

# Particle Traces for Detecting Divergent Robot Behavior

Samuel Zapolsky\* and Evan Drumwright†

**Abstract**—The motion of robots and objects in our world is often highly dependent upon contact. When contact is expected but does not occur or when contact is not expected but does occur, robot behavior diverges from plan, often disastrously. This paper describes an approach that uses simulation to detect possible such behavioral divergences on real robots. This approach, and others like it, could be applied to validation of robot behaviors, mechanism design, and even online planning.

The *particle trace* approach samples robot modeling parameters, sensory readings, and state estimates to evaluate a robot’s behavior statistically over a range of conditions. We demonstrate that combining even coarse estimates of state and modeling parameters with fast multibody simulation can be sufficient to detect divergent robot behavior and characterize robot performance in the real world. Correspondingly, this approach could be used to assess risk and find and analyze likely failures, given the extensive data that such simulations can generate.

We assess this approach on actuated, high degree-of-freedom robot locomotion examples, a picking task with a fixed-base manipulator, and an unpowered passive dynamic walker. This research works toward understanding how multi-rigid body simulations can better characterize the behavior of non-soft robots.

## I. INTRODUCTION

The standard approach to validating robot behavior is simulated testing followed by *in situ* testing. This approach does not inspire confidence as simulations often fail to reflect real world behavior and *in situ* testing is tedious and slow. This problem has instigated research into formal verification methods for robotics (e.g., [13], [25]), which appears promising; intense study is currently underway to scale these approaches to higher degree of freedom systems. This paper explores an alternative path that is straightforward, easily implemented, and uses techniques already familiar to many roboticists to bridge the extremes of isolated physical simulation tests and full-on testing on real robotic hardware.

Our approach focuses specifically on robots that physically interact with their environment via contact (i.e., manipulation and locomotion). Contact is a governing factor for the movement of legged robots about their environment and for the manner in which robot hands pick up, move, operate, and otherwise manipulate objects in their environment. As the photos in Figure 1 depict, the unexpected presence or absence of contact can cause catastrophic failure.<sup>1</sup> We seek

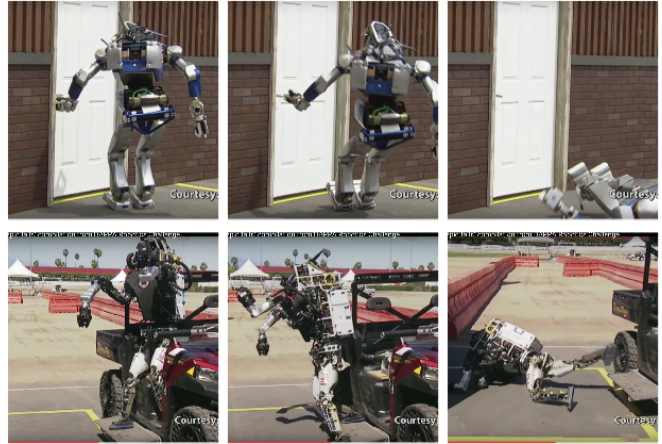


Fig. 1. Two robots performing tasks with anticipated contact (images captured from a video of DARPA’s recent robotics challenge). Both control strategies—(top) turning a door handle, (bottom) standing up from a seat—quickly diverge from plan without the anticipated contact, and the robots fall catastrophically. The particle trace approach can be applied to identify brittle aspects of a plan.

to identify policies and designs that are sensitive to modeling and estimation errors. We perform this task by detecting novel, nonsmooth, bifurcating events that appear between simulations of perturbed robot models (henceforth denoted *particle traces*). Our results indicate that novel, divergent behavior can be readily identified by locating grazing and bifurcating events, with even a small number of samples (see Section III-D). We seek to build upon these results by developing tools to gauge the robustness of controlled nonsmooth mechanical systems and discover ways in which they might fail.

We assess the particle trace approach using (1) a quadrupedal robot and a physically simulated model of this robot performing locomotion tasks; (2) a virtual manipulator robot performing a picking action; and (3) a physically simulated passive dynamic walker [8], for which we assess the walking stability empirically over perturbations in modeling parameters. This last task demonstrates the computational efficiency of the sampling approach—particularly in relation to ongoing related work in assessing stability of hybrid dynamical systems that has proven difficult to scale to higher dimensional systems.

## II. BACKGROUND AND RELATED WORK

Our present work considers robot dynamics that are well modeled by rigid bodies and rigid or nearly rigid contact. The approach is not necessarily predicated on these assumptions,

\*Samuel Zapolsky with the Computer Science Dept., George Washington University, Washington, DC, USA samzapo@gwu.edu

†Evan Drumwright is faculty in the Computer Science Dept., George Washington University, Washington, DC, USA drum@gwu.edu

<sup>1</sup>Russ Tedrake claimed that this problem was a dominant cause of failure of the robots in DARPA’s Robotics Challenge in a plenary session at Humanoids 2015.

but the necessary simulations must be sufficiently fast (likely precluding most deformable body simulations).

#### A. Nonsmooth mechanics

In addition to the challenges of analyzing nonlinear dynamics stability (of multi-rigid body systems), the problem discussed in this paper requires consideration of nonsmooth mechanical systems [5], for which velocities can change discontinuously due to impacts and even non-impacting contact with Coulomb friction [30]. Multibody dynamics with rigid contact and Coulomb friction—which captures important stick-slip transitions—can be modeled as a piecewise differential algebraic equation (DAE):

$$\ddot{\mathbf{q}} = \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u}) \quad (1)$$

$$\mathbf{0} = \phi(\mathbf{q}) \quad (2)$$

where  $\phi(\cdot)$  is a set of *active* algebraic constraints, out of  $m$  total constraints. Some constraints are always active, like bilateral joint constraints. Other constraints are only active if certain conditions are met; e.g., a contact constraint between two polyhedra would only be active when the bodies are in contact at that point and they would otherwise (i.e., without the constraint in place) interpenetrate at that point. These kind of problems can also be formalized as a differential complementarity problem [23].

#### B. Stability analysis and control of nonsmooth systems

A number of researchers have studied stability analysis and control of nonsmooth mechanical systems [5], [31], [32], [27], [26], [16], [17], [24], [25], for which hybrid dynamical systems have been a common formal model. These systems have been applied toward the study of walking machines and robots, which we have also used as an illustrative application.

#### C. Computer Animation

[33] uses *visual plausibility*, qualitatively undetectable perturbations to collision parameters, to generate a set of possible “worlds”. Their idea is essentially the inverse of ours: where we focus on using various perturbations to a simulation to characterize robotic behavior and identify possible divergences, they perturb simulations to try to find plausible, but low probability events.

#### D. Robust Control

Robust controllers seek control policies that are effective given bounded errors [3], which generally appear as initial state and control signal perturbations; work in robust model predictive control (MPC) also accounts for error in the system model. Such systems are tested and/or designed by stress-testing that composes perturbing modeling assumptions on initial state, control signal, environment geometry, or contact data and selecting a controller that performs best across all cases. Robust control has been used for improving the reliability or reducing uncertainty of locomoting systems [34], [22], [28], [7], effecting grasping behaviors [14], [20], [35], [38], and planning trajectories that reduce system uncertainty [12]. Validating such robust controllers through

stress-testing, commonly known as *falsification*, seeks to find counter examples to the robustness claims of a controller. The nonsmooth mechanical systems that we consider preclude use of existing control theoretic tools for falsification (like [4], [11], [1]). Our present work approaches falsification using state of the art nonsmooth dynamics simulation.

#### E. Monte Carlo method and particle approaches

There exists a tenuous relationship between our approach and Monte Carlo and particle-based approaches for state estimation of nonsmooth systems [10], [37], [15], [19], [18]. The extent of the similarity is that both use stochasticity in addition to probability distributions over state to generate time series datasets (*traces*) for each perturbation to dynamical system (*particle*) parameters. These works expose the interaction between dynamics and rigid contact mechanics, as developed in theory of linear complementarity systems [29].

### III. APPROACH

The particle trace approach attempts to locate novel contact events, task failures, and other nonsmooth hybrid state transitions that can have a large impact on system state.

#### A. Sampling approach

The fundamental unit of the sampling approach is the *particle*, which describes an instance of the robot (i.e., kinematic, dynamic, geometric, control, sensing, and communication parameters), and its state to the simulator software as a set of parameters (see Figure 2).

**Figure 2: Particle parameters**

*(Parameters determined at the start of a particle trace)*

**Model:**

link density: {1×base, 4×hip, 4×thigh, 4×shin, 4×foot}

link length: {4×hip, 4×thigh, 4×shin}

link radius: {4×hip, 4×thigh, 4×shin, 4×foot}

joint axis (conical error): 12×actuated joints

**Environment:**

surface friction, surface compliance, contact model,

surface geometry

**Initial state:**

$x, y, z, \psi, \phi, \theta, q_1 \cdots q_{12}, \dot{x}, \dot{y}, \dot{z}, \dot{\psi}, \dot{\phi}, \dot{\theta}, \dot{q}_1 \cdots \dot{q}_{12}$

**Other:**

control lag, actuated joint transmission/gearing, link compliance

*(Parameters determined during particle trace execution)*

**Encoder noise:**  $q_1 \cdots q_{12}, \dot{q}_1 \cdots \dot{q}_{12}, \ddot{q}_1 \cdots \ddot{q}_{12}$

**Force sensor noise:**  $u_1 \cdots u_{12}$

**IMU noise:**  $\ddot{x}, \ddot{y}, \ddot{z}, \dot{\psi}, \dot{\phi}, \dot{\theta}$

**GPS noise:**  $x, y$

**Magnetometer noise:**  $\psi, \phi, \theta$

**State Estimation noise:**  $z, \dot{x}, \dot{y}, \dot{z}, \dot{\psi}, \dot{\phi}, \dot{\theta}$

**Sensed actuator torque noise:**  $u_{des,1} \cdots u_{des,12}$

**Other:** control lag jitter

Fig. 2. A non-exhaustive list of parameters that should be considered when simulating the uncertainty in the computational model of the *Links* robot and its environment.

Each particle is generated by pseudo-random sampling on each of the uncertain elements of a robotic simulation and will differ slightly from the expected model of the robot.

Each particle is then “traced” over a user-specified time by simulating the sensing, dynamics, and control policy (see Figure 3). This paper focuses on multi-rigid body dynamics with rigid contact because these models capture the first-order effects without excessive parameter tuning and because the models’ dynamics can be simulated orders of magnitude more rapidly than the next more representative set of models (i.e., deformable bodies).

### B. Generating particles

A particle’s parameterization determines the evolution of the simulated robotic system and serves as the only cause of diverging behavior between the individual particle traces. *The perturbation to the “true” model is not observed by the robot’s planning, control, and state estimation systems, which generally assume a known robot model (henceforth denoted the expected model) when, e.g., calculating dynamic and kinematic information.*

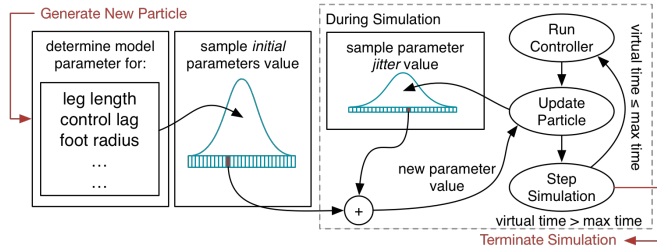


Fig. 3. Example of a *particle trace*. When a new particle is generated: (1) Initial values of parameters are determined before the first simulation update. (2) Parameters that experience noise online (at each simulation update) are modified from their initial value by sampling a jitter value.

We sample a random perturbation to each particle parameter from an estimated distribution on its uncertainty (usually limiting samples to within three standard deviations, excluding the tails). Each model and estimation parameter of the particle is then offset from its expected model value using the sampled perturbation *before* starting the simulation. While a particle is traced as the robot follows its control policy over the course of simulation, additional perturbations to sensor noise and control lag jitter<sup>2</sup> are sampled *on each control loop iteration*. Figure 3 illustrates this sampling process for a single particle parameter.

We have used normally or uniformly distributed uncertainty on particle parameters (link dimension, link density, joint axes, control lag), and homoscedastic (fixed over time), normally distributed uncertainty for parameter noise (contact friction, control lag jitter, sensor data). The experiments described in §IV used Gaussian and uniform distributions over wide ranges (to effect a safety factor), and we did not attempt to tune the distribution parameters. The number of distributions and parameters (see Figures 2 & 4) would likely make such tuning infeasible in any case. Further work will be necessary to assess the ramifications of modeling parameters, state estimates, and sensory noise distributions that tend to

<sup>2</sup>Control lag jitter is a small delay that is added/subtracted from the control lag and randomly selected on each control loop iteration.

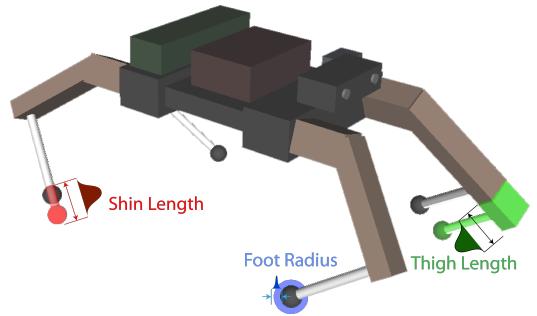


Fig. 4. A depiction of the probabilistic geometric parameters of a legged robot: shin length, thigh length, and foot radius.

follow heteroscedastic (varying over time), leptokurtic (fat-tailed), or skewed distributions.

### C. Computation Time

Each particle can be integrated stably in the *Moby* simulator at approximately real-time speed: each second of time in simulation (virtual time) takes about a second to compute (wall time). Pseudorandom sampling decouples allowing producing any number of particle traces  $s$  in parallel in linear time  $\mathcal{O}(\frac{sT}{cm})$  with respect to the number of particle traces ( $s$ ), where  $m$  is the real-time factor of the simulation ( $m > 1$  being faster than real-time),  $T$  is the duration of the experiment in virtual time, and  $c$  is the number of processor cores available on the machine. With enough cores ( $c \geq s$ ), this algorithm can run in constant time.

### D. Identifying divergent behaviors

We hypothesize that a robot’s behavior is likely to be predictable if no nonsmooth events occur or if nonsmooth events occur at similar times between particles. Similarly, we hypothesize that behavior is harder to predict if some particle traces experience novel nonsmooth events or nonsmooth events occurring in novel sequences. *Event* is used to denote a mode switch in a hybrid dynamical system, which can occur in our application upon impacts and upon switching between sliding, sticking, and rolling contact.

Accordingly, our approach detects both “grazing” events (events likely to occur in only very particular conditions. e.g., an accidental foot scuff or finger brushing against an object when only intending to pass near the object) and near-miss events (i.e., a narrowly avoided collision or other grazing event). Such events are known in the hybrid dynamical systems literature as “grazing bifurcations” [6]. Our second hypothesis anticipates that outcomes will be challenging to predict when a robot operates around such regions of state space. As examples, a slightly longer leg than is modeled might cause the foot to scuff the floor unexpectedly, and a foot heavier than its model might be unable to clear the top of a step when climbing stairs.

Given the immense computational resources that can be applied to produce the particle traces, we require a means of identifying divergent behavior in potentially massive amounts of generated data. We can automatically identify

divergent behavior by searching for high-variance or multi-modality in the distribution of final states from all particle traces. We can also detect a grazing event even before divergent behavior might occur by analyzing event ordering. A sequence of events indicative of a grazing event might identified by novel events, a novel *sequence* of events, or novel outcomes to similar events (between particle traces).

Divergent behavior can also be detected at a goal-oriented level (i.e., rather than detecting novel events) by, e.g., searching for failure to perform a specific task: falls during locomotion, or dropped objects in a pick-and-place task, etc. Normal behavior can also be determined through consensus, where an outlier would be indicative of divergent behavior. Our demonstrations in §IV detect divergent behavior strictly through task failure detection. Further study will focus on efficient detection of detect divergent behavior between particle traces.

#### IV. EXPERIMENTS

We assessed the particle trace approach using high-dimensional, non-smooth robotics scenarios. We assess the approach on locomotion scenarios—which feature frequent contact events and probable unexpected, or poorly timed, destabilizing collisions—and a manipulation scenario, because simulation-based plans for grasping have repeatedly proven to be brittle *in situ* (as depicted in Figure 1). Our experiments use the multi-body dynamics simulator *Moby*, which has been shown to produce behavior consistent with real robots [2], because it uses continuous collision detection [21] allowing it to locate contact events precisely (see [36]).

The following subsections demonstrate how particle traces can be used to determine the sensitivity of a robot to modeling uncertainty (§IV-A); efficiently locate bifurcating events (§IV-C); and assess plan robustness (§IV-B). §IV-D shows how the aggregate particle behavior can accurately characterize *in situ* robot behavior.

##### A. Passive dynamic walker

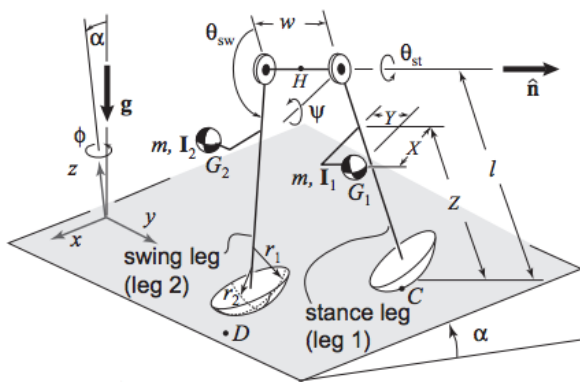


Fig. 5. Model and state parameters for a passive dynamic walker from Coleman and Ruina [9]. This system is stable, but the system’s region of stability is small.

We applied the particle trace method to assess the parametric (modeling) limits with respect to stability of a bipedal

passive dynamic walker [9] (illustrated in Figure 5). We sampled over the inertial and kinematic parameters of the walker, resulting in four hundred different particles. A fixed point cycle computation process (as described in [9]) was applied to each particle to compute initial conditions that yield a walking cycle. Each particle trace was simulated for sufficient duration to allow the biped to walk up to 20 steps.

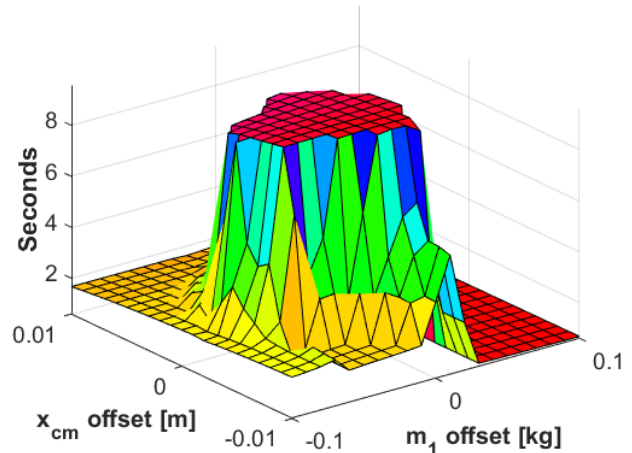


Fig. 6. Stability region for the passive dynamic walker. Upright time over an 8s walk for varied c.o.m. positions ( $x_{cm}$ ) and foot mass ( $m_1$ ).  $x_{cm}$  is offset from an initial value of 0.0 and  $m_1$  is offset from an initial value of 1.0 kg (initially matching  $m_2$ ).

All modeling parameters described in Figure 5 were perturbed in our assessment:  $\{I_{xx}, I_{yy}, I_{zz}, I_{xy}, I_{xz}, I_{yz}, m_1, \alpha, X, Y, Z, l, w, r, m_2\} \sim \mathcal{N}(\mu_{param}, \sigma_{param})$  where  $\mu_{param}$  is the mean (expected) parameter value for a working simulated system and standard deviation  $\sigma_{param}$  is 5% of  $\mu_{param}$ . Region(s) of feasible walker parameters can be determined by examining the duration that parameterizations remain upright.

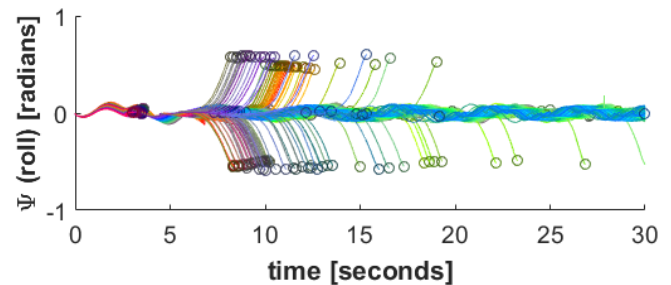


Fig. 7. 400 randomly sampled particles of the passive walker simulated over 30s (virtual time); the walker’s roll orientation is plotted over the course of the experiment. Termination points (falls) are marked with circles.

**Results:** Figure 6 depicts the stability region of the passive walker with respect to a grid sampling over the c.o.m. offset (front-back) and left leg mass. Grid sampling becomes costly when attempted over all parameters due to the curse-of-dimensionality. A pseudorandom sampling over all fifteen parameters detects similar points of failure for the walking



robot, corresponding to the steep ledges seen in Figure 6. Figure 7 displays one state value (roll) with respect to time. State data was collected from four hundred particle traces for this plot. Falls occur in clusters (at 2s and 8s) as the walker passes through a region of its state space with a grazing bifurcation (usually a scuffed or stubbed foot for this system).

Figure 7 depicts a single bifurcating event, scuffing a foot on the ground, generating a compact cluster of state data between particles. Figure 6 indicates that a more robust walker can be obtained by increasing  $m_1$  slightly ( $\approx 0.02$  kg).

### B. Assessing grasping plan robustness

We simulated an 11 DoF fixed-base manipulator robot performing a picking task (reaching, grasping, and lifting) on a ball within its reach. Two distinct plans were generated to achieve the tasks: Path A corresponds to the gripper approaching from above the ball then moving in a straight line between the gripper’s initial position and the expected position of the ball; Path B corresponds to moving sideways from above the ball during the approach to grasp the ball from its side. We ran two trials, with each trial generating forty particles. The trials were differentiated only by their approach trajectory and resulting grasp orientation on the ball. A trial was considered to be successful if the final position of the ball matched that planned. We expected that the relative success rates of the particle traces executing the two plans would reveal the more robust one; we posit that the absolute success rate of a plan would indicate the robustness of a plan executed *in situ*, but we did not test this hypothesis.

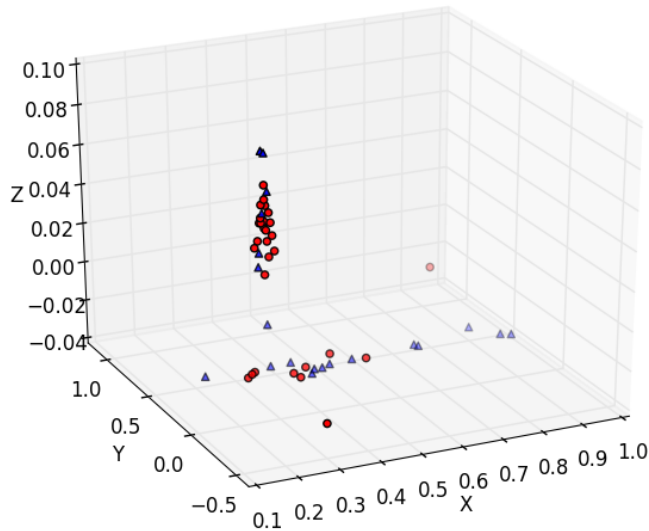


Fig. 8. Final position of the ball after the pick behavior following Path A (red circles) and Path B (blue triangles). Points arrayed along the bottom of the plot are resting on the ground plane, indicating failure of the pick task. A greater number of particle traces for Path B, implying that Path B is less robust.

**Results:** Both plans (Path A and Path B) successfully perform the action of picking up the ball in a well modeled simulation, which would seem to indicate that both paths are equally good. Particle traces show otherwise. Only upon observing the distribution of final states for each particle

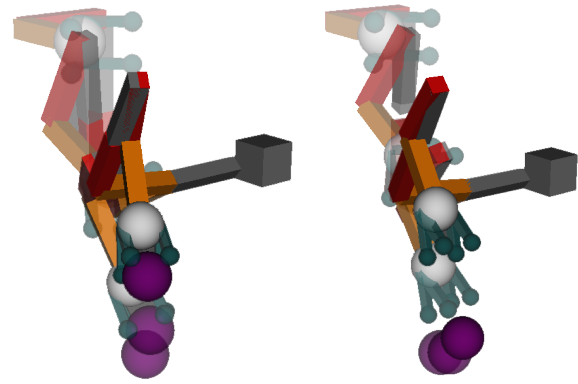
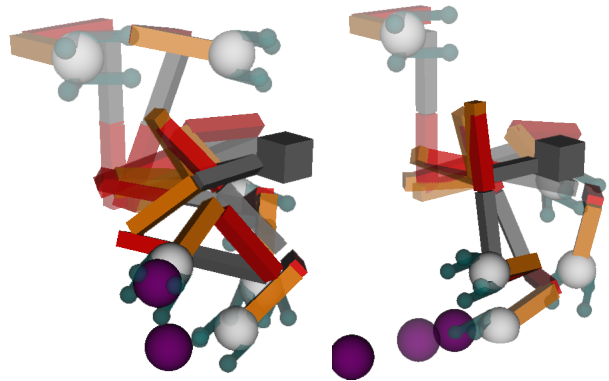


Fig. 9. Path A (top) and Path B (bottom) attempting to grasp the ball. Successes (left) maintain hold on the ball and failures (right) drop the ball or push it away. Colors are meant only to differentiate the numerous rigid bodies in the experiment from one another.



trace did we find that Path B resulted in more brittle (less robust) behavior than Path A. Path A resulted in a 88% success rate while Path B successfully completed the picking task only 63% of the time (a 40% performance differential). A finger brushing the ball (a grazing event) before beginning the grasp caused the ball to roll out of reach in many of the particle traces, resulting in failure for Path B. Figure 8 shows the final position of the ball in each of the particle traces. Examples of successful and failing attempts using each trajectory are depicted in Figure 9. Figure 9 depicts a visual realization of what a particle trace might look like within this framework.

### C. Detecting grazing bifurcation for a quadruped

We simulated placing a 18 DoF quadrupedal robot model (*Links*) next to a curb obstacle, and directed the quadruped to step over the curb while performing with variable step height (but with parameters otherwise fixed). A time-lapse depiction of the diverging behavior is shown in Figure 10. The robot clearly will collide with the curb if the robot does not step high enough. We predict that a grazing bifurcation will occur if the step height is approximately equal to the curb height: small changes in initial conditions, modeling parameters, or sensing (of, e.g., curb geometry) will determine whether or not the robot strikes the obstacle. We ran three trials, each of which uses one of three preset gait control policies that

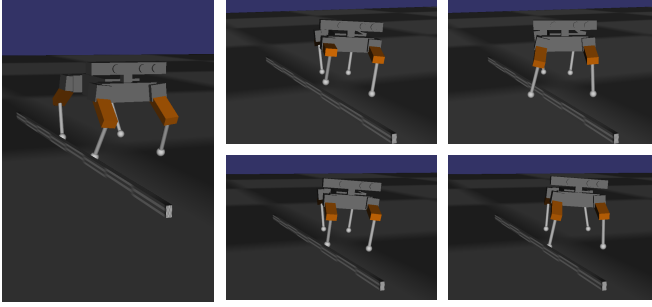


Fig. 10. A time-lapse of the virtual *Links* robot stepping over (top) or into (bottom) an obstacle given high and low step heights, respectively.

attempt two, three, and four cm step heights. The curb height was fixed at three cm.

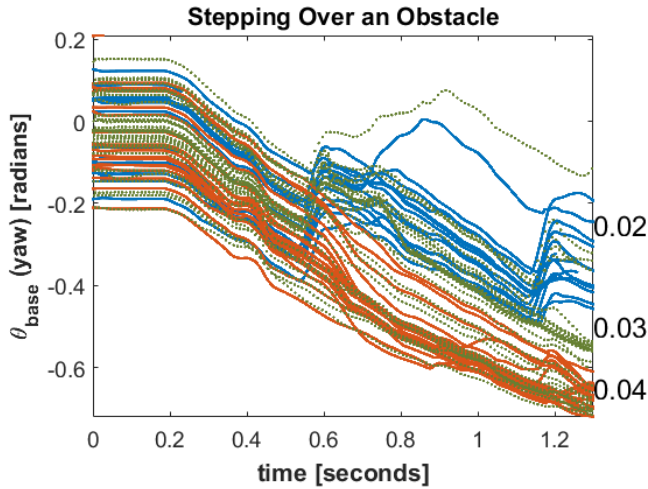


Fig. 11. Virtual quadrupedal robot base yaw when turning into a 3 cm tall curb obstacle. Each line represents a particle, and each color represents a policy. **Red** particles following a 4 cm step height policy (marked as 0.04 m on plot) step over the curb and continue to turn. **Blue** particles following a 2 cm step height policy (marked as 0.02 m on plot) strike the curb and are prevented from turning. **Green, dotted** particles following a 3 cm step height policy (marked as 0.03 m on plot), where step height matches the curb height (3 cm), experience bifurcation by only occasionally striking the obstacle. The behavior of these particles is less predictable.

**Results:** Grazing bifurcation drives the traces using the three cm step-height policy into one of two distinct clusters of states that correspond to robot models clearing and striking the curb, correspondingly (see Figure 11). Comparing the sequence of events in each particle trace for this policy, the cluster that corresponds to robots stepping over the obstacle contains the contact sequence { RH/ground, LH/ground, RF/ground, LF/ground }, and the cluster that corresponds to robots striking the obstacle contains the contact sequence { RH/ground, LH/ground, RF/obstacle, RF/ground, LF/ground }. The sampling strategy efficiently uncovers the grazing bifurcation, as the variance of the green paths in Figure 11 indicate; a grid search over the quadruped’s 36 dimensional state space is intractable. We expect that the robot’s *in situ* performance would be difficult to predict under this policy because the robot’s behavior is sensitive

to modeling and estimation uncertainty.

In contrast, the four cm step height policy allowed the quadruped to step over the curb for all traces and the two cm step height policy caused the robot to collide with the curb in all traces. It is reasonable to expect that the real robot would behave predictably under both of these policies.

#### D. Statistical behavior of a rigid robot from particle traces

The degree of correlation between the behavior of robots simulated over a timespan of seconds or minutes and those robots’ physically situated counterparts depends on many factors. A flexible robot may evince little of the behavior of its virtual counterpart simulated using multi-body dynamics, for example. This section focuses on the correlation between simulated and *in situ* behavior for a scenario that *should* be well modeled using fast simulation tools. This issue is important because one can only expect grazing bifurcations located in simulation to be informative if there is some correlation between simulated and *in situ* behavior.

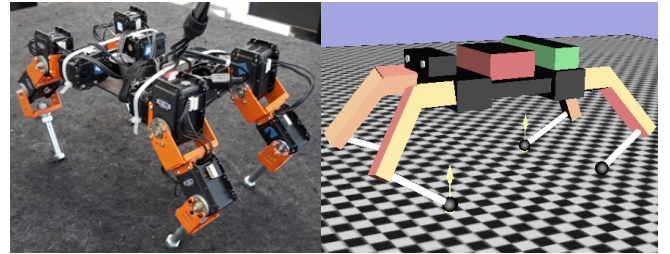


Fig. 13. (Left) the *Links* robot in position to begin a walking experiment. (Right) A snapshot of our physical model of *Links* in simulation.

*a) Robot:* The robot used in physical trials, *Links*, is an 18 degree-of-freedom (12 actuated) quadruped robot constructed from Dynamixel actuators and steel links (see Figure 13). Base orientation is recorded by IMU that produces updates at 100 Hz. Modeling uncertainties and errors on even such a small robot are legion and include, but are not limited to, the rigid body assumption, gear backlash, communication delay, IMU sensing delay, the rigid contact assumption, and back EMF. The modeling and state parameters sampled are listed in Figure 3.

*b) Dynamics model:* The virtual quadrupedal robot was modeled using a box geometry for the base link inertia and geometry, cylinders for the limb link inertias and geometries, and spheres for foot inertia and geometries. Modeling parameters for the virtual quadruped were set from measurements collected from the *Links* robot. There are no compliant elements in the structure of *Links* (unless one counts the transmission), allowing it to be readily modeled as a multi-rigid body.

*c) Control Policy:* We used a simple gait planning system to move the robot in a walking gait around a one foot diameter circle. The desired planar motion of the base of the robot  $\{\dot{x}, \dot{y}, \dot{\theta}\}$  is input to the planner, which produces a trajectory for each foot that attempts to drive the robot base toward the intended operational-space configuration while maintaining sticking contact with the ground.

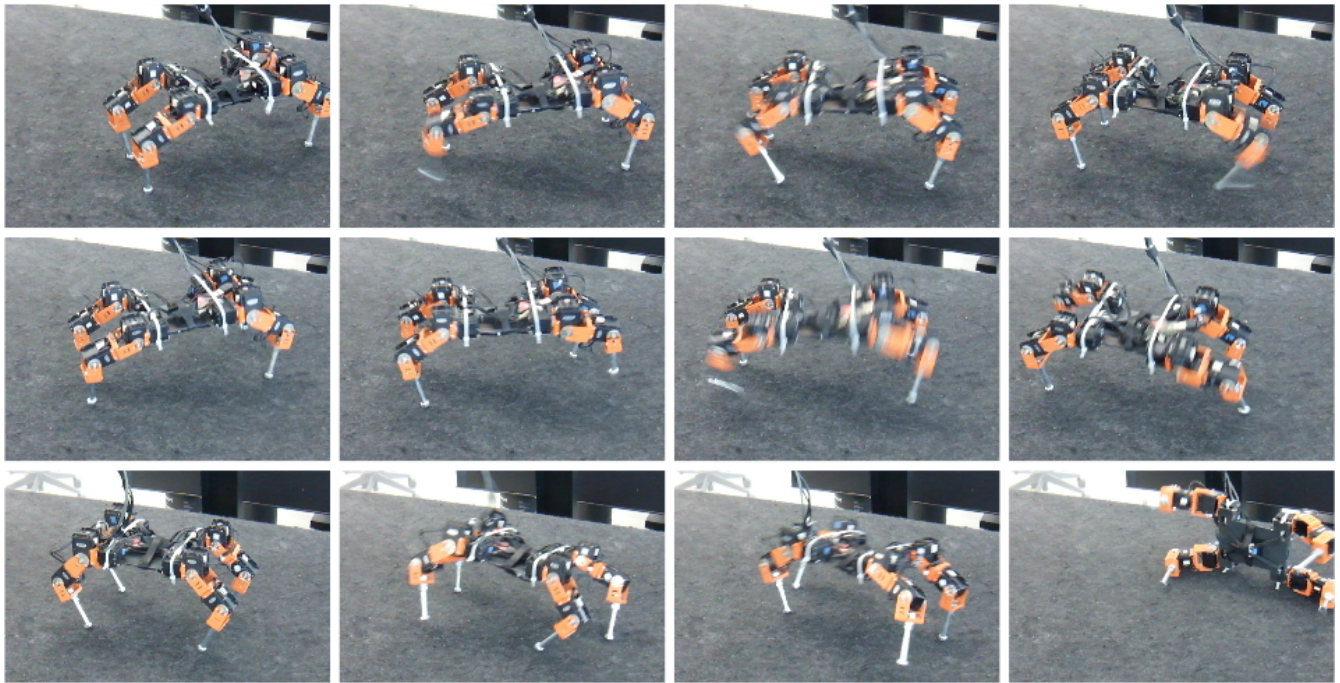


Fig. 12. A two second time-lapse of *Links* walking with a gait period duration of: 0.6 seconds (Top); 1.0 seconds (Middle); 1.5 seconds (Bottom). The robot became progressively less stable as the gait period duration increased.

We adjusted a single gait parameter (gait period duration) and observed how it affected the behavior of the robot. Gait period duration was adjusted from an empirically observed stable value of 0.6 seconds per gait cycle, upward to a value where we had previously observed definite failure: 1.5 seconds per gait cycle. Each particle was traced over 20s of virtual time or until a fall, and *Links* was permitted to walk for 20s of wall time.

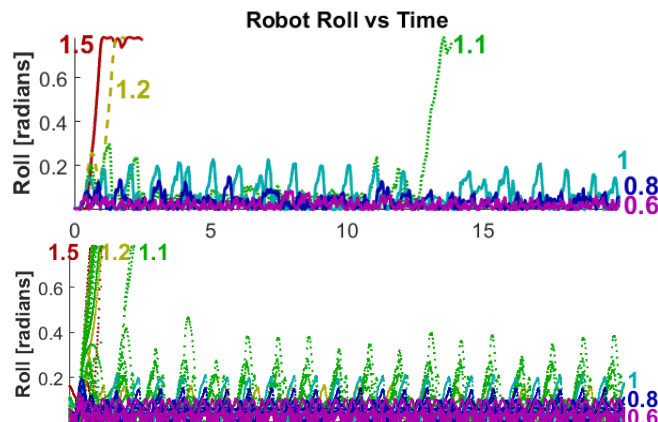


Fig. 14. Roll orientation data for the *Links* robot walking in a circle: (Top) *in situ* and (Bottom) *in sim*. Each line is labeled with its corresponding value for the gait period duration.

**Results:** A time-lapse depiction of this experiment is presented in Figure 12. We recorded the roll orientation of the robot base and labeled a configuration a fall if the roll exceeded  $\frac{\pi}{2}$  radians from vertical. We observed that *Links* completed the 20s walk *in situ* without falling when no

particle traces exhibited a fall. When the robots fell in some traces, *Links* walked for 10s *in situ* before falling. When the models in all traces fell (after the first step), *Links* fell on its first step *in situ* as well (see Figure 14). While a simulation of the robot from modeling and estimates might not have exhibited the robot’s *in situ* behavior for a given policy, the aggregate behavior over all particle traces matched the *in situ* performance well (see Figure 15).

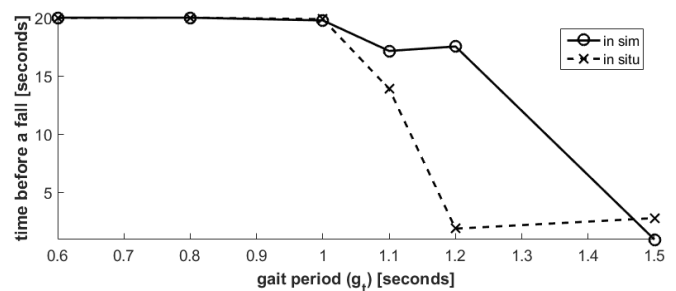


Fig. 15. Duration of time until a fall of the locomoting robot plotted with respect to the gait period duration parameter. (× mark, dotted line) Duration of wall time until a fall of the *Links* robot. (○ mark, solid line) Average duration of virtual time until a fall for all particle traces.

## V. DISCUSSION

We have been able to locate seemingly hard to locate grazing bifurcations (i.e., ones that lie in small volumes in state space) for high dimensional systems with very few particle traces. Since each particle trace is completely independent, particle traces can be generated in an “embarrassingly parallel” manner. So not only is the particle trace approach versatile and simple to implement, it can be quite fast given sufficient computational resources.



We believe the following questions now require much deeper investigation: (1) What scenarios can be constructed for which grazing bifurcations are computationally demanding to find (much like the “bugtrap” scenarios for motion planning)?; (2) What other dynamic scenarios can statistical ensembles of physically simulated robots reliably characterize (and where will such simulations fail to characterize behavior)?; (3) Since simulations are capable of generating huge quantities of data, how can such state space telemetry data be efficiently “mined” to locate clusters of similar high-level behavior?

## REFERENCES

- [1] H. Abbas, G. Fainekos, S. Sankaranarayanan, F. Ivancic, and A. Gupta. Probabilistic temporal logic falsification of cyber-physical systems. *ACM Transactions on Embedded Computing Systems*, 12(2), May 2013.
- [2] Daniel M. Aukes, Barrett Heyneman, John Ulmen, Hannah Stuart, Mark R. Cutkosky, Susan Kim, Pablo Garcia, and Aaron Edsinger. Design and testing of a selectively compliant underactuated hand. *Int. J. Robot. Res.*, 33(5), 2014.
- [3] Alberto Bemporad and Manfred Morari. *Robust model predictive control: A survey*, volume 245 of *Lecture Notes in Control and Information Sciences*, chapter Robust model predictive control: A survey, pages 207–226. Springer London, London, Sep. 2007.
- [4] M. S. Branicky, M. M. Curtiss, J. Levine, and S. Morgan. Sampling-based planning, control and verification of hybrid systems. In *IEE Proceedings - Control Theory and Applications*, Sep. 2006.
- [5] Bernard Brogliato. *Nonsmooth Impact Mechanics: Models, Dynamics, and Control*. Springer-Verlag, London, 1996.
- [6] C. J. Budd. Non-smooth dynamical systems and the grazing bifurcation. In *Nonlinear Mathematics and its Applications*, pages 219–235. Cambridge Univ. Press, 1996.
- [7] Samuel A. Burden, s. Shankar Sastry, Daniel E. Koditschek, and Shai Revzen. Near-simultaneous footfalls lend stability to multi-legged gaits. In *Dynamic Walking*, 2015.
- [8] Michael J. Coleman, Mariano Garcia, Katja Mombaur, and Andy Ruina. Prediction of stable walking for a toy that cannot stand. *Phys. Review E*, 64(2), 2001.
- [9] Michael J. Coleman and Andy Ruina. An uncontrolled toy that can walk but cannot stand still. *Phys. Review Lett.*, 80(16):3658–3661, 1998.
- [10] D. J. Duff, T. Morwald, R. Stolkin, and J. Wyatt. Physical simulation for monocular 3D model based tracking. In *Proc. IEEE Intl. Conf. Robot. Autom. (ICRA)*, 2011.
- [11] Joel M. Esposito, Jongwoo Kim, and Vijay Kumar. *Algorithmic Foundations of Robotics VI*, volume 17 of *Springer Tracts in Advanced Robotics*, chapter Adaptive RRTs for Validating Hybrid Robotic Control Systems, pages 107–121. Springer, October 2005.
- [12] Aaron M. Johnson, Jennifer E. King, and Siddhartha Srinivasa. Convergent planning. *IEEE Robotics and Automation Letters*, 1(2):1044–1051, 2016.
- [13] B. Johnson and Hadas Kress-Gazit. Analyzing and revising synthesized controllers for robots with sensing and actuation errors. *Int. J. Robot. Res.*, 34:816–832, 2015.
- [14] Junggon Kim, Kunihiro Iwamoto, James J. Kuffner, Yasuhiro Ota, and Nancy S. Pollard. Physically based grasp quality evaluation under pose uncertainty. *IEEE TRANSACTIONS ON ROBOTICS*, 29(6), 2013.
- [15] Michael Koval, Mehmet Dogar, Nancy S. Pollard, and Siddhartha S. Srinivasa. Pose estimation for contact manipulation with manifold particle filters. In *Proc. IEEE/RSJ Intl. Conf. Intell. Robots & Systems (IROS)*, 2013.
- [16] R. Leine and N. van de Wouw. *Stability and convergence of mechanical systems with unilateral constraints*. Springer Verlag, 2008.
- [17] R. Leine and N. van de Wouw. Stability properties of equilibrium sets of non-linear mechanical systems with dry friction and impact. *Nonlinear Dynamics*, 51(4):551–583, 2008.
- [18] S. Li, S. Lyu, Jeff C. Trinkle, and Wolfram Burgard. A comparative study of contact models for contact-aware state estimation. In *Proc. IEEE/RSJ Intl. Conf. Intell. Robots & Systems (IROS)*, 2015.
- [19] Shuai Li, Siwei Lyu, and Jeff Trinkle. State estimation for dynamic systems with intermittent contact. In *Proc. IEEE Intl. Conf. Robot. Autom. (ICRA)*, 2015.
- [20] Jeffrey Mahler, Sachin Patil, Ben Kehoe, Jur van den Berg, Matei Ciocarlie, Pieter Abbeel, and Ken Goldberg. Gp-gpis-opt: Grasp planning with shape uncertainty using gaussian process implicit surfaces and sequential convex programming. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2015.
- [21] Brian Mirtich. *Impulse-based Dynamic Simulation of Rigid Body Systems*. PhD thesis, University of California, Berkeley, 1996.
- [22] K. Mombaur, R. W. Longman, H. G. Bock, and J. P. Schlöder. Open-loop stable running. *Robotica*, 23(1), 2005.
- [23] Jong-Shi Pang and David E. Stewart. Differential variational inequalities. *Math. Program., Ser. A*, 113:345–424, 2008.
- [24] A. Papachristodoulou and S. Prajna. Robust stability analysis of nonlinear hybrid systems. *IEEE Trans. Autom. Control*, 54(5), 2009.
- [25] Michael Posa, Mark Tobenkin, and Russ Tedrake. Stability analysis and control of rigid body systems with impacts and friction. *IEEE Trans. Autom. Control*, 2015.
- [26] Stephen Prajna, Ali Jadbabaie, and George J. Pappas. A framework for worst-case and stochastic safety verification using barrier certificates. *IEEE Trans. Autom. Control*, 52(8), 2007.
- [27] Stephen Prajna and Anders Rantzer. Convex programs for temporal verification of nonlinear dynamical systems. *SIAM J. Control Optim.*, 2007.
- [28] Oguz Saglam and Katie Byl. Robust policies via meshing for metastable rough terrain walking. In *Robotics: Science and Systems (RSS)*, 2014.
- [29] J. Shen and J. S. Pang. Linear complementarity systems: Zeno states. *SIAM J. on Control and Optimization*, 44(3):1040–1066, 2005.
- [30] David E. Stewart. Rigid-body dynamics with friction and impact. *SIAM Review*, 42(1):3–39, Mar 2000.
- [31] Claire J. Tomlin, John Lygeros, and Shankar Sastry. A game theoretic approach to controller design for hybrid systems. *Proc. IEEE*, 88:949–969, Jul 2000.
- [32] Claire J. Tomlin, Ian Mitchell, Alexandre M. Bayen, and Meeko Oishi. Computational techniques for the verification of hybrid systems. *Proc. of the IEEE*, 91(7), 2003.
- [33] Christopher D. Twigg and Doug L. James. Many-worlds browsing for control of multibody dynamics. *ACM Trans. on Graphics*, 26(3), July 2007.
- [34] J. M. Wang, D. J. Fleet, and A. Hertzmann. A optimizing walking controllers. *ACM Trans. on Graphics (Proc. of SIGGRAPH Asia)*, 168, 2009.
- [35] Jonathan Weisz and Peter K. Allen. Pose error robust grasping from contact wrench space metrics. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2012.
- [36] Samuel Zapolsky and Evan M. Drumwright. Adaptive integration for controlling speed vs. accuracy in multi-rigid body simulation. In *Proc. IEEE/RSJ Intl. Conf. Intell. Robots & Systems (IROS)*, 2015.
- [37] L. E. Zhang, S. Lyu, and J. C. Trinkle. A dynamic bayesian approach to real-time estimation and filtering in grasp acquisition. In *Proc. IEEE Intl. Conf. Robot. Autom. (ICRA)*, 2013.
- [38] Yu Zheng and Wen-Han Qian. Coping with the grasping uncertainties in force-closure analysis. *Int. J. Rob. Res.*, 24(4):311–327, April 2005.