Adaptive Integration for Controlling Speed vs. Accuracy in Multi-Rigid Body Simulation

Samuel Zapolsky[1] and Evan M. Drumwright[2]

Abstract— **Most current approaches for simulating robots in 3D with rigid body dynamics and contact operate by taking fixed, first order integration steps. In this paper, we investigate the viability of various integration approaches (semi implicit and fully explicit first-order, fourth order Runge Kutta, and variable step first order) for maximizing computational efficiency (accuracy and stability *vs.* running time) on typical robotics manipulation and locomotion applications.**

After arguing that first-order accuracy to the dynamics models is sufficient for most manipulator and locomotion robotics applications, we focus our investigation on methods for efficient, stable dynamic simulations. We describe a novel algorithm that attempts to provide smooth convergence to the true dynamics solution, which allows us to estimate the error in kinetic energy around an integration step size (as a proxy for the simulation stability). We use this algorithm to evaluate multiple hypotheses using simulation-based experiments with two virtual robot models toward developing methods for faster dynamic simulations.

I. INTRODUCTION

Physical interaction with an environment is a central facet of robotics. Accurate modeling of these interactions can allow researchers to predict the performance of and optimize robotic systems [10]; can be used to generate better controls for robots [30], [29]; and can plan grasps and footholds, among other possibilities. DARPA's recent robotics challenge was first conducted in simulation, bringing attention to the power and limitations of 3D dynamics "engines" for simulating robots; the past two years has seen the introduction of several new simulation engines (`DART`, `MuJoCo`, and `RPI-Sim`), all of which—like `ODE` and `Bullet`—use a *time stepping approach* [26] that takes fixed integration steps. The sudden proliferation of engines by robotics researchers indicates clear interest in producing faster simulations *without physical artifacts* and targeted to robotics applications.

This paper focuses on robotic simulations applied to robotic locomotion and manipulation applications. Given those foci, this paper hopes to provide data toward answering the following questions: (1) How can we know when a multi-rigid body dynamics simulation is sufficiently accurate?; (2) What type of integration scheme should be selected to maximize performance (numerical accuracy or numerical stability *vs.* speed) over state of the art time stepping approaches?; and (3) How much can dissipation approaches increase simulation stability and at what cost to accuracy?

II. BACKGROUND

We next describe background on simulating multi-rigid body systems subject to impact and friction (§II-A), including discussion on higher order schemes (§II-B), and the effects of first order integration on robotic applications (§II-C).

A. Integration of multi-rigid body systems subject to impact and friction

Robust and accurate simulation of multi-rigid bodies with impacts, contacts, and friction is challenging because rigid body contact problems with friction may require impulsive forces even without collisions [26] (exemplified by, *e.g.*, Painlevé's Paradox [23]), because constraints (joints and contacts) often become violated during the simulation, and because Zeno type behaviors (where the simulation is unable to progress in time) can occur. The result is that standard integration algorithms for ODEs/DAEs with "events" (*event driven methods* [6], depicted in Figure 1) have been mostly supplanted with approaches (*time stepping methods* [26], see Figure 2) that discretize the Differential Complementarity Problem formulation [24] and solve for impulsive forces that satisfy constraints (friction limits, non-interpenetration, *etc.*)

Most recent multi-rigid-body dynamics engines are based on a time stepping approach discovered by Moreau [22] which avoids solving for precise times of contact; locating event times can make event driven simulation intractable when many contacts occur over a small interval of time, *e.g.*, as in simulations of granular media. And Stewart showed that, under certain assumptions, the time-stepping simulation converges to the continuous time solution of the dynamics as the integration step size becomes sufficiently small [25]. *We are not aware of any approaches that can indicate when convergence is near.* However, we describe an algorithm that aims to provide smooth convergence to the continuous time solution for a nominal integration step size.

The relatively isolated nature of contact interactions (*i.e.*, there are generally few pairs of rigid bodies contacting at any one time) in many applications in Robotics makes finding exact times of contact tractable, which minimizes interpenetration between rigid bodies. Additionally, no research group has yet discovered a time stepping method that can guarantee that contacts are not missed. This paper attempts to provide such a method.

B. Some higher order schemes

Higher-order integration approaches are a common approach to obtain higher performance (via larger step sizes to offset the increased number of function evaluations) on

George Washington University, Washington, DC 20052, USA
Positronics Lab: `robotics.gwu.edu/positronics`
{[1]`samzapo` [2]`drum`}`@gwu.edu`

initial value problems for ODEs/DAEs. We next describe a few such schemes applied to multi-rigid-body simulation.

Mirtich's event-driven type simulation scheme [21] is not subject to Painlevé's Paradox, but is still subject to Zeno behavior from modeling sustained contacts as impacts. Mirtich mitigates this problem by ensuring that bodies only contact over short time intervals (thereby introducing new problems as Mirtich acknowledges). Arbitrary integrators can be applied to the ODEs/DAEs between impacts. Mirtich's approach ensures that all impacts are detected (to the accuracy of the integration scheme).

Acary describes a hybrid event driven / time stepping scheme that uses event driven methods over time intervals exhibiting smooth dynamics and first-order time stepping otherwise [1]; one of the hardest problems that Acary addresses is that of ensuring that the resulting hybrid approach is consistent. Acary does not investigate how to ensure that accuracy is sufficient for the low order time stepping scheme.

Studer *et al.* [27] note that, "Event-driven methods are very well suited for systems with few contacts and long smooth parts of the motion, while time-stepping schemes have advantages for systems with many contacts..." Studer *et al.* then present a method that attempts to combine the advantages of both: ODEs/DAEs are integrated to "switching points" where the ODEs/DAEs would change, as when a contact transitions from sliding to sticking or when an impact occurs, but time stepping is used to mitigate Zeno behavior. The resulting approach is of arbitrary order. Studer *et al.*'s work is usable for conducting the necessary experiments for this paper, but the inability of the approach (as described) to prevent tunneling[1] led us to develop the method described in Section III.

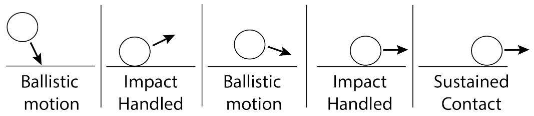

Fig. 1. *Depiction of an event driven approach to simulation. The ODEs of motion are integrated up to the time of an event (an impact), the impact is modeled, and the ODEs are integrated until the next impact.*

C. *Effects of first order integration on robotics applications*

Typical robot control loops run between 100 Hz and 10,000 Hz. Assuming that the simulation must treat the robot's control loop as a black box and that the controls are not necessarily smooth, the maximum integration step will thereby be limited to the inverse of the control frequency. If we integrate using a first-order approach at steps of $\Delta t = 0.01$ seconds, then the truncated terms in a Taylor Series approximation would be of a scale $\frac{1}{100}$ that of the present terms. Therefore, *higher order effects must be quite significant for Euler integration to be inefficient at simulating typical legged robots and manipulators.*

III. ADAPTIVE TIME STEPPING SIMULATION

This section describes two algorithms. The first algorithm, presented as Algorithm 1, simulates a multi-rigid body system over time interval $[t_0, t_f]$ while ensuring that no impacts or contacts are missed to the accuracy of the integrator within that interval. We will build on this algorithm, yielding Algorithm 2, to estimate error through smooth convergence to the continuous time solution (for a nominal step size). Without loss of generality, this section assumes any bilateral joint constraints are explicit (those that can be represented by a body's independent coordinates). Neither do we consider limits on the range of motion of bilateral constraints though such limits are easily incorporated into this formulation.

A. *Simulation without missing impacts*

We assume that the multi-rigid body dynamics are integrated over interval h using a first order approach like:

$$\boldsymbol{q}_{i+1} \leftarrow \boldsymbol{q}_i + h\boldsymbol{v}_i \tag{1}$$

$$\boldsymbol{v}_{i+1} \leftarrow \boldsymbol{v}_i + h\boldsymbol{f}_{\boldsymbol{q}_{i+1},\boldsymbol{v}_i} \tag{2}$$

where $\boldsymbol{q}$ corresponds to generalized coordinates, $\boldsymbol{v}$ corresponds to generalized velocities, and $\boldsymbol{f}(.)$ corresponds to an ordinary differential equation. We compute $\boldsymbol{f}(.)$ with respect to the new positions and the old velocities, but such a choice is arbitrary (we could also compute $\boldsymbol{f}(.)$ using the old positions and old velocities). If $\boldsymbol{q}$ contains unit quaternion components to store rigid body orientation intermediate steps may be taken: (1) converting the spatial velocity components in $\boldsymbol{v}_i$ to a differential quaternion and (2) projecting the quaternion components in $\boldsymbol{q}_{i+1}$ back onto SO(3), as described in [20]. Such details can be omitted from further discussion without loss of generality.

1) Conservative advancement plus time stepping: The idea behind our approach, presented as Algorithm 1, is to combine Mirtich's conservative advancement approach [21]—which determines integration steps such that the first time of impact between bodies (if any) is found[2]—

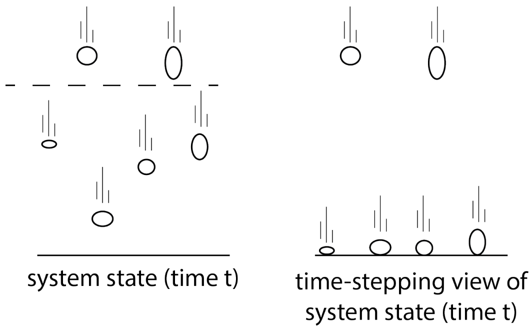

Fig. 2. *A depiction of the way a time stepping simulation works. At left, all grains below the dotted line will impact the ground plane within time interval Δt. A time stepping approach might treat all of those grains as impacting simultaneously at time t (as shown at right) when determining state $t + \Delta t$.*

[1] Tunneling occurs when a moving body passes completely through another body [14] due to insufficiently granular discrete geometric intersection (collision detection) checking.

[2] Applying generic root finding schemes to this problem as in [27] makes the simulation susceptible to missing the first impact (finding a later impact instead) or reporting no impacts when one or more exists (*i.e.*, tunneling).

with a first order time stepping scheme that "adaptively" steps to impacts and contacts. Lines 3–15 integrate repeatedly until an impact event is encountered. Using our first order scheme with conservative advancement,

$$r(\boldsymbol{q},\boldsymbol{v}) \equiv ||\boldsymbol{d}||/(\hat{\boldsymbol{d}}^{\mathsf{T}}(\dot{\boldsymbol{x}}_A - \dot{\boldsymbol{x}}_B) + |\hat{\boldsymbol{d}} \times \boldsymbol{\omega}_A| r_{\max_A} + \ldots \quad (3)$$
$$|\hat{\boldsymbol{d}} \times \boldsymbol{\omega}_B| r_{\max_B}),$$

allows us to assume constant velocity over the time interval, which speeds the conservative advancement process considerably ($\boldsymbol{d}$ is the vector between the closest point on the first rigid body to the closest point on the second body, $\hat{\boldsymbol{d}} \equiv \boldsymbol{d}/||\boldsymbol{d}||$, $\dot{\boldsymbol{x}}$ is the linear velocity of a body, $\boldsymbol{\omega}$ is the angular velocity of a body, and r_{max} is the maximum distance of any point on a body from the body's center of mass—see [21], [31] for further details). Note that the commonly implemented time stepping method (see, *e.g.*, [4]), which takes a fixed step of size Δt (thereby permitting some interpenetration) can be obtained by setting h_{min} (the minimum step size) to Δt.

The conservative advancement process is repeated until a pair of bodies either come into contact or the positions have been integrated over the full interval (of size Δt). After our approach integrates the positions, it then integrates velocities (Line 9), which includes application of an impact model (*e.g.*, [12]) on Line 11. The impact model (applied via function $k(.)$) serves identically to standard time stepping mechanisms; using, *e.g.*, the contact model described in [4] as an impact model would yield identical results between Algorithm 1 (assuming $h_{\text{min}} = \Delta t$) and [4]. The impact model applies no impulses if $\{\boldsymbol{q}_{i+1}, \boldsymbol{v}^*_{i+1}\}$ does not correspond to an impacting state: $\boldsymbol{v}_{i+1}$ then equals $\boldsymbol{v}^*_{i+1}$.

2) Ensuring that $t_c > 0$: For conservative advancement to return a nonzero value, (some small) positive distance must be maintained between each pair of bodies. Thus, on a call to SIMULATE, at least the *first* computation of t_c on Line 4 must return a positive value. A simple way to effect this is to apply the projection approach of Cline and Pai [7] directly after SIMULATE to keep the bodies separated by that small distance. Also, notice how Algorithm 1 handles bodies in sustained contact. For sustained contacts—with the time derivative of the signed distance function being zero, based on the current state of the system—the conservative advancement process should determine that the bodies move along, rather than into, the contact manifold, thus implying that $r(\boldsymbol{q}_{i+1}, \boldsymbol{v}_i) > 0$. Forces that act to push the bodies together (*e.g.*, gravity) on Line 9 can be counteracted by the impact model on Line 10 to keep the bodies in sustained contact (as appropriate).

To conclude discussion of this algorithm, note that tunneling will not be observed (for sufficiently small h_{min}): the approach takes small steps when necessary (*i.e.*, for accuracy or to keep bodies from interpenetrating) and large steps otherwise.

B. Simulation with conservative conservative advancement

This section will build on Algorithm 1 by adding integration with error control. We can use standard techniques for estimating integration error (see [17]), which compare the result from taking one full integration step against the result from taking two half steps. However, care is necessary because a naïve approach might detect and model an impact at the first half step that was not detected at the full step, for example, thereby invalidating the necessary smoothness conditions for the error estimates to hold.

This algorithm attempts to provide smooth convergence to the continuous time solution to the dynamics *for a nominal step size*. The idea is to subdivide the desired integration interval as necessary until every impacts has been detected and modeled. The strongest assumption is that the first-order error estimate (recommended by Hairer *et al.* [17]) is a sufficiently conservative bound; this approach has been applied successfully in existing integration schemes (*e.g.*, [1]).

1) Conservative conservative advancement: Algorithm 2 operates using a process we call conservative conservative advancement (C^{2}A). The idea is that integration step size can be decreased, repeatedly if necessary, until (1) the estimated local kinetic error in energy from integration (see Lines 18 and 19) is within the desired absolute (for simplicity of presentation) tolerance (ϵ) on Line 22 *and* (2) the interval that the bodies were integrated over is free of impacts (using bounds on velocity). These velocity bounds are derived from error estimates computed by the integration algorithm, and are accounted for via $s(.)$, a modification of $r(.)$ that leverages these velocity bounds:

$$\begin{aligned} s(\boldsymbol{q},\boldsymbol{v},\boldsymbol{e}_{\boldsymbol{v}}) \equiv & ||\boldsymbol{d}||/(\hat{\boldsymbol{d}}^{\mathsf{T}}(\dot{\boldsymbol{x}}_A - \dot{\boldsymbol{x}}_B) + \\ & |\hat{\boldsymbol{d}}|^{\mathsf{T}}(|\boldsymbol{e}_{\dot{\boldsymbol{x}}_A}| + |\boldsymbol{e}_{\dot{\boldsymbol{x}}_B}|) + |\hat{\boldsymbol{d}} \times g(\boldsymbol{\omega}_A, \boldsymbol{e}_{\boldsymbol{\omega}_A})| r_{\max_A} + \\ & |\hat{\boldsymbol{d}} \times g(\boldsymbol{\omega}_B, \boldsymbol{e}_{\boldsymbol{\omega}_B})| r_{\max_B}). \end{aligned} \quad (4)$$

where $\boldsymbol{e}_{\boldsymbol{v}} = \{\boldsymbol{e}_{\dot{\boldsymbol{x}}}, \boldsymbol{e}_{\boldsymbol{\omega}}\}$ are error estimates for the linear and angular velocities of a rigid body, $|.|$ is applied element-wise for vectors, and $g(.)$ is defined as:

$$g(\boldsymbol{\omega}, \boldsymbol{e}_{\boldsymbol{\omega}}) \equiv \begin{bmatrix} \begin{cases} \omega_x + e_{\omega_x} & \text{if } |\omega_x + e_{\omega_x}| > |\omega_x - e_{\omega_x}| \\ \omega_x - e_{\omega_x} & \text{if } |\omega_x + e_{\omega_x}| \leq |\omega_x - e_{\omega_x}| \end{cases} \\ \begin{cases} \omega_y + e_{\omega_y} & \text{if } |\omega_y + e_{\omega_y}| > |\omega_y - e_{\omega_y}| \\ \omega_y - e_{\omega_y} & \text{if } |\omega_y + e_{\omega_y}| \leq |\omega_y - e_{\omega_y}| \end{cases} \\ \begin{cases} \omega_z + e_{\omega_z} & \text{if } |\omega_y + e_{\omega_z}| > |\omega_z - e_{\omega_z}| \\ \omega_z - e_{\omega_z} & \text{if } |\omega_z + e_{\omega_z}| \leq |\omega_z - e_{\omega_z}| \end{cases} \end{bmatrix} \quad (5)$$

Finally, Lines 3–6 of Algorithm 2 allow the algorithm to be used in purely time stepping form, if desired (for very small step sizes). Richardson Extrapolation (Lines 29–30) is used to obtain a second order solution, as described in [17].

2) Integration error as kinetic energy error: This algorithm allows distilling integration error into a scalar (rather than the usual absolute and relative error tolerances—with physically incomparable units along different dimensions—used in generic integration schemes with error control). We will argue that this estimated kinetic energy error can be used to assess stability: if the relative error in energy is sufficiently large, the system is more likely to become unstable (see first experiment, below).

Algorithm 1 SIMULATE($t_0, \Delta t$) Simulates a system of multi-rigid bodies from time t_0 to time $t_f \leq t_0 + \Delta t$ using the largest possible step size such that impact events are not missed.

```
h ← 0
$\boldsymbol{q}_{i+1} \leftarrow \boldsymbol{q}_i$
**while** $h \leq \Delta t$ **do**
    $t_c \leftarrow \min(\max(r(\boldsymbol{q}_{i+1}, \boldsymbol{v}_i), h_{\text{min}}), h)$    ▷ Compute conservative step
    **if** $t_c > 0$ **then**    ▷ See whether simulation can be advanced
        $h \leftarrow h + t_c$
        $\boldsymbol{q}_{i+1} \leftarrow \boldsymbol{q}_i + h\boldsymbol{v}_i$    ▷ Integrate position forward
    **else**    ▷ Contact/impact at the current time
        $\boldsymbol{v}^*_{i+1} \leftarrow \boldsymbol{v}^*_i + h\boldsymbol{f}_{\boldsymbol{q}_{i+1},\boldsymbol{v}_i}$    ▷ Integrate velocity forward
        $\boldsymbol{v}_{i+1} \leftarrow k(\boldsymbol{v}^*_{i+1})$    ▷ Apply impact model
        **break**
**return** $\{h, \boldsymbol{q}_{i+1}, \boldsymbol{v}_{i+1}\}$
```

Algorithm 2 SIMULATE-C^{2}A($t_0, \Delta t$) Simulates a system of multi-rigid bodies from generalized position $\boldsymbol{q}_i$, generalized velocity $\boldsymbol{v}$, and time t_0 to time $t_f \leq t_0 + \Delta t$, position $\boldsymbol{q}_{i+1}$, and velocity $\boldsymbol{v}_{i+1}$ using conservative conservative advancement (C^{2}A) and error control.

```
 1: $t_{c_i} \leftarrow \min(r(\boldsymbol{q}, \boldsymbol{v}_i), \Delta t)$    ▷ Compute conservative step
 2: $h \leftarrow t_{c_i}$
 3: **if** $h < h_{\text{min}}$ **then**
 4:     $h \leftarrow h_{\text{min}}$
 5:     $\boldsymbol{q}_{i+1} \leftarrow \boldsymbol{q}_i + h\boldsymbol{v}_i$    ▷ Take one Euler step of size $h$
 6:     $\boldsymbol{v}_{i+1} \leftarrow k(\boldsymbol{v}_i + h\boldsymbol{f}_{\boldsymbol{q}_{i+1},\boldsymbol{v}_i})$    ▷ Integrate velocity forward and apply impact model
 7: **else**
 8:     $\boldsymbol{q}'_{i+1} \leftarrow \boldsymbol{q}_i + h\boldsymbol{v}_i$    ▷ Take one Euler step of size $h$
 9:     $\boldsymbol{v}'_{i+1} \leftarrow k(\boldsymbol{v}_i + h\boldsymbol{f}_{\boldsymbol{q}'_{i+1},\boldsymbol{v}_i})$
10:     $\boldsymbol{q}''_{i+\frac{1}{2}} \leftarrow \boldsymbol{q}_i + \frac{h}{2}\boldsymbol{v}_i$    ▷ Take first Euler step of size $\frac{h}{2}$
11:     $\boldsymbol{v}''_{i+\frac{1}{2}} \leftarrow k(\boldsymbol{v}_i + \frac{h}{2}\boldsymbol{f}_{\boldsymbol{q}''_{i+\frac{1}{2}},\boldsymbol{v}_i})$
12:     $t_{c_{i+\frac{1}{2}}} \leftarrow \min(r(\boldsymbol{q}''_{i+\frac{1}{2}}, \boldsymbol{v}''_{i+\frac{1}{2}}), \frac{h}{2})$    ▷ Compute new conservative step
13:     **if** $t_{c_{i+\frac{1}{2}}} < \frac{t_{c_i}}{2}$ **then**
14:         $t_{c_i} \leftarrow 2t_{c_{i+\frac{1}{2}}}$
15:         **goto** Line 2    ▷ Step not sufficiently conservative
16:     $\boldsymbol{q}''_{i+1} \leftarrow \boldsymbol{q}''_{i+\frac{1}{2}} + \frac{h}{2}\boldsymbol{v}''_{i+\frac{1}{2}}$    ▷ Take second Euler step
17:     $\boldsymbol{v}''_{i+1} \leftarrow k(\boldsymbol{v}''_{i+\frac{1}{2}} + \frac{h}{2}\boldsymbol{f}_{\boldsymbol{q}''_{i+\frac{1}{2}},\boldsymbol{v}''_{i+\frac{1}{2}}})$
18:     $E' \leftarrow \frac{1}{2}\boldsymbol{v}'^{\mathsf{T}}_{i+1}\mathbf{M}_{\boldsymbol{q}'_{i+1}}\boldsymbol{v}'_{i+1}$    ▷ K.E. after one step
19:     $E'' \leftarrow \frac{1}{2}\boldsymbol{v}''^{\mathsf{T}}_{i+1}\mathbf{M}_{\boldsymbol{q}''_{i+1}}\boldsymbol{v}''_{i+1}$    ▷ K.E. after two half steps
20:     $e_E \leftarrow E' - E''$    ▷ Estimate energy error
21:     $e_{\boldsymbol{v}} \leftarrow \boldsymbol{v}'_{i+1} - \boldsymbol{v}''_{i+1}$
22:     **if** $e_E > \varepsilon$ **then**    ▷ See whether simulation can be advanced
23:         $h \leftarrow .9|\epsilon|/|e_E|h$    ▷ Scale w/ absolute error tolerance $\epsilon$
24:         **goto** Line 3
25:     $t^*_{c_i} \leftarrow \min(s(\boldsymbol{q}_i, \boldsymbol{v}_i, \boldsymbol{e}_{\boldsymbol{v}}), h)$    ▷ Compute conservative conservative advancement (C$^2$A) step
26:     **if** $t^*_{c_i} < t_{c_i}$ **then**    ▷ Check whether conservative step holds
27:         $t_{c_i} \leftarrow t^*_{c_i}$
28:         **goto** Line 2
29:     $\boldsymbol{q}_{i+1} \leftarrow 2\boldsymbol{q}''_{i+1} - \boldsymbol{q}'_{i+1}$    ▷ Richardson Extrapolation of position (renormalize any quaternion components)
30:     $\boldsymbol{v}^*_{i+1} \leftarrow 2\boldsymbol{v}''_{i+1} - \boldsymbol{v}'_{i+1}$    ▷ Richardson Extrapolation
31: **return** $\{h, \boldsymbol{q}_{i+1}, \boldsymbol{v}_{i+1}\}$
```

IV. EXPERIMENTS

Our experiments use two virtual robotic platforms: the seven degree-of-freedom passive dynamic walker described in [8] (simulated in `MATLAB`) and an 18 degree-of-freedom quadrupedal robot (controlled using `Pacer` [28] and simulated with `Moby` [11]). Both robots are represented in minimal coordinates to avoid issues of constraint stabilization (see [5]) and associated parameter tuning. Contact forces are computed using the approach described in [12] (treating the impact model as a contact model, as in [2]); a direct solver is used to avoid the issues of parameter tuning and convergence [18] for iterative solvers.

Experimental platforms: We experiment using the virtual passive dynamic walker because the simulation is highly accurate (verified to approximately fifteen significant figures[3] using multiple numerical codes) and because it is straightforward to check how well a solution maintains energy on this conservative system. We experiment using the virtual quadrupedal robot because its number of degrees of freedom and contact geometry (point contacts at the feet) are comparable to current legged robots and models. When locomoting, the robot carries out a trotting gait with a step height of 3 cm; the stance phase of each foot accounts for 75% of the gait cycle duration (0.225 of every 0.3 seconds are spent in stance phase). The quadruped locomotes on a planar surface for a duration of five seconds for each experiment (thirty-three steps).

For simulating our quadrupedal robot, we generally use a 1 ms integration step, which we determined (before this present research) yields stable simulations. However, we will show that we can control the quadrupedal robot effectively at a mean step size of approximately 2.3 milliseconds (§IV-D) on certain tasks.

Assessing stability: We assess stability for the passive dynamic walker using deviation in total energy. We observe the simulation having a high tendency toward instability when energy deviates 0.01% from its nominal value. We build in a safety margin by requiring the model's total energy to deviate no more than 0.001% from its nominal value to remain stable.

The quadruped's virtual actuators make checking that total system energy is conserved meaningless. However, we do use error estimates (computed to first-order accuracy) from the integration process to assess stability. Like the passive walker, we observe the simulation having a high tendency toward instability when local kinetic energy error is estimated at 0.5% or higher. We build in a safety margin by requiring the model's kinetic energy to deviate no more than 0.25%

[3] Personal communication with A. Ruina.

from its nominal value to remain stable.

We observe that the particular safety margins described above for integration with error control do not transfer between models: bounds on relative error that lead to stable simulations must be determined in the above manner for each new robot.

A. Testing hypothesis that greater local kinetic energy error is a proxy for simulation stability

The kinetic energy of a dynamically simulated system will become unbounded after a simulation has become unstable. At larger simulation step sizes, a simulated robotic platform will tend to exhibit instability in fewer steps. We demonstrate this by measuring the number of steps taken before the robot's kinetic energy far exceeds normal values (10^7 J in our experiment) under various step sizes. This experiment used the virtual quadrupedal robot in a simulation with fixed step sizes $\Delta t \in \{1{,}1.5{,}1.75{,}2{,}3{,}4{,}5{,}6{,}7{,}8{,}9{,}10\}$ ms). Figure 5 shows that the number of steps a simulator can take before becoming unstable is negatively correlated with step size.

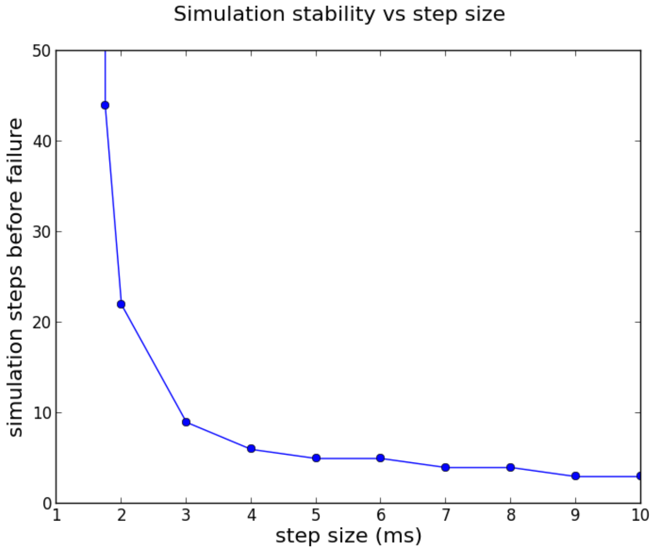

Fig. 3. *Inverse relationship between step size and stability. Failure not observed for values above bounds of plot.*

We conclude that **local kinetic energy error can be used to indicate simulation stability**. Controlling the growth of this error is likely beneficial in stabilizing the simulation of a dynamic robot: larger simulation step sizes requires combatting energy increases in the robotic system. We demonstrate one such method that uses exponential dissipation (§IV-E).

B. Testing hypothesis that RK4 is fast enough to make up for the increased number of steps

Runge Kutta schemes that exhibit fourth order global error (henceforth denoted RK4) are proven techniques. We wished to see whether RK4 could lead to faster integration for legged locomotion applications. This experiment, which uses the passive dynamic walker model, determines the largest step size using two explicit integration schemes such that the deviation in system energy satisfies the tolerances described in §IV. The maximum step size for explicit Euler and RK4 integration was determined to be 3.5×10^{-3} and 2.8×10^{-1} seconds, respectively. RK4 requires four times as many function evaluations per integration step, so the total speedup moving from explicit Euler to RK4 is approximately 20.0 times for this model. Our experience indicates that the integration step size for the Euler integrator lies in the range commonly used for articulated models.

There is no standard Runge-Kutta approach for time stepping. The only RK4 approach applied to time stepping of which we are aware is referred to in [13], but details of the approach are unpublished. The considerable challenges of developing any such higher order time stepping schemes are described in [3]. Assuming that a collocation type approach (*i.e.*, computing contact forces at all intermediate points over the integration interval simultaneously) is taken, mathematical programming variables in the time stepping problem will grow by a factor of three. If we generously assume cubic time complexity for solving the linear complementarity problem (LCP) / nonlinear complementarity problem[4], the extra variables will make one of the most computationally expensive aspects of the simulation process at least 27 times slower to solve. Thus, the computation saved by the ability to take larger steps is similar to the computation required to solve additional LCPs: **higher order integration is likely not worth the effort of implementation for common robotic manipulation and locomotion scenarios**.

C. Testing hypothesis that symplectic integration does not yield more stable simulation of non-Hamiltonian systems

Symplectic integrators preserve the numerical Hamiltonian and are generally are known to produce superior accuracy and stability for Hamiltonian systems [16]. Stewart notes the challenges with adapting such schemes to applications with contact and friction [26]. Regardless, documentation for `ODE` (among other software) touts the semi-implicit first-order scheme—position is integrated using the new velocity, rather than the old velocity as in the explicit method—as tending to conserve energy. We ran an experiment to test whether the semi-implicit Euler algorithm (also known as Störmer-Verlet and symplectic-Euler [16]) possesses better stability properties than the explicit Euler integration algorithm on generic (*i.e.*, not necessarily Hamiltonian) multi-rigid-body systems with contact; we used the virtual quadrupedal robot for this experiment.

Algorithm 2 implements a semi-implicit Euler scheme, which is also a symplectic integrator for Hamiltonian systems [16]. We ran an experiment against the fully explicit version of the same integrator by modifying $\boldsymbol{f}$ appropriately (using current positions to compute accelerations) on Lines 6, 9, 11, and 17 of Algorithm 2. Using both schemes, we computed the estimated local kinetic energy error over each 1 ms time interval while simulating a quadrupedal robot locomoting on a planar environment. Estimated kinetic

[4] Cottle *et al.* claim that Lemke's Algorithm [19] for solving copositive LCPs generally requires n pivots [9]. Using low-rank updates, this assumes a best case time complexity of $O(n^3)$.

energy errors and total kinetic energy were averaged over the experiment interval (five thousand samples over five seconds). Relative errors were calculated at each time step, then the resulting percentage measurements were averaged over the duration of the experiment.

We observed that integration with explicit Euler integration produced a kinetic energy error of 1.2306×10^{-4} J out of a total 0.13494 J on average (mean 0.77389% relative error at each time step) for the walking robotic system. We compared these results against semi-implicit integration, which produced a kinetic energy error of 1.5029×10^{-4} J out of a mean total 0.12875 J (mean 1.3904% relative error at each time step). Using an explicit Euler first order integration scheme resulted in nearly half the average relative error, though the difference is minor in absolute terms. In conclusion, **we find that explicit Euler integration may *increase* simulation stability over semi-explicit Euler integration.**

D. Testing hypothesis that integration with error control only yields improvements in running time for slow-moving robotic systems

We used the quadrupedal robot model for this experiment and disabled the C^{2}A process to determine an upper bound on improvements in running time: the knowledge that the ground is planar precludes tunneling phenomena. We set 5 ms as the nominal step size and 0.5 ms to be the minimum step size limit (h_{min}). Algorithm 2 was then used to select the largest time step to take at each simulator iteration such that local kinetic energy error remains within tolerances. We set the experimentally-derived local kinetic energy relative error limit at 0.25% for these trials. Step size selection was managed by Algorithm 2. About two thousand simulation steps were taken to simulate five seconds of walking and standing.

The mean step size was 2.3427 ms while using Algorithm 2. Analysis indicated that the stability of the simulation was limited by the robot's feedback control system performing poorly in the presence of significant errors; in other words, the control system (not the robot dynamics) reduces stability. Over intervals consisting of the robot walking, error control yielded a mean step size of 2.1646 ms. This data tells us that 1 ms is a reasonable time step for this robot, which is consistent with our experience with this model. We quantify the simulation stability when the robot is in motion in Figure 4. We found that simulation stability decreases substantially when the robot is taking a step versus standing in place (indicated by an increase in estimated local kinetic energy error during the stepping portions of the trotting gait).

Finally, given the extra computation required by these methods, **integration with error control is unlikely to result in significant speed increases even on less dynamic robots, and the possibility of *increased* computation time as well as software bugs from additional implementation complexity make it hard to recommend investigating such approaches further.**

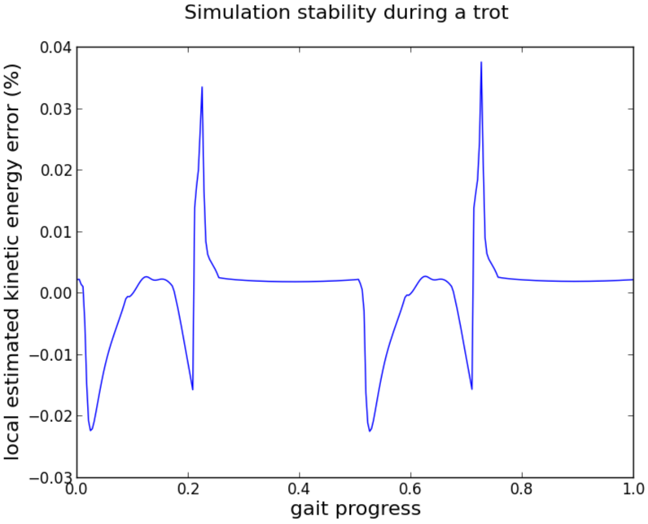

Fig. 4. *Simulation stability (inversely proportional to relative, local kinetic energy error) decreases when the quadruped is in motion versus standing in place. Data is presented over one gait cycle ({ step, stand, step, stand}). Simulation stability is reduced during a step, but remains high when the robot is stationary.*

E. Testing hypothesis that exponential dissipation can increase step size considerably

Given the relationship between dynamic stability of ODEs and numerical stability of algorithms for solving them, it is clear that one can use mechanisms for energy dissipation to increase stability. We showed earlier (§IV-A) that instability introduced by taking larger step sizes is correlated with increasing kinetic energy. We hypothesize that dissipating this energy will allow us to take larger step sizes, leading to faster simulation at the cost of accuracy. We wished to test the hypothesis using an exponential dissipation function. The function is simple: after every Algorithm 1 or 2 call, the generalized velocity is scaled by $\lambda \in [0, 1]$ (where $\lambda = 0$ is equivalent to no dissipation and $\lambda = 1$ brings the system to rest instantaneously). We used this approach after experimenting with Rayleigh dissipation [15], which produced less dramatic results and uses a less intuitive parameterization.

We simulated the quadrupedal robot with exponential dissipation coefficients $\lambda \in \{0, 0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$. We simulated the walking quadruped with a fixed step size of 0.005 (unstable with no dissipation, see Figure 5) and counting the number of iterations until failure (triggered when kinetic energy exceeds 10^7 J).

We observed that **adding dissipation to an actuated multi-rigid-body simulation leads to increased stability and allows us to take larger integration steps**, which lends evidence to our hypothesis (see Figure 5). These results indicate that dissipation provides a beneficial tradeoff between stability and accuracy. Furthermore, we assume that this tradeoff can be used to speed simulation of dynamically controlled robots at the cost of some accuracy.

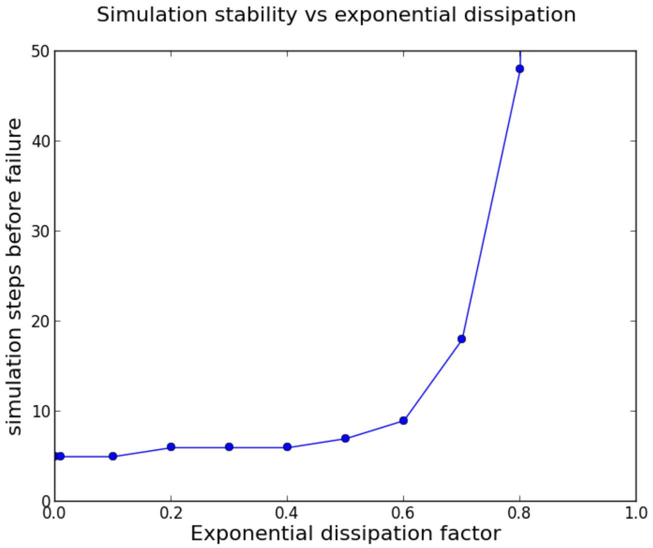

Fig. 5. *Relationship between exponential dissipation and simulation stability for the walking quadruped simulated at steps of 0.005 s. Failure is not observed for values above vertical bounds of plot.*

V. CONCLUSIONS

We identified local kinetic energy error as a useful proxy of simulation stability for dynamically moving robots and presented an algorithm for adaptively adjusting simulation step size to satisfy a constraint on local kinetic energy error. We observed that reducing kinetic energy through dissipation may be a viable strategy for speeding simulation at the cost of some physical accuracy.

VI. FUTURE WORK

Future work will focus on verifying mathematically that our algorithm converges smoothly to the continuous time solution as step sizes decrease.

ACKNOWLEDGMENT

This work was partially supported by NSF CMMI-110532 and Open Source Robotics Foundation. We thank Michael Coleman for providing us with his excellent passive dynamic walker simulation.

REFERENCES

[1] V. Acary. Toward higher order event-capturing schemes and adaptive time-step strategie for nonsmooth multibody systems. Technical Report RR-7151, INRIA, 2009.

[2] M. Anitescu. Optimization-based simulation of nonsmooth dynamics. *Mathematical Programming, Series A*, 105:113–143, 2006.

[3] M. Anitescu and G. D. Hart. A constraint-stabilized time-stepping approach for rigid multibody dynamics with joints, contacts, and friction. *Intl. Journal for Numerical Methods in Engineering*, 60(14):2335–2371, 2004.

[4] M. Anitescu and F. A. Potra. Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. *Nonlinear Dynamics*, 14:231–247, 1997.

[5] U. M. Ascher, H. Chin, L. R. Petzold, and S. Reich. Stabilization of constrained mechanical systems with DAEs and invariant manifolds. *J. Mech. Struct. Machines*, 23:135–158, 1995.

[6] B. Brogliato, A. A. ten Dam, L. Paoli, F. Génot, and S. Abadie. Numerical simulation of finite dimensional multibody nonsmooth mechanical systems. *ASME Appl. Mech. Reviews*, 55(2):107–150, March 2002.

[7] M. B. Cline and D. K. Pai. Post-stabilization for rigid body simulation with contact and constraints. In *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3744–3751, 2003.

[8] M. J. Coleman, M. Garcia, K. Mombaur, and A. Ruina. Prediction of stable walking for a toy that cannot stand. *Phys. Review. E*, 64(2), 2001.

[9] R. W. Cottle, J.-S. Pang, and R. Stone. *The Linear Complementarity Problem*. Academic Press, Boston, 1992.

[10] K. M. Digumarti, C. Gehring, S. Coros, J. Hwangbo, and R. Siegwart. Concurrent optimization of mechanical design and locomotion control of a legged robot. In *Proc. Intl. Conf. Climbing Walking Robots (CLAWAR)*, 2014.

[11] E. Drumwright. Moby. http:://github.com/edrumwri/Moby.

[12] E. Drumwright and D. A. Shell. Modeling contact friction and joint friction in dynamic robotic simulation using the principle of maximum dissipation. In *Proc. of Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2010.

[13] T. Erez, Y. Tassa, and E. Todorov. Simulation tools for model-based robotics: comparison of Bullet, Havok, MuJoCo, ODE, and PhysX. In *Proc. IEEE Intl. Conf. Robot. Autom. (ICRA)*, 2015.

[14] C. Ericson. *Real-Time Collision Detection*. Morgan Kaufmann, 2005.

[15] H. Goldstein, C. Poole, and J. Safko. *Classical Mechanics, 3rd ed.* Addison Wesley, 2001.

[16] E. Hairer, C. Lubich, and G. Wanner. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer, 2nd edition, 2006.

[17] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems, 2nd. ed.* Springer, 2008.

[18] C. Lacoursière. Splitting methods for dry frictional contact problems in rigid multibody systems: Preliminary performance results. In M. Ollila, editor, *Proc. of SIGRAD*, pages 11–16, Nov 2003.

[19] C. E. Lemke. Bimatrix equilibrium points and mathematical programming. *Management Science*, 11:681–689, 1965.

[20] A. T. Miller. *GraspIt: A Versatile Simulator for Robotic Grasping*. PhD thesis, Columbia University, 2001.

[21] B. Mirtich. *Impulse-based Dynamic Simulation of Rigid Body Systems*. PhD thesis, University of California, Berkeley, 1996.

[22] J. J. Moreau. *Standard inelastic shocks and the dynamics of unilateral constraints*, pages 173–221. Springer-Verlag, New York, 1983.

[23] P. Painlevé. Sur le lois du frottement de glissement. *C. R. Académie des Sciences Paris*, 121:112–115, 1895.

[24] J.-S. Pang and D. E. Stewart. Differential variational inequalities. *Math. Program., Ser. A*, 113:345–424, 2008.

[25] D. E. Stewart. Convergence of a time-stepping scheme for rigid-body dynamics and resolution of Painlevé's problem. *Arch. Ration. Mech. Anal.*, 145:215–260, 1998.

[26] D. E. Stewart. Rigid-body dynamics with friction and impact. *SIAM Review*, 42(1):3–39, Mar 2000.

[27] C. Studer, R. I. Leine, and C. Glocker. Step size adjustment and extrapolation for time-stepping schemes in non-smooth dynamics. *Intl. J. Numerical Methods in Engineering*, 76:1747–1781, 2008.

[28] S. Zapolsky. Pacer. https://github.com/PositronicsLab/Pacer.

[29] S. Zapolsky and E. Drumwright. Quadratic programming-based inverse dynamics control for legged robots with sticking and slipping frictional contacts. In *Proc. IEEE/RSJ Intl. Conf. Intell. Robots & Systems (IROS)*, 2014.

[30] S. Zapolsky, E. Drumwright, I. Havoutis, J. Buchli, and C. Semini. Inverse dynamics for a quadruped robot locomoting on slippery surfaces. In *Proc. Intl. Conf. Climbing Walking Robots (CLAWAR)*, Sydney, Australia, 2013.

[31] X. Zhang, M. Lee, and Y. J. Kim. Interactive continuous collision detection for non-convex polyhedra. *The Visual Computer (Proc. of Pacific Graphics)*, 22(9–11), 2006.