Actuated SLIP Model: Partial Feedback Linearization and Two-Part Control Strategy

Giulia Piovan and Katie Byl
Robotics Laboratory, University of California, Santa Barbara

Introduction

Biological data suggest that legs regulate energy production and removal via muscle activation: in this work we consider the active SLIP model, an energetically non-conservative version of the SLIP model with series actuation.

We propose a strategy for actuator displacement in order to:
- add/remove energy from the system,
- analytically solve part of its dynamics,
- online computation of actuator displacement and leg positioning to drive the system to a desired state, even in the presence of terrain perturbation.

Passive SLIP model

Point mass, \( M \), attached to a massless spring leg, with length \( \ell \) and spring stiffness constant \( k \).

Running dynamics consist of two phases: the flight phase and the stance phase.

Equations of motion during stance not analytically solvable:

\[
\dot{\ell}(t) = -\frac{k}{M} (\ell(t) - \ell_0 - \ell_{act}(t)) - g \sin \theta + \ell \dot{\theta}^2,
\]

\[
\dot{\theta} = -2 \frac{\ell}{\ell} \dot{\theta} - \frac{g}{\ell} \cos \theta.
\]

Active SLIP model

- Piston-like actuator \( \ell_{act} \) added in series with the spring.
- Running dynamics consist of two phases: the flight phase and the stance phase.

Applications: Recovery from perturbations

We test the recovery capabilities of our controller for unexpected (positive or negative) perturbations on the terrain height of up to 50% of the leg length.

Foothold Placement Control

- Terrain where only a specific set of \( N \) footholds is allowed: minimizing the distance between the landing points and the desired footholds.
- The longer the horizon \( N \), the better the performance. The horizon length affects computational time. Optimal path via approximation reduces computational time and makes possible to extend the planning horizon. But the approximation, as such, carries an error.
- Trade-off between horizon length/computation time, and foothold error.

Applications: Hopping on Rough Terrain

- Goal: maintain the same forward velocity and the same distance from the terrain with respect to the last step.
- Assume that the terrain measurements are faulty

\[
J = \sqrt{PE_y^2 + PE_x^2}
\]

Error Reduction

What are the benefits of having a partial analytical solution to the stance phase?

We compute the error reduction as a percentage of the distance to the desired apex state:

\[
PE_y = 100 \| y_{real} - y_{appr} \| / \ell_0,
\]

\[
PE_x = 100 \| x_{real} - x_{appr} \| / \ell_0,
\]

and as a function of the relative spring stiffness \( \gamma = \ell_{act}/\ell_0 \).

Quadripeds and bipeds: \( \gamma \approx 10 - 20 \) per leg.

Choice of Actuator Constant Value

- Divide the stance phase in two parts, separated by the point of maximal leg compression.
- Chose two constant values for \( \ell_{act} \): one for the first part, \( \ell_{act1} \), and one for the second part \( \ell_{act2} \) of the stance phase.

- Using optimization algorithm, chose best actuator values and touch-down angle to reach a desired apex state.
- In presence of unexpected sensing error on terrain: upon touching the ground, recompute the actuator value for the second part of the stance phase based on the new terrain information.

Approximating the stance phase dynamics through partial feedback linearization

Divide total actuator displacement in two parts: \( \ell_{act} = \ell_{act1} + \ell_{act2} \), such as:

- \( \ell_{act1} \), has the purpose of cancelling the nonlinear terms in (1):

\[
\ell_{act1}(t) = \frac{M}{k} [g \sin \theta(t) - \ell(t)] \dot{\theta}(t)^2.
\]

- We drive the second term, \( \ell_{act2} \), to a constant value \( \ell_c \), moving with constant velocity.

- We are then able to solve analytically the o.e.m of the leg length \( \ell(t) \), and we use an approximation for the dynamics of the angle \( \theta(t) \).

References and Acknowledgments

Part of this work appeared as:
- G. Piovan and K. Byl, “Two-Element Control for the Active SLIP Model,” in IEEE Int. Conf. on Robotics and Automation, Karlsruhe, Germany, May 2013, pp. 5636–5642
- This work is supported by DARPA (Grant No. W911NF-11-1-007)