A MULTI-THREADED COMPUTING ALGORITHM FOR
PURE SIMULATION OF COMPLEX SYSTEMS IN
SIMULINK

Master’s Thesis Presentation

Benjamin C. Snyder
October 26, 2006

Advisor: Dr. Sohel Anwar

Department of Mechanical Engineering
Indiana University Purdue University Indianapolis
Cummins Inc.
Dynamic System Modeling

Diesel Engine Development

- Gas Flow
- Chemical Reactions
- Material Deformations
- Kinematics
Modeling Limitations
System Simulation Example

- Time Step: 2 ms
  - SimTime = 1 hour
  - RealTime = 3.8 hours

- Time Step: 20 ms
  - SimTime = 1 hour
  - RealTime = 0.7 hours

Driveline Oscillation
- 2000 RPM, 6 cylinders
- Frequency = 100 Hz

Time Step $\leq 5$ ms
Main Objective

Develop a means of executing a Simulink model as several “light” or “parallel” processes with multiple sampling rates
Current Method
Single Rate, Single Thread

A = B = 20 ms

Time (ms):  20  40  60  80  100  120  140
Process:   AB  AB  AB  AB  AB  AB  AB  AB
Multi-Rate, Single Thread

A = 60 ms  
B = 20 ms

Time (ms): 20  40  60  80  100  120  140
Process: B B AB B B AB B B
Multi-Rate, Multi-Thread

A = 60 ms
B = 20 ms

Time: 20 40 60 80 100 120 140
Process: B B B B B B B B B

A A
Previous Work

- CFD: grid computing
- MATLAB-P: MIT research program
- Proprietary: Cummins real time threading
Model Development

### Parameters

<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Mass (kg)</th>
<th>Spring Coefficient (N/m)</th>
<th>Damping Coefficient (Ns/m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass 1</td>
<td>1</td>
<td>100</td>
<td>2500</td>
</tr>
<tr>
<td>Mass 2</td>
<td>1</td>
<td>10</td>
<td>25</td>
</tr>
<tr>
<td>Mass 3</td>
<td>1</td>
<td>1</td>
<td>0.25</td>
</tr>
</tbody>
</table>

![Diagram of Mass Subsystems](image)
Step Response
(f1 = 100 N)

\[
\begin{cases}
x_1 = 0.0254 & -0.0683 & 0.0030 & 0 & 0 & 0 \\
x_2 = -0.0036 & 0.0317 & -0.0854 & 0.0154 & 0.0035 & -0.0017 \\
x_3 = 0 & -0.0011 & 0.0112 & -0.0068 & -0.2071 & 0.1636 \\
\end{cases}
\]

\[
\begin{bmatrix}
e^{-0.5741 t} \\
e^{-0.4546 t} \\
e^{-3.0058 t} \\
e^{-8.3467 t} \\
e^{-29.9536 t} \\
e^{-79.6652 t} \\
\end{bmatrix}
+ 0.004 u(t) \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}
\]

Exact Step Response

- Blue: Mass 1
- Green: Mass 2
- Red: Mass 3

Displacement (m)

Time (s)
Sinusoidal Response
(f1 = 10,000\times\text{sine}(20t) \text{ N})

\begin{align*}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} &= \begin{bmatrix} -0.5995 & 3.1562 & -0.1059 & 0.0005 & -0.0001 & 0 \\ 0.0842 & -1.4638 & 3.0364 & -0.2269 & -0.0101 & 0.0037 \\ -0.0011 & 0.0501 & -0.3993 & 0.0996 & 0.5939 & -0.3718 \end{bmatrix} + \begin{bmatrix} e^{-79.6652t} \\ e^{-29.9536t} \\ e^{-8.3467t} \\ e^{-3.0058t} \\ e^{-0.5741t} \\ e^{-0.4546t} \end{bmatrix} + \begin{bmatrix} -2.4512 \\ 1.4236 \\ 0.0285 \\ 2.2948 \\ -0.6240 \\ -0.0723 \end{bmatrix} 
\end{align*}
Simulink Model

\[ m\ddot{x} + \left( c_{\text{above}} + c_{\text{below}} \right) \dot{x} - c_{\text{above}} \dot{x}_{\text{above}} - c_{\text{below}} \dot{x}_{\text{below}} + \left( k_{\text{above}} + k_{\text{below}} \right) x - k_{\text{above}} x_{\text{above}} - k_{\text{below}} x_{\text{below}} = f \]
Simulink Model
Simulink Model
Step Response Error
Simulink Model
Sinusoidal Response Error

![Percent Error in Responses](chart_image)
Multi-Rate Model Development

- Integrator
- Discrete-Time Integrator
- Rate Transition

Equation of Motion

Library: massspringdamper/library/Multi Rate Discrete Integrator Equation of Motion
Multi-Rate Model Development
Multi-Rate Model Results

Percent Error in Responses

- Mass 1
- Mass 2
- Mass 3

Percent Difference in Position Response vs. Time (s)
Multi-Rate Model
Results

Mass 1 Response

Displacement (m)

Time (s)
Parallelizing Simulink

- Interpretive Mode
- Compiled Mode
Compiled Mode

Model Source Code
- model.c
- model.h
- model_data.c
- etc...

Common Source Code
- rsim_main.c
- rsim_engine.h
- rsim_mat.c
- etc...

Code Generation

Compilation

model.exe
Parallelizing the Model

Common Source Code
- rsim_main.c
- rsim_engine.h
- rsim_mat.c
- etc...

Model Source Code
- model.c
- model.h
- model_data.c
- etc...
Program Architecture

Main Function

- Onestep Function
- Initialization Tasks
- Start Clock Loop
  - Sets Trigger Vector
  - Call to Onestep
- End Clock Loop

Model Functions

- Solve a portion of the model for a single time step
- Calls Model Functions Based on Trigger Vector
- Initialization Tasks
- Start Clock Loop
  - Sets Trigger Vector
  - Call to Onestep
- End Clock Loop

Program Architecture

<table>
<thead>
<tr>
<th>Time (ms)</th>
<th>Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 40 60 80 100 120 140</td>
<td>B B AB B B AB B</td>
</tr>
</tbody>
</table>

[0,1] [0,1] [1,1] [0,1] [0,1] [1,1] [0,1]
Modificed Program Architecture

- Wrapper Function
  - Start Local Clock
  - Loop
  - Call onesteponestep
  - Thread Management
  - End Local Clock
  - Loop

- Model Functions
  - No Change!

- Main Function
  - ((rsim_main.c))
  - Initialization Tasks
  - Create Threads (wrapper)
  - Start Clock Loop
    - Sets Trigger Vector
    - Thread Management
  - End Local Clock Loop
Evaluation

Criteria: overall processing time
(100 sec SimTime)

Test Cases:
- Single step size (0.001), Single thread
- Single step size (0.010), Single thread
- Multi-step sizes (0.001, 0.010, 0.100), Single thread
- Multi-step sizes (0.001, 0.010, 0.100), Multi-threads

Varying artificial computational load
Thread Confirmation

Single Thread

Multiple Threads
Results

Model Evaluation
Processing Time

Artificial Computational Load vs. Processing Time

- Single Rate (0.010), Single Thread
- Single Rate (0.001), Single Thread
- Multi-Rate, Single Thread
- Multi-Rate, Multi-Thread
Conclusions

- At a given level of detail, the implementation of multiple rates and multiple threads show substantial processing time savings.
- Multiple rates and multiple threads allow for increased detail at a smaller processing time cost.
- Rate transition blocks bring a considerable amount of processing load to the model.
Acknowledgements

Cherian Olikara
Chief Systems Architect
CyberMetrix, Inc.

Marten Dane
Senior Engineer
Cummins Inc.
Questions?
POSIX pthread Tools

- Defined Types
  - pthread_t
  - pthread_cond_t
  - pthread_mutex_t

- Functions
  - pthread_create
  - pthread_mutex_lock
  - pthread_mutex_unlock
  - pthread_cond_wait
  - pthread_cond_signal
  - pthread_join
Numerical Methods

Forward Euler: (first order accurate)

\[ y_{n+1} = y_n + hf(t_n, y_n) \]

Trapezoid Method: (second order accurate, but implicit)

\[ y_{n+1} = y_n + \frac{h}{2} \left[ f(t_n, y_n) + f(t_{n+1}, y_{n+1}) \right] \]

Heun’s Method: (Combination of trapezoid and Forward Euler)

\[ y_{n+1} = y_n + \frac{h}{2} \left[ f(t_n, y_n) + f(t_{n+1}, y_n + hf(t_n, y_n)) \right] \]