Motion planning with efficient spring-mass behavior

Patrick Clary and Jonathan Hurst
{claryp, jhurst}@oregonstate.edu

Abstract

Energy-efficient, agile, and robust legged locomotion has been demonstrated on robots using reactive spring-mass behavior to determine foot placement and motor activation. However, these control methods are not compatible with common robotic motion planning methods, which rely on using feedback controllers to follow a planned trajectory through the robot's configuration space.

Trajectory following with feedback control is not ideal for spring-mass robots, which purposefully limit their control authority to allow natural spring-mass dynamics to take effect. In the presence of moderate disturbances, a straightforward feedback controller will be physically unable to bring the robot back to its desired trajectory.

To address this, we propose that a more suitable way of planning efficient legged locomotion is to plan through the ***action space*** of efficient ***reactive spring-mass behaviors***. The elements of this space consist of controllers for different periodic gaits and transient actions that can arbitrarily trade off efficiency and robustness. We show in a planar walking domain that planning in terms of reactive spring-mass behaviors is feasible for robust navigation through complex environments.

What is reactive spring-mass behavior?

Reactive spring-mass behavior is observed in robots that use their natural *spring-mass* dynamics to produce walking and running behavior, and which stabilize and control the system by *reacting* to changes in terrain or commanded velocity as if they are disturbances rather than by planning for them in advance.

Controllers for this type of behavior work in tandem with the mechanical design of the robot, lightly guiding the spring-mass dynamics by adding and removing energy and choosing where to place the robot's feet.

Cassie's predecessor ATRIAS uses a spring-mass walking controller that reactively stabilizes it after being kicked.

Robots that use reactive spring-mass behavior are capable of robust blind walking and running over rough terrain, and can be kicked and shoved while maintaining dynamic stability. However, they are not yet able to handle environments and obstacles that require some planning in advance for successful traversal.

Action spaces for legged motion planning

Motion planners compose their plans within a particular ***action space***. State trajectories are a common action space for motion planning, but this choice is not ideal for spring-mass legged robots.

- Underactuation, friction constraints, and motor limits make following trajectories difficult for spring-mass robots, but these are unavoidable or important for efficiency
- Choosing good foothold locations requires reasoning about obstacles and dynamics simultaneously, which is a challenge for most motion planning architectures

Instead, we propose that reactive spring-mass behavior can be composed into a more appropriate action space for spring-mass legged robots.

Cassie is an agile bipedal robot built for efficient spring-mass walking and running. Here, Cassie is shown performing a dynamic task that requires planned interaction with the environment. Several logical steps are shown, but what is the ***action space*** through which the planner communicates these steps to the controller?

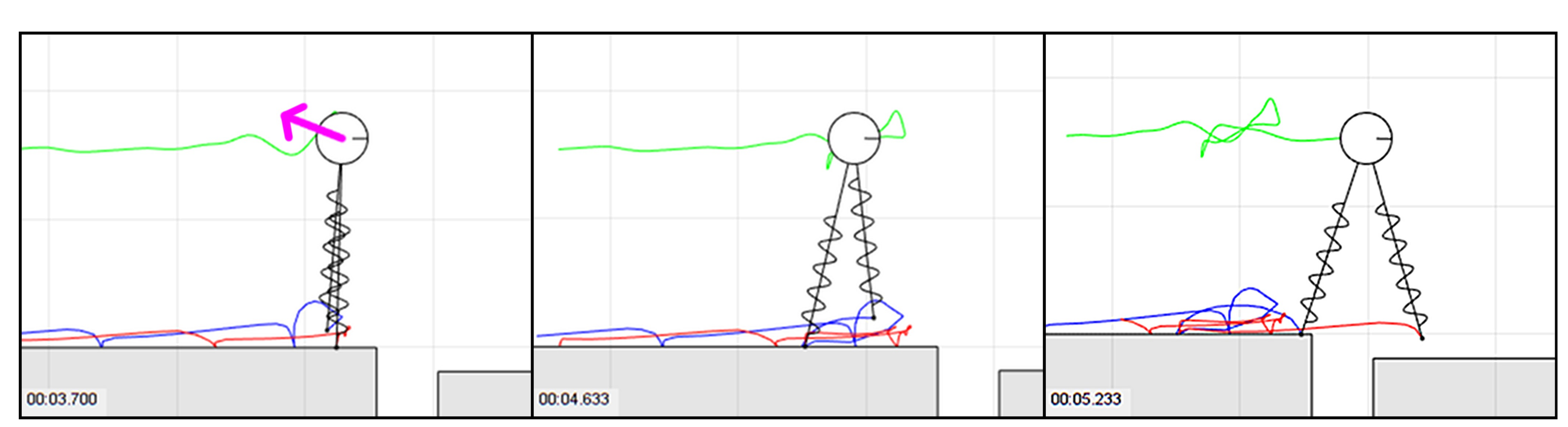
Planning with reactive spring-mass behavior

A planning action space based on reactive spring mass behavior is composed of a set of different *actions*, or controllers, that all exhibit reactive stability based on spring-mass dynamics, but that produce qualitatively different movement. Examples include controllers that cause the robot to move at different speeds or take higher steps.

In the absence of large disturbances, these actions produce easily predictable results. If the robot slips or is pushed, the underlying reactivity will cause its motion to deviate in a way that stabilizes the robot, even if it diverges from the original plan.

A motion planner can find a sequence of reactive controllers to execute that will result in the robot traversing an upcoming obstacle, assuming that no major disturbances occur. When large disturbances do occur, the reactive controller will take immediate action to keep the robot from falling. This gives the planner time to evaluate the situation and find a new plan that reaches its goal.

As a proof-of-concept, a short-horizon **Monte-Carlo planner** was used to plan the motion of a planar biped using an action space based on reactive spring-mass behavior. The biped's physical parameters were chosen to match Cassie, an agile and efficient spring-mass bipedal robot developed by our lab.

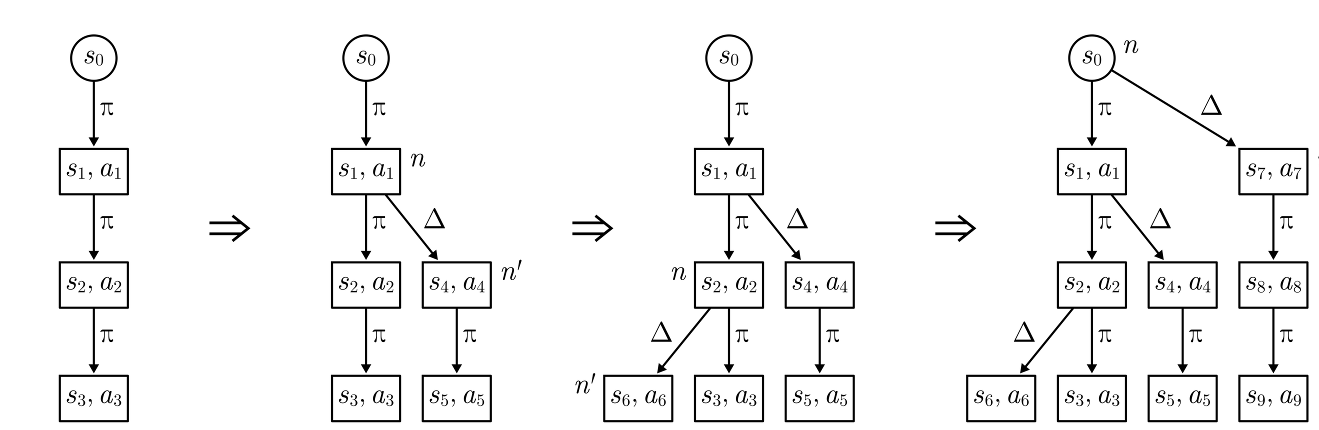


After an unexpected impulse, reactive behavior keeps the robot from falling. The planner adjusts to the new situation after some delay.

Despite the planner running at a slow update rate and with significant latency, the biped was able to navigate through complex terrain even in the presence of large disturbances. The biped was directed to walk forward over terrain consisting of varied-height platforms with gaps between them. Impulses were applied to the biped's body at random intervals, large enough to throw the biped into the air, but reactive controllers were able to keep the robot from falling for long enough to re-plan.

Monte-Carlo planning

Using reactive spring-mass behavior as the action space for motion planning breaks some of the assumptions made by standard motion planning algorithms like Rapidly-exploring Random Trees or online nonlinear optimization. To plan through this space, we developed a Monte-Carlo planner that sparsely samples actions out to a fixed horizon.



A sequence of search trees generated by our Monte-Carlo planner.

This planner works by predicting the result of a sequence of actions, which execute a particular reactive behavior controller for a fixed amount of time. It tries a branching tree of potential action sequences, and selects the sequence with the best predicted outcome. The first action in the sequence is executed on the robot, and the planning process restarts to select the next action.

We use a horizon of two seconds, with each action lasting for one third of a second. Outcomes are scored by giving a large penalty to states where the robot is about to fall over and rewarding velocities that are close to the target velocity.

Acknowledgments

This research is funded by DARPA Award W911NF-16-1-0002 and an ARCS Foundation scholar award.