

Report: SLIP models for controlling articulated robotic legs

Stefan Schneyer and Fabian Kolb

I. INTRODUCTION

Bio inspired robotics and especially humanoid robots play an increasing role in society and science. Robotic solutions with a structural similarity to humans are especially suitable for tasks which are up to now mainly solved by humans. For moving in challenging terrain with higher velocities hopping is an interesting approach. For simplicity hopping can be simulated with one leg. SLIP models describe the spring-like leg behaviour of animals and humans in a very simple and abstract way. The model reduces the leg to a massless spring with a point mass attached to the top. This abstract SLIP model behaviour is then projected on the dynamics of an articulated leg. In a first stage a closed loop dynamics is generated based on the SLIP model. In the second stage the high order robotic leg tracks this closed loop dynamics and uses a feedback controller to adapt to deviations. The robot state is modelled as automaton. A continuous state captures the position of the robot in the 3D space, the hip angle and knee angle. A discrete state models the transition of flight to stance and stance to flight phase. A control structure is implemented for both the flight and the stance phase. For the flight phase a cascade control consisting of PI and PID controller is used. When touching the ground, the SLIP dynamics can be projected onto the CoG motion of the robotic leg. From there the joint actuator torques which are necessary to generate the operational space forces is calculated. Real robotic legs differ from the simplified SLIP model because the mass and inertia of the leg's segments influence the the robot's motion. Real robots suffer from the impact when the foot strikes the ground. This leads to a loss of kinetic energy which is incorporated in the SLIP model by an energy compensation. Different types of integration are implemented in order to study their effect on the controller stability. The project is implemented in Python. The RBDL library is used, which offers efficient rigid body dynamics algorithms. For visualization Meshup, a tool that was developed with RBDL is used.

II. SLIP MODEL

The SLIP model is an abstraction capturing the locomotion behavior of robots with light, compliant legs. In this model the body is reduced to a point mass m . The leg is represented by a massless spring with stiffness k_0 and a rest length l_0 . For this system the equations of motion are the following:

$$\begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} + \begin{bmatrix} 0 \\ mg \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \end{bmatrix} \quad (1)$$

where x and y are the coordinates of the center of mass (COM). F_x and F_y are the ground reaction forces (GRF). The GRFs are 0 during flight and depend on the length of the

spring l during compression and the landing angle α of the foot.

$$\begin{bmatrix} F_x \\ F_y \end{bmatrix} = k_0(l_0 - l) \begin{bmatrix} -\cos(\alpha) \\ \sin(\alpha) \end{bmatrix} \quad (2)$$

The SLIP model describes running as a series of apex heights y_i

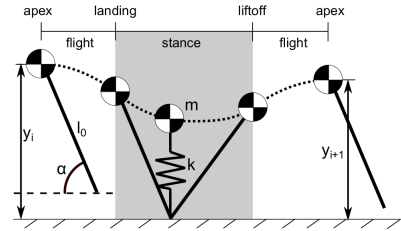


Fig. 1. flight and stance phases of SLIP model [1]

In Fig. 1 it becomes clear how the robot trajectory can be separated in flight and stance phases. The gait behavior is characterized by the forward speed and the apex height. The SLIP model is a conservative system without any friction or other mechanism to dissipate the dynamics, and thus, their phase space does not shrink over time. Therefore, the apex height and forwards speed are coupled through the system energy and the trajectory is fully characterized by the apex height. The transition between two consecutive apex heights is defined by the landing angle α :

$$f_{yi+1} = f_{yi}|(\alpha) \quad (3)$$

The landing angle is used to achieve deadbeat stability of the system [1] [2].

III. CONTROL

A. Higher order model: ATRIAS

The ATRIAS robot has lightweight carbon fiber legs. The robots total mass is concentrated mostly in the trunk. The legs are constructed as a four bar structure actuated by two motors through a leaf spring. The leaf springs and the four bar structure of each leg generate spring like forces in the legs. These spring like forces and the fact that the trunk outweigh the legs make the SMM abstraction especially useful for the ATRIAS robot.

The pair of motors allow changing the orientation and length of the leg (Fig. 2). The motors can generate linear actuator outputting forces F_i and rotatory torques τ_i .

In contrast to the SMM abstraction, the ATRIAS robot also needs to balance the trunk, generate swing motion in order to move the legs during flight phase and consider slip

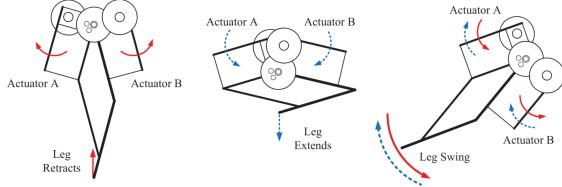


Fig. 2. ATRIAS leg actuation [3]

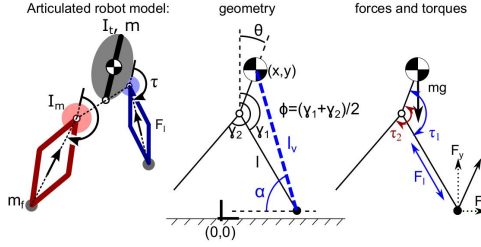


Fig. 3. ATRIAS model [1]

at the feed dissipating the system energy. The model of the ATRIAS robot is shown in Fig. 3. In vertical direction a GRF is generated corresponding to a spring damper system. In horizontal direction a sliding force proportional to the vertical GRF is generated opposing the motion of the robot. The dynamics of the ATRIAS robot can be described by the equation of motion:

$$M(q)\ddot{q} + h(q, \dot{q}) = S\tau + J^T F \quad (4)$$

where M is the mass matrix, q is the coordinate vector, h accounts for Coriolis, centrifugal and gravitational forces, the selection matrix S assigns control inputs τ to q . The coordinate vector consists of the global position and orientation of the trunk x, y, θ and the global orientation and length of each leg γ_i, l_i . The control vector τ consists of motor torques τ_i and linear force F_i of each leg [1] [3].

IV. MAPPING AND CONTROL DESIGN

For controlling the ATRIAS robot a few assumptions were applied. Since the mass of the feet is clearly smaller than the mass of the robot, the inertial effects of the feet can be neglected. This assumption does eliminate the dynamics of the leg during flight. The force for swinging the leg during flight is controlled separately and is not part of the paper. Besides that it is assumed that there is no slip during the whole stance phase [1].

A. Control Goals

Based on the SMM model the leg placement policy $\alpha(t)$ provides deadbeat tracking of the robot gait, which is fully determined by the apex height. The ATRIAS robot is able to achieve the same performance when tracking the landing

angle and generating GRFs through the center of mass (COM) corresponding to the spring forces in the SMM. To prevent unstable trunk pitch rates, the angular momentum $H = I_t\dot{\theta} + I_m\dot{\gamma}_1 I_m\dot{\gamma}_2$ around the COM should be stabilized at zero in average. This is achieved by keeping the trunk pitch constant at θ^* and mirroring the hip angle velocities $\dot{\gamma}_1 = -\dot{\gamma}_2$. Also, the translational Energy $E_{Trans} = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) + mgy$ should match the system energy E^* that parametrizes the SMM. Since the inertial effects of the feet are ignored H and E_{Trans} can only be regulated during stance. Besides that the robot feet should not be trailed back too far $\gamma_2 = 2\pi - \gamma_1$. The average hip angle is therefore $\frac{\gamma_1 + \gamma_2}{2} = \pi$ [1].

B. Flight

During flight the COM follows on the calculated trajectory and no external forces are applied on the robot. The control focuses on the angular motions of the trunk, hips and legs and minimizes their error to $\theta^*, \phi^*, \alpha^*$. An LQR controller is used to find an optimal trade off between these goals [1].

C. Stance

During the stance phase the trunk pitch θ , average hip angle ϕ and translational energy E_{Trans} are tracked. The GRFs are calculated from the SMM and also tracked. To achieve the control goals, a cascade control design is used [1].

1) *Outer loop*: Target GRFs are aimed such that they produce work reducing the system energy error between E_{Trans} and E^* . Since spring-leg behavior is assumed virtual spring forces are calculated to model the robot's GRFs [1].

2) *Inner loop*: Actuator commands are calculated from these virtual spring forces to produce the actual robot GRFs defined in the outer loop. To do so, the virtual spring forces are substituted in the equations of motion to solve for the feed-forward actuation $[\tau_{1,ff}, \tau_{2,ff}, F_{1,ff}]$. These produce the work opposing the estimated energy error from the outer loop. In a feedback loop torques are calculated driving the average hip angle ϕ and trunk pitch θ towards their stationary references π and θ^* .

$$\begin{aligned} \tau_{2,fb} &= K_{p\phi}(2\pi\gamma_1 - \gamma_2) - K_{d\phi}(\dot{\gamma}_1 + \dot{\gamma}_2) \\ \tau_{1,fb} &= -\tau_{2,fb} + K_{p\theta}(\theta_* - \theta)K_{d\theta}(\dot{\theta}) \end{aligned} \quad (5)$$

For the motor torques the feed-forward and feedback results are added $\tau_i = \tau_{i,ff} + \tau_{i,fb}$ and for the leg force just the feedforward actuation is used $F_i = F_{i,ff}$ [1].

V. RESULTS

Fig. 4 shows the performance of the control design for the ATRIAS robot on flat ground. C and E show that the controller is able to stabilize the trunk pitch θ and the average hip angle ϕ in two steps. These can still deviate from their goal angle in order to stabilize the angular momentum H . The gait behavior described by the apex height in A and forward velocity in B was achieved by matching the landing condition $\alpha^*(t)$. When the foot slips (red shaded in G) maximum torque is applied in order to oppose the motion of the foot. When the foot is stationary (gray shaded in G) the applied torques generate

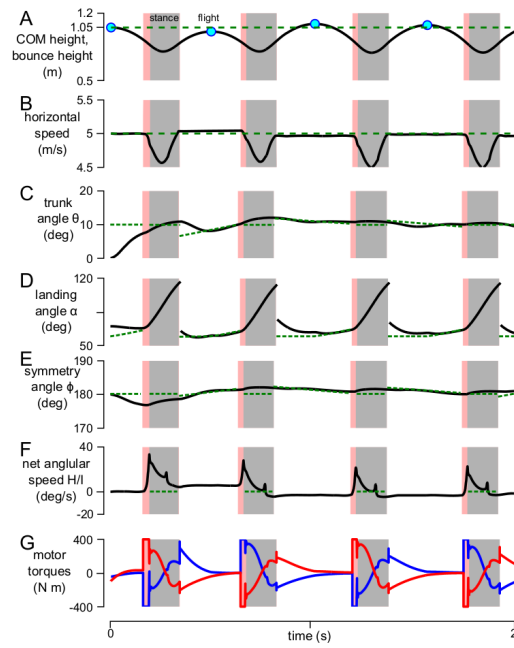


Fig. 4. First two seconds of running in flat terrain [1]

REFERENCES

- [1] A. Wu and H. Geyer, "Highly robust running of articulated bipeds in unobserved terrain," *IEEE International Conference on Intelligent Robots and Systems*, pp. 2558–2565, Sept 2014.
- [2] M. Hutter, C. D. Remy, M. A. Höpflinger, and R. Siegwart, "Slip running with an articulated robotic leg," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 4934–4939.
- [3] J. Grimes and J. Hurst, *THE DESIGN OF ATRIAS 1.0 A UNIQUE MONOPOD, HOPPING ROBOT*, 09 2012, pp. 548–554.

GRFs in order to regulate the system energy and to drive the angular momentum H towards zero [1].