Evaluation of Real-time Emulators for Future Development of Fire Control Applications

John W. Kelly
David D'Onofrio
James.R.Bates
George Khadar
Vincent Marcopoli
General Dynamics Land Systems Corporation

Patrick O'Heron
MSC Software
Marcella Haghgooie
Applied Dynamics International

ABSTRACT

In order to highlight General Dynamics Land Systems' rapid-prototyping methology, a demonstration of real-time system emulation was conducted. The system emulated was the dynamic behavior of a 1/10th scaled model of a tank turret. A physics based turret model was realized using ADAMS/Real Time. This virtual model was implemented on the Applied Dynamics International (ADI) multi-processor emulation platform. The emulator provided real-time feedback signals to a "COTS" processor which executed the control algorithm using, real-time Linux (RTLinux) operating system. The control algorithm was developed and then validated against the virtual physics model using co-simulation between ADAMS and Simulink. The control program was then embedded into the real-time Linux space via Math Work's Real-Time Workshop. In order to verify the fidelity of the virtual model a physical model, 1/10th scaled, was constructed. A second identical controller drove the physical model. Comparative performance tests were conducted between the physical and virtual model. Presented in this paper is an overview of the development process used in this demonstration. Basic technologies of model development, rapid prototyping and real-time emulation are reviewed. Validation of these technologies and the overall development process is accomplished by performance comparisons between the virtual and physical systems. The results of these comparisons are presented.

INTRODUCTION

In general, a mathematical-model based emulator provides the real-time dynamic response about the bandwidth of interest of the actual subsystem. Benefits of using real-time system emulators are reduction in development time and cost. Therefore, real-time emulators are anticipated to play a critical role in the development of future fire control application at General Dynamics Land Systems.

The benefits of an emulator can be realized both in product development and subsequent product improvement. By emulating the dynamic response of a subsystem, that is not yet fully developed, an emulator expedites the integration and development of the overall system by allowing for simultaneous development of dependent subsystems. In the case of a future gun turret system, a physics-based, real-time emulator allows for the prototyping of the objective controller. Prototyping of the controller is done concurrently with the development and construction of the physical turret. As a result development time is shortened. Ready to be implemented upon the completion of a turret are 1) a physical controller with developed I/O and developed sensor integration and 2) a mature control strategy that accounts for such real-time physical constraints such as system time delay, limited feedback signal bandwidth, etc.

Once the subsystem is fully developed, the real-time emulator can then be utilized in continuing development situations, i.e. product improvement. For future control development and diagnostic analysis it is not practical in terms of cost and resources to have a full-scale turret in the laboratory environment. From a bench-top emulator, a controls engineer can efficiently realize refinements and improvement to the control strategy. Confidence in those refinements and improvement is increased by comparison and validation of the emulator model to the physical plant. From the same laboratory setup, a systems engineer can trouble-shoot other dependent subsystems through their integration with the plant emulator in *hardware in the loop*.

In order to demonstrate usefulness and the effectiveness of real-time emulators in future fire control applications, an emulator was developed and tested for real-time emulation of an artificial gun turret. This artificial gun turret was roughly based on the Future Combat System-Mounted Combat System (FCS–MCS), however, it is one-tenth the size and possesses a structural composition of ABS plastic. A dynamic mathematical-based model was created for this artificial gun turret. This model was then realized in a commercially available emulation platform. In order to validate its real-time emulation, an actual physical model of the artificial gun turret was fabricated.

Presented in this paper is the development methology for the realization of a general real-time emulator. This process is illustrated with the creation of a real-time emulator for the mock gun turret system. In addition, comparative results of the dynamic response between the virtual mode and physical model are presented

THE PROCESS

I. Development of the plant model

For the artificial gun turret the modeling began with building of a structural model using the solid modeling software program Pro/Engineer. The model consists of the turret, the gun, the elevation drive motor inertia and the turret basket. The material properties in the Pro/Engineer model were chosen to be that of ABS plastic. Once the mass properties and geometry configuration were correct, the Pro/Engineer model was ready to be exported to the physics based modeling environment called Adams.

II. Conversion the structural model into a dynamic or physics based model

The turret structural model was converted using the program MechPRO. Once in the ADAMS environment, kinematic joints, couplers and forces were added to enable degrees of freedom to both elevate the gun and transverse the turret as well as allow rotational movement of the gear motor drives systems.

The drive systems for the physical gun/turret were designed to approximate as closely as possible a well behaved linear system. A turret ring gear and coupler were built to allow traverse motion of the turret and a no slip belt drive was constructed between the gear motor and trunnions to elevate the gun. The gear motors chosen were dc motors with zero backlash gear drives.

For the motor model low voltage DC motors mated to a zero backlash gear drive was chosen. This was done for both axes. The motor models were split into two parts. The electrical part consisting of the electrical time constant, back emf and torque constant and summing node were modeled in Matlab Simulink environment. The motor rotor and gearbox were modeled in the ADAMS mechanical environment.

Once the complete gun-turret model was described the MSC.ADAMS software formulated the equations of motion. Using the Euler-Lagrange method, dynamic coupling was achieved between the two degrees of freedom, azimuth and elevation.

III. Control Design and Co-Simulation

With the realization of a virtual gun turret prototype, control development could begin. Controls development was done in MATLAB's SIMULINK. A feature of MSC.ADAMS toolset is the ability to perform co-simulation between a virtual model and a SIMULINK model. Once the controller was designed, it was combined with the motor model in the Simulink environment and co-simulated with the mechanical model. Co-simulation allows the controller to be simulated in its native (Simulink) environment and the physics based model in ADAMS to be solved in its native environments. The benefit of this type of simulation allows each environment to solve its models using the best type of integrators that are well suited for that class of problem. For example, ADAMS has a set of integrators that are tuned for mechanical systems that are intrinsically stiff. Simulink on the other hand is ideally set-up for discrete solutions that in our case had to be solved for the discrete controller. Upon receiving the torque command from Simulink, the ADAMS solver calculates the solutions to its equations resulting in forces and motions of the turret and gun. Motor position and velocities are fed back to Simulink and then used to calculate the controller effort which is then fed to ADAMS and the process repeats. Successful co-simulation results ensured that the controller performance was meeting specification when connected to a higher fidelity plant model and enabled the next phase consisting of generating real time code.

IV. Realization of the simulation model in emulation hardware

For this demonstration the approach taken was to take the ADAMS, gun/turret model and convert it to real time C code and embedded in a Simulink S-function. This S-function was then embedded in the Applied Dynamics International's (ADI) Emulator using ADI's SIMsystem's Advantage software and MATLAB's Real-Time Workshop. This emulator was designed as an open architecture platform specifically designed to implement solutions in real-time. MSC Software was contracted to convert the ADAMS models into real time code. ADI was contracted to work with MSC to ensure that the code would run in real time. Once the ADAMS code was running on the emulator, the motor models were incorporated in the emulator as well. Off line simulations suggested that the sample time of the ADAMS model would have to run at a rate much higher then the controller. The electrical time constant of the motor suggested an even higher sampling frequency that proved to be incompatible with maintaining real time operation. As a result of this the electrical time constant proved to be an order of magnitude greater then the ADAMS code sampling time and therefore was omitted.

V. Integration between virtual plant and controller

For the gun turret demonstration a commercial off the shelf (COTS) controller processor was integrated with ADI platform. Simulink's Real-Time Workshop was used to auto-code the Simulink model of the compensator.

Once the auto-code was embedded in the controller the controls engineer could begin testing the control strategy in real-time.

VI. Validation and Model Refinement

In order to evaluate the performance of the emulator a comparison was made with a physical gun turret. The material for the one-tenth scaled physical model is ABS plastic. It was built by a process called fused Deposition modeling. The model had both an azimuth and gun elevation drive. Motor position and later motor current signals were feedback to the control. The model was capable of continuous revolution. Based on the inertia and weight of the plastic turret, the motors and the gear-train were designed to match the same require specification, i.e. slew rate, as a full size metal turret.

As a result of the comparison between the dynamic responses of the physical plant and virtual model, refinements and or changes can be made to the model to increase its fidelity. For this demonstration the model was improved by the addition of friction. The following sections detail the experimental setup, comparison data and the resulting improvement to the model.

EXPERIMENTAL SETUP

Figure 1 shows the experimental setup for the validation demonstration. On left side are the physical model and its controller and on the right side are the real-time emulator and its controller. Also shown are the gunner's handle and visualization hardware.

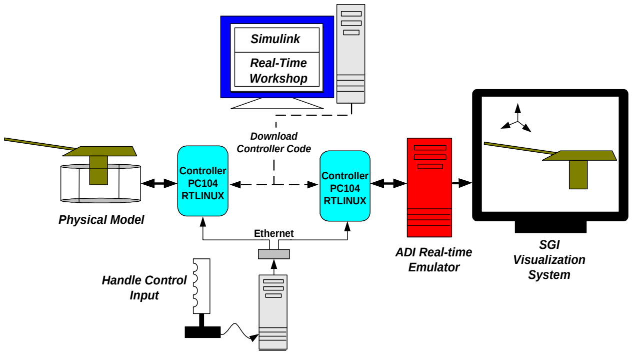


Figure 1. Experimental Setup

Interfaced with the ADI emulator is Silicon Graphics' (SGI) Onyx4 visualization system. This system provided real-time animation of the virtual gun turret. This allowed for quick validation with respect to the physical model. One could simultaneously view both the virtual and physical models dynamic response. (Data was collected for more comprehensive validation and is presented in the next section.) For relatively slow dynamic systems such as the gun turret, visualization provides a rapid and tangible means of evaluation.

The virtual and physical model controllers are exactly the same and can be interchanged. The COTS controller chosen for this project was the PC104 controller module. The PC104 module consisted of a Pentium III 700 MHz processor, built-in internet controller and a data acquisition card (A/D, D/A and quadratue inputs). The operating system was Red Hat Linux patched with FSM Labs' real-time kernel.

The controller for each drive consists of a second order speed observer, deck clearance algorithm, rate loop with compensator. (Later a position loop was added for stabilization.) Real-time Workshop generated auto-code from the Simulink control diagram. Using a custom target configuration designed for real-time Linux, the control source code was built from the Target Language Compiler. This auto-code was embedded in the generic control real-time thread as shown in Figure 2. The three main components of the generic control real-time thread are: the data acquisition component, the previously mentioned embedded controller component and the data output component. The frequency of this time-determinant thread was set to actual sampling rates of existing gun-turret electric-drives

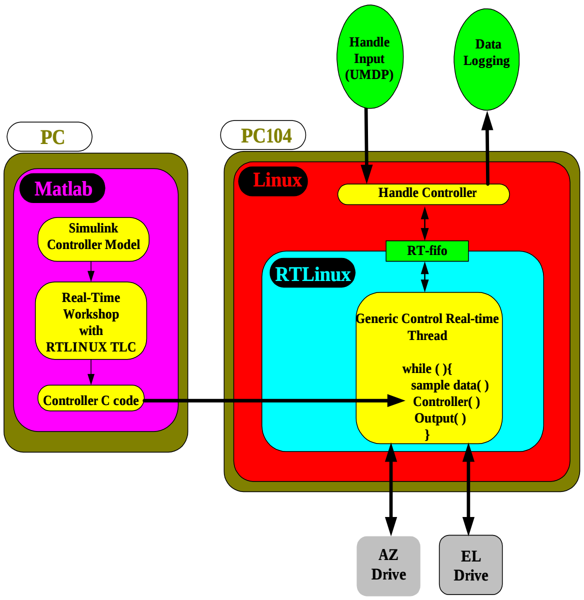


Figure 2. Controller Architecture

Rate or position commands were generated in two ways. The first method as shown in figure 2 was manual commands generated from the gunner's handle. The handle generated commands for both azimuth and evaluation. These commands were simultaneously broadcast to the both controls using UMDP. Real-time fifos were used to import handle data in to the real-time environment. (Also the real-time fifos provided a means for data logging.)The second method for providing commands was to include commands in the Simulink controller block diagram. Both input signals and the controller program were then auto-coded and embedded in the controller. This was the primary method for dynamic response testing. Commands such as step and sinusoidal inputs were built using Simulink Source tools and auto-coded along with the control program.

MODEL VALIDATION

Comparing the response of the physical model to the virtual model it was discovered that the effects of friction were not negligible and had to be accounted for in the model. Figure 3 shows the addition of friction model to the system.

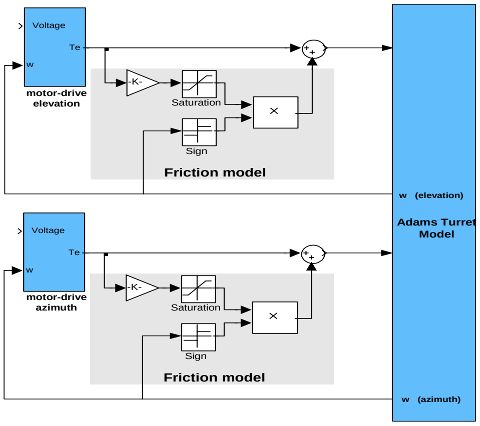


Figure 3. Addition of friction to the model

For expediency the friction models were added to the Simulink part of the model along with the electric motors. In the future the friction models will be part of the mechanical, ADAMS model.

The step response for both the elevation and azimuth were studied. In figures 4-7 the normalized rate of the turret (azimuth) and the gun (elevation) are plotted. Figure 4 compares the azimuth response of the real-time emulator, the physical model and co-simulation model. Figure 5 is close up of figure 4 about the zero rate mark.

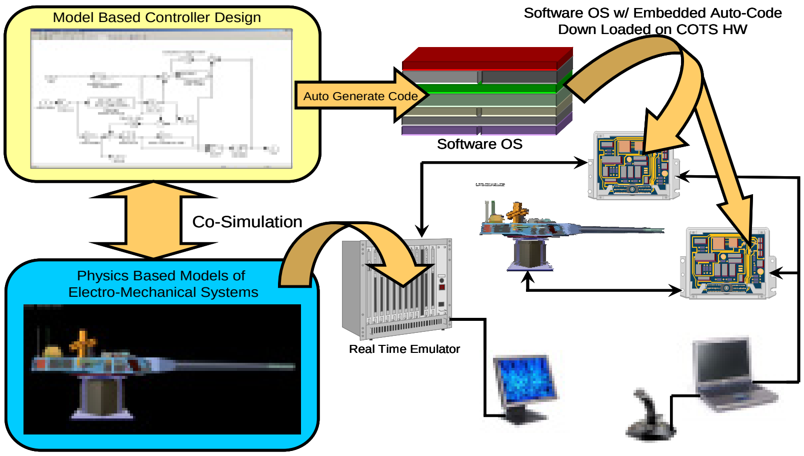

Model Based Controller Design
Auto Generate Code
Software OS w/ Embedded Auto-Code Down Loaded on COTS HW
Software OS
Co-Simulation
Physics Based Models of Electro-Mechanical Systems
Real Time Emulator

Figure 8. Emulator and Control Development Process

As shown the response of the real-time emulator is very similar to the response of the physical system. Figure 6 compares the elevation (gun) response of the real-time emulator, the physical model and co-simulation model. Figure 7 is close up of figure 6 about the zero rate mark.

CONCLUSION

The results shown in figures 4-7 lend credibility to the real-time emulator and its dynamic based model. Therefore a controls engineer would have a degree of confidence in the emulator's response to modified/improved control strategy.

The controllers for both the physical turret and real-time emulator were identical and could be interchanged. Therefore a hardware engineer would have a degree of confidence in testing new hardware or diagnosing current hardware with the real-time emulator. The real-time emulator interfacing with actual hardware forms a *hardware in the loop* type setup.

In summary the process for real-time emulator development is as follows:

1) ***development of the plant model***
2) ***conversion of the structural model into a dynamic or physics based model***
3) ***realization of the model in emulation hardware***.
4) ***comparison with actual system and updating model if necessary***

This process is summarized in figure 8.

REFERENCES

1) C. Dougan, Matt Sherer, “RTLinux POSIX API for IO onReal-time FIFOs and shared Memory”, FSMLabs website
2) Real-Time Workshop for Use with Simulink, July 2002, The MathWorks, Inc

CONTACTS

John W. Kelly
Sr. Engineer Specialist
General Dynamics Land Systems
38500 Mound
Sterling Hts., Mi 48310

David J D'onofrio
Sr. Engineer Specialist
General Dynamics Land Systems
38500 Mound
Sterling Hts., Mi 48310
Phone: 586-825-8919
donofrid@gdls.com

Marcella Haghgooie
Business Development Consultant
Applied Dynamics International
www.adi.com
Phone: 734-973-1300 x302
mmh@adi.com

Patrick O’Heron
Business Development Consultant
MSC Software
Phone: 734-913-2548
patrick.oheron@mscsoftware.com

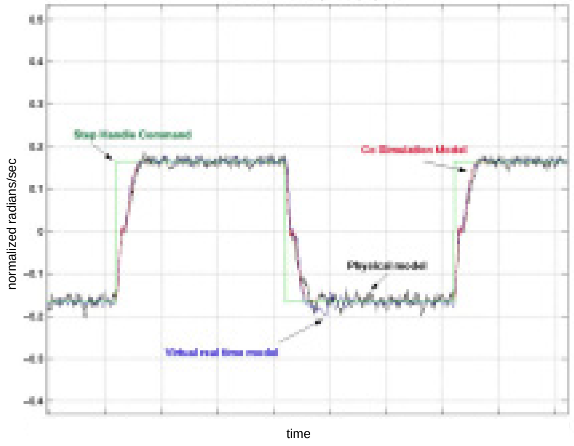


Figure 4 Azimuth (turret) step response

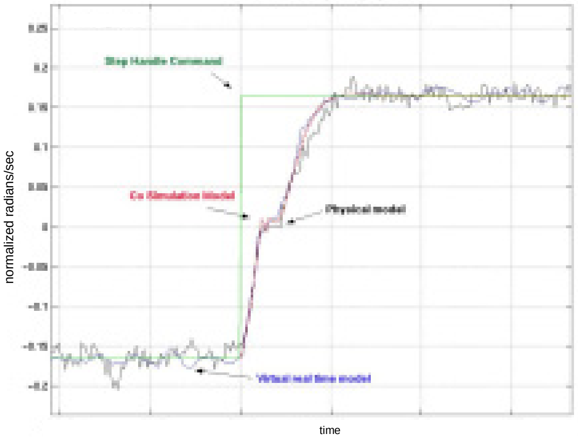
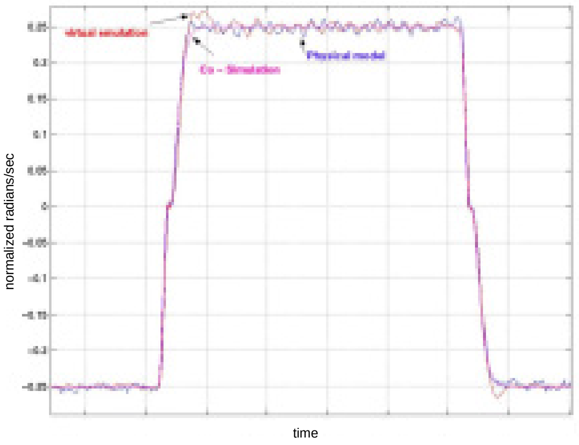


Figure 5. Close-up Azimuth (turret) step response



[illegible]

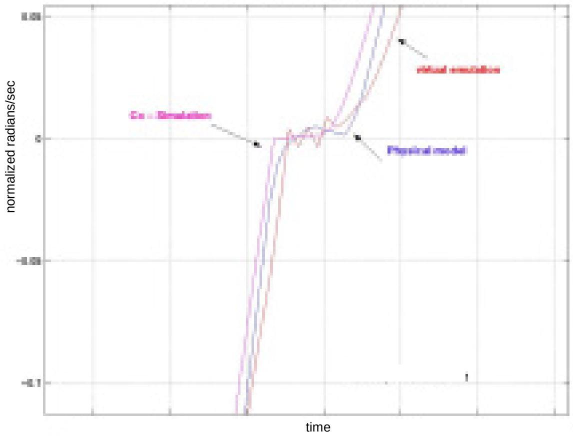


Figure 7. Close-up Elevation (Gun) step response