Full State Control of a SLIP Model by Touchdown Detection

Author(s):
Hutter, Marco (iD); Remy, C.David; Hoepflinger, Mark A.; Siegwart, Roland

FULL STATE CONTROL OF A SLIP MODEL BY TOUCHDOWN DETECTION[*]

MARCO HUTTER, C. DAVID REMY, MARK A. HOEPFLINGER,
ROLAND SIEGWART

Autonomous Systems Lab, Institute of Robotics and Intelligent Systems, Swiss Federal Institute of Technology (ETHZ), Zurich, Switzerland, mahutter@ethz.ch

To enhance the well known, self-stabilizing effects of Spring Loaded Inverted Pendulum (SLIP) models, researchers have proposed a variety of dead-beat controllers that adjust model parameters (angel of attack, spring stiffness), such that a disturbance is rejected within a single step. While such laws can be nicely encoded for disturbances in hopping height (by using the time of flight as a measure of vertical position), they suffer from substantial drift due to the missing information about the forward velocity and inaccuracies in the actual system model. Without requiring additional complex sensors, we propose a method to estimate the forward velocity of a SLIP model based solely on measuring the time of stance. This method is additionally able to perform realtime parameter estimation, which paves the road to implement a full state dead-beat controller that can reject arbitrary disturbances even in the presence of model and sensor errors.

1. Introduction

Biomechanical studies suggest that the dynamics of the center of mass in running gaits (including bounding, trotting, or galloping) can be described by a spring loaded inverted pendulum (SLIP) [1, 2], which combines a prismatic, mass-less, and elastic leg with a point mass at the Center of Gravity (CoG) (fig. 1). This simplified model allows fundamental theoretical research to explain several aspects of locomotion. It was, for example, found that for certain parameter selections, the SLIP model is dynamically stable for single leg hopping [3-6]. On the control side, different studies focused on dead beat control structures that allow the rejection of an arbitrary disturbance within one single hop by adjusting the angle of attack α or the spring stiffness k, allowing to continuously maintain constant velocity and apex height, even in extremely unstructured terrain [7]. Measuring the time of flight allows for registration of a varying ground level height and hence, due to the energy conservation properties of a SLIP model, the implementation of a constant-height hopping motion.
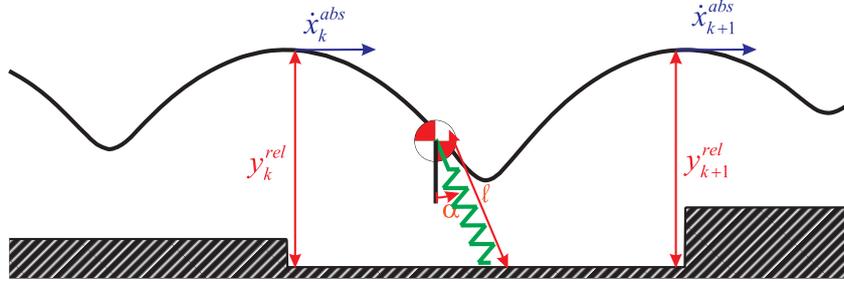
Figure 1 SLIP model hopping on uneven terrain. Since the energy within one single step remains constant, constant speed running can be interpreted as keeping apex height constant in every step, or directly controlling the absolute forward velocity.

Transferring these controllers to a real robotic device remains a challenging task. While the available control structures can result in instabilities due to model and sensor uncertainties, we are presenting a novel model based absolute speed estimation and realtime model parameter identification for improved controller robustness without the need for additional exteroceptive sensing capabilities.

2. Methods

2.1. *Return map based dead beat controller*

Established dead beat control strategies try to keep hopping height and forward speed constant from step to step by adapting either the angle of attack α or the spring stiffness k such that they create a fix point in the single step Poincaré map, which is defined at apex transit:

$$\left(y_{k+1}, \dot{x}_{k+1}\right) = P\left(y_k, \dot{x}_k, m, k, \alpha\right) \tag{1}$$

Due to energy preservation,

$$E_{k+1} = 0.5m\dot{x}_{k+1} + mgy_{k+1} = 0.5m\dot{x}_k + mgy_k = E_k \tag{2}$$

Forward velocity and apex height are not controllable independently and hence keeping only one of the values constant

$$y_{k+1} = P_y\left(y_k, \dot{x}_k, m, k, \alpha\right) \overset{!}{=} y_k$$
$$\dot{x}_{k+1} = P_x\left(y_k, \dot{x}_k, m, k, \alpha\right) \overset{!}{=} \dot{x}_k \tag{3}$$

automatically implies that the motion is periodic; i.e., it represents a fixed-point of (1). In this paper we want to focus on three different controller

implementations based on (3), and their stability against zero-energy disturbances. [NOTE: Since disturbances that change the energy-level can not be rejected in an energetically conservative system, they will be unaffected by any controller. This means that the first Eigenvalue ($1^{st} EV$) of the system is invariably 1. Such disturbances are hence not considered in the remainder of this paper].

CTRL1:
The control problem (3) can simply be solved by measuring the previous height $y_k = y_k^{meas}$ and under the assumption of $\dot{x}_k = \dot{x}_{des}$. While an error in hopping height is observed and controlled, errors in velocity direction are invisible for the controller and hence lead to an erroneous fix point search, that most often leads to a $2^{nd} EV > 1$. Consequently a large drift occurs even for a small random disturbance (0.5%) in the measurement of y_k^{meas} (fig. 2a).

CTRL2:
An improved control strategy additionally relies on measured absolute forward velocities $\dot{x}_k = \dot{x}_k^{meas}$. This allows solving problem (3) based on a full state measurement, and hence avoids the consequent errors in the search for a fix point. This controller shows an indifferent behavior ($2^{nd} EV = 1$) to measurement noise (fig. 2a) since errors both in apex height and velocity direction are observed but due to the single-step dead beat control structure not "destroyed" but rather carried over to the next step. Model errors (e.g. $\pm 0.5\%$ spring stiffness, fig. 2b) result in a drift that will never be compensated with both single step controllers CTRL1 and CTRL2.

CTRL3:
Since the absolute velocity should be kept constant and the full state is available, the dead beat control problem (3) can be re-formulated according to:

$$\dot{x}_{k+1} = P_{\dot{x}} \left(y_k^{meas}, \dot{x}_k^{meas}, m, k, \alpha \right) \overset{!}{=} \dot{x}_{des} \qquad (4)$$

This implementation shows a $2^{nd} EV = 0$, since both errors in velocity or height direction never lead to an error in forward velocity. Hence neither sensor noise nor model parameter errors lead to a velocity drift. Both are limited by an upper and lower boundary velocity defined by the fix point that belongs to the corresponding model parameter. This is based on the principle that velocity estimates are absolute, whereas height measurements are only relative to the (changing) ground.
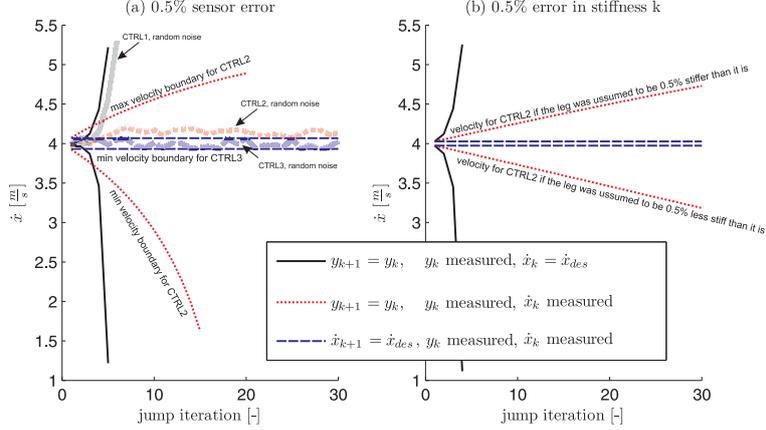
4

Figure 2 Velocity evolution over successive steps using different controllers under (a) sensor noise and (b) model errors. Having a random error (±0.5%) in measuring height or velocity results in a drift of CTRL 1 due to an instable pole, an indifferent behavior of CTRL 2 restricted in the region between the upper and lower limit and a dead beat behavior of the absolute velocity controller. Equal effects occur due to modeling errors (b).

The presented strategies depend on a correct measurement of height y_k^{meas} and/or forward velocity $\dot{x}_k^{meas}$. The varying relative hopping can be encoded very elegantly in a time law of the ballistic flight curve [7, 8]

$$y_k^{meas} = \frac{1}{2} g t_{fall}^2 + l \sin(\alpha) \tag{5}$$

A method for velocity determination based on internal states remains missing, since energy constancy is inapplicable due to the varying ground level and hence varying energy level.

2.2. *Absolute velocity estimation*

Without additional exteroceptive sensors, the absolute forward velocity $\dot{x}_k$ can be found by solving a boundary value problem (BVP) during stance phase.

$$\dot{x}_k^{meas} = BVP(\alpha_{TD}, \alpha_{LO}, t_s) \tag{6}$$

Similar to the time encoded height measurement, the stance time t_s is recorded. Using the touchdown (α_{TD}) and lift-off (α_{LO}) angles and the spring length $l_{TD} = l_{LO} = l_0$, the following equation for the ground contact (stance) dynamics $f(q,m,k)$ can be stated:

$$\ddot{q} = \begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} = f(q,m,k), \quad q(0) = \begin{pmatrix} -l_0 \sin(\alpha_{TD}) \\ l_0 \cos(\alpha_{TD}) \end{pmatrix}, \quad q(t_s) = \begin{pmatrix} -l_0 \sin(\alpha_{LO}) \\ l_0 \cos(\alpha_{LO}) \end{pmatrix} \tag{7}$$

Knowing that $\dot{x}_{TD} > 0$ and $\dot{y}_{TD} < 0$, this problem returns a unique solution and hence the absolute velocity $\dot{x}_k^{meas}$ can be calculated based on a time measurement. We implemented such a controller using the standard BVP solver in Matlab [9].

2.3. *Run time model parameter estimation - robustness*

In a real robotic system we have to deal with inaccuracy on the sensor signals (y_k^{meas} and $\dot{x}_k^{meas}$) as well as with modeling errors in m and k. When assuming that sensor signals are only noisy without additional offset, the effects of these random disturbances on the steady state behavior of the controlled system are negligible. More problematic for our dead beat controller are modeling errors that lead to wrong model based speed estimations and hence initiate an erroneous fix point search.

Due to the simplicity of our model, additional boundary conditions can be used for realtime parameter estimations to overcome this problem. The equations of motion (7) are only depending on the parameter quotient k/m. Based on flight time measurement, the pre-impact vertical velocity $\dot{y}_{TD} = -gt_{flight}$ is used to augment the BVP problem to

$$\left(\dot{x}_k^{meas}, \left(\frac{k}{m} \right)^{meas} \right) = BVP\left(\alpha_{TD}, \alpha_{LO}, t_s, \dot{y}_{TD} \right) \tag{8}$$

This allows parallel determination of speed and model parameters, again only depending on internal states, and time measurements.

2.4. *Approximate analytical solution*

So far, the return map based dead beat control problem (1) as well as the speed and parameter estimation BVP problems (6) and (8) require numerical solutions of the well known but unsolvable stance phase dynamics. This can be avoided by using approximate analytical solutions [10, 11]. In the following, we will focus on an implementation of [11], especially on the following two adapted equations:

$$\alpha(t) = \alpha_{TD} + (1 - 2h)\dot{\alpha}_{TD}t + \frac{2\dot{\alpha}_{TD}}{\hat{\omega}_0}\left[h\sin(\hat{\omega}_0 t) + \frac{|\dot{r}_{TD}|}{\hat{\omega}_0 l_0}(1 - \cos(\hat{\omega}_0 t)) \right]$$

$$r(t) = l_0 - \frac{|\dot{r}_{TD}|}{\hat{\omega}_0}\sin(\hat{\omega}_0 t) + \frac{\dot{\alpha}_{TD}^2 l_0 - g}{\hat{\omega}_0^2}(1 - \cos(\hat{\omega}_0 t)) \tag{9}$$

with $\hat{\omega}_0 = \sqrt{k/m + 3\dot{\alpha}_{TD}^2}$ and $h = \dfrac{\dot{\alpha}_{TD}^2 - g/l_0}{\hat{\omega}_0^2}$. The boundary conditions at $t = 0$ are inherently fulfilled, while the final conditions $\alpha(t_s) = \alpha_{LO}$ and $r(t_s) = l_0$ result in a nonlinear system of equations that can be solved for the polar velocities $\dot{\alpha}_{TD} = \dot{\alpha}_{LO}$ and $\dot{r}_{TD} = -\dot{r}_{LO}$ which are directly connected to the forward velocity $\dot{x}_k$:

$$\dot{x}_k = -\dot{r}_{LO}\sin(\alpha_{LO}) - l_0\dot{\alpha}_{LO}\cos(\alpha_{LO}) \tag{10}$$

In addition to this analytical BVP problem solution, the approximate solution in combination with the ballistic flight curve can also be used for the return map based fix point search.

3. Results

3.1. *Model parameter estimation*

The previous results showed that the step-to-step dead beat controllers even in case of perfect knowledge about forward velocity are not applicable when we have to deal with modeling errors. Defining the robustness against model error with the region around the nominal parameter m_0 and k_0 where the $2^{nd}\,EV$ of the closed loop system remains smaller 1, the controller including BVP speed estimation (6) and absolute velocity return map based angle of attack adjustment (4) can deal with errors of $k/m \in [0.88, 1.002] \cdot k_0/m_0$ for a nominal parameter fraction of $k_0/m_0 = 150$, which lies in the range of human running. In addition to this stability loss, the parameter offset results in a undesired divergence of the steady state velocity in the range of $\dot{x}_{fix} \in [0.95, 1.15] \cdot \dot{x}_{des}$. In contrast thereto, simulations have shown, that parameter errors of nearly arbitrary size in both mass and stiffness direction are perfectly estimated with (8). Consequently, the $2^{nd}\,EV$ is kept zero independent on the model accuracy.

3.2. *Approximate BVP implementation*

The transfer from the numerical to an analytical approximate controller (including BVP speed estimation and return map based fix point search) entails two major problems. The approximate BVP does not give the correct actual velocity value and consequently the fix point search of the return map based
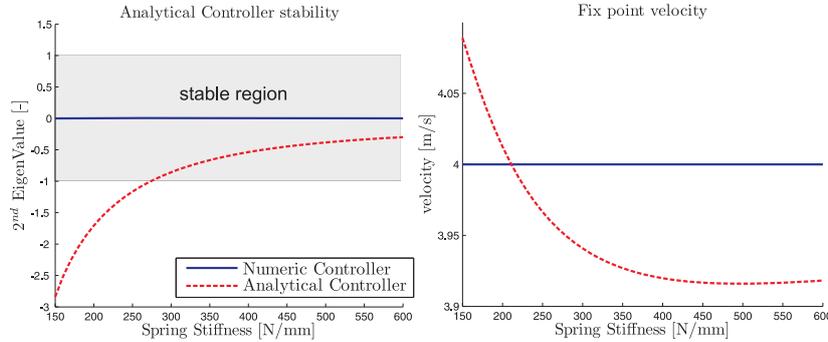
Figure 3 (a) Stability region of the analytical controller: While the numeric controller (solid blue) shows no parameter sensitivity, an increase in the stiffness improves the quality of the approximate solution and hence a decrease of the 2nd Eigenvalue below 1. (b) The velocity approximation in the BVP solution results in a (rather small) steady state error of the absolute forward velocity.

controller not only refers to approximate solutions but also works with inaccurate initial conditions.

The nature of the approximate solution shows better agreement of stiffer systems that have by default smaller radial spring deflections and angular swept. Therefore, even in case of a perfect model, closed loop stability largely depends on the spring stiffness. Fig. 3a depicts the expected dependency: the stiffer the system, the closer we are at the actual solution and consequently the smaller is the $2^{nd} EV$. Due to the approximation, steady state solutions have an offset to the actual demand velocity (fig. 3b).

4. Discussion

The presented paper demonstrates a way of expanding the established single-step dead beat controllers towards absolute velocity control. Solving a boundary value problem based on internal states and time measurement during ground contact allows for a model based estimation of the velocity. In addition, the numeric implementation is augmented with a realtime parameter estimation. The solution is a global dead beat behavior with a $2^{nd} EV$ always equal to zero, even in the presence of lager modeling errors.

Additionally, an approximate controller with the BVP speed estimation and return map fix point search based on analytical stance dynamics is presented. The closed loop stability and robustness is highly dependent of the accuracy of the analytical solution, and is better for stiffer systems. Overall, a numerical implementation of the BVP seems to be preferable. Various simulations with sensor noise and/or modeling errors showed large robustness of the closed-loop

system and great performance of the algorithm. The implemented problem is solvable in real time, such that the inaccuracies of the analytical approximate solution can be avoided.

Although a final experimental validation is still missing, the presented control strategies show great potential for a solution that can deal with model and sensor errors. This is an important step towards the application of biomechanically inspired control strategies in real robotic devices.

References

1. R. M. Alexander, "3 Uses for Springs in Legged Locomotion," *International Journal of Robotics Research,* vol. 9, April 1990.
2. R. Blickhan, "The spring-mass model for running and hopping," *Journal of Biomechanics,* vol. 22, 1989.
3. J. Schmitt and P. Holmes, "Mechanical models for insect locomotion: dynamics and stability in the horizontal plane I. Theory," *Biological Cybernetics,* vol. 83, 2000.
4. R. M. Ghigliazza, R. Altendorfer, P. Holmes, and D. Koditschek, "A simply stabilized running model," *Siam Review,* vol. 47, Sep 2005.
5. A. Seyfarth, H. Geyer, M. Gunther, and R. Blickhan, "A movement criterion for running," *Journal of Biomechanics,* vol. 35, May 2002.
6. H. Geyer, R. Blickhan, and A. Seyfarth, "Natural dynamics of spring-like running: emergence of selfstability," in *International Conference on Climbing and Walking Robots (CLAWAR)*, 2002
7. M. Ernst, H. Geyer, and R. Blickhan, "Spring-Legged Locomotion on Uneven Ground: A Control Approach to Keep the Running Speed Constant," in *International Conference on Climbing and Walking Robots (CLAWAR)*, 2009
8. A. Seyfarth and H. Geyer, "Natural control of spring-like running - optimized self-stabilization," in *International Conference on Climbing and Walking Robots (CLAWAR 2002)*, 2002
9. L.F. Shampine, J. Kierzenka, and M. W. Reichelt, "Solving boundary value problems for ordinary differential equations in Matlab with bvp4c," The MathWorks, Technical report, 2000.
10. W. J. Schwind and D. E. Koditschek, "Approximating the Stance Map of a 2-DOF Monoped Runner," *Journal of Nonlinear Science,* vol. 10, 2000.
11. H. Geyer, A. Seyfarth, and R. Blickhan, "Spring-mass running: simple approximate solution and application to gait stability," *Journal of Theoretical Biology,* vol. 232, 2005.