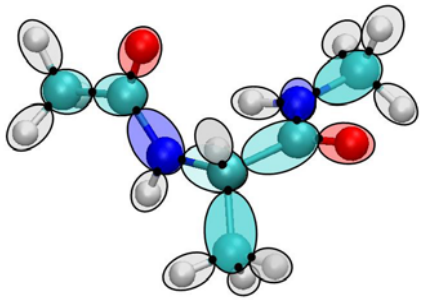
Adaptive RNA Simulation using Molmodel

Kishor D. Bhalerao

Computational Dynamics Lab
Department of Mechanical, Aerospace, and Nuclear Engineering
Rensselaer Polytechnic Institute, Troy, NY 12180-3590

kishor8dm@gmail.com

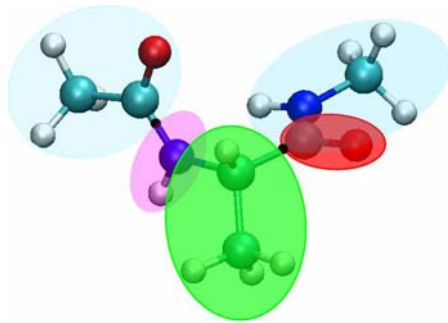
Adaptive MD: Metrics



Reduce fidelity

Change model resolution as required

Increase fidelity

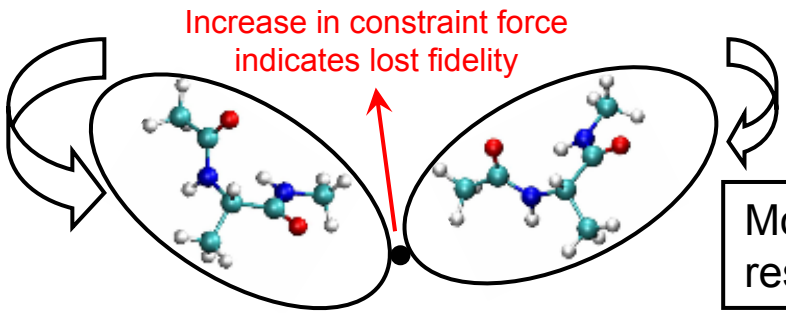


Reduce Fidelity

Spatial configuration determined by the state vector of the system (joint angles). If standard deviation of a joint angle calculated over a moving window of user specified time-length is less than a user specified threshold, the joint is locked.

Increase Fidelity

Activate or release a locked joint if the constraint forces are higher than a user specified threshold.



Motion enforced by rest of the system

Thresholds can be adaptively changed as the simulation progresses. The goal is to focus computational effort at active locations.

kishor8dm@gmail.com

Adaptive RNA Simulation with Molmodel

Molmodel

Define RNA structure using Molmodel interface

SimTK::CompoundSystem

SimTK::SimbodyMatterSubsystem

SimTK::DuMMForceFieldSubsystem

SimTK::RNA

Set initial coarse grain structure using

RNA::setCompoundBondMobility()

RNA::setRedisueBondMobility()

Set simulation parameters using functions from the class SimTK::DuMMForceFieldSubsystem

User defined values in main()

Real *AngleThreshold*

Real *ForceThreshold*

int *LengthofMovingWindow*

Adaptive DCA Solver

system, matter, state

kishor8dm@gmail.com

Adaptive RNA Simulation with Molmodel

Adaptive DCA Solver

DCAPreprocessor::Init()

Extract mass and inertia properties using MobilizedBody::getBodyMassProperties()

Two-handle equations for body k

$$\mathcal{A}_1^k = \zeta_{11}^k \mathcal{F}_{1c}^k + \zeta_{12}^k \mathcal{F}_{2c}^k + \zeta_{13}^k$$
$$\mathcal{A}_2^k = \zeta_{21}^k \mathcal{F}_{1c}^k + \zeta_{22}^k \mathcal{F}_{2c}^k + \zeta_{23}^k$$

Matrices ζ_{ij}^k, $(i,j) = (1,2)$ are constant in the Mobilizer frame (Simbody Theory Manual, pg. 33, figure 4) for each body for the entire course of simulation. These are computed and stored as a preprocessing step.

The seeding operation is carried out for the *WindowLength* specified by the user. Once, sufficient data is obtained, adaptive dynamics can be started.

Simbody

Perform Kinematic sweep MultibodySystem::realize(state,Stage::Velocity)

Two Handle equations

Calculate matrices ζ_{13}^k, ζ_{23}^k for all the bodies in the system

Forces due to molecular forcefields obtained using getRigidBodyForces()

DCA

Calculate $\dot{u}$ for the entire system

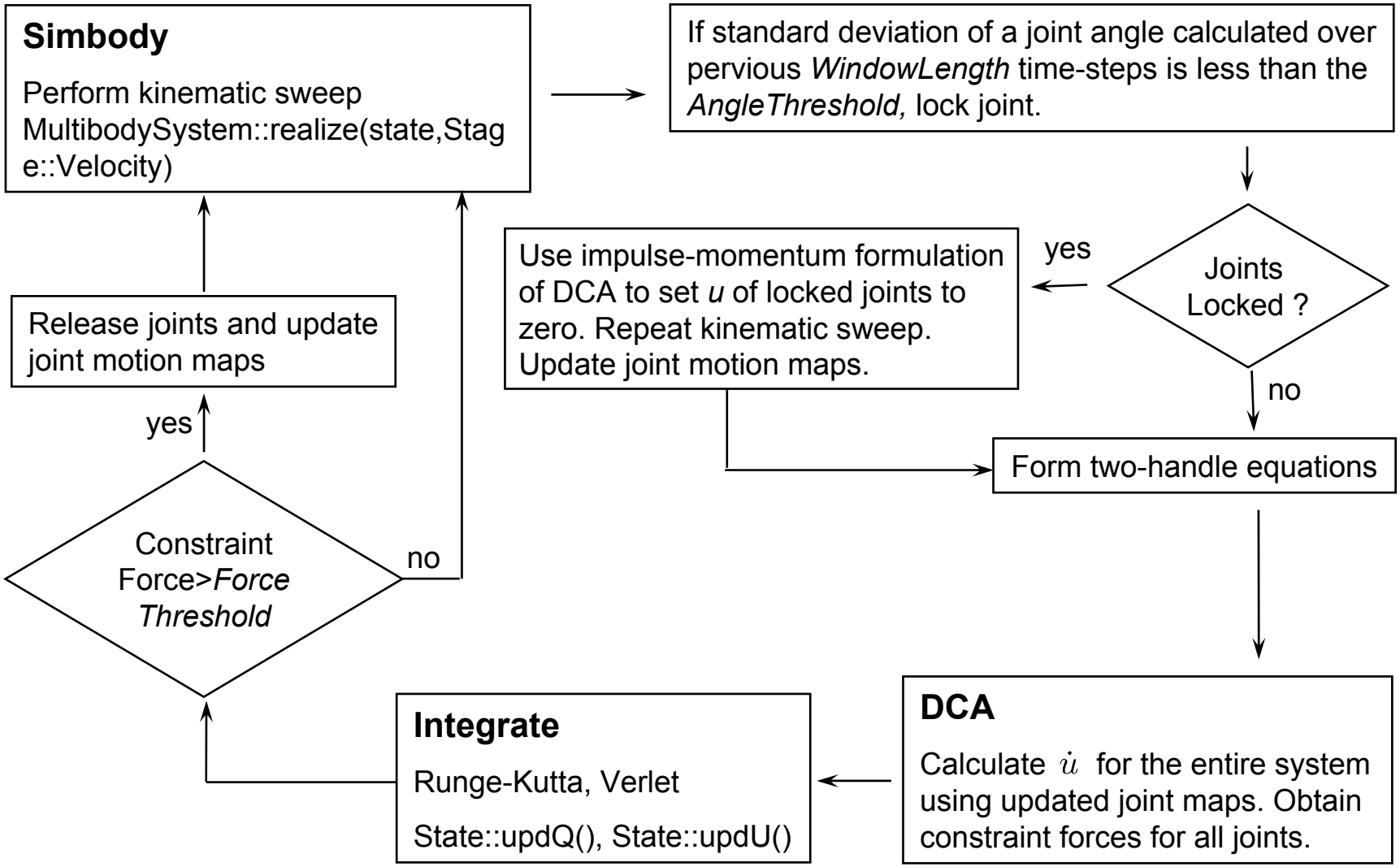
Integrate

Runge-Kutta, Verlet

State::updQ(), State::updU()

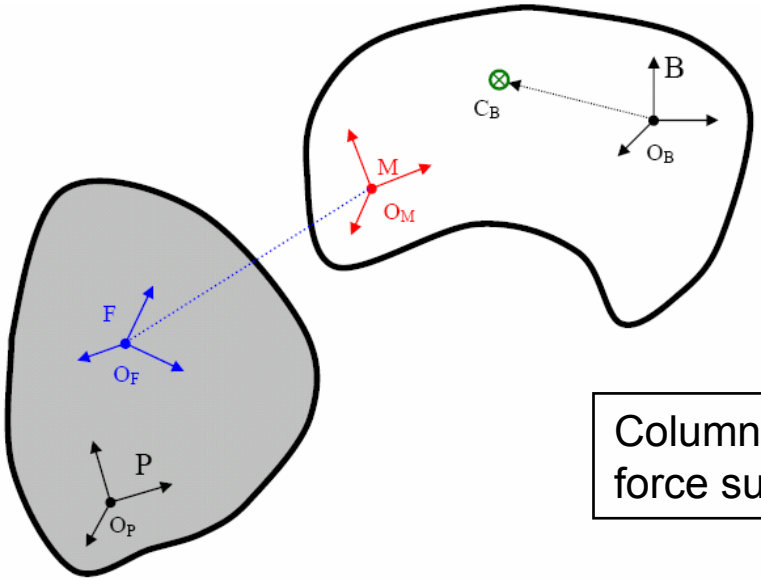
kishor8dm@gmail.com

Adaptive RNA Simulation with Molmodel



©Kishor Bhalerao
kishor8dm@gmail.com

Adaptive RNA Simulation with Molmodel



Joint Motion Maps

The spatial velocity of frames M and F are related by the equation $\mathcal{V}_M = \mathcal{V}_F + H\mathbf{u}$

Define matrix D such that $D^T \cdot H = \mathbf{0}$

Columns of matrix D span the constraint force subspace of the joint

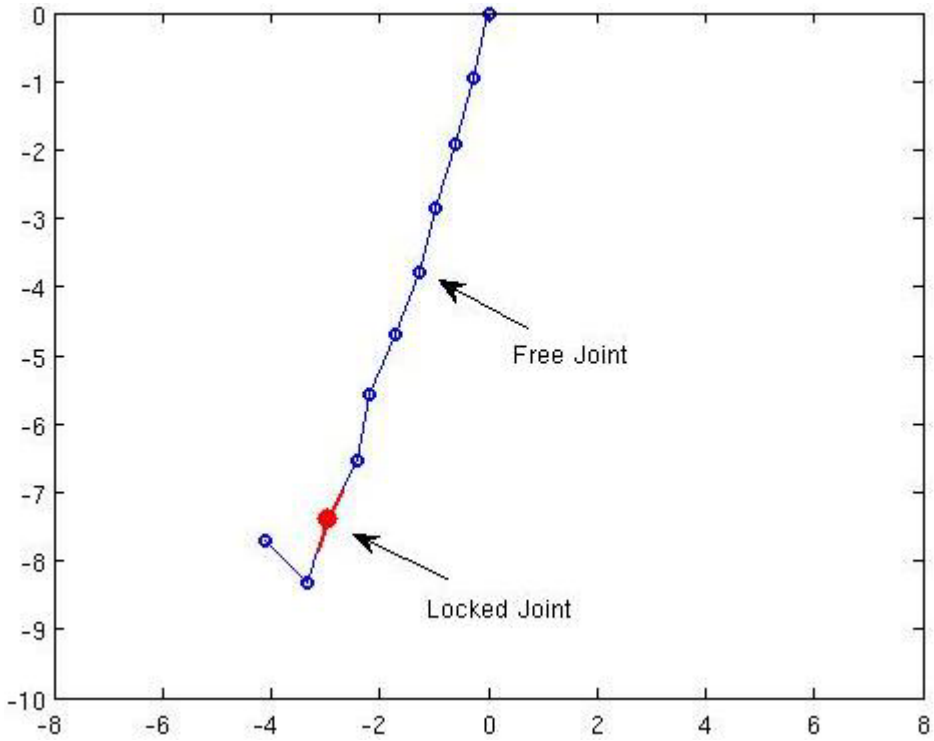
e.g. for a revolute joint about z axis

$$H = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}_{6\times 1}, \quad D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{6\times 5}$$

If a joint is locked, the matrix D is a 6 x 6 identity matrix. In DCA, to turn an active joint to a rigid joint and vice versa, just the appropriate D matrix is required. Rest of the simulation code is identical. This makes the adaptive implementation easier.

kishor8dm@gmail.com

Example



Simulation parameters

AngleThreshold: 3^o

Force Threshold: 10

WidthofMovingWindow: 1000

Integrator: Runge-Kutta 4^{th} order

Time-step: 0.001

Animation can be found at the following location

https://simtk.org/docman/view.php/327/1350/pend.gif

kishor8dm@gmail.com