CLC _____ Number _____

UDC _____ Available for reference ☐Yes ☐No

Undergraduate Thesis

Thesis Title: **Model-based Control of SOPHIE with Dynamic Overconstrained Locomotion**

Student Name: **Wang Baiyue**

Student ID: **11711217**

Department: **Department of Mechanical and Energy Engineering**

Program: **Robotics Engineering**

Thesis Advisor: **Song Chaoyang**

Date: March 10, 2021

COMMITMENT OF HONESTY

1. I solemnly promise that the paper presented comes from my independent research work under my supervisor's supervision. All statistics and images are real and reliable.
2. Except for the annotated reference, the paper contents no other published work or achievement by person or group. All people making important contributions to the study of the paper have been indicated clearly in the paper.
3. I promise that I did not plagiarize other people's research achievement or forge related data in the process of designing topic and research content.
4. If there is violation of any intellectual property right, I will take legal responsibility myself.

Signature:

Date:

Model-based Control of SOPHIE with Dynamic Overconstrained Locomotion

Wang Baiyue

(Department of Mechanical and Energy Engineering Instructor: Song Chaoyang)

[ABSTRACT]: Legged robots with 2 DOF per leg with parallel actuation were usually designed with planar legged locomotion, and their stable locomotion were also in a sagittal plane when they are not turning. Those 2 DOF leg that realized 3D locomotion, however, lacks jumping agility. Our goal in designing SOPHIE lies in reducing the Degree of Freedom (DOF) of the whole robot and focusing on investigating trade-offs between the ability of locomotion in frontal plane and in sagittal plane. This work tired to expand the 2 DOF locomotion to 3 dimensional by designing an overconstrained leg using a spatial linkage, who generates a customizable spatial surface, giving possibilities to realizing both vertical jumping and 3D walking at the same time. The expanded workspace of that leg is exploited to achieve dynamic movements: hopping in the sagittal plane and walking in 3D space, by using different controllers. The sagittal hopping controller was proved to be stable in theory in the case of parallel legs, and here we extend the controller to overconstrained leg, so is its verification in the simulation. The biped locomotion is derived in theory using Hybrid Zero Dynamics (HZD), but the existence of solutions is not yet proved. Such overconstrained leg design is therefore expected to be applicable to a real robot, leveraging the potential of generating multiple modes of movement from only 2 actuators.

[Keywords]: Overconstrained Locomotion, Dynamic Control

目录

1. Introduction

Previous legged locomotion merely explored beyond limits from their workspace. Specifically, a 2 degrees of freedom (DOF) leg with parallel actuation[1-4] never tired to expand their motion from a planar surface to a 3 dimensional space, which is left only to a leg with 3 or more DOF[5, 6], or a 2 DOF leg that does not allow vertical jumping[7]. At the same time, leg design had always been conservative according to the limitation of number of DOF. There were design principles concerning energy and vertical abilities[3, 4], and mechanism explorations that enhance vertical jumping[8, 9], but there were no exploration on the 3 dimensional movement as well as vertical jump ability that could be possibly generated by a 2 DOF leg. Our question therefore became: is it possible for a 2 DOF leg to achieve multi-directional movement in 3D space while also achieves agile vertical motions?

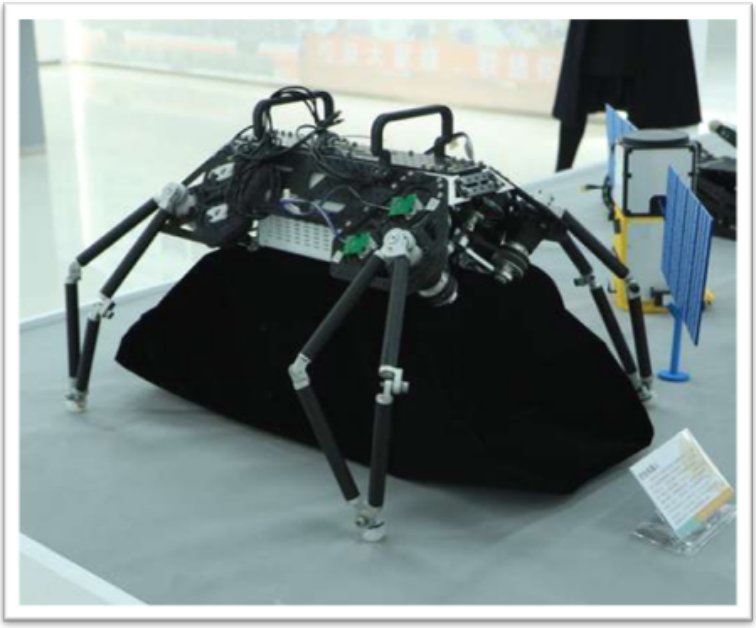
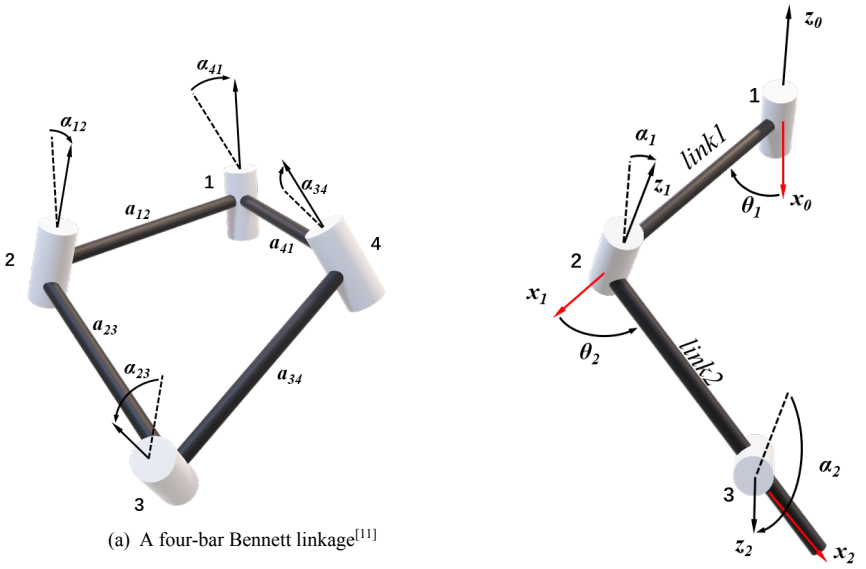


Fig. 1: A conceptual version of SOPHIE that showcase its leg mechanism

Then we came to our design of SUSTech Lobster Robot for Learning in Amphibian Environment (SOPHIE, Fig. 1). This quadruped robot is aimed to realize amphibian working scenario with omni-direction locomotion, while using leg mechanism with reduced DOF. Reducing one DOF per leg means reducing four DOF in total, relieving weight, cost and computational burden. There were some pre-existed quadruped with 2 DOF leg, such as[2, 10], but their legs are all parallel, making it almost impossible to realize lateral locomotion.

In this work, a leg with only 2 DOF who achieves 3 dimensional workspace (Fig. 2(b)) is proposed by the inspiration of spatial linkages[11, 12] (Fig. 2(a)). The overconstrained leg was tested on a monopod robot in simulation and a biped robot in theory. Both sagittal hopping and biped locomotion were explored with specific control method, and the results of our analysis and simulations showed that such leg design is capable of different modes of motion.



(a) A four-bar Bennett linkage[11]

(b) 2 DOF overconstrained leg

Fig. 2: From inspiration to our design: Bennett's linkage and Overconstrained Leg

1.1 Related Work

A leg that has 2 dimensional plane movement had been successfully developed and analysed. Through actuator selection, leg mechanism, embedded elastic element, such leg could exhibit dynamical locomotion[1, 2], energy conservation[3], superior vertical agility[4, 8, 9]. The actuators from those leg mechanism were all parallel, which is the key reason why their motion was limited in a sagittal plane. However, if we add a twist angle between two joints of the leg, like what had been done in some spatial linkages[11, 12], a 3 dimensional plane could be generated at the end-effector where the foot touches the ground. As an example, the overconstrained leg mechanism is shown in Fig. 2.

The problem of utilizing this mechanism lies in control. Although the overconstrained leg design generate a spatial surface of workspace, a special controller is needed to exploit the workspace for a 3 dimensional working scenario. Previous research focused on the kinematics[12], but dynamics is rarely seen, nor is its application in a legged robot. This work is going to derive the dynamics of the overconstrained legged locomotion using possible solutions, and then apply the control method to a monopod in simulation, and a biped robot in theory.

There were some fully developed control methods for legged robot. The first well-known heuristic controller[5] and its analysis[13] elaborated to us the stability of that control. We think that a controller that combine different templates[10] is suitable to the overconstrained legged hopping locomotion, because the workspace of overconstrained leg contains a large area in the sagittal plane. To control the one-legged locomotion, we chose a SLIP model with fore-aft speed control[14-16] to demonstrate a possible hopping solution. Furthermore, we also tried out a biped locomotion solution with inspiration from[7, 17, 18], where biped robots with 2 DOF leg were discussed.

1.2 Organization

This paper is organized as followed: in Section. 2 the mechanism of the overconstrained leg is demonstrated, so as the kinematics and dynamics that is used in the later content. Section. 3 explains our chosen template for forward-backward locomotion and analysed the stability of the template theoretically. Section. 4 explains a possible solution for a biped overconstrained robot to walk stability. The sagittal jumping was ran in simulation and is demonstrated in Section. 5 to strengthen the possibility of its monopodial, and further quadrupedal locomotion in real life. This paper will be discussed and future work will be included in Section. 6.

2. Overconstrained Leg Design

In this section, the previous design of leg mechanism will be reviewed and analysed, and their common design limitations is pointed out from our observation. A generalization of their limitations then leads to our proposed solution: a novel overconstrained leg design. A thorough explanation will be given, followed by its analytical derivation of the kinematics and dynamics that are vital to its application in later sections. The overconstrained leg mechanism pushes forward the mechanical dimensionality that could be possibly reached by only two DOF, revealing a new genre in the field of legged locomotion.

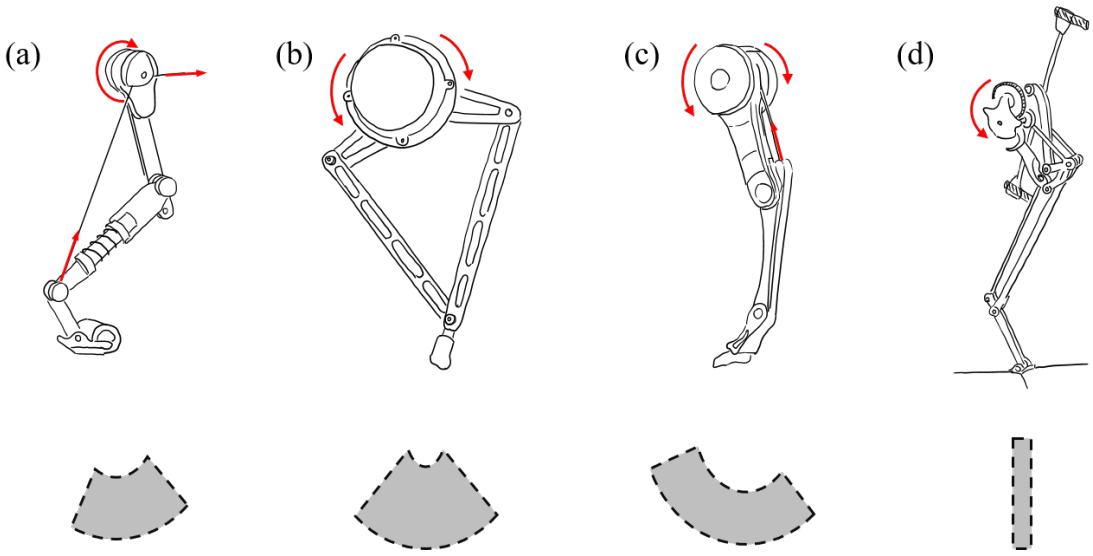


Fig. 3: Some previous leg designs and their planar workspace (below). (a) Leg design of Cheetah-Cub[1] with 2 DOF, one is pulling the string to actuate the shank, and the other one rotates the thigh. (b) Parallelogram structure of the 2 DOF leg from Ghost Minitaur[3], the 2 DOF are co-axis and fixed at the hip. (c) MIT Cheetah[6] with 2 DOF and one is driving the shank through a string, similar to (a). (d) Salto[8] with only one DOF, linkages guide the rotary output to an almost linear workspace.

2.1 Previous leg designs with 2 or less DOF

As a good comparison, leg designs with 2 or less than 2 DOF is considered in our review, because a leg with 3 or more DOF can easily reach their 3-D workspace thanks to their extra DOF. But our concerns focus on the cases when the DOF of a leg is restricted and therefore does not

have a 3-D workspace. Apparently, previous leg designs with 2 or less DOF are limited in the 2-D workspace.

The first three leg designs in Fig. 3 (a), (b), and (c) have 2 DOF. Cheetah-Cub[1] and MIT Cheetah[6] both used one rotary actuation to drive the thigh and another string actuation to drive the shank. Both of them showed great agility and robustness in dynamic walking. Minitaur from Ghost[3] expands its workspace through a parallelogram structure that shifts the two actuators to co-axis, and minimizes leg weight. The forth leg design of Salto[8] has only one actuation for leg locomotion, and complex links design guides the rotary actuation to a almost linear workspace. All of the four designs lift the actuator to above their hips in order to minimize leg weights and generate agile movements.

However, these legs with 2 or less DOF never expand their workspace to beyond 2 dimensional. As illustrated in Fig. 3, their workspace are either a sector shape or a straight line. Such workspace does not guarantee the control of the robot's locomotion outside the sagittal plane. Therefore, we will introduce our leg design that could expand its workspace to a 3 dimensional surface with only two DOF.

2.2 Workspace and Kinematics of Overconstrained Leg

The overconstrained design concept is introduced in Section 1. The Bennett's linkage invented a long time ago[11] is the inspiration of our design. In this work, we choose a half of the 4-bar Bennett's linkage to be our leg, where the first two joints are chosen to be actuation (joint 1 and 2 in Fig. 2 (b)). To derive the kinematics and dynamics of this mechanism, we can list out the D-H parameters in Table 1:

Joint	a_i	α_i	d_i	θ_i
1	a_1	α_1	0	θ_1
2	a_2	α_2	0	θ_2

Table 1: D-H parameters of overconstrained leg

Here, v and ω are the linear and angular velocity of the foot relative to o_0, τ is the joint output torque, and F is the 3-D force at the foot.

Define the translational matrix:

$$A_1^0 = R_{z,\theta_1} T_{x,a_1} R_{x,\alpha_1} = \begin{bmatrix} R_1^0 & o_1 \\ 0 & 1 \end{bmatrix}, \tag{1}$$

$$A_2^1 = R_{z,\theta_2} T_{x,a_2} R_{x,\alpha_2} = \begin{bmatrix} R_2^0 & o_2 \\ 0 & 1 \end{bmatrix}, \tag{2}$$

$$A_3^2 = T_{x,a_3}, \tag{3}$$

where A_1^0 is the translation from $joint1$ to $joint2$, A_2^1 is the translation from $joint2$ to $joint3$, and A_3^2 is the translation from $joint3$ to the end of the leg. Now we have the Jacobian matrix that maps

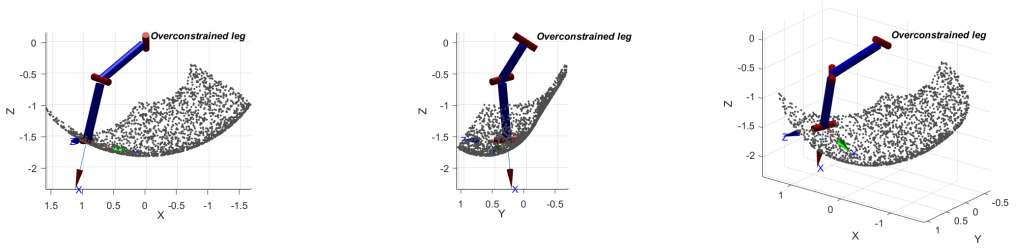
the joint space to the workspace force and velocity:

$$J(q) = \begin{bmatrix} z_0 \times (o_3 - o_0) & z_1 \times (o_3 - o_1) \\ z_0 & z_1 \end{bmatrix}, \tag{4}$$

$$\begin{bmatrix} v_2^0 \\ \omega_2^0 \end{bmatrix} = J \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}, \tag{5}$$

$$\tau = J^T(q)F. \tag{6}$$

If we choose some random points in the joint space and calculate the forward kinematics, the workspace of this overconstrained leg shows a 3 dimensional surface as in Fig. 4.



(a) Sagittal plane view (b) Frontal plane view (c) 3D view

Fig. 4: Plotted workspace of the overconstrained leg. The leg is fixed at joint 1 and some random joint angles of joint 1 and 2 were chosen. The workspace exhibits a 3-D surface. (a) the 3-D view, (b) the projection on $x - z$ plane and (c) the projection on the $y - z$ plane.

By changing the parameters a_i and α_i in Table 1, the overconstrained leg will generate different 3D surface. In this work, we select the parameter to be $a_i = 1$m, $\alpha_i = 1$rad. The selection of parameters defines the possible ability of moving in the sagittal and frontal plane. In this work, we only seek to realize a solution for vertical jumping and biped waling, but not to include parameter optimization, which is left to our future work.

Having the extended workspace is easy, controlling and making use of the given workspace is hard. In the following section, we will try to control and analyze a monopod robot with overconstrained leg, and the methods will be tested in simulations.

3. Sagittal Plane SLIP Hopping

The first locomotion we tried out on the overconstrained leg is one legged hopping, which will be a good start for future extension in biped and quadruped jumping. The world's first heuristically controlled monopod by Raibert[5] jumped stably in the 3D space, and the stability of that controller is proved in[13]. The analysis showed that Raibert's controller has its domain of attraction that stabilized the robot's state. Unlike hybrid zero dynamics[18] that required accurate modeling of each robot state, Raibert's controller that simplified the whole high dimensional system to a Spring Loaded Inverted Pendulum (SLIP) became much lighter and easier in the derivation of equations.

We believe it's possible for the monopod with overconstrained leg to jump stably in a sagittal plane, because the leg has a large area of workspace in the vertical direction of that plane (Fig. 4 (b)). If we control the leg to be a virtual spring model, the overconstrained leg will have a vertical force

on the robot's body. However, during jumping, the leg will also have a lateral displacement relative to the ground that is not beneficial to the lateral stability. This problem is cannot be ignored. In this work we only consider, and verify the stable jumping in sagittal plane, and the lateral movement is restricted by extra mechanisms.

3.1 Vertical Hopping of Overconstrained Leg

The recent work[10] separated the controllers of a tail-energized hopping robot to different templates, which is proved to be stable and is universal in legged robots. They treated the vertical hopping as a spring-mass-damper system with spring deflection χ. Then we have

$$\ddot{\chi} + 2\omega\beta\dot{\chi} + \omega^2\chi = \tau, \tag{7}$$

where ω is the natural frequency, and β the damping coefficient. τ is the vertical force exerted on the robot. We can choose τ to be

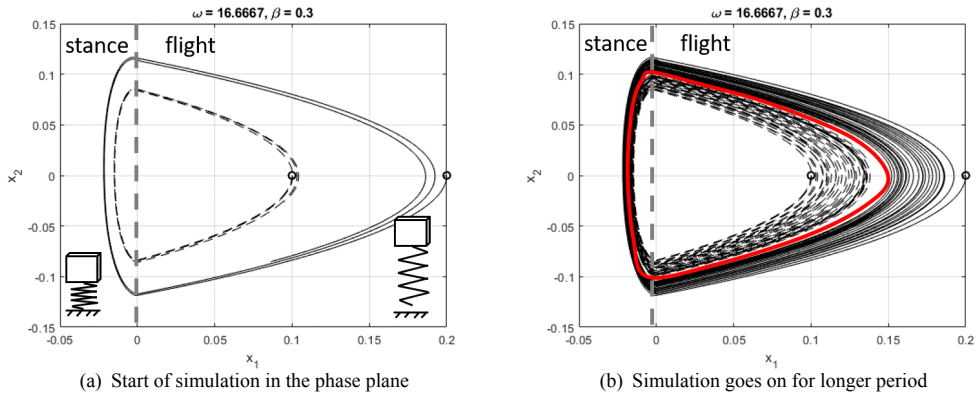
$$\tau := \frac{k_s x_1 + k_v x_2}{||x|| + \epsilon} \tag{8}$$

where $x_1 := \chi$, $x_2 := \dot{\chi}/\omega$ changes the robot state to a phase coordinate. If we write the robot state in the state space,

$$\begin{aligned} \dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} \omega x_2 \\ -2\omega\beta x_2 - \omega x_1 + \tau/\omega \end{bmatrix}, \\ &= -\omega \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} (-2\beta\omega x_2 + \tau/\omega). \end{aligned} \tag{9}$$

This template is proved mathematically to has a *unique attracting periodic orbit*[10]. That means, if we plot the vector field of this phase plane and select some random points as the initial state, the state will eventually converge to a periodic orbit. A simulation of the robot in its phase plane was run to demonstrate this orbit. In that simulation, which will be explained as followed, the robot state evolves through the gradient of the phase space in Equation (9).

Shown in Fig. 5(a), the robot starts from two initial states plotted by 'o'. The controller drives



(a) Start of simulation in the phase plane

(b) Simulation goes on for longer period

Fig. 5: Simulations of vertical hopping control template. The two 'o' points are two random initial states, and they both converge to a periodic orbit.

the robot towards a periodic orbit (b). When $x_1 < 0$, the robot is set to be touching the ground, under the force of τ; when $x_1 > 0$, the robot is on the fly with only gravitational force. After running the simulation for adequate step, the robot eventually converge to a periodic orbit, drawn in red in Fig. 5(b).

In order for the overconstrained leg to have the same jumping ability, we only have to control the leg to have the same vertical jumping force.

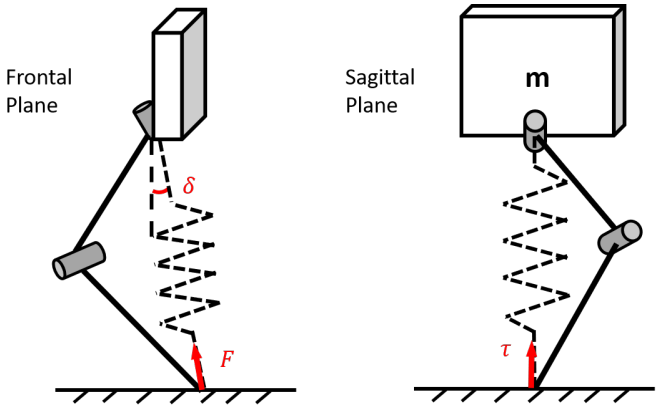


Fig. 6: Virtual model of the overconstrained leg. The force exerted by the leg on the vertical direction (left) has the same effect as a vertical spring (right).

Suppose that the angle between the virtual spring of the leg and the vertical line in the frontal plane is δ (Fig. 6 (a)), then we can simply control the fore of the leg to be:

$$F = \tau \arccos(\delta), \tag{10}$$

and the robot will have the same vertical jumping as the virtual template. The loss of energy caused by friction on the ground is large compared with those parallel legs, but it is still small relative to the energy provided by force F. So we decided to ignore the effect of friction and see the initial result.

3.2 Fore-Aft Speed Control

One-legged hopping robot uses its flight phase to control the fore-aft speed[5, 10, 15]. During the stance phase of SLIP, there exist a neutral touchdown angle β^* that stabilizes speed such that $||v^+|| = ||v^-||$. To find this neutral angle, we have a small-angle assumption when treating the angle swept by the leg. So the length traveled by the robot body equals to the arc length of the circle:

$$vT_s = 2\beta^* L_{spring}, \tag{11}$$

$$\beta^*(v) = \frac{vT_s}{2L_{spring}}, \tag{12}$$

where T_s is the duration of stance, and L_{spring} is the virtual spring length. v is the velocity in the sagittal plane.

If we apply the Raibert stepping controller to control the robot to a desired speed v^*:

$$\beta : v \mapsto \beta^*(v) + k_{sv}(v - v^*), \tag{13}$$

the robot will be stabilized to the forward speed v^*[10]. Here, the controlled touchdwon angle is a

function of v, and it maps to a function that contains our prediction of equilibrium (Equation (12)) and an adjustable gain k_{sv}.

4. Biped Stable walking

A 3D ellipsoid workspace of overconstrained leg makes possible a stable biped walking gait. As a comparison, biped robot with 2 DOF at the hip of each leg could achieve stable walking using hybrid zero dynamics (HZD) by applying to the robot a specified virtual constraint[7, 18]. Those legs with 2 DOF at their hips generate a spherical workspace, whereas the overconstrained leg generates an ellipsoid workspace. This similarity motivates us to try out HZD on an overconstrained biped.

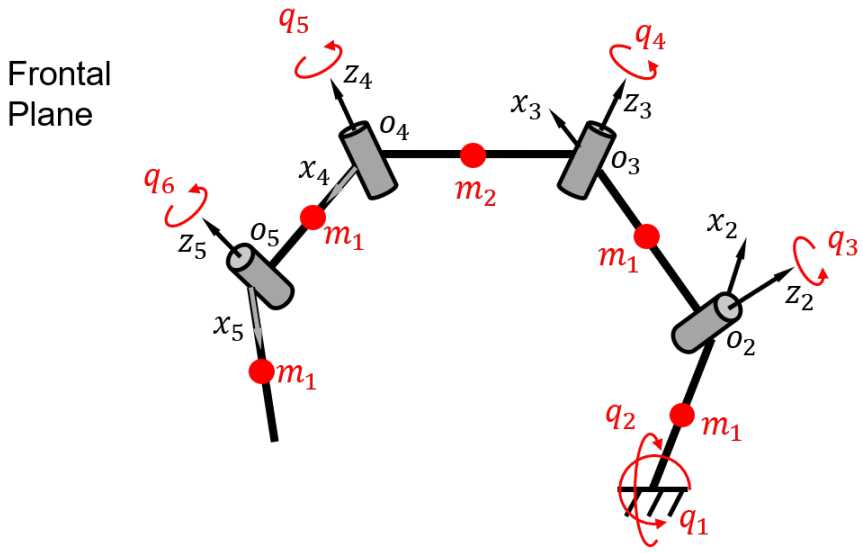


Fig. 7: A biped robot with overconstrained legs in swing phase. q_3 to q_6 represents the four DOF driven by an actuator, q_1 and q_2 are two underactuated joints at the stance foot. m_1 is the mass of each leg and m_2 is the body mass.

4.1 Swing Phase Model

During the stance phase, the biped robot is considered as a 5-link robot whose base is located at the stance feet, where two underactuated DOF lie in (q_1, q_2 in Fig. 7). Each leg of the robot has mass m_1 and length l, and the body of the robot has mass m_2 and length l_b. Coordinates that define the configuration is noted in the figure. Derive the translation between each coordinate, we have

$$
\begin{aligned}
A_1^0 &= R_{y,q_1}, \\
A_2^1 &= R_{z,q_2} T_{x,l} R_{x,-\alpha}, \\
A_3^2 &= R_{z,q_3} T_{x,l} R_{x,\alpha} R_{z,-q_4}, \\
A_4^3 &= R_{y,\beta} T_{x,l_b} R_{y,\beta} R_{z,q_5}, \\
A_5^4 &= T_{x,l} R_{x,-\alpha} R_{z,q_6}, \\
A_6^5 &= T_{x,l/2}.
\end{aligned}
$$

It is important to note the location of the center of mass of each link:

$$
\begin{aligned}
o_{c_1} &= (o_0 + o_2)/2, \\
o_{c_2} &= (o_2 + o_3)/2, \\
o_{c_3} &= (o_3 + o_4)/2, \\
o_{c_4} &= (o_4 + o_5)/2, \\
o_{c_5} &= o_6.
\end{aligned}
$$

For each joint, the linear and angular velocity Jacobian matrix is,

$$
\begin{aligned}
J_{v_i}^{3\times 1} &= z_{i-1} \times (o_5 - o_{i-1}), \\
J_{\omega}^{3\times 1} &= z_{i-1},
\end{aligned}
$$

from which we can derive the Jacobian matrix of each link:

$$
\begin{aligned}
J_{v_{c1}}^{3\times 6} &= \left[z_0 \times (o_{c1} - o_0) \quad z_1 \times (o_{c1} - o_1) \quad 0_{3\times 4}\right], \\
J_{v_{c2}}^{3\times 6} &= \left[z_0 \times (o_{c2} - o_0) \quad z_1 \times (o_{c2} - o_1) \quad z_2 \times (o_{c2} - o_2) \quad 0_{3\times 3}\right], \\
J_{v_{c3}}^{3\times 6} &= \left[z_0 \times (o_{c3} - o_0) \quad z_1 \times (o_{c3} - o_1) \quad z_2 \times (o_{c3} - o_2) \quad z_3 \times (o_{c3} - o_3) \quad 0_{3\times 2}\right], \\
J_{v_{c4}}^{3\times 6} &= [z_0 \times (o_{c4} - o_0) \quad z_1 \times (o_{c4} - o_1) \quad z_2 \times (o_{c4} - o_2) \quad z_3 \times (o_{c4} - o_3) \\
&\qquad z_4 \times (o_{c4} - o_4) \quad 0_{3\times 1}], \\
J_{v_{c5}}^{3\times 6} &= [z_0 \times (o_{c5} - o_0) \quad z_1 \times (o_{c5} - o_1) \quad z_2 \times (o_{c5} - o_2) \quad z_3 \times (o_{c5} - o_3) \\
&\qquad z_4 \times (o_{c5} - o_4) \quad z_5 \times (o_{c5} - o_5)]. \\
J_{\omega_{c1}}^{3\times 6} &= \left[z_0 \quad z_1 \quad 0_{3\times 4}\right], \\
J_{\omega_{c2}}^{3\times 6} &= \left[z_0 \quad z_1 \quad z_2 \quad 0_{3\times 3}\right], \\
J_{\omega_{c3}}^{3\times 6} &= \left[z_0 \quad z_1 \quad z_2 \quad z_3 \quad 0_{3\times 2}\right], \\
J_{\omega_{c4}}^{3\times 6} &= [z_0 \quad z_1 \quad z_2 \quad z_3 \quad z_4 \quad 0_{3\times 1}], \\
J_{\omega_{c5}}^{3\times 6} &= [z_0 \quad z_1 \quad z_2 \quad z_3 \quad z_4 \quad z_5].
\end{aligned}
$$

Therefore, we can compute the 6×6 mass-inertia matrix

$$
D_s^{6\times 6}(q) = \sum_i^n \{m_i J_{v_i}(q)^T J_{v_i}(q) + J_{\omega_i}(q)^T R_i(q) I_i R_i(q)^T J_{\omega_i}(q)\} = \begin{bmatrix} d_{11} & \cdots & d_{16} \\ \vdots & \ddots & \vdots \\ d_{61} & \cdots & d_{66} \end{bmatrix}, \quad (14)
$$

where R_i is the rotary matrix from base to the i-th link, I_i is the inertia of the i-th link about its center of mass. And the Coriolis matrix

$$
C_s^{6\times 6}(q, \dot{q}) = \begin{bmatrix} c_{11} & \cdots & c_{16} \\ \vdots & \ddots & \vdots \\ c_{61} & \cdots & c_{66} \end{bmatrix}, \quad (15)
$$

$$
c_{kj} = \sum_i^6 \frac{1}{2} \left(\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{kj}}{\partial q_i} - \frac{\partial d_{ij}}{\partial q_k} \right) \dot{q}_i,
$$

and the gravity term,

$$g_s(q) = \begin{bmatrix} \frac{\partial P}{\partial q_1} & \frac{\partial P}{\partial q_2} & \cdots & \frac{\partial P}{\partial q_6} \end{bmatrix}^T, \tag{16}$$
$$P = \sum_i^5 m_{ci} g o_{ci}.$$

The dynamics of the whole system during stance phase is then calculated from Equation (14), (15), (16)

$$D_s^{6\times 6}(q)\ddot{q} + C_s(q,\dot{q})\dot{q} + g_s(q) = \begin{bmatrix} 0_{2\times 4} \\ I_{4\times 4} \end{bmatrix} u_{4\times 1}. \tag{17}$$

because the first two DOF are underactuated.

We can write the model in state space by defining $x := (q_s \quad \dot{q}_s)^T$,

$$\begin{aligned} \dot{x} &= \begin{bmatrix} \dot{q} \\ D_s^{-1}(q)[-C_s(q,\dot{q})\dot{q} - G_s(q) + B_s(q)u] \end{bmatrix}, \\ &=: f_s(x) + g_s(x)u. \end{aligned} \tag{18}$$

4.2 Impact Model

The impact happens in double support phase, therefore requires four additional spatial variables[7] to extend the generalized coordinates, which becomes $q_e = [q^T \quad x_1 \quad y_1 \quad z_1 \quad q_{st}]^T$. The four additional variables are the Cartesian coordinates and the orientation of the stance foot. With this extension, the dynamic model should also be extended, start with Jacobian matrices:

$$\begin{aligned} J_{ev_{c1}}^{3\times 10} &= [J_{v_{c1}} \quad x_0 \quad y_0 \quad z_0 \quad z_0 \times (o_{c1} - o_0)], \\ &\vdots \\ J_{ev_{c6}}^{3\times 10} &= [J_{v_{c6}} \quad x_0 \quad y_0 \quad z_0 \quad z_0 \times (o_{c6} - o_0)]. \\ J_{e\omega_{c1}}^{3\times 10} &= [J_{\omega_{c1}} \quad 0_{3\times 3} \quad z_0], \\ &\vdots \\ J_{e\omega_{c6}}^{3\times 10} &= [J_{\omega_{c6}} \quad 0_{3\times 3} \quad z_0]. \end{aligned}$$

After renewing the dynamic model,

$$D_e^{10\times 10}(q_e)\ddot{q}_e + C_e^{10\times 10}(q_e,\dot{q}_e)\dot{q}_e + g_e^{3\times 10}(q_e) = B_e^{6\times 4}u + \delta F_{ext}, \tag{19}$$

where δF_{ext} is the vector of external forces. From the principle of virtual work[17],

$$\begin{bmatrix} D_e(q_e^-) & -E_2(q_e^-)^T \\ E_2(q_e^-) & 0_{2\times 2} \end{bmatrix} \begin{bmatrix} q_e^+ \\ F_2 \end{bmatrix} = \begin{bmatrix} D_e(q_e^-)\dot{q}_e^- \\ 0_{2\times 1} \end{bmatrix},$$

where $E_2(q_e) := \frac{\partial}{\partial q_e}[x_2 \quad y_2 \quad z_2 \quad q_{sw}]^T$ is the Jacobian matrix for the swing foot, F_2 is the reaction force at the end of the swing leg. The overall impact model could be written as[7],

$$\begin{aligned} q^+ &= \Delta_q(q^-), \\ \dot{q}^+ &= \Delta_{\dot{q}}(\dot{q}^-). \end{aligned}$$

If we define state variables as $x = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}$, then the complete periodic walking can be expressed as

$$\sum : \begin{cases} \dot{x} = f_x(x) + g_s(x)u & x^- \notin S \\ x^+ = \Delta(x^-) & x^- \in S \end{cases} \tag{20}$$

where $S := \{(q_s, \dot{q}_s) \in TQ_s | p_2^v(q) = 0, p_2^h(q) > 0\}$ is the switching set that determines the robot phase.

4.3 Virtual Constraint Controller Design

In his book,[17] proposed a virtual constraint method on the entire robot that restrict the robot motion to a desired evolution of the actuated variables. The form of the constraint is

$$y = h(q) = q_a - h_d(\theta). \tag{21}$$

q_a is the actuated coordinates, and θ is to parameterize the periodic motion. By defining the first, second and third derivative of the output y to be zero, the required torque can be computed to be

$$u^* = (\frac{\partial h(q)}{\partial} D^{-1} B)^{-1} \left(\frac{\partial^2 h_d(\theta)}{\partial \theta^2} \dot{\theta}^2(t) + \frac{\partial h(q)}{\partial q} D^{-1}(C(q, \dot{q})\dot{q} + G) \right), \tag{22}$$

$$:= (L_g L_f h)^{-1}(v - L_f^2 h). \tag{23}$$

The design of virtual constraint $q_a = h_d(\theta)$ is usually parameterized with Bezier polynomials. Whether or not the Beizer polynomials is applicable in overconstrained leg is a question that awaits future investigation. Mathmatically, let $Z = \{(q, \dot{q}) \in \chi | h(q) = 0, L_f h(q) = 0\}$ denotes the zero dynamics manifold and S is the switching set defined in Equation (20), the conditions required for stable walking is[19]:

(1) $S \cap Z$ is a smooth submanifold of χ;

(2) the decoupling matrix $L_g L_f h$ is invertible;

(3) the convergence time of the controller is strictly less than the time of a single step of the robot.

5. Simulation & Results

Simulations on sagittal plane hopping were conducted in the simulation platform Webots [1]. Experiments on vertical hopping and sagittal plane hopping both showed stability in the motion of overconstrained leg. The details of experiment setup and results are shown in the following sections.

5.1 Sagittal Plane Hopping

As we have mentioned in Section 3, hopping in the sagittal plane requires a combination of two controller templates, namely, vertical control template and fore-aft speed control template. The simulation environment is set up with a robot and its restriction. The robot's body weights $5kg$, and

[1] The source code of this simulation can be found on https://github.com/Wangbaiyue007/overconstrained-locomotion

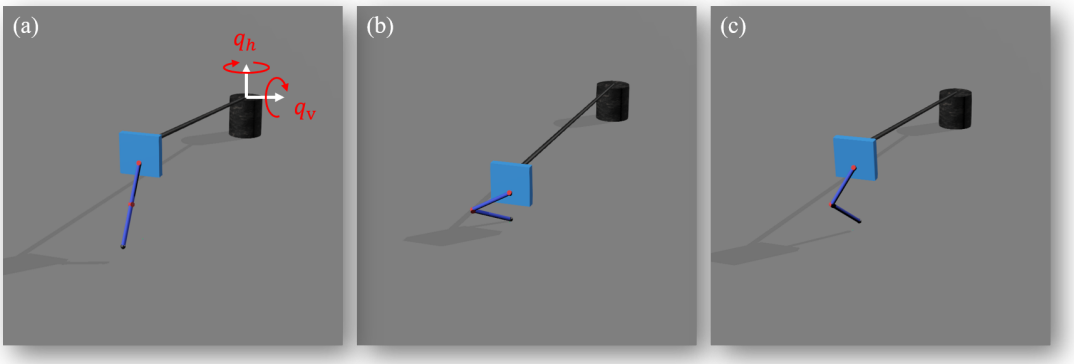


Fig. 8: Overconstrained leg jumping simulation in Webots. In this setup, the robot is connected by a stick with two DOF. The three figures show three different phases of the jumping motion. (a) is the initial status of the robot when the joints configurations are all zero, (b) is when the robot reaches the lowest point during stance phase, and (c) is the flight phase.

each leg weights $0.2kg$. The robot is connected to a stick whose other end is attached to a heavy cylinder with much greater mass. The stick is designed to have 2 rotary DOF around the cylinder, shown in Fig. 8 (a). With such physical restriction, the monopod then moves in a sagittal plane.

5.1.1 Vertical Hopping

We first tested the vertical hopping by limiting the robot's motion to 1 vertical DOF. That is, the stick that connects the robot only rotates around q_v in Fig. 8 (a). (a) to (c) is three frames selected from one continuous jump.

The algorithm of the controller for vertical jumping can be divided into the following parts:

- *main()*: the algorithm that enclose the whole simulation program.
- *UpdateRobotState()*: the algorithm that uses interface from Webots to estimate the robot state.
  - *FootTranslation()*: calculate the forward kinematics of overconstrained leg at current time.
  - *StateUpdate()*: update the robot's state machine that describe the robot by **STANCE** and **FLIGHT**.
  - *LegUpdate()*: calculate the force that should be generated from the overconstrained leg.
  - *JacobianUpdate()*: update the Jacobian matrix used in the dynamic control.
- *RobotControl()*: calculate the output torque of each motor from the results of *UpdateRobotState()*.

The entire simulation runs in *main()* (Algorithm 1). Within the *UpdateRobotState()* function,

there are four functions listed below in Algorithm 2, 3, 4, 5.

Algorithm 1: main()

```
while simulation is running do
    UpdateRobotState() ;
    RobotControl() ;
end
```

Algorithm 2: FootTranslation()

input : Two motor angles θ_1, θ_2; two twist angles α_1, α_2; two leg lengths l

output: Translational matrix A_1^0, A_2^0

```
A_1^0 ← rotz(θ_1)*translx(l)*rotx(α_1) ;
A_2^1 ← rotz(θ_2)*translx(l)*rotx(α_2) ;
A_2^0 ← A_1^0 · A_2^1 ;
```

Algorithm 3: StateUpdate()

Data: Virtual spring current length l_c; predefined spring relax length l_0; stance time T_s; previous stance time T_{sp}; robot state $state$

Result: Update robot state machine

```
if l_c < l_0 and IsFootTouching() is true then
    // IsFootTouching returns a result from the touch sensor located
       at the end of the leg, it is true if contact is detected
    T_s ← T_s + 0.002 ;
    state ← STANCE ;
else
    if state is STANCE then T_s ← T_s + T_sp;
    T_s ← 0 ;
    state ← FLIGHT ;
end
```

Algorithm 4: LegUpdate()

Data: Virtual spring current length l_c; predefined spring relax length l_0; virtual spring first derivative $\dot{l}_c$; spring displacement Δl; robot state $state$; leg reactional force τ; δ from Equation (10); 3 dimensional force vector $vector$

Result: Update Leg Reactional Force

```
Δl ← l_c − l_0 ;
if state is FLIGHT then
    τ ← SpringWithDamping(Δl, l̇_c) ;
    // spring force with constant stiffness and damping ratio
else
    τ ← Controller(Δl, l̇_c) ;
    // Controller from Equation (8)
    τ ← τ · arccos δ
end
vector ← Project(τ) ;
```

During the first ten jump, the height of the robot is plotted against time in Fig. 9.

Algorithm 5: RobotControl()

input : 3 dimensional force vector $vector$; Jacobian matrix J
output: Motor torques T

$T \leftarrow J^T \cdot vector$;
if `IsNotSingular()` *is true* **then**
    `// IsNotSingular() decide from the configuration of the robot whether the robot is in its singularities`
    `SetTorque(`T`)` ;
else
    `SetTorque(`*0.01*`)` ;
    `// set small torque to avoid sigularities`
end

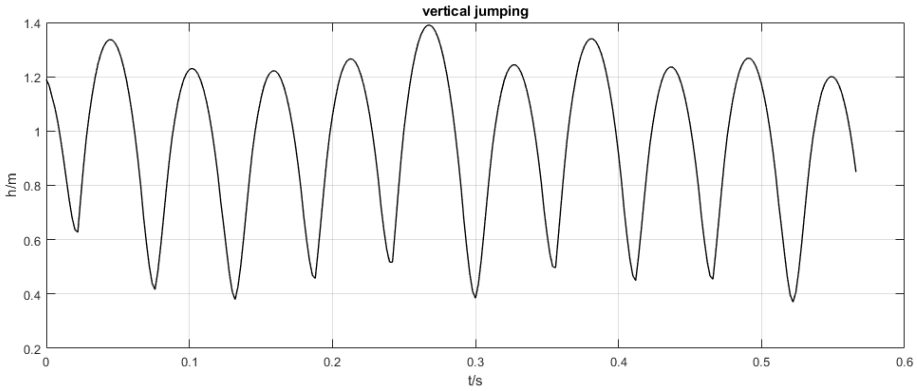


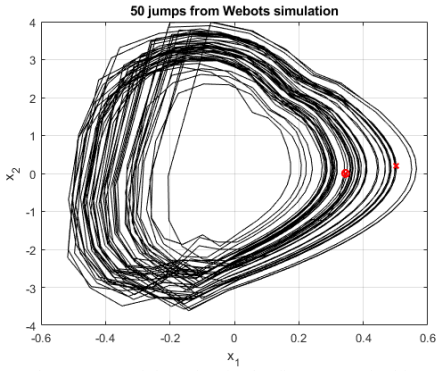
Fig. 9: Height versus time during the vertical jumping

We can tell from Fig. 9 that the robot realized stable jumping at approximately constant height, at a range of $[1.2m, 1.4m]$, and thus proves the stability of the vertical hopping template. Fig. 10(a) is the phase space of the robot during the first 50 jumps simulation in Webots. From this figure, the robot encounters some disturbances when it touches the ground, but its motion still follows a periodic orbit, as modeled in theory in Fig. 5.

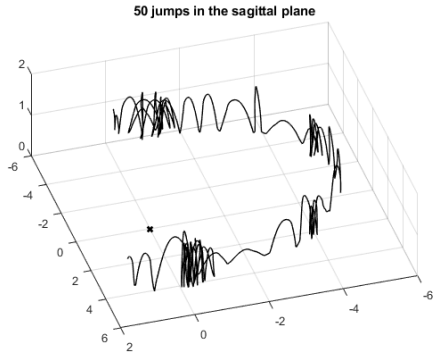
5.1.2 Sagittal Plane Hopping

Hopping in the sagittal plane requires at least 2-DOF motion, so both q_h and q_v in Fig. 8 (a) are enabled. The controller now is a combination of vertical hopping (10) and fore-aft speed control (13). Specifically in our code, a *VelocityControl()* (Algorithm 6) function is added in *UpdateRobotState()*. During our implementation, *VelocityControl()* is placed before *LegUpdate()*, and the *LegUpdate()* function now has a PD gain to track the desired touchdown angle β^*.

We track the robot's coordinates in the 3D space as shown in Fig. 10(b). Form our observation, the monopod sometimes becomes unstable due to the slight relative motion between stance foot and the ground, making it difficult to control horizontal speed in the sagittal plane. However, most of the other time, the controller stabilized the horizontal speed from becoming too big, generating an overlapping trajectories. The overlapping trajectories in Fig. 10(b) represents those stabilized repetitive motions.



(a) Phase space of the robot in the first 50 vertical jumps.



(b) Robot's position during the first 50 jumps in the sagittal plane.

Fig. 10: Simulation results from Webots.

Algorithm 6: VelocityControl()

input : Robot state $state$; stance time T_s; 3 dimensional position vector o; tangential velocity v_t; radial velocity v_r; current touchdown angle β; its first derivative $\dot{\beta}$; δ from Equation (10); virtual spring current length and its first derivative l_c, $\dot{l_c}$

output: Robot forward velocity v_f; desired touchdown angle β^*

$v_t \leftarrow -\dot{\beta} \cdot l_c$;

$v_r \leftarrow -\dot{l_c}$;

if *state is STANCE* **then**

  $\delta \leftarrow$ `acos(ProjectYZ(`o`)/`l_c`)`-1 ;

  `// the coordinates of the leg is projected onto the Y-Z plane, 1 is the angle of` α

  $v_f \leftarrow$ (v_r`sin(`β`)` $+v_t$ `cos(`β`))*cos(`δ`)` ;

end

Update β, $\dot{\beta}$;

$\beta^* \leftarrow (v_f + T_s)/$(`2`$l_0$`*cos(`$\delta$`)`) ;

`// refer to Equation` (12)

6. Discussion & Future Work

In this work, a novel overconstrained leg structure is proposed and analyzed for SOPHIE. The overconstrained leg utilizes the twist angle between two joints to generate a 3 dimensional ellipsoid workspace, which is rarely seen in a 2 DOF robotic leg. Being not only special in its workspace, this overconstrained leg also exhibits potentials in multiple locomotion modes. Explained and derived in this work includes monopod hopping and biped stable walking.

The monopod hopping mode is verified in simulation to be stable in a sagittal plane, giving rises to possibilities of applications in quadruped robots. The biped walking mode is theoretically derived, and is expected to work in simulations and in real life, but the problem is left to selecting an realizable virtual constraints.

This is only the beginning of the investigation. Theoretically, the overconstrained leg is capable of jumping as well as walking, but none of them has been verified in real life. In our future work, a jumping mode should be tested, especially under the unknown of contact friction that could be a major effects on motions. A biped mode should also be considered further to derive the entire HZD controller with specified virtual constraint, and therefore be verified in real life.

References

[1] BADRI-SPRÖWITZ A, TULEU A, VESPIGNANI M, **andothers**. Towards Dynamic Trot Gait Locomotion: Design, Control, and Experiments with Cheetah-cub, a Compliant Quadruped Robot[J]. The International Journal of Robotics Research, 2013, 32. DOI: 10.1177/0278364913489205.

[2] KAU N, SCHULTZ A, FERRANTE N, **andothers**. Stanford Doggo: An Open-Source, Quasi-Direct-Drive Quadruped[C]//. [S.l. : s.n.], 2019: 6309-6315. DOI: 10.1109/ICRA.2019.8794436.

[3] KENNEALLY G, KODITSCHEK D E. Leg design for energy management in an electromechanical robot[C]//2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). [S.l. : s.n.], 2015: 5712-5718. DOI: 10.1109/IROS.2015.7354188.

[4] KENNEALLY G, DE A, KODITSCHEK D E. Design Principles for a Family of Direct-Drive Legged Robots[J]. IEEE Robotics and Automation Letters, 2016, 1(2): 900-907. DOI: 10.1109/LRA.2016.2528294.

[5] RAIBERT M H, H. BENJAMIN BROWN J, CHEPPONIS M. Experiments in Balance with a 3D One-Legged Hopping Machine[J]. The International Journal of Robotics Research, 1984, 3(2): 75-92. DOI: 10.1177/027836498400300207.

[6] BLEDT G, POWELL M J, KATZ B, **andothers**. MIT Cheetah 3: Design and Control of a Robust, Dynamic Quadruped Robot[C]//2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). [S.l. : s.n.], 2018: 2245-2252. DOI: 10.1109/IROS.2018.8593885.

[7] SHIH C L, GRIZZLE J, CHEVALLEREAU C. Asymptotically Stable Walking of a Simple Underactuated 3D Bipedal Robot[C]//IECON 2007 - 33rd Annual Conference of the IEEE Industrial Electronics Society. [S.l. : s.n.], 2007: 2766-2771. DOI: 10.1109/IECON.2007.4460177.

[8] PLECNIK M M, HALDANE D W, YIM J K, **andothers**. Design Exploration and Kinematic Tuning of a Power Modulating Jumping Monopod[J]. Journal of Mechanisms and Robotics, 2016, 9(1). DOI: 10.1115/1.4035117.

[9] HALDANE D W, PLECNIK M M, YIM J K, **andothers**. Robotic vertical jumping agility via series-elastic power modulation[J]. Science Robotics, 2016, 1(1). DOI: 10.1126/scirobotics.aag2048.

[10] DE A, KODITSCHEK D E. Parallel composition of templates for tail-energized planar hopping[C]//2015 IEEE International Conference on Robotics and Automation (ICRA). [S.l. : s.n.], 2015: 4562-4569. DOI: 10.1109/ICRA.2015.7139831.

[11] HON-CHEUNG Y. The Bennett linkage, its associated tetrahedron and the hyperboloid of its axes[J]. Mechanism and Machine Theory, 1981, 16(2): 105-114. DOI: https://doi.org/10.1016/0094-114X(81)90056-2.

[12] SONG C, CHEN Y, CHEN I M. A 6R linkage reconfigurable between the line-symmetric Bricard linkage and the Bennett linkage[J]. Mechanism and Machine Theory, 2013, 70: 278-292. DOI: https://doi.org/10.1016/j.mechmachtheory.2013.07.013.

[13] KODITSCHEK D E, BÜHLER M. Analysis of a Simplified Hopping Robot[J]. The International Journal of Robotics Research, 1991, 10(6): 587-605. DOI: 10.1177/027836499101000601.

[14] SCHWIND W, KODITSCHEK D. Control of forward velocity for a simplified planar hopping robot[C]//Proceedings of 1995 IEEE International Conference on Robotics and Automation: **volume** 1. [S.l. : s.n.], 1995: 691-696 vol.1. DOI: 10.1109/ROBOT.1995.525364.

[15] ARSLAN Ö, SARANLI U. Reactive Planning and Control of Planar Spring–Mass Running on Rough Terrain[J]. IEEE Transactions on Robotics, 2012, 28(3): 567-579. DOI: 10.1109/TRO.2011.2178134.

[16] DE A, KODITSCHEK D E. Vertical hopper compositions for preflexive and feedback-stabilized quadrupedal bounding, pacing, pronking, and trotting[J]. The International Journal of Robotics Research, 2018, 37(7): 743-778. DOI: 10.1177/0278364918779874.

[17] WESTERVELT E R, GRIZZLE J W, CHEVALLEREAU C, **andothers**. Feedback Control of Dynamic Bipedal Robot Locomotion[Z]. 2007.

[18] WESTERVELT E, GRIZZLE J, KODITSCHEK D. Hybrid zero dynamics of planar biped walkers[J]. IEEE Transactions on Automatic Control, 2003, 48(1): 42-56. DOI: 10.1109/TAC.2002.806653.

[19] PLESTAN F, GRIZZLE J, WESTERVELT E, **andothers**. Stable walking of a 7-DOF biped robot[J]. IEEE Transactions on Robotics and Automation, 2003, 19(4): 653-668. DOI: 10.1109/TRA.2003.814514.

Acknowledgement

- This work is not possible if there's no support form my tutor, Prof. Song Chaoyang. I would like to thank his brilliant ideas on pushing forward the industry and his support in my original ideas.
- Thanks to my schoolmates and friends for giving me interesting ideas as well as emotional support during the development of this work.
- I would also like to thank Southern University for providing us such a dynamic research environment, and a freedom of choosing the thesis topic that eventually lies in my research interest.
- Thanks to my family for supporting me to persuade my study in SUSTech. Thanks to my cat Huihui for accompany. Thanks to my computer for staying up late with me and for running the rigorous calculations.