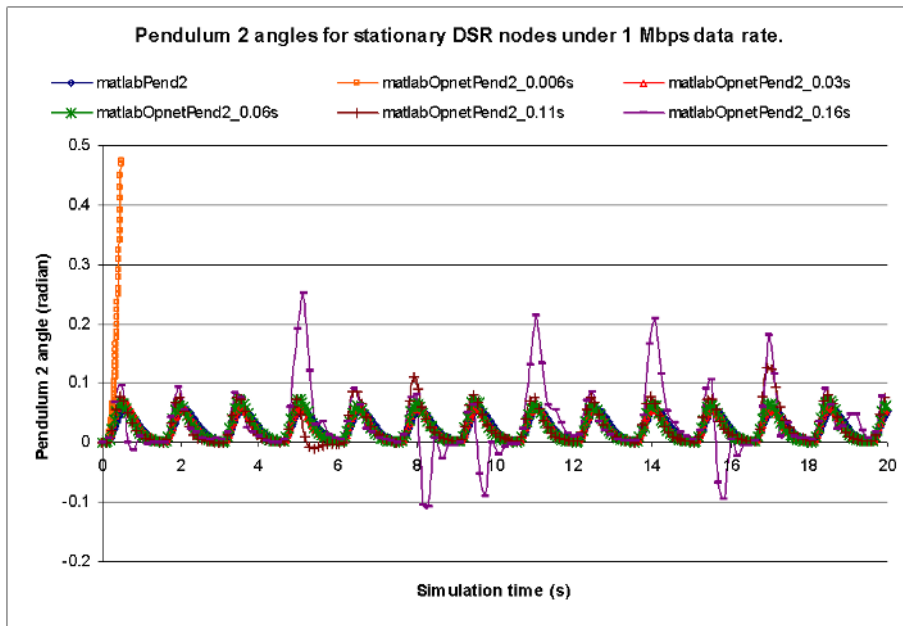
Figure 4: Pendulum tracking performance for thirteen stationary DSR nodes under various sampling periods at 1 Mbps data rate.

The packet routes and delays produced by OPNET were extracted for analysis purposes. According to Table II, the receiver can receive low power signals properly at the data rate of 1 Mbps, so most of the packets can reach the controller or plant directly. Therefore, packet delay and drop probability exhibit low values.

The total delays of the closed-loop control are shown in Figure 5. In most cases, control packets arrived at the plant before the next sampling started and therefore satisfied the constraint stated in assumption section II.G. Some packets experienced delays larger than the sampling periods. These control packets were discarded by the plant, and therefore represent packet drops/losses. Figure 5 also reveals that as the total delays are larger than the sampling period (0.005s, 0.006s) many control packets were dropped. Hence, a sampling period of (0.005s, 0.006s) made the plant unstable.
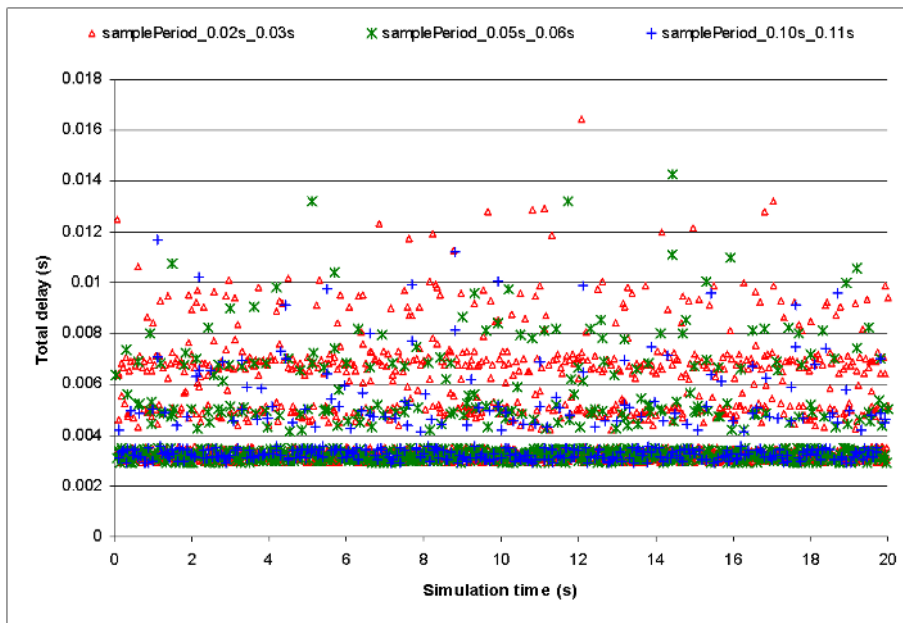
Figure 5: Total closed loop delay for thirteen stationary DSR nodes under 1 Mbps data rate.

The top row of Figure 6 shows the pendulum angles for the sampling periods of (0.02, 0.03s), (0.05, 0.06s) and (0.10s, 0.11s) at the data rate of 1 Mbps. The sampling period of (0.02s, 0.03s) produced the closest performance to

direct control as it was the highest that the WNCS could support among the applied sampling rates. However, the sampling period of (0.10s, 0.11s) exhibited some spikes which might be caused by the large sampling period.

B. *Effect of data rate*

This section presents how the stability of the double pendulum plant is affected as the data rate of the MANET is increased to 2, 5.5 and 11 Mbps.
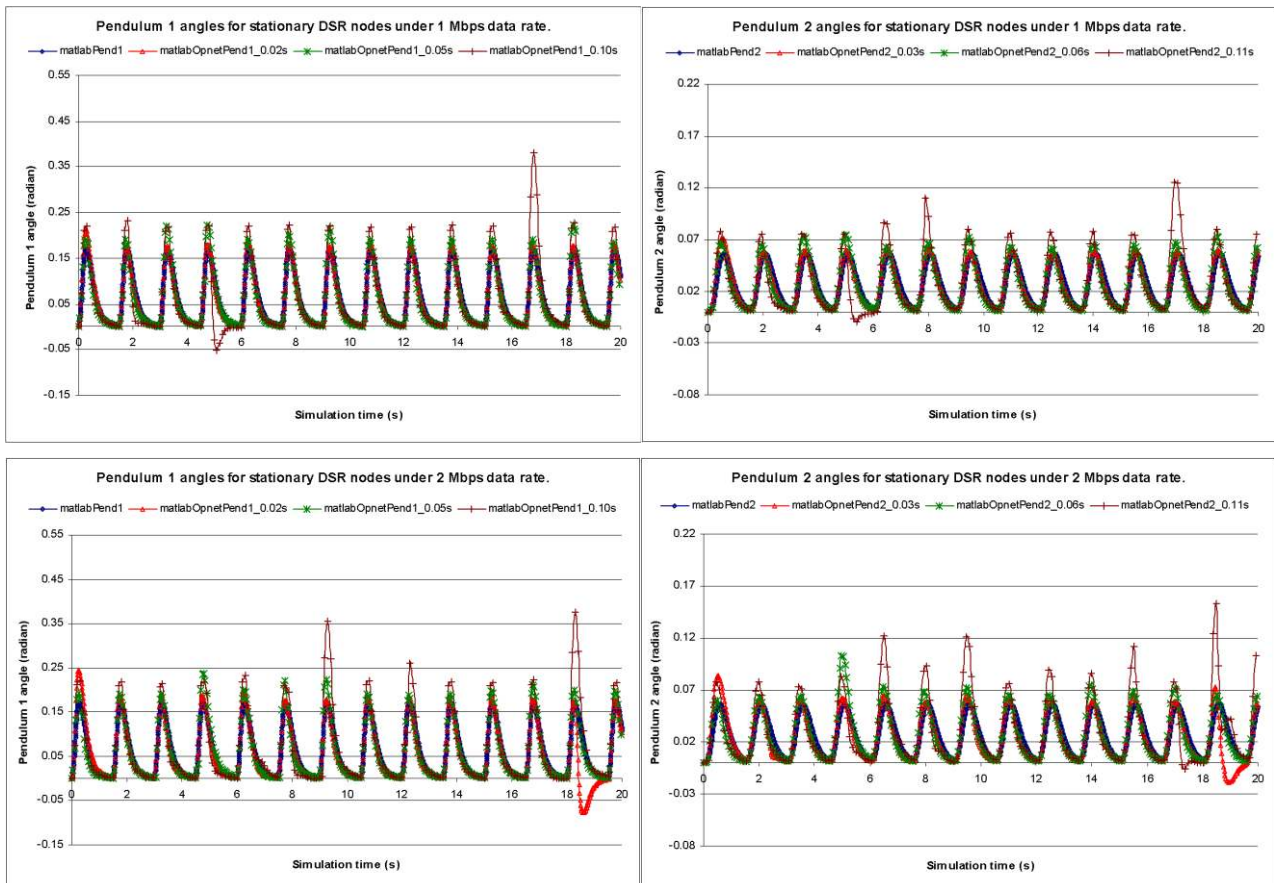
1) *2 Mbps data rate*

As data rate increases, transmission range becomes smaller and packets need to travel via more intermediate nodes to reach the destination. This can be seen from the packet route analysis which revealed that most of the packets needed to travel via one intermediate to reach the plant or the controller. This increased the overall packet delay. The pendulum angles are shown in the second row of Figure 6. The sampling period of (0.10s, 0.11s) is suffering from occasional overshoots.

2) *5.5 Mbps data rate*

After analysing the packet routes, it is noted that all packets travelled via one intermediate node that increased the total delay and the packet loss probability. For the 5.5 Mbps data rate, the sampling period (0.02s, 0.03s) is no longer supported. The pendulum angles are depicted in the third row of Figure 6. The angles showed spikes at the beginning of the simulations as the MANET was going through the route discovery phase.

3) *11 Mbps data rate*

Some packets experienced two intermediate nodes before reaching their destination which caused a significant increase in total delay and packet loss probability. The 11 Mbps data rate only supported the sampling period of (0.10s, 0.11s) and the bottom row of Figure 6 depicts the pendulum angles. The pendulums exhibited occasional spikes in angles because of high packet delay and packet loss.
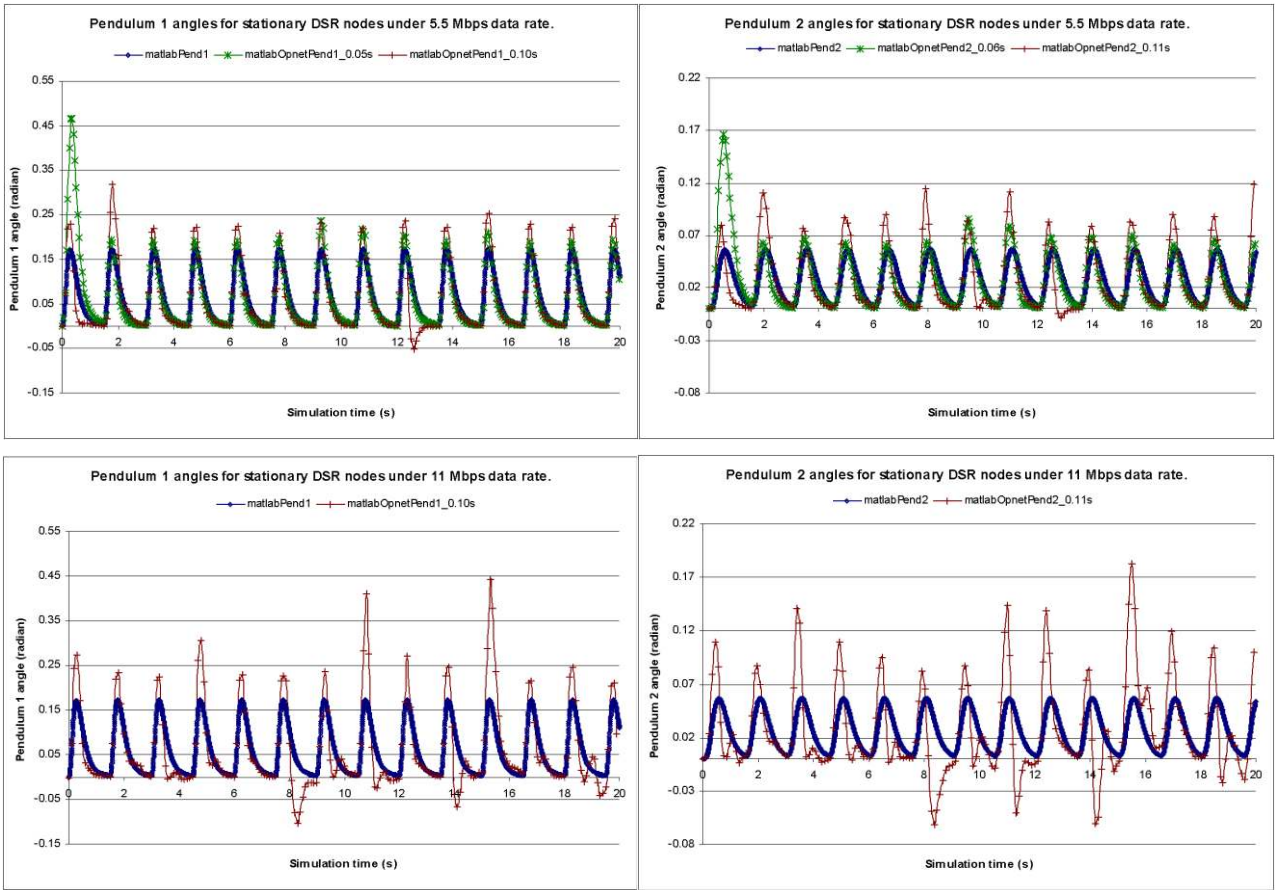
Figure 6: Pendulum angles for thirteen stationary DSR nodes under 1, 2, 5.5 and 11 Mbps data rate.

C. Effect of node movement and network size

The summary of plant stability for stationary and mobile six, ten, thirteen, twenty and twenty six DSR nodes is given in Table III and Table IV respectively. It is noted that the data rates of 1 and 2 Mbps kept the plant stable for all five network sizes under both stationary and mobile conditions. This is because the plant and controller could communicate with each other directly or via a maximum of one intermediate node at these data rates.

For stationary networks it is clear that the six node network did not support data rates of 5.5 and 11 Mbps at all. This is because the small network could not establish an intermediate node that is required between the plant and the controller to cover 174m × 174m according to Table II. The ten, thirteen, twenty and twenty six node networks have adequate nodes and showed similar performances. However, extra nodes in the twenty and twenty six node network did not produce any better routes than the ten and thirteen node networks.

All mobile networks except the thirteen node scenario supported the data rates of 1, 2 and 5.5 Mbps for all sampling periods. It might be that the random node movements established such routes that supported the delay constraint of the WNCS properly. For the stationary node scenario, the data rate of 5.5 Mbps did not support the plant stability under any sampling period in the case of the six node network and the plant remained stable for only one sampling period, i.e., (0.10s, 0.11s) in the case of the twenty six node network. As the node movement is introduced, the plant stayed stable for the networks with six and twenty six nodes under all three sampling periods at the data rate of 5.5 Mbps. However, the stability window became narrower for thirteen nodes as node movement was introduced. For instance, under a sampling period of (0.10s, 0.11s), the plant is stable at all data rates in the case of stationary nodes. But, plant stability is supported only for data rates of 1 and 2 Mbps for mobile nodes.

In general, the sampling period of (0.02s, 0.03s) is not supported by the data rate of 11 Mbps at all. This is because the packets needed to travel through multiple intermediate nodes before reaching their destination, which produced longer delays and higher drop probability. The stability performance was not significantly improved by increasing the number of nodes above ten and hence ten is the optimal network size for this particular scenario.

Table III: Plant stability for stationary DSR nodes.

Network type	Network size	Data rate	Pendulum 1: 0.02s Pendulum 2: 0.03s	Pendulum 1: 0.05s Pendulum 2: 0.06s	Pendulum 1: 0.10s Pendulum 2: 0.11s
Stationary	Six nodes	1 Mbps	☑	☑	☑
		2 Mbps	☑	☑	☑
		5.5 Mbps			
		11 Mbps			
	Ten nodes	1 Mbps	☑	☑	☑
		2 Mbps	☑	☑	☑
		5.5 Mbps	☑	☑	☑
		11 Mbps			☑
	Thirteen nodes	1 Mbps	☑	☑	☑
		2 Mbps	☑	☑	☑
		5.5 Mbps		☑	☑
		11 Mbps			☑
	Twenty nodes	1 Mbps	☑	☑	☑
		2 Mbps	☑	☑	☑
		5.5 Mbps	☑	☑	☑
		11 Mbps			
	Twenty six nodes	1 Mbps	☑	☑	☑
		2 Mbps	☑	☑	☑
		5.5 Mbps			☑
		11 Mbps			☑

Table IV: Plant stability for mobile DSR nodes.

Network type	Network size	Data rate	Pendulum 1: 0.02s Pendulum 2: 0.03s	Pendulum 1: 0.05s Pendulum 2: 0.06s	Pendulum 1: 0.10s Pendulum 2: 0.11s
Mobile	Six nodes	1 Mbps	☑	☑	☑
		2 Mbps	☑	☑	☑
		5.5 Mbps	☑	☑	☑
		11 Mbps		☑	☑
	Ten nodes	1 Mbps	☑	☑	☑
		2 Mbps	☑	☑	☑
		5.5 Mbps	☑	☑	☑
		11 Mbps			
	Thirteen nodes	1 Mbps	☑	☑	☑
		2 Mbps	☑	☑	☑
		5.5 Mbps			
		11 Mbps			
	Twenty nodes	1 Mbps	☑	☑	☑
		2 Mbps	☑	☑	☑
		5.5 Mbps	☑	☑	☑
		11 Mbps			
	Twenty six nodes	1 Mbps	☑	☑	☑
		2 Mbps	☑	☑	☑
		5.5 Mbps	☑	☑	☑
		11 Mbps			

IV. CONCLUSION

This paper focuses on the development of interactive co-simulation of SIMULINK and OPNET to investigate the impact of network size on the performance of WNCS over MANET. It was found that the main challenges of such a system are to maintain acceptable packet delays and drops. An increased network data rate makes the transmission range smaller. Therefore, packets need to travel via more intermediate nodes to reach their destinations. Both smaller transmission range and route re-establishment could cause higher packet delay and drop probability. Increased number of network nodes does not always guarantee reliable route establishment, i.e. overall system stability. Though, in general, the mobile node scenario showed better stability performance than the stationary scenario, random node movement might make the system stability region narrower as the MANET needs to establish new routes. As future work, the following cases can be considered.

- Applying other control methods such as model predictive control.

- Comparison of proactive and reactive routing protocol efficiency e.g., AODV.

V. REFERENCES

[1]   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.

[2]   M. Conti and S. Giordano, "Multihop Ad Hoc Networking: The Theory," *IEEE Communications Magazine*, vol. 45, no. 4, pp. 76-86, 2007.

[3]   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, 2004.

[4]   J. R. Hartman, "Networked Control System Co-simulation for co-design: Theory and experiments," in *Electrical Engineering and Computer Science*, Master of Science thesis, Case Western Reserve University, Cleveland, Ohio, USA, 2004,

[5]   K. E. Årzén and A. Cervin, "Control and Embedded Computing: Survey of Research Directions," presented at the 16th IFAC World Congress, Prague, Czech Republic, 2005.

[6]   J. Colandairaj, G. W. Irwin, and W. G. Scanlon, "An integrated approach to wireless feedback control," presented at the UKACC International Control Conference, Glasgow, UK, 2006.

[7]   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, 2005.

[8]   Z. Chen, L. Liu, and J. Zhang, "Observer Based Networked Control Systems with Network-induced Time Delay," presented at the IEEE international Conference on Systems, Man and Cybernetics, Hague, The Netherlands, pp. 3333-3337, 2004.

[9]   G. P. Liu, D. Rees, and S. C. Chai, "Design and practical implementation of networked predictive control systems," presented at the IEEE international conference on Networking, Sensing and Control, Arizona, USA, pp. 336- 341, 2005.

[10]   Y. Yang, Y. Wang, and S. H. Yang, "A Networked Control System with Stochastically Varying Transmission Delay and Uncertain Process Parameters," presented at the 16th IFAC World Congress, Prague, Czech Republic, 2005.

[11]   F. L. Lian, J. R. Moyne, and D. M. Tilbury, "Performance Evaluation of Control Networks: Ethernet, ControlNet and DeviceNet," *IEEE Control Systems Magazine*, vol. 21, no. 1, pp. 66-83, 2001.

[12]   W. Zhang, M. S. Branicky, and S. M. Phillips, "Stability of Networked Control Systems," *IEEE Control Systems Magazine*, vol. 21, no. 1, pp. 84-99, 2001.

[13]   D. Henriksson, "Flexible Scheduling Methods and Tools for Real Time Control Systems," in *Automatic Control*, PhD thesis, Lund Institute of Technology, Lund, Sweden, 2003,

[14]   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, 2003.

[15]   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, 2005.

[16]   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, 2002.

[17]   A. Cervin, "Integrated Control and Real time Scheduling," in *Automatic Control*, PhD thesis, Lund Institute of Technology, Lund, Sweden, 2003, http://www.lucas.lth.se/events/doc2003/030425.pdf.

[18]   L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, and H. Yu, "Advances in network simulation," *Computer*, vol. 33, no. 5, pp. 59-67, 2000.

[19]   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.

[20]   X. Chang, "Network simulations with OPNET," presented at the 31st conference on Winter simulation: Simulation – a bridge to the future, Arizona, USA, vol. 1, pp. 307-314, 1999.

[21]   M. S. Branicky, V. Liberatore, and S. M. Phillips, "Networked Control System Co-Simulation for Co-Design," presented at the American Control Conference, Denver, 2003.

[22]   M. S. Hasan, C. Harding, H. Yu, and A. Griffiths, "Modelling delay and packet drop in Networked Control Systems using Network Simulator NS2," *the International Journal of Automation and Computing (IJAC)*, vol. 2, no. 2, pp. 187-194, 2005.

[23]   M. S. Hasan, H. Yu, A. Griffiths, and T. C. Yang, "Simulation of Distributed Wireless Networked Control Systems over MANET using OPNET," presented at the IEEE International Conference on Networking, Sensing and Control, London, UK, pp. 699-704, 2007.

[24]   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.

[25]   A. Al-Hammouri, V. Liberatore, H. Al-Omari, Z. Al-Qudah, M. S. Branicky, and D. Agrawal, "A co-simulation platform for actuator networks," presented at the 5th international conference on Embedded networked sensor systems, Sydney, Australia, pp. 383-384, 2007.

[26]   M. S. Hasan, H. Yu, A. Griffiths, and T. C. Yang, "Interactive co-simulation of MATLAB and OPNET for Networked Control Systems," presented at the 13th International Conference on Automation and Computing, Stafford, UK, pp. 237-242, 2007.

[27]   M. S. Hasan, H. Yu, A. Griffiths, and T. C. Yang, "Co-simulation framework for Networked Control Systems over multi-hop mobile ad-hoc networks," presented at the 17th IFAC World Congress, the International Federation of Automatic Control, Seoul, Korea, pp. 12552-12557, 2008.

[28]   M. Ikeda and D. D. Siljak, "Optimality and robustness of linear quadratic control for nonlinear systems," *Automatica*, vol. 26, no. 3, pp. 499-511, 1990.

[29]   D. D. Siljak, *Decentralized control of complex systems*. Boston, MA: Academic Press, 1991.

[30]   N. J. Ploplys, "Wireless feedback control of mechanical systems," in *Mechanical Engineering*, MSc thesis, University of Illinois, Champaign, 2003, http://mr-roboto.me.uiuc.edu/ARGWeb/theses/NicksThesis/Ploplys_MS2003.pdf.

[31]   N. J. Ploplys, P. A. Kawka, and A. G. Alleyne, "Closed-loop Control over Wireless Networks," *IEEE Control Systems Magazine*, vol. 24, no. 3, pp. 58-71, 2004.

[32]   S. Zampieri, "Trends in Networked Control Systems," presented at the 17th World Congress, the International Federation of Automatic Control, Seoul, Korea, pp. 2886-2894, 2008.

[33]   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), 2004.

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

[35]   "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.

[36]   M. Conti and S. Giordano, "Multihop Ad Hoc Networking: The Reality," *IEEE Communications Magazine*, vol. 45, no. 4, pp. 88-95, 2007.

[37]   H. Yu, C. Harding, and M. S. Hasan, "Overview of Networks and Control," presented at the International Conference on Instrumentation, Control and Information Technology, Japan, pp. 205-210, 2005.

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

[39]   C. E. Perkins and E. M. Royer, "Ad hoc On-Demand Distance Vector Routing," presented at the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 99), New Orleans, LA, pp. 90-100, 1999.

[40]   E. Altman and T. Jimenez, "NS Simulator for beginners," 2003, http://www-sop.inria.fr/maestro/personnel/Eitan.Altman/COURS-NS/n3.pdf, Mar 2006.

[41]   F. D. Pellegrini, D. Miorandi, S. Vitturi, and A. Zanella, "On the Use of Wireless Networks at Low Level of Factory Automation Systems," *IEEE Transactions on Industrial Informatics*, vol. 2, no. 2, pp. 129-143, 2006.

[42]   J. Nilsson, "Real-time Control Systems with Delays," in *Automatic Control*, thesis, Lund Institute of Technology, Lund, Sweden, 1998, http://www.control.lth.se/documents/1998/nilj98dis.pdf.

[43]   P. Albertos, A. Crespo, M. Valles, and I. Ripoll, "Embedded Control Systems: Some Issues and Solutions," presented at the 16th IFAC World Congress, Prague, Czech Republic, 2005.

[44]   T. C. Yang, H. Yu, M. R. Fei, and L. X. Li, "Networked control systems: a historical review and current research topics," *the Journal of the Institute of Measurement and Control*, vol. 38, no. 1, pp. 12-16, 2005.