Morphologically Self-Stable 3D Legged Robot using the Single Rigid Body Model for High Speed Running

Gabriel García[1], Robert Griffin[1], and Jerry Pratt[1]

Abstract— **Spring-Mass models are a classical approach for modeling running robots. Biomechanical studies find an incredible match between these models and the human body data while running, but most of the studies consists only on 2D data in the saggital plane, and very few of them deal with the 3D case. Additionally, a trunk for modeling the orientation of the body is sometimes considered (as the T-SLIP model), but his 3D analogous has not been explored yet. In this work, we present a 3D Legged Robot which is self-stable at high speed-running, i.e., there exists a limit cycle for the hybrid system consisting on stance and flight phase, and this limit cycle is stable after strides, suggesting that stability in this robot is consequence of the geometric shape of the robot with no feedback involved. This work shows the power of well-placed and tuned springs and dampers in mechanical systems and their effectiveness in the reduction of consumption of energy in dynamic motions of robots.**

I. INTRODUCTION

Running at high speeds in robots have been an interesting topic in robotics in the last thirty (?) years. MIT Legged Laboratory developed early models of biped robots which were able to walk and run. When running robot models became popular, one of the most studied approach was the Spring Loaded Inverted-Pendulum (SLIP), also known as the Spring-Mass model,because of its big connection with biomechanics: The SLIP model approximates very well the biped running in humans.

In [1] and [2], authors studied the biomechanics of walking. The most important result they obtain is that the angular momentum around the Center of Mass is constant. Because of this, a model mass can approximate well the angular momentum mechanics (Considering a body with no inertia.

In [3], [4] and [5], authors studied the SLIP model and the leg behaviour that makes the running at high speeds stable. An important finding they made was that a retraction rate in the swinging leg starting at the apex helps in the stability of the running and it acts as a time-based controller against perturbations in the groud height. Using the SLIP previous studies, Hurst. et al. designed and build ATRIAS ([6], [7]) a robot that emulates SLIP running in the saggital plane with successful results.

The inclusion of an orientation for considering the angular momentum is also studied in the literature. In 2D, this model is called T-SLIP (Trunk-SLIP). In [8] and [9], authors are focused in understanding how the direction of the forces affects the locomotion of the T-SLIP model from the biomechanical point of view. They study the called Virtual Pivot Point (VPP) or just Virtual Point (VP), how its position affects the system and they regulate the trunk orientation of the T-SLIP with the VP. In [10], authors also use the VPP to control the orientation of the trunk with a PD controller. This was enhanced later in [11] with a LQR for achieving disturbance rejection and with a feedback linerization for a variable spring for achieving CoM tracking. In [12], the author is focused on finding stable limit cycles for given parameters of the T-SLIP model using Stochastic Optimization for finding the locally most stable limit cycle in the eigenvalue sense after strides.

There is little to no literature in self-stable running in 3D with orientation. Without considering orientation, the SLIP model was extended to its 3D version in [13] where the swing leg has two degrees of freedom in spherical coordinates. They mention that a 3D-SLIP model with a spring pointing to the CoM results in a motion constrained to a plane in stance phase, and they showed that the system is unstable to perturbations in the normal direction of that plane. In consequence an additional control law is necessary and if no feedback is desired, then it is necessary another spring-damper component. In [14], authors find the swing leg angles that produces a deadbeat controller in local coordinates for the 3D-SLIP model.

The main contribution of this work is presented:

- Show that there exists a 3D Running Model considering orientation which is stable without feedback at high-speeds and it does not use gyroscopical effect.

Note that the 3D HexRunner developed at IHMC Robotics already shows that 3D self-stable running without feedback is possible, but one concern about this robot is that its legs are always spinning in one direction, so the gyroscopical effect is a possible reason of its stability.

Although the present model represents more a biped robot rather than a quadruped, but we prefer to keep the term legged for generality purposes (HexRunner is a legged robot).

II. MATHEMATICAL MODEL

In this section we are going to derive the equations of motion of the 3D FastRunner. This robot has the following properties:

- Massless legs
- 3-D Trunk with Inertia
- Point-feet only
- Symmetric left-right leg configuration.

*This work was supported by IHMC

[1]Authors are with The Florida Institute for Human and Machine Cognition, Pensacola, FL, USA {ggarcia, rgriffin, jpratt}@ihmc.us

- Two running phases: Stance and Flight
- Only one actuator provides power through a constant torque
- No actuation in other joints: Just springs and dampers

Regarding the first property, we assume that this work is the approximation of a real robot when ratio mass of the legs/mass of the trunk goes to zero. The second property generalizes the T-SLIP model to 3D. The third property fixes the Center of Pressure to a constant point. This assumption is almost universal in the running models. The fourth assumption allows us to simplify the stability analysis to only one stride. The fifth assumption ensembles that biomechanical property of running. The last two properties shows that the present model does not use feedback: Springs and dampers are going to stabilize the motion and the constant torque is only for compensating the energy loss due to the dampers.

A. Equations of Motion

The centroidal dynamics of any Multi-body System under a single point-foot contact evolves as:

$$\begin{bmatrix} \ddot{\mathbf{r}} = \frac{1}{m}\mathbf{f} + \mathbf{g} \\ \dot{\mathbf{L}} = (\mathbf{r}_p - \mathbf{r}) \times \mathbf{f} \end{bmatrix} \tag{1}$$

Where $\mathbf{r}$ is the position of the CoM, $\mathbf{f}$ is the vectorial ground reaction force, $\mathbf{r}_p$ is the position of the point-foot contact, m is the mass of the robot and $\mathbf{g}$ is the gravity vector.

When the system has only one link with mass (i.e. body, the legs are massless), the angular momentum $\mathbf{L}$ is reduced to

$$\mathbf{L} = \mathbf{I}\omega = \mathbf{R} I_B \mathbf{R}^T \omega \tag{2}$$

where I_B represents the Inertia tensor of the body in a local frame (constant), $\mathbf{I}$ represents the Inertia in the world coordinates, $\mathbf{R}$ is the rotation matrix from the local to the world coordinates of the body, and ω represents the angular velocity of the body. We parameterize the $\mathbf{R}$ matrix in euler angles. We require for this problem that the robot's heading direction does not point up, so the singularity in euler angles will never be reached.

$$\mathbf{R}(\Theta) = R_z(\Theta_1) R_y(\Theta_2) R_x(\Theta_3) \tag{3}$$

The dynamics of $\mathbf{R}$ hold:

$$\dot{\mathbf{R}} = [\omega]_\times \mathbf{R} \tag{4}$$

where $[\bullet]_\times$ denotes the matrix form of the cross product.

Taking time derivative of (3) and plugging (3) into (4) we obtain an nonlinear matrix $M(\Theta) \in \mathbb{R}^{3\times 3}$ which defines the kinematics of the Euler angles as:

$$\dot{\Theta} = M(\Theta)\omega \tag{5}$$

Note that $M(\Theta)$ is not well-defined when $\cos(\Theta_2) = 0$. From (2) we obtain

$$\omega = \mathbf{R}(\Theta) I_B^{-1} \mathbf{R}(\Theta)^T \mathbf{L} \tag{6}$$

If we combine (6) with (5) and (1), we obtain the dynamics of the SRBM in Euler angles:

$$\begin{bmatrix} \ddot{\mathbf{r}} = \frac{1}{m}\mathbf{f} + \mathbf{g} \\ \dot{\mathbf{L}} = (\mathbf{r}_p - \mathbf{r}) \times \mathbf{f} \\ \dot{\Theta} = M(\Theta)\mathbf{R}(\Theta) I_B^{-1} \mathbf{R}(\Theta)^T \mathbf{L} \end{bmatrix} \tag{7}$$

B. Structure and Dynamics

In this section we will define a particular explicit representation of FastRunner. In body coordinates (Considering the CoM position as 0, and the rotation as the identity matrix) we define a kinematic configuration for the virtual leg as a 3 joint, mass-less and inertia-less chain. Starting at the CoM position ($[0,0,0]^T$) we define a the position of the 'hip' H in any direction, whose coordinates are parameters. The point H is the base of the leg and it's coordinates are given by $\mathbf{r}_H$. We place the first joint in the y-axis as an abduction-adduction joint q_1. The second joint is placed in the x-axis (after q_1) as flexion-extension q_2. The third joint is in the hip-foot direction, defined as a prismatic joint q_3. Note that when the point H is placed at the CoM, this model is the 3D SLIP model, and when the H point is above the CoM and the joint q_1 is set to 0, we obtain the 2D T-SLIP model (2D SLIP enhanced with orientation).

We define two phases of locomotion for the 3D FastRunner as the classical "stance" and "flight" phase.

In order to obtain the forces applied to the ground for the robot we will define the position of the foot in local body coordinates as:

$$\mathbf{r}_{f_{\mathscr{B}}} = \mathbf{r}_H - R_y(-\mathrm{q}_1) R_x(\mathrm{q}_2) \mathbf{e_3} \mathrm{q}_3 \tag{8}$$

where $R_w(\theta)$ is the matrix corresponding to a rotation of θ rads around the w-axis, $\mathbf{e}_3 = [0,0,1]^T$ and $\mathbf{r}_H$ is the vector containing the shift of the Hip respect to the CoM (i.e. the H vector). Note also that respect to the world we have:

$$\mathbf{r}_{f_{\mathscr{B}}} = \mathbf{R}^T(\mathbf{r}_p - \mathbf{r}) \tag{9}$$

An important remark is that we can obtain an expression for the joints $\mathbf{q}$ as a function of the $\mathbf{x}$ variables by inverting (8) and replacing (9) on it.

$$\mathbf{q} = \mathbf{q}(\mathbf{x}) \tag{10}$$

Because the legs are massless, we can define now the forces applied to the ground in local coordinates as a function of the generalized torques τ_g of the joints q_i as:

$$\tau_g = \frac{\partial \mathbf{r}_{f_{\mathscr{B}}}^T}{\partial \mathbf{q}} \boldsymbol{f}_{\mathscr{B}} \tag{11}$$

$$\boldsymbol{f}_{\mathscr{B}} = \frac{\partial \mathbf{r}_{f_{\mathscr{B}}}^T}{\partial \mathbf{q}}^{-1} \tau_g \tag{12}$$

Where $\mathbf{q} = (q_1, q_2, q_3)$ is the vector containing the stacked joint positions. Note that $\left| \frac{\partial \mathbf{r}_{f_{\mathscr{B}}}^T}{\partial \mathbf{q}} \right| = -q_3^2 \cos(q_2)$, so (12) is only valid when $q_3 \cos(q_2) \neq 0$. In global coordinates, the reaction force applied to the robot by the ground is given by:

$$\mathbf{f} = -\mathbf{R}\boldsymbol{f}_{\mathscr{B}} \tag{13}$$

The first and the third joint only contain springs and dampers. The second joint contains a motor with damping, so τ_g is given by:

$$\tau_g = -K_p(\mathbf{q} - \mathbf{q}_{ss}) - K_d\dot{\mathbf{q}} + \tau_0\mathbf{e}_2 \tag{14}$$

Where $K_p = \text{diag}([k_{p1}, 0, k_{p3}])$ with k_{p1} and k_{p3} are constants, K_d is a diagonal positive-definite matrix, $\mathbf{e}_2 = [0,1,0]^T$, τ_0 is a scalar representing a constant torque, and $\mathbf{q}_{ss} = [q_{ss1}, 0, q_{ss3}]$ with q_{ss1} and q_{ss3} being the nominal joint positions when the springs are not compressed.

By plugging (13) with (7) together we obtain the stance phase dynamics. Note that $\boldsymbol{f}_{\mathscr{B}}$ is a function of the joints $\mathbf{q}$ and the torque τ_g. (8) is invertible as long as $q_3\cos(q_2) \neq 0$, so we can write the joints q as a function of the local foot position $\mathbf{r}_{f_{\mathscr{B}}}$, and this in turn can be written as a function of the world position and the body rotation $\mathbf{r}$ and $\mathbf{R}$ as in (9). In consequence, the dynamics of (7) in stance phase can be written directly as a function of its variables $\mathbf{x} = (\mathbf{r}, \mathbf{R}, \dot{\mathbf{r}}, \mathbf{L})$, so (7) can be written as:

$$\dot{\mathbf{x}} = \mathbf{f}_{\mathbf{St}}(\mathbf{x}) \tag{15}$$

In flight phase the dynamics are straightforward, since no forces act on the robot. (7) with $\mathbf{f} = \mathbf{0}$ can be written as:

$$\begin{bmatrix} \ddot{\mathbf{r}} = \mathbf{g} \\ \dot{\mathbf{L}} = \mathbf{0} \\ \dot{\Theta} = M(\Theta)\mathbf{R}(\Theta)I_B^{-1}\mathbf{R}(\Theta)^T\mathbf{L} \end{bmatrix} \tag{16}$$

or equivalently

$$\dot{\mathbf{x}} = \mathbf{f}_{\mathbf{Fl}}(\mathbf{x}) \tag{17}$$

In order to obtain the equations of motion of the joints respect to the body, we are going to use the fact that the legs are massless. The Euler-Lagrange equation vanishes (Due to no kinetic energy and the potential energy of the spring is casted directly into τ_g as showed in (14)) and generalized torque τ_g y remains, together with the joint motor, springs and dampings ():

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial L}{\partial \mathbf{q}} = \tau_g \tag{18}$$

$$0 = -K_p(\mathbf{q} - \mathbf{q}_{ss}) - K_d\dot{\mathbf{q}} + \tau_0\mathbf{e}_2 \tag{19}$$

The dynamics of the first and third joints are given by:

$$\dot{q}_i = -\frac{k_{pi}}{k_{di}}(q_i - q_{ssi}) \tag{20}$$

$$q_i(t) = q_{ssi} + (q_i(0) - q_{ssi})e^{-\lambda_i t} \tag{21}$$

for i=1,3 and with $\lambda_i = \frac{k_{pi}}{k_{di}}$. Now, the dynamics of the second joint is given by:

$$\dot{q}_2 = \frac{\tau_0}{k_{d2}} \tag{22}$$

$$q_2(t) = q_2(0) + \frac{\tau_0}{k_{d2}}t \tag{23}$$

The real equations of motion of the joints approaches (19) when the leg mass approaches to zero. Now, we are going to make 2 assumptions. Regarding (20), we are going to assume that the states remain close to their steady state values in stance phase, and that there is enough time in flight for the joints to be reset. So we have now directly:

$$q_i(t) = q_{ssi} \tag{24}$$

for i=1,3; i.e, the springs and dampers will *instantaneously* reset the position of the joints in the air to their steady-state values. This is done in order to take out this variables and simplify the Poincare Map analysis that we are going to do in Section III.

The second assumption is respect to (23). The value of $\frac{\tau_0}{k_{d2}}$ is a constant corresponding to the classical ω_r "swing retraction rate". We are going to assume that. This is done in order to obtain more generality: If this 3D FastRunner is used as an approximate model of a buildable robot, then the swing retraction rate can be changed to another value as a parameter. Otherwise in flight we should enforce $\omega_r = \frac{\tau_0}{k_{d2}}$.

For now, we are going to define the phase of the leg in flight phase as a linear function of time, i.e:

$$\dot{\phi}_{leg} = \omega_r \tag{25}$$

where ω_r is a constant and it is known as "swing retraction rate".

Note also that, compared to classical SLIP works, we are starting the dynamics of retraction rate when a transition from stance to flight starts instead of the classical apex event. This is done in order to keep consistency with an open-loop robot, where the apex event cannot be detected.

C. Domains, Guards and Reset Maps

We are going to define the domain of the stance phase, recalling the need of $q_3\cos(q_2) \neq 0$ for the solution of eq. (8) and for the inversion of (11). In particular, the operational space C_0 for the joint configuration in stance phase will be:

$$C_0 = \{\mathbf{q} \in \mathbb{R}^3 : q_1 \in (-\pi, \pi] \wedge q_2 \in (-\frac{\pi}{2}, \frac{\pi}{2}) \wedge q_3 > 0\} \tag{26}$$

In the stance phase, the ground reaction force has a non-negative vertical component. We are going to define the domain of the stance phase as:

$$D_0 = \{\mathbf{x} \in X : F_{vert}(\mathbf{x}) \geq 0 \wedge \mathbf{q}(\mathbf{x}) \in C_0\} \tag{27}$$

Where $\mathbf{q}(\mathbf{x})$ is the inversion of (8) when $\mathbf{r}_{f_{\mathscr{B}}} = \mathbf{R}^T(\mathbf{r}_p - \mathbf{r}) = \mathbf{r}_{f_{\mathscr{B}}}(\mathbf{x})$ and F_{vert} is the vertical component of the ground reaction force $\mathbf{f}(\mathbf{x})$ from (13), and it is given by:

$$F_{vert}(\mathbf{x}) = \mathbf{e}_3^T\mathbf{f}(\mathbf{x}) \tag{28}$$

The transition from stance to flight phase occurs when the vertical force becomes zero and its time derivative is negative. In this case, the contact is broken and the robot starts flying. The guard from stance to flight is given by:

$$G_{St\to Fl} = \{\mathbf{x} \in X : F_{vert}(\mathbf{x}) = 0 \wedge \dot{F}_{vert}(\mathbf{x}) < 0\} \tag{29}$$

When a switch from stance to flight occurs the state variables in the SRBM $(\mathbf{r},\mathbf{R},\dot{\mathbf{r}},\mathbf{L})$ don't change. For now, we are going to define a change in the phase leg ϕ_{leg} as:

$$\phi_{leg}^{+} = \phi_{leg}^{-} + \alpha_r \tag{30}$$

Where ϕ_{leg}^{+} is the initial phase leg in flight phase (phase leg after the transition) and ϕ_{leg}^{+} is the final phase leg in stance phase (phase leg before the transition). The parameter α_r corresponds to a constant representing a shift in the phase between legs. Compared to classical SLIP works where the parameter α_r represents a reset "angle", but (30) considers the correlation between legs that may be connected each other. The 2D "Elliptical Runner" or the 3D "HexRunner" have legs physically connected. This parameter is discussed later.

So, the reset map from stance to flight is given by:

$$R_{Fl\to St}(\mathbf{x}) = [\mathbf{r}^T,\mathbf{R}^T,\dot{\mathbf{r}}^T,\mathbf{L}^T,0,0,\phi_{leg}^{+}]^T \tag{31}$$

In the flight phase, we are going to make an assumption. We will be looking for natural running, where the point-foot is always placed under the CoM position. Note that, even though other solutions to high-speed running may exists, they have higher changes to be unstable. So, the domain is defined as follows:

$$D_1 = \{x \in X : \mathbf{e}_3^T \bar{\mathbf{r}}_{fFl}(\mathbf{x},\mathbf{q}) \geq 0\} \tag{32}$$

where $\mathbf{r}_{fFl}(\mathbf{x},\mathbf{q})$ represents the foot position in the space, given by:

$$\bar{\mathbf{r}}_{fFl}(\mathbf{x},\mathbf{q}) = \mathbf{r} + \mathbf{R}\mathbf{R}_{Mirr}\mathbf{r}_{f_{\mathscr{B}}} \tag{33}$$

$$\mathbf{r}_{fFl}(\mathbf{x},\mathbf{q}) = \mathbf{r} + \mathbf{R}\mathbf{R}_{Mirr}(\mathbf{r}_H - R_y(-\mathrm{q}_{ss1})R_x\big(\mathrm{q}_2(t)\big)\mathbf{e_3}\mathrm{q}_{ss3}) \tag{34}$$

where we have used (8) with the flight assumptions given in (24) and (25), and $\mathbf{R}_{Mirr} = \mathrm{diag}([-1,1,1])$ is a mirror matrix reflecting the leg in the x-axis in local coordinates (assuming that the robot is running along the y-axis).

The transition from flight to stance phase occurs when the z-component of the swing foot $\bar{\mathbf{r}}_{fFl}(\mathbf{x},\mathbf{q})$ in world coordinates is zero and its derivative is negative. The guard from flight to stance is given by:

$$G_{Fl\to St} = \{\mathbf{x} \in X : \mathbf{e}_3^T \bar{\mathbf{r}}_{fFl}(\mathbf{x},\mathbf{q}) = 0 \wedge \mathbf{e}_3^T \dot{\bar{\mathbf{r}}}_{fFl}(\mathbf{x},\mathbf{q}) < 0\} \tag{35}$$

The reset map from flight to stance is defined using a reflection on the x-axis.

$$R_{Fl\to St}(\mathbf{x}) = \mathbf{R}_{MirrAug}[\mathbf{r}^T,\Theta^T,\dot{\mathbf{r}}^T,\mathbf{L}^T,0,0,\phi_{leg}^{+}]^T \tag{36}$$

Where $\mathbf{R}_{Aug}$ is matrix encoding the mirror augmentations applied to the state variables for a reflection in the x-axis.

D. 3D FastRunner as a Hybrid System

We can define the 3D FastRunner as a Hybrid Dynamical System (Notation taken from [15]):

$$\mathscr{H} = (Q,X,f,In,D,E,G,R) \tag{37}$$

with:

$$\begin{aligned}
Q &= \{q : q \in \{0,1\}\} \\
X &= \mathbb{R}^{15} \\
f(q,x) &= \begin{cases} f_{St}(x) & \text{if } q=0 \\ f_{Fl}(x) & \text{if } q=1 \end{cases} \\
I_0 &= \{0,x_0\} \\
D &= \begin{cases} D_0 & \text{if } q=0 \\ D_1 & \text{if } q=1 \end{cases} \\
E &= \{[0,1],[1,0]\} \\
G(E) &= \begin{cases} G_{St\to Fl} & \text{if } E=[0,1] \\ G_{Fl\to St} & \text{if } E=[1,0] \end{cases} \\
R(E,x) &= \begin{cases} R_{St\to Fl}(x) & \text{if } E=[0,1] \\ R_{Fl\to St}(x) & \text{if } E=[1,0] \end{cases}
\end{aligned} \tag{38}$$

III. POINCARE MAP AND OPTIMIZATION SET-UP

A. Poincare Map Section

We define a Poincare section S at the beginning of the stance phase of the hybrid system $\mathscr{H}$. $\mathscr{H}$ holds the following properties, which will allows us to reduce the dimension of 14:

- Springs uncompressed.
- (10) holds for $\mathbf{r}_P = 0$.

The first property comes from (24), where we assumed that the springs return to their uncompressed positions during flight phase so $q_1 = q_{ss1} \wedge q_3 = q_{ss3}$ and we can take out this components of the Poincare section. The second property implies that the local position of foot respect to the CoM is a function of only q_2, so we can write explicitly:

$$\mathbf{r} = \mathbf{r}_p - \mathbf{R}(\mathbf{r}_H - R_y(-\mathrm{q}_{ss1})R_x(\mathrm{q}_2)\mathbf{e_3}\mathrm{q}_{ss3}) \tag{39}$$

Because in the Poincare Section S the initial position $\mathbf{r}$ is always a function of the rotation matrix $\mathbf{R}$ and the joint position q_2, we can take out the position $\mathbf{r}$. The Poincare Section S now has 10 variables $(q_2,\Theta,\dot{\mathbf{r}},\mathbf{L})$ instead of the original 15, but there is still one component that we can take out.

Any robot that does not take information from the world cannot recover from orientation changes in the heading direction, i.e, rotations about any z-axis.

In that sense, if a limit cycle exists, then for any yaw rotation of the robot there exists initial conditions that produce a limit cycle (more precisely, the velocity and angular momentum in the Poincare Map must be rotated). So, we can enforce directly $\Theta_1 = 0$. If we don't want to make this reduction, then we will always have an eigenvalue equal to 1 in the Poincare Map, corresponding to rotations on the z axis.

The final Poincare map S has now 9 variables $(\mathrm{q}_2, \Theta_{2,3}, \dot{\mathbf{r}}, \mathbf{L})$, and we are going to optimize stability in the Floquet multiplier sense.

Let $P(\mathbf{x}_r) : S \to S$ be the Poincare map of the hybrid system $\mathscr{H}$ at a point $\mathbf{x}_r \in S$

$$\mathbf{x}_{r_{k+1}} = P(\mathbf{x}_{r_k}) \tag{40}$$

A fixed point $\mathbf{x}_\mathbf{r}^*$ under the application P (i.e. $\mathbf{x}_\mathbf{r}^*$ holds $\mathbf{x}_\mathbf{r}^* = \mathbf{P}(\mathbf{x}_\mathbf{r}^*)$ represents the initial conditions in stance phase of a limit cycle. The linearization of $P(x)$ around $\mathbf{x}_{r_{k0}}$ is given by:

$$\mathbf{x}_{r_{k+1}} = P(\mathbf{x}_{r_{k0}}) + \frac{\partial P}{\partial \mathbf{x}_{r_k}^T}(\mathbf{x}_{r_k} - \mathbf{x}_{r_{k0}}) \tag{41}$$

When the eigenvalues of $\frac{\partial P}{\partial \mathbf{x}_{r_k}^T}$ are less than one, then the limit cycle is locally stable (given that (41) is a discrete map). In general the Poincare Map is going to depend of parameters p of the Hybrid System $\mathscr{H}$, so we are going to write $\mathbf{x}_{r_{k+1}} = P(\mathbf{x}_{r_k}, p)$.

B. Poincare Map Optimization

Let λ_i be an eigenvalue of $\frac{\partial P}{\partial \mathbf{x}_{r_k}^T}$. We are going to perform the following optimization:

$$\mathbb{P}_1 : \min_{p, \mathbf{x}_{r_{k0}}} \max_i |\lambda_i|, \ \textbf{s.t.} \tag{42}$$

$$\mathbf{x}_{r_{k0}} = P(\mathbf{x}_{r_{k0}}, p)$$

Where $\mathbf{x}_{r_{k0}} \in S$ and p is the set of parameters of the Hybrid System, in this particular case:

$$p = [\mathbf{r}_H; k_{p1}; k_{p3}; q_{ss1}; \tau_0; \mathrm{diag}\, K_d; \mathrm{diag}\, I_B; \omega_r; \alpha_r] \tag{43}$$

where the symbol ";" produces a vertical stacking of the components of p and the operator $\mathrm{diag}(M) : \mathbb{R}^{3\times 3} \to \mathbb{R}^3$ returns a vector with the diagonal components of the square matrix M. Note that we are excluding the mass and the leg length of the robot from the parameters in p. This is done because of dimensional analysis. The values of mass, gravity and leg length already contains the three magnitudes present in the dynamics of the robot (M, L and T), so we can safely fix them and then scale all the magnitudes properly as we desire.

Let q_{2TD} be the value of the joint q_2 at the touchdown after the flight phase in a limit cycle. The floquet multipliers of the poincare map $P(\mathbf{x}_{rk})$ does not depend on the higher terms of the Taylor expansion of $q_2(t)$ around the touchdown time t_{TD} more than the first order term.

$$q_2(t) \approx q_{2TD} + \frac{\partial q_2}{\partial t}(t - t_{TD}) + \frac{1}{2}\frac{\partial^2 q_2}{\partial t^2}(t - t_{TD})^2 + O((t - t_{TD})^3) \tag{44}$$

In virtue to this, optimization $\mathbb{P}_1$ leaves the coefficients of the Taylor expansion of $q_2(t)$ free to choose. We can solve the following optimization, which avoids the eigenvalues to be not very sensible to disturbance after a stride:

$$\min_{p, \alpha_\mathbf{r} r_{k0}} \max_i \left\| \frac{\partial \lambda_i}{\partial \mathbf{x}^T} \right\|, \ \textbf{s.t.} \tag{45}$$

$$\mathbf{x}_{r_{k0}} = P(\mathbf{x}_{r_{k0}}, p)$$

Given that λ_i depends only on the jacobian of the Poincare Map, $\frac{\partial \lambda_i}{\partial \mathbf{x}^T}$ depends on the Hessian of the Poincare Map. In consequence, the objective function of (45) depends only on the quadratic term of q_2. The polynomial coefficients of q_2 will affect their respective n-derivative of the Poincare Map, so if we want to include them in an optimization, then we need to approximate higher order derivatives of $P(\mathbf{x}_{rk})$. The number of variables of these derivatives grows exponentially with the order, so this scales bad with the number of variables. Instead, we will use an optimization with an heuristic objective function for finding good coefficients for the polynomial swing retraction angle $q_2(t)$.

$$\mathbb{P}_{2pre} : \min_{\alpha_\mathbf{r}} \max_{x_{rk} \in R} \|P(x_{rk0}^*) + A(x_{rk} - x_{rk0}^*) - P(x_{rk}, p)\|_P \tag{46}$$

$$\mathbf{x}_{r_{k0}} = P(\mathbf{x}_{r_{k0}}, p)$$

where R is a region of interest (around the fixed point $\mathbf{x}_{rk0}^*$), the matrix $A = \frac{\partial P}{\partial \mathbf{x}^T}\Big|_{\mathbf{x}=\mathbf{x}_{rk0}^*}$ and $\| \bullet \|_P$ represents the P-norm of a vector with P some matrix. A very small region R is going to be almost unaffected by the higher order terms of $q_2(t)$, and a very large region will be difficult or impossible to stabilize with only retraction rate. The operator max can be changed to an integral over a region of interest and for simplicity we are going to chose random points near the fixed point, and we are going to weight them.

$$\mathbb{P}_2 : \min_{\alpha_\mathbf{r}} \frac{1}{n} \sum_{x_{rk} \in R} \|P(x_{rk0}^*) + A(x_{rk} - x_{rk0}^*) - P(x_{rk}, p)\|_P \tag{47}$$

In particular, we are going to choose the region R as a "hypercube" around the euler angles (can be changed depending on what do we want the robot to be robust to) of side equal to 0.08 and the matrix P as a quadratic region around the fixed point, defined by the discrete Lyapunov Equation, which approximates locally the Region of Attraction of the fixed point on the discrete map.

$$A^T P A - P = -Q \tag{48}$$

with Q being any positive-definite matrix. In this case it will be only a diagonal with ones.

IV. RESULTS

The mass, the leg length and gravity are defined as follows:

TABLE I

FIXED VALUES IN OPTIMIZATION

Parameter	Value
Mass	3.7664
Leg length	0.7103
Gravity	9.81

Because we are letting the optimization $\mathbb{P}_1$ find the robot configuration and parameters over the whole possible set, and because the optimization is only trying to find the most stable

limit cycle without any other objective, we obtain interesting and somewhat strange results. The first observation is that the optimization tries to increase its inertia to high values. A possible explanation for that is that high inertia values produce less changes in the angular acceleration due to forces on the robot. Because of this, we bound each component of the inertia matrix using a maximum radius of gyration.

TABLE II

VALUES OBTAINED THROUGH THE OPTIMIZATION

Parameter	Value Opt. 1	Value Opt. 2
CoM-Hip Shift	[0.708; -0.292; 0.202]	[0.356; -0.238; 0.149]
Leg Spring	6134.9	1550.6
Azimuthal Spring	65.691	51.654
Azimuthal Angle	-0.269	0.114
Constant torque	-37.28	-20.36
Azimuthal Damper	8.22	2.675
Damping in q_2	6.783	4.954
Leg Damper	296.48	243.57
Inertia Diagonal	[0.3; 0.446; 0.422]	[0.847; 0.744; 0.055]
Retraction Rate	-1.134	-5.355
Touchdown Angle	0.483	0.046
Max. Eigenvalue	0.1172	0.2157

After this change we obtain the values shown in Table II in the column Value Opt.1. We observe a second interesting result: the optimization was trying to increase the separation of the legs for obtaining small eigenvalues. At some point the leg separation was about two times the leg length, as it is shown in the first component of the CoM-Hip Shift in Table II in the column Value Opt.1 . A possible explanation for that may be related with the support region theory for walking robots: A big leg separation implies a big support region for stability. We can bound the distance between legs to maximum value, because a big leg separation may be topologically problematic for a robot this second optimization is shwon in the column Value Opt. 2. It is important to note that, given that this is nonlinear optimization, values shown in Table II are just local minima for the maximum eigenvalue of the Poincare Map of the strides.

A last interesting result is that in the saggital plane, this model behaves similar to its analogous 2D T-SLIP model: In this plane, the force consequence of the leg spring points always to a point above and behind the CoM (similar to the VPP studies), but in the frontal plane, the force tries to stay vertically instead of pointing strictly to a point above and close to the CoM.

In Fig. 1 we can see energy evolving in the system in the limit cycle. In stance phase the actuator pumps energy to the system, while in flight phase, this energy is mostly dissipated due to the damping and decompression of the springs. It is important to note that the optimization also tries to be as less as it is possible in the air, so the flight time ($\approx 0.01s$) is small compared to the Stance time ($\approx 0.4s$).

In Fig. 2 we can see a representation of the 3D Fastrunner with the stance leg plotted.

In Fig. 3 we can see the visualization of the Swing Leg Trajectory in Body Coordinates. The most impressive result

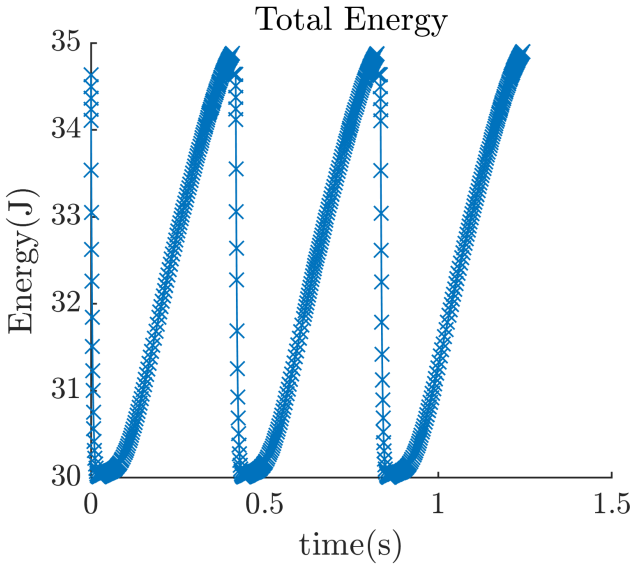

Fig. 1. Energy plot of the system

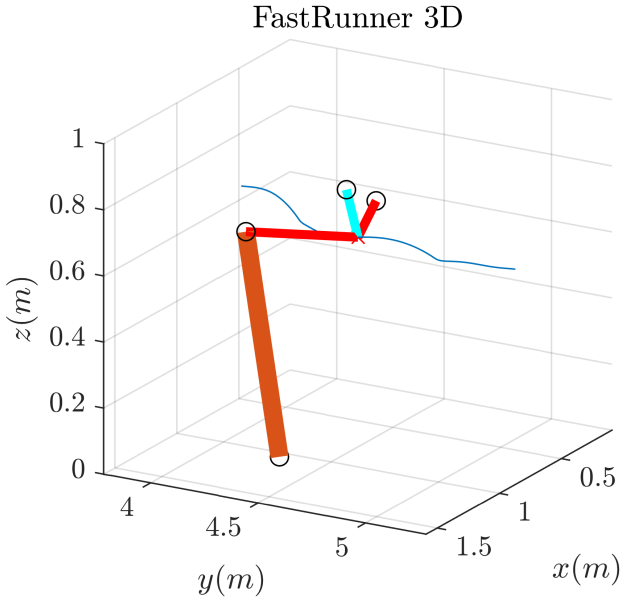

Fig. 2. 3D FastRunner visualization

here is that a locally, most stable configuration for running has a shape which pretty much represents the 3D Hexrunner built at IHMC.

Respect to $\mathbb{P}_2$ we can see that, the improvement in the square root of the objective function is near 25% (from $9.731 \cdot 10^{-4}$ to $7.147 \cdot 10^{-4}$). Additional testings of $\mathbb{P}_2$ on a 2D T-SLIP model also shows a maximum improvement near to 25%. This shows that swing retraction taylor expansion is not as important in the 3D model with Trunk as it is in the 2D-SLIP, where $\mathbb{P}_2$ is able to find the taylor approximation of the time-based deadbeat controller around the touchdown time. In any case $\mathbb{P}_2$ helps when the designer wants the robot to be more robust to a given desired disturbance. For smaller regions of interest $\mathbb{P}_2$ is less useful, but bigger regions may

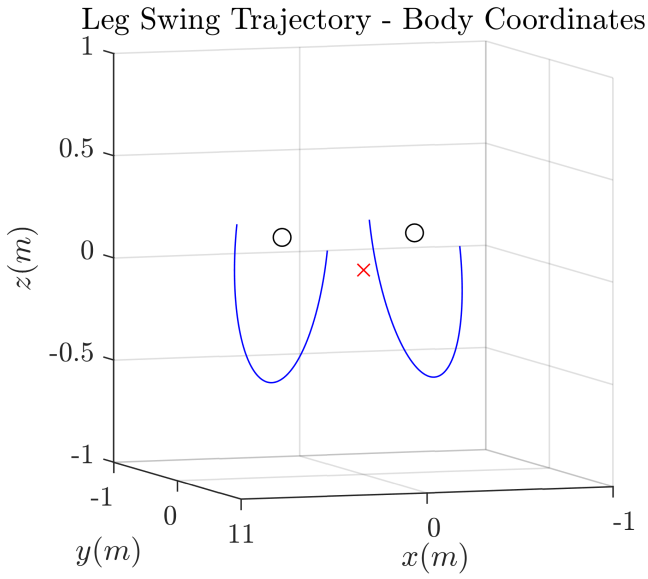

Fig. 3. FastRunner 3D Swing Leg Trajectory in Body Coordinates

include points that cannot be stabilized with retraction rate. An important fact in this application is that coefficients of the cubic term (fourth coefficient of Taylor expansion) and higher are no longer useful for the optimization because the time is close to the touchdown time, so we can just consider the quadratic term (third coefficient of Taylor expansion), i.e. the angular acceleration of the leg.

TABLE III
RETRACTION RATE COEFFICIENTS IN OPTIMIZATION

Parameter	Value
Third coefficient	-0.0436
Fourth coefficient	2863.84

V. CONCLUSIONS

By finding parameters and initial conditions showed in Table II in the last section, we conclude that there exists a 3D model that generalizes the SLIP model with Trunk to 3D under the assumptions written at the beginning of Section II. The massless legs assumptions may be a strong one, but the dynamics of the robot are almost the same when ratio between the mass of the legs and the trunk is small. Because there is not a component in the robot which is spinning in a constant (nor near constant) direction, we can affirm that there is not gyroscopical effect, and in consequence, running with no feedback is possible without gyroscopes. Also, test in the optimization $\mathbb{P}_2$ shows that the angular velocity at the touchdown time is an important factor for stabilization and it is more important the shape of the leg swing trajectory (which in this case is a circle with a variable inclination determined by the optimization) rather than its acceleration or higher linearization terms around the touchdown time. Although feedback will always outperform passive, semi-passive, feed-forward or time based systems (as long as the measures have low noises) this work shows the power of springs and dampers in the stabilization of a complicated hybrid system like a running robot, and this exemplifies a way in which optimization can be used in order to obtain parameters like inertia, angles or legs in the mechanical design of a robot for performing an specific task like high stability or by just changing the objetive, we can obtain stable running with low energy consumption.

REFERENCES

[1] M. Popovic, A. Hofmann, and H. Herr, "Angular momentum regulation during human walking: biomechanics and control," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 3, 2004, pp. 2405–2411 Vol.3.

[2] H. Herr and M. Popovic, "Angular momentum in human walking," *Journal of Experimental Biology*, vol. 211, no. 4, pp. 467–481, 2008. [Online]. Available: https://jeb.biologists.org/content/211/4/467

[3] A. Seyfarth, H. Geyer, and H. Herr, "Swing-leg retraction: a simple control model for stable running," *The Journal of experimental biology*, vol. 206, pp. 2547–2555, 09 2003.

[4] H. Geyer, A. Seyfarth, and R. Blickhan, "Compliant leg behavior explains basic dynamics of walking and running," *Proceedings. Biological sciences / The Royal Society*, vol. 273, pp. 2861–2867, 11 2006.

[5] W. C. Martin, A. Wu, and H. Geyer, "Robust spring mass model running for a physical bipedal robot," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 6307–6312.

[6] D. Renjewski, A. Spröwitz, A. Peekema, M. Jones, and J. Hurst, "Exciting engineered passive dynamics in a bipedal robot," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1244–1251, 2015.

[7] C. Hubicki, J. Grimes, M. Jones, D. Renjewski, A. Spröwitz, A. Abate, and J. Hurst, "Atrias: Design and validation of a tether-free 3d-capable spring-mass bipedal robot," *The International Journal of Robotics Research*, vol. 35, no. 12, pp. 1497–1521, 2016.

[8] Ö. Drama and A. Badri-Spröwitz, "Trunk pitch oscillations for energy trade-offs in bipedal running birds and robots," *Bioinspiration & Biomimetics*, vol. 15, no. 3, p. 036013, Mar 2020.

[9] O. Drama and A. Badri-Spröwitz, "Trunk pitch oscillations for joint load redistribution in humans and humanoid robots," in *Proceedings International Conference on Humanoid Robots*, Sep. 2019. [Online]. Available: http://arxiv.org/abs/1909.03687

[10] J. Lee, M. Vu, and Y. Oh, "A control method for bipedal trunk spring loaded inverted pendulum model," 05 2017.

[11] M. N. Vu, J. Lee, and Y. Oh, "Control strategy for stabilization of the biped trunk-slip walking model," in *2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 2017, pp. 1–6.

[12] M. van den Broek, *Fast Self-Stable Planar Bipedal Running*. TUDelft Thesis Repository - Master Thesis, 2019. [Online]. Available: http://resolver.tudelft.nl/uuid:3b31bc33-1f67-4fda-85db-5dace4835bcf

[13] J. E. Seipel and P. Holmes, "Running in three dimensions: Analysis of a point-mass sprung-leg model," *The International Journal of Robotics Research*, vol. 24, pp. 657 – 674, 2005.

[14] A. Wu and H. Geyer, "The 3-d spring–mass model reveals a time-based deadbeat control for highly robust running and steering in uncertain environments," *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1114–1124, 2013.

[15] J. Lygeros, *Lecture Notes on Hybrid Systems*. University of Patras, 2004. [Online]. Available: https://people.eecs.berkeley.edu/ sastry/ee291e/lygeros.pdf