Decoupling of Natural Systems in Multi-Rate Parallel Simulations

Rodrigo Leonard (leonard@engr.sc.edu), Roger A. Dougal (dougal@engr.sc.edu)
University of South Carolina
Columbia, SC 29208

Keywords: decoupling, distributed, multi-rate, parallel simulation.

Abstract
We describe a method of decoupling networks for the purpose of running a simulation in parallel with each separate section executing with different time steps. The speed advantages are system-dependent, but a particular scenario where the method is applicable is studied, resulting in a speedup of more than 5 when running on a cluster of two computers.

1. INTRODUCTION
A model-based design environment for large systems must execute large simulation models rapidly and efficiently or many of the advantages of simulation-based design are devalued. One method for speeding up a simulation is to run the computation in parallel on multiple computers. Most parallelization methods are focused on "embarrassingly parallel" problems; for example MATLAB’s distributed computing toolbox can automate the process of running multiple versions of one simulation, each having different inputs [1], or by requiring the user to apply his own algorithm to the problem at hand via MPI [5]. In our research, we use the Virtual Test Bed (VTB) software, and explore methods for performing a single simulation in parallel, by developing methods to decouple a large system into separate parts that can be run on separate computers. Other examples of uncoupling methods for multi-rate simulation can be found in literature, such as the simultaneous-solution direct integration methods described by Moreira in [2]. Non-iterative simultaneous-solution methods such as the one described by Moreira depend on interpolating the missing steps of the systems with slower dynamics. The method described here proposes a different link with the possibility of a more accurate interpolation for systems with known first-order behavior at the coupling point.

A particular focus of our work is on the case when one part of the simulation has fast dynamics while other parts have slow dynamics. Thus, different time steps can be used in the separate simulators. Given sufficient system complexity, this method has the potential to greatly increase net simulation speed.

2. SIMULATION INTERFACE
2.1. Signal-Based Coupling
Signal coupled couplings are connections where each model is responsible for its outputs based solely on its received inputs. For the case where the system is separated at a point of signal coupling, little difficulty is encountered, since there conservation of energy is not inherent in the connection point. VTB sends and receives values between the involved simulations before the stepping process. This necessarily introduces a single step delay at the coupling point, and it is thus desirable to perform the decoupling at a place where this delay already exists, thus replacing a delay block, as shown in Figure 1. The delay should correspond to one time step as seen by the part of the system originating the signal. It is up to the system designer to determine where such delay blocks may be added with minimum impact to the system, if a delay block does not already exist, and its impact may be confirmed in a monolithic simulation.

Figure 1. Decoupling a signal system into two separate systems

In a multi-rate simulation, values received from a simulation with a smaller time step are averaged, while values received from a simulation with a larger time step behave as a zero-order hold. It is sometimes the case, especially if the signal-coupling point occurs at a feedback loop, that this naïve solution is insufficient to maintain system stability. However, given knowledge of a specific system, a user may create VTB blocks for different coupling interfaces, which may use assumptions to enhance accuracy and stability, as well as compensate for the delay. The same is true for natural-based coupling, and one such interface is described in the following sections.
2.2. Natural-Based Coupling

2.2.1. Power-Hardware-in-the-Loop Interface

The natural-based coupling studied here is a close adaptation of a method previously used in VTB for power hardware-in-the-loop simulations (real-time hardware-in-the-loop simulations that maintain conservation of power at the coupling point between the simulation and hardware portions of the system). We represent the terminal response of a subsystem by a first-order approximation. Two versions of the coupling interfaces are necessary, one for the case in which the across variable can be approximated as a first-order system, and another for which the through variable can be approximated in the first-order system form [3]. Without loss of generality, the case in which the through variable is approximated as a first-order system is explained in this paper.

In an electrical system, the through variable is current and the across variable is voltage. The interface is described in Figure 2. The values for equivalent resistance and current source are dynamically computed based on the independent solution of the decoupled subsystems.

![Figure 2. Time-variant first-order nature coupling interface](image)

The current at the coupling point (I_2 in Figure 2) can be approximated by the form in Equation 1. The software implementation solves the equation using the trapezoidal integration method. The solution is given in Equations 2, 3, and 4.

\[
\frac{di_2}{dt} = a \cdot i_2 + b \cdot v_1
\]

\[
i_2 = \alpha \cdot v_1[k-1] + \beta \cdot i_2[k-1] =
\]

\[= G_{eq}[k] \cdot v_1[k] + I_{eq}[k]
\]

\[
\alpha = \frac{b \cdot T_s}{1 - a \cdot T_s}
\]

\[
\beta = \frac{1 + a \cdot T_s}{1 - a \cdot T_s}
\]

\[G_{eq}[k] = \alpha
\]

\[I_{eq}[k] = \beta \cdot i_2[k-1]
\]

In addition, a compensation current-source is used, for when the system reaches steady-state and numerical issues arise [4]. This current source is given by Equation 5, where \(e[k]\) is the difference in current at the interface point between the two portions of the simulation and \(K_i\) is a constant of the same magnitude as the determinant of the matrix shown as Equation 6.

\[I_c[k + 1] = I_c[k] + K_i \cdot \frac{e[k] + e[k - 1]}{2} \cdot T_s
\]

\[
\begin{bmatrix}
  v_1[k - 1] & i_2[k - 1] \\
  v_1[k - 2] & i_2[k - 2]
\end{bmatrix}
\]

2.2.2. Multi-Step Simulation Additions

Since the interface described above is well understood, we can use the same behavior to interpolate missing data points when the time steps between the subsystems are not equal. The portion of the simulation with the greater time-step merely performs the coupling as above, but with averaged values received from the portion with smaller time-steps. The portion of the simulation with the smaller time-steps, however, uses the first-order approximation to determine the values from the other simulation that occur in between its larger time-steps. It accomplishes this by using Equation 1, where \(a\) and \(b\) can be solved from Equation 2, using the greater time-step as \(T_s\).

3. PERFORMANCE ANALYSIS OF THE COUPLING SCHEME

3.1. Ideal Case

The performance of the system and its scalability is dependent on multiple factors. For every two sections of a system, the factors are the time it takes to process a single time step, \(t_p\), the difference in time steps \(t_s\) in between two systems, and the communication time \(t_c\) between the clustered computers performing the simulation which depends on bandwidth, latency, and overhead associated with scheduling the communication between the subsystems. The overhead increases with the number of total systems and the number of connections between the systems.

The parallel execution will be faster than the sequential execution if \(t_c\) does not dominate execution time. In addition, the parallelization is most efficient when the condition in Equation 7 is satisfied.
\[ \frac{I_{p1}}{I_{p2}} = \frac{I_{s1}}{I_{s2}} \] (7)

3.2. Example System

A system that demonstrates the potential of this multi-step coupling system in speeding up simulations is shown in Figure 3. The system consists of a speed control for a DC motor, since the dynamics of the motor provides us with ideal conditions for the method described here. The current behaves as a first-order system, and its slower dynamics means that the plant itself can run at a larger time-step without great loss in accuracy. In contrast, the switching control requires a smaller time-step. Additional motors exist in the plant simulation to artificially increase computational complexity.

Figure 3. Un-partitioned example system. This system will be broken up into a plant section and control section for the distributed multi-step simulation.

The system above can be divided using the first-order natural coupling system at the interface between the buck converter and the controlled DC motor. In addition, a signal-coupling port is used between the speed sensor and its respective control input. The resulting subsystems are shown in Figures 4 and 5.

Figure 4. Control portion of partitioned system. The control portion of the system in Figure 3, which must execute at a smaller time step than the plant portion.

Figure 5. Plant portion of partitioned system. The slower dynamics portion of the system in Figure 3, which may be executed at a greater time step.

The simulation takes 156.150 seconds to complete un-partitioned. The result is included in Figure 6.

Figure 6. Reference and motor speed simulation result of un-partitioned system. The figure shows the behavior of the system in the original simulation, before partitioning. This system will be used to compare the results with the distributed and multi-step simulations.

3.3. Partitioned System Results

The result of the simulation with the partitioned system with both sections running under the same time-step of 0.0001 seconds is shown in Figure 7. The simulation takes 129.328 seconds to complete.
Figure 7. Speed simulation result from partitioned system with same time step for both portions. The result of the un-partitioned system is in blue, but is obscured by the result of the partitioned system which perfectly tracks it.

The small speed advantage is due to the lack of computational complexity on the control portion compared to the plant portion. A performance benefit can still be seen, and a system where the performance complexity of each portion is comparable would scale better, as long as the amount of coupling points is kept small.

Now the same simulation is performed, but the time-step of the plant is increased to 0.001 seconds. The result is shown in Figure 8, and the simulation takes 26.797 seconds to complete.

The magnitude of the steady-state ripple remains approximately the same in all three simulations. The large control system error is due to the naïve signal-coupled feedback loop. The natural coupling algorithm can be seen to perform extremely well by the example in Figure 9 which shows the open-loop behavior of the motor in a multi-step decoupled system at the same time steps. The motor shaft speed in the multistep partitioned system completely overlaps the value acquired in the un-partitioned system.

Figure 8. Speed simulation result from partitioned system with larger time step for plant portion. The greater-time step distributed system is in red, the un-partitioned original system is in blue.

Figure 9. Open-loop motor speed simulation result from partitioned system with larger time step for plant portion. Demonstrates the error in response-time in the example simulation is due to the naïve signal-coupled algorithm and not the natural-coupled one. The greater-time step distributed system is in red, the un-partitioned original system is in blue.

We can also observe the performance of the closed-loop un-partitioned vs. partitioned system in tracking a sinusoidal reference in Figure 10. The systems perform almost identically, with slightly more ripple from the distributed system with larger time step.
Figure 10. Speed simulation result from partitioned system with larger time step for plant portion following a sinusoidal reference. The greater-time step distributed system is in red, the un-partitioned original system is in blue.

As we can see, the performance of the system remains reasonably accurate, and the speedup is significantly greater, due to the factor of 10 increase in time step in the more computationally complex portion of the system.

4. CONCLUSIONS

This paper contains the description of a method for uncoupling a system for parallel simulation with multiple time-steps. Performance benefits are highly dependent upon the system being used, but the example simulation performed demonstrates the potential of this method for suitable systems.

The speedup provided by the parallel simulation was only 1.21 due to the difference in computational complexity between the two sections, but the time-step increase in the plant resulted in a 5.83 speedup, which can be expected anytime the complexity of the system is greater than the overhead involved in performing the communication between the partitioned system.

Acknowledgements

This work was supported in part by the U.S. Office of Naval Research under grant N00014-02-1-0623 and N00014-03-1-0434.

References


Parallel Programming with MatlabMPI.