MultiBody Dynamics Analysis Software on Real Time Distributed Systems

Pierangelo Masarati
Marco Morandini

Dipartimento di Ingegneria Aerospaziale
Politecnico di Milano (Italy)

One-day meeting on:
RTAI, present and future
July 8th, 2004
Outline

- Objectives and approach:
  - problem description & requirements
  - software requirements
- Key results:
  - software development
  - model rationalization
  - software performances improvement
- Concluding remarks
Objective

- Develop a *General Purpose Real-Time* experiment simulator:
  - hard real-time capabilities
  - minimal modeling limitations
  - commonality of modeling
- Develop real-time models and controls for a wide spectrum of applications
  - model accuracy
  - analysis accuracy
  - distributed control
Real-Time Simulation

• Requirements:
  • exploit OS multitask/memory protection paradigms
  • use generic libraries (algebra, communication, …)
  • use general purpose simulation software
  • use/develop Open-Source software
  • run on "low" cost platforms (dual Athlon)
Real-Time Simulation

- Proposed solution:
  - RTAI (Real-Time Application Interface) for Linux
    http://www.rtai.org/
  - RTnet (Hard Real-Time Networking for Linux/RTAI)
    http://www.rts.uni-hannover.de/rtnet/
  - MBDyn (MultiBody Dynamics)
    http://www.mbdyn.org/
  - RTAILab data acquisition/control code generation
    http://www.rtai.org/
Real-Time Sim.: Status

- Real-time software:
  - Linux/RTAI mature (industry level applications)
  - simulation software real-time enabled

- Multibody software development:
  - MBDyn multibody mature, multidisciplinary
  - development lines drawn, partially investigated

- Multibody model of COMAU SMART robot:
  - redundant coordinate set model (>100 unknowns)
  - dynamic friction models
  - timings estimates available (up to 2KHz on Athlon 2.4 Ghz equivalent)
  - distributed control via RTAILab/RTnet
Real-Time Simulation

Traditional real-time simulations:
- minimal set (ODE)
- explicit integration
- specialized topology
- bound to OS internals

Proposed real-time solution:
- redundant set (DAE)
- implicit integration
- general topology
- POSIX compliant

- fast, fast, fast!
- difficult to extend
- code/tool duplication

- performance limitations
- code/tool commonality
- modeling flexibility
Real-Time Enabled Software

Conventional simulation software

- Avoid / wrap system calls
- Statically preserve stack / initialize resources
- STL containers: memory pools
- Insert few task execution / control statements
- Add real-time I/O communication provisions

~180,000 LOC

Real-time enabled simulation software
Real-Time Enabled Software

- Distributed Real-time software:
  - hard RTnet for control loop
  - soft RTnet for monitoring
MBDyn: Features

- Dynamic analysis of deformable multibody mechanical systems
- Integrated analysis of hydraulic, electric and generic, control related components
- Aeroelastic, rotorcraft-oriented analysis
- Open-Source http://www.mbdyn.org/
MBDyn: Analysis Description

- Dynamics analysis: solution of Initial Value problems (IVP)
- Static/kinematic analysis modeled as downgraded dynamics analysis
- Eigenanalysis by Proper Orthogonal Decomposition of time series
- Typical analysis procedure consists in performing "virtual experiments"
MBDyn: Analysis Description

- **Bodies**
  - Rigid
  - Deformable

- **Joints**

- **Rigidity**
  - Lumped
  - Beams
  - Modal

- **Libraries:**
  - Mechanical
  - Aerodynamic
  - Hydraulic
  - Controls

- **Absolute/relative position, orientation, vel., acc.**
- **Nonlinearities:** geometry, contacts, tires, friction
- **Interaction with deformable bodies**

- **Rigid/deformable blade element**
- **Rotor inflow**
- **State-space on modal bodies**

- **CFD coupling**

- **Pipes, actuators, valves, sources, sinks**

- **Electric motors**
- **Strain, displacement, acceleration sensors**
- **System modeling/signal processing elements**
- **Programmable elements**
Real-Time Experiment

Virtual Experiment as a cheap and safe replacement of experiment setup

Real-Time Virtual Experiment
Simulation Improvements

Software:
- sparse matrix/vector handling during assembly
- sparse solution handling
- assembly parallelization
- solution parallelization
- hardware improvements

Model:
- topology rationalization
- constraints idealization
- deformability reduction (and elimination)
Software Improvements

Guidelines:
- beneficial for standard solution as well
- assembly/solution: ~ 70-80 vs. 25-15 % CPU time

Results: speed-up
- assembly in compressed matrix form: -10/-20 %
- multi-threaded assembly: -0/15 %
- dense solver: +50/-40 %
- iterative solver (worst case limitation)
- multi-threaded solution (n.a. yet)

Overall: -20/-45 %
Comau robot model

Model description:

- rigid bodies, plane rotation joints
- friction
- 6 DOFs, 120 unknowns
- from 500 Hz to 2kHz
  (Athlon XP 2400+)
Comau control and monitor
Concluding Remarks

- General purpose real-time multibody simulation is feasible
- Performances on robot simulations meet requirements
- There is room for software/model improvements:
  - electric motors
  - feed-forward
  - parallel solution
- Real-time software optimization beneficial for conventional GP simulations as well
Previous works


- Acknowledgements to Matteo Martegani, Michele Attolico and Giuseppe Quaranta.