

# Quadrature Encoder Signal Emulation for Integrated Real-Virtual Environments and Systems

Mohamed Hussein<sup>1</sup>, Philip Moore<sup>2</sup>, Jun-Sheng Pu<sup>2</sup> and Chi-Biu Wong<sup>2</sup>

<sup>1</sup>Faculty of Mechanical Engineering, Universiti Teknologi Malaysia (UTM), Malaysia.  
email: [mohamed@fkm.utm.my](mailto:mohamed@fkm.utm.my)

<sup>2</sup>Mechatronics Research Group, DeMontfort University (DMU), UK.  
email: [prmoore@dmu.ac.uk](mailto:prmoore@dmu.ac.uk)

**Abstract-Integrated Real-Virtuality (R-V) Environments and Systems** involve the investigation on how 3-D virtual environments can be used for the integration of design, simulation and controller program development of a typical manufacturing system. The study evolved around the adoption of 3-D simulation models being used as virtual machines and being controlled by a real controller. The controller can be programmed, tested, verified and validated in real time and the action of the machine is realised in the 3-D model. A new method which employs the use of a control signal emulator has been proposed. Within the study a novel technique for emulating quadrature encoder signals to provide virtual closed loop control of servomotors has been formulated. The deployment of a control signal emulator technique makes the system unique and removes its dependency on specific hardware.

## I. INTRODUCTION

A key challenge facing machine builders is to develop advanced control systems with proven operation in the most timely and cost effective manner, which can then be commissioned on the customer site. The difficulty lies with the fact that more often than not the machine target systems to be controlled are not available to enable a full verification and testing of the control system developed. It can easily take the manufacturer several months for a new production system installation to operate smoothly to specification. The control system can obviously play a very significant role in minimising the down time in the use phase of the production line or the machine.

Having a fully tested control system is vital [1]. The only certain way to test how a control system handles every possible situation is to write the code, install it in the controller, and try it out on the system. Normally, testing takes place during the start-up phase of the system to be controlled which is an expensive, risky and error-prone way of developing control systems [2]. This is also a very costly means of uncovering design flaws. A control system failure can shut down the plant and require expensive modifications. In the worst case, the entire control system may have to be scrapped, redesigned, and rebuilt [3].

Advancement in 3-D modelling allows designed machines or systems being viewed in highly realistic ways. Thus, detailed 3-D models coupled with matching machine system realism, helps in presenting designs to potential customers

even before prototypes are available. Simulation techniques together with some 3-D visualisation can provide tools for the verification of specific designs [4]. The designed machine or system behaviour is observed and studied as it evolves over time. Simulated models help to get the feel of the real system early in the design stage before the system is built.

Bringing reality into simulation is not an easy task. The term emulation is normally used to describe a simulated environment with real time capabilities. Coupling real hardware with the simulation is one of the techniques to provide an emulated environment. However emulation may not necessarily involve the use of hardware. Emulation differs from simulation in several ways. Emulation substitutes software and special hardware for parts of a product's architecture, which permits examining the product or its subsystems in real time, including the function of the control software [5]. An emulator is a "device" that mimics the operation behaviour of some target system. Research has also shown the benefit of using emulator in system control related application. According to LeBaron and Thompson [6], the difference between a simulator and an emulator in system control application is that the emulator is an exact representation of the real controller. Since the emulator is an exact representation of the real controller is it easy to exchange logic between the real controller and the emulator [7]. The use of emulation has allowed testing of control systems faster than in real-time and under safe conditions.

## II. THE REAL-VIRTUALITY SYSTEM

Design, simulation and control of manufacturing machinery and related systems can be seen as a collection of technologies, which if integrated properly can provide highly convenient interfaces between humans, computers and machines. Attempts have been made such that the activities involved were more intuitive to use [8]. The term intuitive reflects an ambition of Real-Virtuality. The term Real-Virtuality (R-V) at first sight, might appear a contradiction of terms and hence needs some explanation. Traditionally, "real" suggests an artefact or physical presence of an item e.g. machinery, components, etc while "virtual" usually refers to computer software model or synthetic environment which is capable of mimicking the behaviour of a chosen system. However, certain "virtual"

models can provide almost perfect behaviour patterns of chosen machinery and hence provide users with what is referred to as a transparent interface, making navigation through, and interaction with virtual environment as real as possible. Here, “real” will refer to actual artefacts, while “R-V” will be used for a combination of real machinery and software models which enable complex procedures to be done with high levels of confidence.

The R-V system is a highly integrated design, simulation, verification and validation environment for realising virtual testing, commissioning and training of a system. In the integrated R-V system the dynamics of the machine system is not a concern. What are of concern are the real-time interactions between the target machine system to be controlled and the control system. In this case, both the target machine system and the associated control system are of interest. The primary concern is the behaviour and controllability of the target machine or system, not the control system. In adopting an underlying integration paradigm, R-V seeks to facilitate functional requirements from different levels and perspectives, explicitly:

- to realise a real-time virtual environment
- to provide a platform for control system design confirmation
- to lift the gap between simulation environment and real environment of machine system

The ultimate objective was to realise machine system solutions with the receptivity, rapidity and agility required by control system developers and machine builders. The basic Real-Virtuality system architecture is shown in Fig. 1.

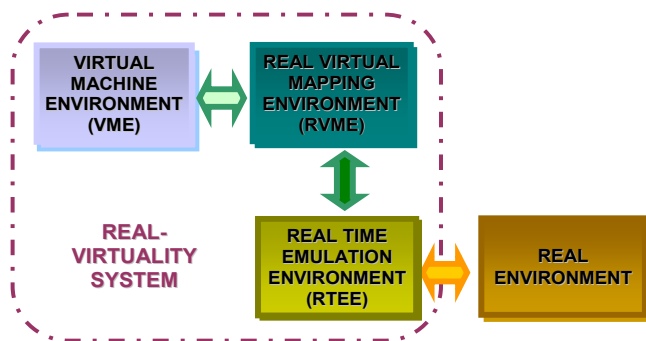


Fig.1 The Real-Virtuality system architecture

The development of the Real-Virtuality (R-V) system focuses on machine level application which involves both discrete and continuous operations. The study involves bridging the real controller (real) with machine simulation (virtual); a technique to bridge between the two has been addressed. In the study, the bridging technique which is based on signal emulation was utilised.

### III. SIGNAL EMULATION

#### A. Signals Classification

Physical system can be classified into either *continuous* or *discrete* or a blend of both. The developed R-V system provides support to both continuous and discrete systems. Thus, it should be able to accommodate all types of sensor and transducer signals. The type of signal existing in machine operation can be divided into three types and are classified into three categories. They are:

- *Discrete Digital Signal*

These are signals from sensors which are normally used for discrete events. They only have two states; ON and OFF or HIGH and LOW. Examples of this type of signal are from limit switches, proximity sensors, push buttons, etc.

- *Analogue Signal*

These are sensors/transducers signals which typically vary smoothly and continuously over time. In most control applications, analogue signals range continuously over a specified current or voltage range. Examples of these types of signals are from potentiometers, tachogenerators, LVDTs, etc.

- *Continuous Digital Signal*

These are signals from sensors which have two states HIGH and LOW but are produced continuously. Examples of these types of signals are from incremental encoders, absolute encoders, etc.

Emulation of the first two types of signals is relatively straight forward, the emulation of the third is found to be very challenging. Continuous digital signals are considered more important than analogue signal for machine applications especially when involving quadrature encoders. Current trends shows that all modern machine uses servo motors which have digital encoders as their feedback element. These digital encoders produce continuous digital signals. It should be noted that the quadrature encoders are most prevalent in industry because of their ability to provide direction of rotation besides its ruggedness and simple wiring [9]. Furthermore, at present<sup>1</sup>, there was no hardware yet available to mimic the continuous digital signals behaviour of quadrature encoder in comparison to analogue signals. Thus, the ability to produce one is considered novel and important to prove the feasibility of the R-V system. In this paper, only the emulation of the quadrature encoder will be discussed.

#### B. Encoder Signal Emulation

For the R-V system to function, it is important to be able to emulate the behaviour of the quadrature incremental encoder. The emulated encoder signal produced is used as

<sup>1</sup> At present there is encoder simulator available in the market. The product is introduced in October 2006 by Plant Control and Automation Pty. Ltd., an Australian company. However the simulator can only be operated manually which is not appropriate for R-V applications.

the input signal to the real controller in continuous feedback operation, a defect signal can easily trigger an error in the controller and crash the whole system. Thus, the ability to mimic the encoder behaviour is crucial.

In order to be able to produce the encoder signal emulation hardware, the quadrature encoder signal behaviour has to be understood. The following are the characteristic of quadrature encoder signal:

- i. the output of an encoder is two continuous 90° shift clock signal, normally named as A and B.
- ii. in a clockwise motor movement, the outputs signal B will lead A and alternately, output signal A will leads B in counter clockwise motor movement.
- iii. encoder signal is normally in the form of square wave with the frequency that may vary between 0 to 80 MHz.
- iv. signal frequencies for both A and B change continuously based on the motor speed. The timing of the frequency change has to be accurate as expected by the controller. Discrepancies between the encoder signal and the expected signal will cause an error to the controller.

In order to achieve the above, the hardware for the encoder signal emulation was designed based on the schematic diagram shown in Fig. 2. Within the circuit, based on a single clock signal, two outputs A and B are generated. Output A is based on the falling edge of the clock signal and B on the rising edge. A signal divider is used in such one output signal is produced after two clock signal edges (either rising or falling) has occurred.

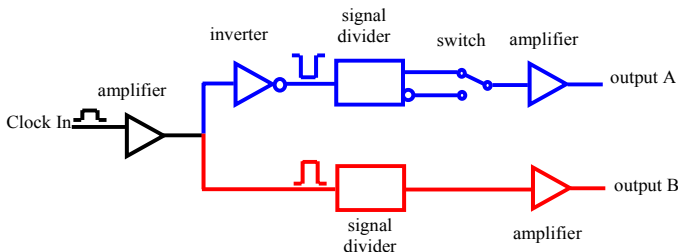


Fig.2 Schematic diagram of encoder signal emulator

A digital switch with external trigger is used to produced an inverted A ( $\bar{A}$ ) signal. An inverted A signal will cause output A leading output B. Hence, emulates the counter clockwise rotation of the encoder. The timing diagram of the clock, output A, output B and inverted output A (i.e.  $\bar{A}$ ) is illustrated in Fig. 3. Amplifiers are used to strengthen the clock signals so that a perfect square wave is maintained throughout the process.

The clock frequency is changing throughout the whole encoder emulation process. Thus, the clock signal generator should able to change its clock frequencies timely and accordingly. A commercial stepper motor driver is used to produce the clock signal for the encoder signal emulator

circuit. The stepper motor driver used is PC-based. The use of stepper motor driver is twofold. Firstly, the stepper motor driver is capable to produce the required clock frequencies. Secondly, there is similarity between stepper motor operation and servo motor operation. Thus, it is capable to adapt with almost all of the motion commands. The circuit diagram for encoder signal generation which is able to emulate four encoder signals is shown in Fig. 4.

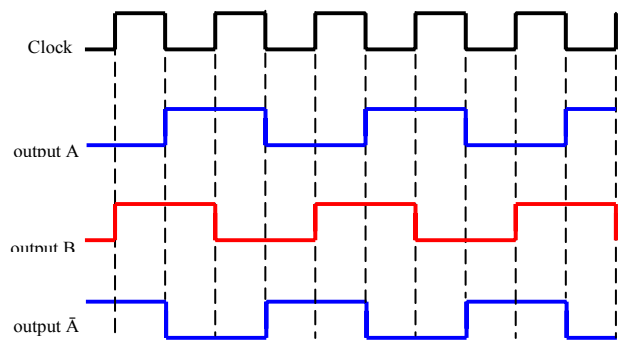


Fig.3 Timing diagram of input clock, output A, output B and output  $\bar{A}$

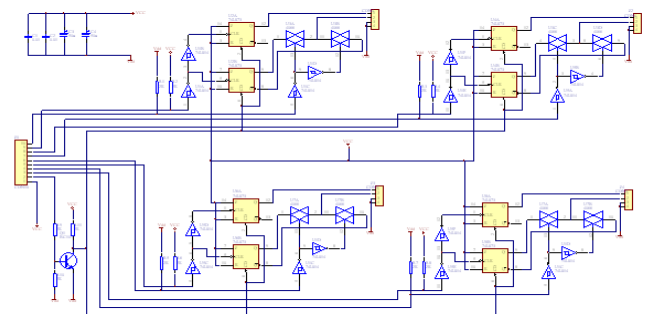


Fig.4 Encoder signal generation circuit

C. Encoder Signal Generation Commands

The industrial controller normally supplies with a set of motion control commands which determine how the motor can be operated. Hence, the R-V system is designed to be able to adapt to all these possibilities. Since the encoder emulation hardware uses an industrial controller stepper motor driver to generate its clock signal, these do not cause any problem. The stepper motor controller has motion control commands which are basically identical to the servomotor motion control commands. Adopting the stepper motor motion control commands allow to provide adequate R-V commands for encoder signals manipulation.

The R-V system prototype is designed to have a set of signal request command. At present, there are six (6) R-V encoder signal manipulations command available. The command identification available and its corresponding function are as shown in Table 1 below.

**Table 1: Encoder emulation command and its respective function**

cmd. id	Motive
0.0	emulates single axis moving with certain speed
0.1	emulates single axis moving with certain acceleration
0.2	emulates single axis moving with certain acceleration
1.0	emulates multiple axes moving in certain relative distance
2.0	emulates multiple axes moving in certain absolute distance
3.0	emulates axes stopping
9.0	reset all encoder parameter
10.0	reconfigure encoder parameter based on new user input

These commands are very basic which is commonly used for servomotor applications. Additional commands can easily be included for the more advance application.

*D. Encoder Emulator Testing and Evaluation*

A demonstrator system was built to illustrate the use of a Real Virtuality (R-V) system. The demonstrator is based on a six degree of freedom (DOF) Silver Reed ARX articulate robot which is available within the Intelligent Machines and Automation Systems (IMAS) laboratory at De Montfort University (DMU). The robot consists of four moving axes (only three was used in the study) and nine sensors, and represents a machine with mixed continuous and discrete operations. The controller used for the study is a NextMove; a commercial PC-based industrial controller produced by Baldor Corporation. Throughout the building of the system, the robot is regarded as a generic machine for industrial applications. IGRIP (Interactive Graphics Robot Instruction Program) simulation package from Delmia Corporation is used for constructing the virtual environment of the machine (the robot). Within the R-V system, the real controller is used to drive the machine 3-D model with the assistance of the emulated control signals.

The test and evaluation on the encoder emulator covers two main aspects, firstly, to test and evaluate the capability of the signal emulators to produce the required signals and secondly, to demonstrate that the encoder emulator is able to provide real-time capabilities to R-V system. The initial test was on the encoder emulator. Each encoder emulator is initiated to produce a motion signal for movement between two positions. The individual emulator encoder speed and position data are then recorded. The configuration for each axis is as shown in Table 2. The speed, acceleration and deceleration units are correspond to the movement type of each individual axis.

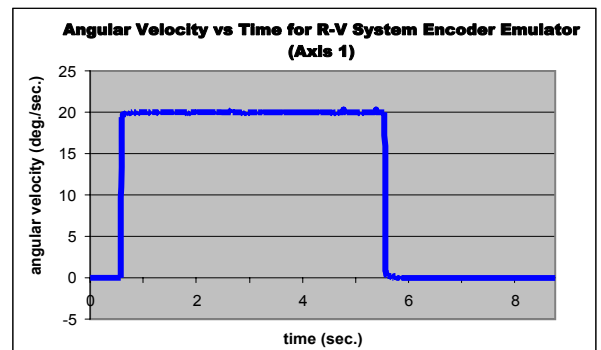
To access the real-time capability of R-V system, two identical NextMove controllers were used to drive the R-V system and the real machine separately. A five loop kinematics test was carried on both the R-V system and real system. Both systems were given the same movement sequences. Comparison was then made between both sets of system data.

**Table 2: Encoder emulator configuration for accuracy evaluation**

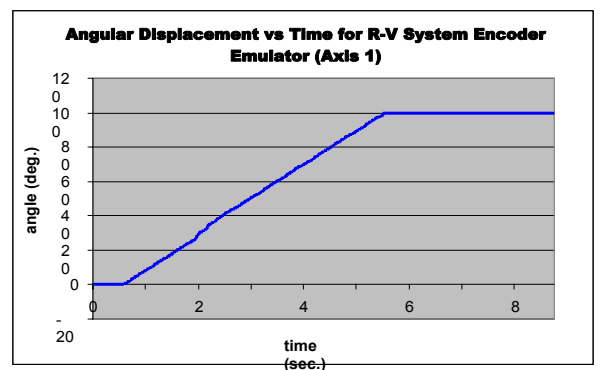
Axis	SPEED	ACCEL.	DECEL.	MOTION
Axis 1	20	1000	1000	0° to 100°
Axis 2	20	500	500	0° to 120°
Axis 3	10	1000	1000	0 mm to 75 mm

IV RESULTS AND DISCUSSION

Fig. 5 to Fig. 7 shows the graphs of velocity and displacement in respect of time for the three axes of the encoder emulators. Comparison is made on the time taken to reach the desired position theoretically and from the tests. The results are tabulated in Table 3.

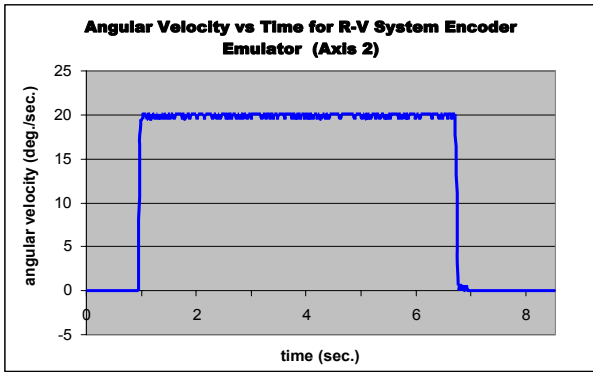


(a)

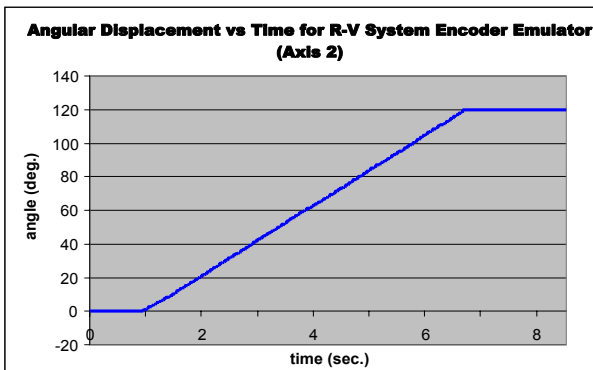


(b)

Fig.5 The (a) angular velocity and (b) displacement profiles of encoder emulator for Axis 1 (0° to 100° movement)

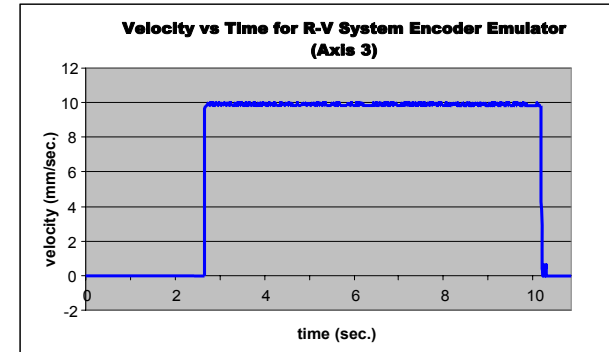


(a)

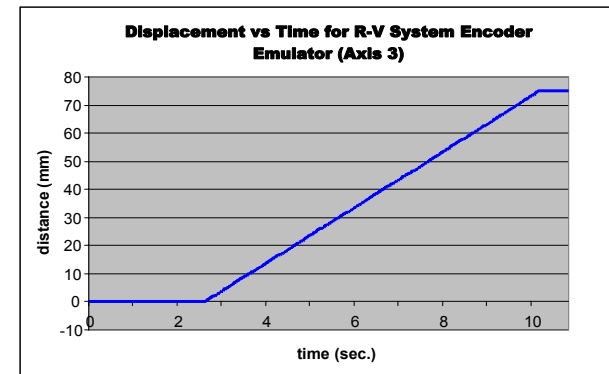


(b)

Fig.6 The (a) angular velocity and (b) displacement profiles of encoder emulator for Axis 2 (0° to 120° movement)



(a)



(b)

Fig.6 The (a) angular velocity and (b) displacement profiles of encoder emulator for Axis 2 (0mm to 75mm movement)

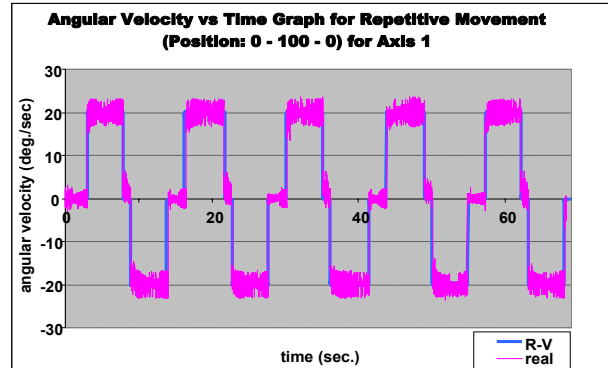
Table 3: Comparison between theoretical and actual time taken for emulated encoder signal to reach a desired position for all axes.

Axis	Movement	$t_{cal.}$ (sec.)	$t_{actual}$ (sec.)	% error
Axis 1	0° to 100°	5.020	5.012	0.159
Axis 2	0° to 120°	6.040	6.135	1.573
Axis 3	0 mm to 75 mm	7.510	7.498	0.160

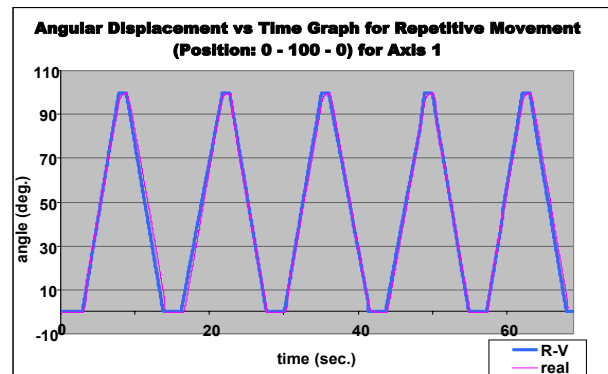
From the comparison it shows that the time difference to reach a target position between theoretical (column 4) and actual (column 5) is not more than 0.1 second with the highest error being 1.573%. If the percentage error can be regarded as the percentage of servo error, then the emulator encoders' capability are considered acceptable because the satisfactory servo error is between 2% to 5% [10].

It should be noted, the encoder emulator for Axis 2 produces the most error. It is apparent in the graph (shown in Fig. 6 (a)) where instability occurs on Axis 2 encoder emulator output. This instability was probably due to electrical noise and/or external interference.

Fig. 8 to Fig. 10 shows the results from five loops tests carried out on both the R-V system and the real system. The position and velocity graphs of the real system are superimposed onto the encoder emulators' position and velocity graphs.

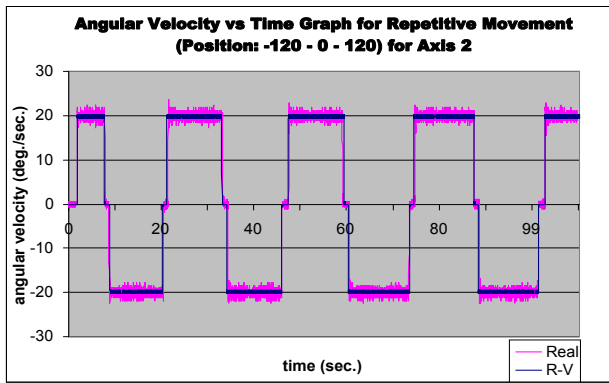


(a)

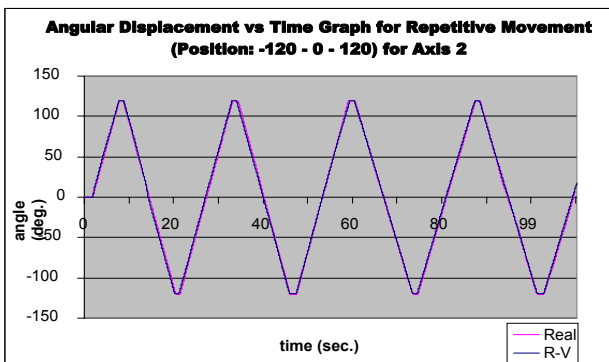


(b)

Fig.8 The (a)angular velocity and (b)displacement profile comparison between R-V system and real system for Axis 1 (0° to 100° movement)

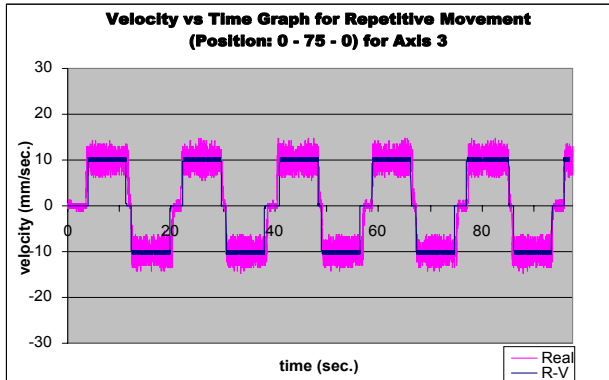


(a)

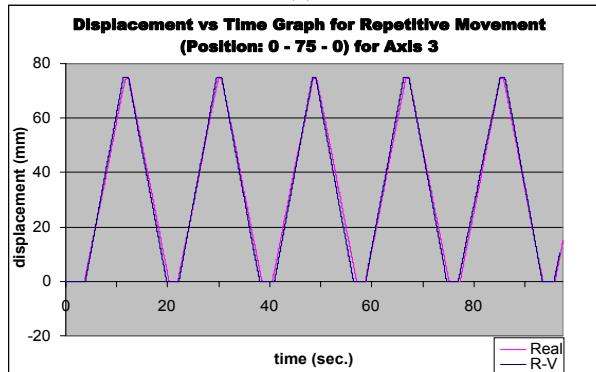


(b)

Fig.9 The (a)angular velocity and (b)displacement profile comparison between the R-V system and the real system for Axis 2 (-120° to 120° movement)



(a)



(b)

Fig.9 The (a)angular velocity and (b)displacement profile comparison between the R-V system and the real system for Axis 3 (0mm to 75mm movement)

From the graphs it clearly shows that the R-V system motion profiles coincide perfectly with the real machine motion profiles for all the three axes. Furthermore, the encoders produce perfectly clean motion profiles when compared to the real system.

The discrepancies in tuning the control parameters (PID parameters) result in unclean motion profile of the real machine; it can be clearly seen in all three sets of graphs. However the servomotor for Axis 2 is seen to be tuned acceptably. Hence, it shows quite a clean motion profile and overlaps perfectly with the R-V system profile.

#### IV CONCLUSION

This paper has describes the general use of R-V system which is intended to help machine designers and machine builders to work with their machine in real-time although the machine itself has not necessarily been built. This new approach is based on control signal emulation. The study has resulted in the formulation of a technique for emulation of quadrature encoder signals. This proposed technique can be used not only for the R-V system environments but also for servomotor control testing and verification. The devised encoder emulator is programmable; hence, it can help to provide the desired signals for the control feedback element.

#### REFERENCES

- [1] F. Auinger, M. Vorderwinkler and G. Buchtela, "Interface driven domain-independent modeling architecture for 'soft-commissioning' and 'reality in the loop' ", Proceedings of the 1999 Winter Simulation Conference, 1999, pp. 798-805.
- [2] A. Pfeiffer, B. Kádár and L. Monostori, "Evaluating and improving production control systems by using emulation", IASTED International Conference on Applied Simulation and Modelling (ASM 2003), Spain, 2003.
- [3] V. Van Doren, "Test your control system with simulation", Control Engineering, September Issue, 2003.
- [4] B. K. Min, A. Huang, Z. J. Pasek, D. Yip-Hoi, F. Husted and S. Marker, "Integration of real-time control simulation to virtual manufacturing environment", Journal of Advanced Manufacturing Systems, Vol. 1, No. 1, 2003, pp. 67-87.
- [5] S. F. Scheiber, "Emulation and control system assurance", Control Engineering, November Issue, 2003.
- [6] T. LeBaron and K. Thompson, "Emulation Of A Material Delivery System", Proceedings of the 1998 Winter Simulation Conference, 1998, pp 1055-1060.
- [7] F. Danielsson and P.R. Moore, Validation, off-Line programming and optimisation of industrial control logic, Mechatronics, Vol. 13. Issue 6, 2003, pp. 571-585.
- [8] R.J. Stone, "Virtual reality: definition, technology and selected engineering applications overview", Mechatronics Forum Paper, May 1998.
- [9] Encoder Products Company (EPC), Product descriptions, <http://www.encoder.com/products.html> [Accessed: 1 March 2007].
- [10] National Instrument (NI), "Understanding Servo Tune", <http://zone.ni.com/devzone/cda/tut/p/id/2923> [Accessed: 6 June 2007]