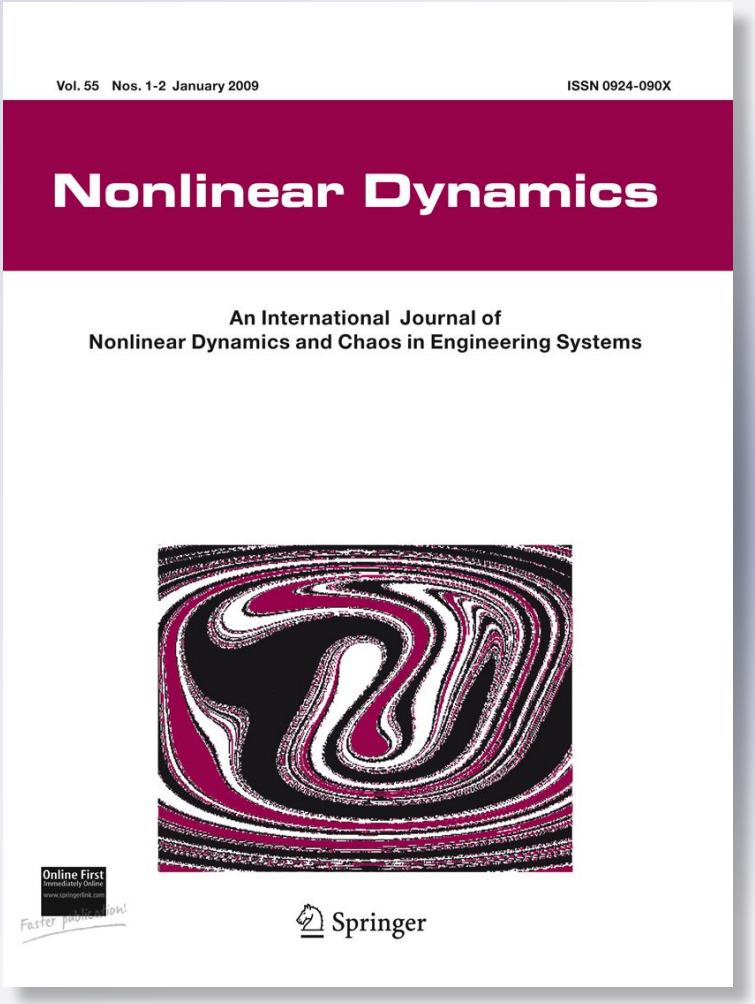
An approximate decoupled dynamics and kinematics analysis of legless locomotion

Ravi Balasubramanian, Alfred A. Rizzi & Matthew T. Mason

Nonlinear Dynamics
An International Journal of Nonlinear Dynamics and Chaos in Engineering Systems

ISSN 0924-090X
Volume 67
Number 3

Nonlinear Dyn (2012) 67:2123-2138
DOI 10.1007/s11071-011-0134-z



Springer

Nonlinear Dyn (2012) 67:2123–2138
DOI 10.1007/s11071-011-0134-z

ORIGINAL PAPER

An approximate decoupled dynamics and kinematics analysis of legless locomotion

Ravi Balasubramanian · Alfred A. Rizzi · Matthew T. Mason

Received: 13 January 2011 / Accepted: 12 June 2011 / Published online: 31 August 2011

Abstract We present a novel analysis technique to understand the dynamics of a recently described locomotion mode called legless locomotion. Legless locomotion is a locomotion mode available to a legged robot when it becomes high-centered, that is, when its legs do not touch the ground. Under these conditions, the robot may still locomote in the plane by swinging its legs in the air, rocking on its body, and taking advantage of the nonholonomic contact constraints. Legless locomotion is unique from all previously studied locomotion modes, since it combines the effect of oscillations due to controls and gravity, nonholonomic contact constraints, and a configuration-dependent inertia. This complex interaction of phenomena makes dynamics analysis and motion planning difficult, and our proposed analysis technique simplifies the problem by decoupling the robot's oscillatory rotational dynamics from its contact kinematics and also decoupling the dynamics along each axis. We show that the decoupled dynamics models are significantly simpler, provide a good approximation of the motion, and offer insight into the robot's dynamics. Finally, we show how the decoupled models help in motion planning for legless locomotion.

Keywords Dynamics approximation · Kinematics approximation · Robotic locomotion · Nonholonomic constraints

R. Balasubramanian (✉)
Oregon State University, Corvallis, OR, USA
e-mail: ravi.balasubramanian@oregonstate.edu

A.A. Rizzi
Boston Dynamics, Waltham, MA, USA
e-mail: arizzi@bostondynamics.com

M.T. Mason
Carnegie Mellon University, Pittsburgh, PA, USA
e-mail: matt.mason@cs.cmu.edu

1 Introduction

The dynamics of a mechanical system can be complex due to the interaction of different phenomena like the coupling between various degrees of freedom, environmental contact, and external forces like gravity. Analysis can become more complex when the system is oscillatory as well. In such situations, simple models even if approximate may provide insights into the dynamics. In this paper, we propose a novel technique to simplify the dynamics of a unique complex locomotion mode called legless locomotion [1].

Legless locomotion was discovered during experiments that explored for novel locomotion modes for mobile-robot error recovery. Legless locomotion provides incremental mobility when a legged robot becomes high-centered on a rock, that is, when the robot's body is perched on a rock and the legs cannot push off the ground. The key idea in legless locomotion is to exploit the dynamic effect of swinging the legs to induce body rotations (assuming a rounded

body) which when coupled with a rolling contact induce translation. We use a robot called the Rocking and Rolling Robot for studying legless locomotion (see Fig. 1). RRRobot is always high-centered since its legs do not touch the ground and can only locomote by swinging its legs. We have shown earlier that legless locomotion offers planar accessibility through gaits that provide straight-line and curved translation [1]. Figure 2 shows an example of the oscillatory translation produced by legless locomotion.

Importantly, legless locomotion presents a new class of dynamic systems for research since the combination of its properties make it entirely unique from all previously studied locomotion. Specifically, legless locomotion's dynamics is continuous, oscillatory, and exploits the interaction between a configuration-dependent system inertia and nonholonomic contact constraints in the presence of gravity. As a result of these properties, legless locomotion is different from legged locomotion gaits such as walking, running, and jumping, which have hybrid dynamics due to intermittent contact. Legless locomotion is also different from typical wheeled locomotion, which is quasistatic, continuous, and non-oscillatory. Table 1 also shows how legless locomotion is different from classes of previously studied dynamic locomotion modes (the rows indicate the variation in mechanism inertia and the columns indicate the variation in contact constraints and the influence of gravity; interestingly, to our knowledge, there are no examples in the literature of a dynamically coupled locomotion mode with constant inertia, contact constraints, and gravitational drift).

While legless locomotion's equations of motion are straightforward to derive using, say, a Lagrangian formulation [2], the mechanics is complex due to the combination of its properties. Its unique characteristics preclude the application of existing techniques for dynamics analysis and developing control algorithms, such as linearization [3, 4] and kinematic reduction [5] (see Sect. 2 for more details). Furthermore, legless locomotion's dynamics structure is difficult to integrate symbolically even for a specific input, thus forcing a numerical analysis.

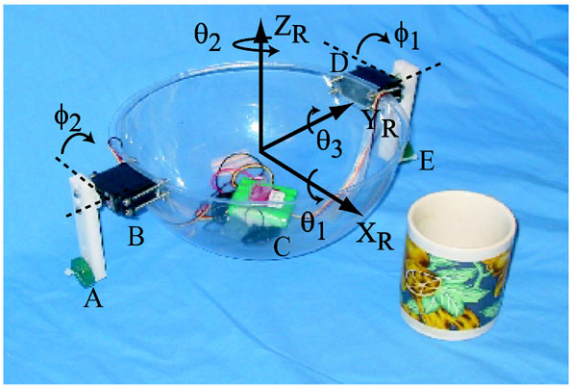


Fig. 1 The Rocking and Rolling Robot (RRRobot), which is used to study legless locomotion, uses halteres to induce body attitude oscillations leading to body translations. RRRobot has two massless legs that are driven by servos and translates by rocking and rolling on the spherical shell. The shell has negligible mass when compared to the masses at the distal ends of the legs (at A and E), the servo mass (at B and D), and the controller and the battery mass at the shell bottom (at C)

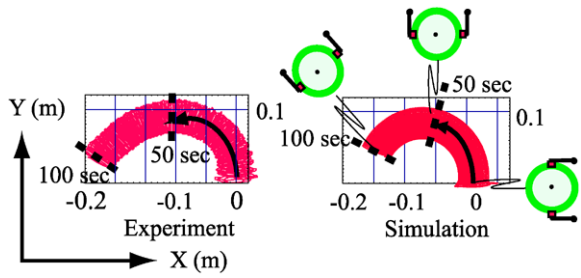


Fig. 2 The path traced by a legless locomotion gait (sinusoidal leg motions) that produces counter-clockwise translation through body roll-pitch-yaw oscillations [16]. The *broad* (*red*) *patch* is a result of the back-and-forth pitch oscillations. The figure also shows RRRobot's changing orientation over time

Table 1 Comparison of properties between legless locomotion and other classes of dynamically coupled locomotion modes

	No nonholonomic contact constraints	With nonholonomic contact constraints and no gravitational drift	With nonholonomic contact constraints and gravitational drift
Constant inertia	Floating rigid bodies (submarines) [6, 7]	Snakes [8], snakeboard [9], and roller racer [10]	
Configuration-dependent inertia	Floating articulated systems (satellites) [11–13]	Trikke [14, 15]	Legless locomotion (RRRobot)

As a result, the motion planning and control problem for legless locomotion is difficult to solve with current methods; that is, it is nontrivial to find the control inputs (leg swing trajectories) that produce the instantaneous desired robot translation and rotation. In this paper, we present a set of simple models that decouple legless locomotion's dynamics from its contact kinematics, but still closely approximate legless locomotion's behavior. We then use the simplified models to solve a simpler version of the legless locomotion control problem, namely an approximate inverse dynamics solution for RRRobot's oscillatory locomotion at steady state. After a brief review of related work in Sect. 2, we present the legless locomotion dynamics models in Sect. 3. We present the simplified legless locomotion models in Sect. 4 and then outline some ideas that lead toward an inverse dynamics solution and control strategy for legless locomotion in Sect. 5.

2 Related work

2.1 Studying dynamic locomotion with constraints

There exist several approaches for studying a mechanical system's motion through deriving its equations of motion, including the Lagrange–Euler [2, 17, 18] and the Newton–Euler methods [19], the differential-variation principle [20], and the generalized D'Alembert principle [21, 22] (see [23] for a review of recent advances in robot dynamics research). In addition, numerous investigators have studied dynamic systems with nonholonomic contact constraints [24–26] such as the *snakeboard* [9, 27] and Trikke [14, 15], kinematic systems such as the *Sphericle* [28] and spherical balls with orthogonal actuators [29], and floating articulated manipulators [12, 13]. However, we have shown earlier in [1] that legless locomotion is unique from all previously studied locomotion modes, since it is dynamic, oscillatory, continuous and exploits the simultaneous interaction of shape changes, a varying inertia, and nonholonomic contact constraints in the presence of gravity. Thus, while we can use the Lagrangian method to derive legless locomotion's equations of motion, the equations include hundreds of coupled terms making analysis complex.

2.2 Dynamics approximation techniques

Several techniques exist in the literature to simplify dynamics analysis. One method is to linearize the control system about an operating point or nominal trajectory [3, 4]. Even though an approximation, linearization provides insights into many control systems. For example, Laumond [30] gives various linearization techniques to control a nonholonomic car-like robot. However, RRRobot's dynamics does not lend itself to linearization. While RRRobot's pitch and roll oscillation dynamics can be approximated by linear systems since the oscillation dynamics are essentially damped pendulums, RRRobot's yaw dynamics is inherently nonlinear. The nonlinearities arising from RRRobot's con-figuration-dependent inertia are essential to produce curved translation through the incremental yaw produced over each cycle.

Other approaches to dynamics simplification include neglecting the interaction between the limbs of a star-like mechanism and the influence of limb motion on the base [31], but this approach is effective only when the base is significantly heavier than the limbs and the limbs have many degrees of freedom. Another possibility is to ignore the contributions of the outer links of a parallel manipulator to the rotational kinetic energy [32] but this approach is valid only when the inertial effects are small compared to gravitational forces. Finally, Chen et al. [33] provide a method to construct the equations of motion for a multilegged robot in a modular fashion, one leg at time, but this approach does not work for legless locomotion since the motion of the legs and body are inherently coupled.

While the aforementioned approaches offer approximations of a mechanical system's dynamics, one approach called kinematic reduction [5] offers an exact simplification. The key idea it to reduce a dynamic system to a kinematic system. So what is the difference between the two systems? A dynamic system has velocity-related terms and force or torque control inputs, which make developing control algorithms difficult, while kinematic systems are drift-free and use velocity inputs. The term "drift" here refers to the velocity-product terms which indicate that the system moves even in the absence of control inputs. Bullo and Lynch [7] provide a direct algorithm for finding kinematic reductions by enforcing the condition that the kinematic system must satisfy the mechanical system's dynamic constraints at arbitrary time scaling,

and Bullo and Lewis [34] provide planning primitives for the Snakeboard by finding a kinematic reduction. In addition, Shammas et al. [35, 36] provide a natural gait generation strategy using height functions for the planar snakeboard and the Trikke which operate in gravity-free environments. Since legless locomotion is influenced by gravity, it is not possible to find a kinematic reduction or develop height-functions for RRRobot in the current form. However, as we will show in Sect. 4.2, kinematic reductions are still useful for developing insight into legless locomotion's simplified models.

2.3 Synchronization in oscillatory systems

The legless locomotion system comprises multiple oscillators, namely, the pitch, roll, and yaw body oscillations, driven by the leg motions. These oscillators are coupled through the system's configuration-dependent inertia and the nonholonomic contact constraints. The steady-state phase difference (or synchronization) between the various oscillators determines the robot's translation in the plane. Such synchronization between oscillators has been studied in detail before (see [37] for a nice introduction to the problem and other references). However, there are unique challenges in studying how the three oscillators in legless locomotion are coupled since rotations in $SE(3)$ and the translation of the contact point in the plane (due to the nonholonomic contact constraints) do not commute [26]. We show through an empirical analysis that decoupling the three body rotations still provides a close approximation of the synchronization between the oscillators.

3 Legless locomotion dynamics

We use a robot called the *Rocking and Rolling Robot* (RRRobot, see Fig. 1) [1] to study the mechanics of legless locomotion. RRRobot is an unconventional bipedal robot: it has a rounded bottom, two actuated legs, but the legs never touch the ground. RRRobot is thus high-centred always, and the legs act only as reaction masses. RRRobot's design helps us explore leg motions that induce locomotion as the robot rolls and rocks on its rounded stomach.

RRRobot's design includes a massless rigid shell of radius r to which are hinged two massless legs of length l. There are five masses on the robot: a reaction mass at the distal end of each leg (M_l), a servo mass where each leg is hinged (M_s), and the battery and controller mass at the bottom of the shell (M_b). Torques τ_1 and τ_2 may be applied at the leg joints, and the rigid shell rolls on the plane without slip at the single point of contact (noncompliant contact). The geometric and inertial parameters used in this paper are provided in Table 2.

Table 2 Geometric and inertial parameter values

Parameter	Value
Leg mass M_l	0.057 kg
Servo mass M_s	0.053 kg
Battery and controller mass M_b	0.3 kg
Shell radius r	0.12 m
Leg length l	0.1 m
Leg motion frequency	8 rad/s
Gravity	9.81 m/s^2

RRRobot's configuration q_{rr} consists of the sphere's position and orientation with respect to a spatial frame and the internal configuration of its legs and can be expressed as $q_{\text{rr}} = (x, y, \theta_r, \theta_p, \theta_y, \phi_1, \phi_2) \in \mathbb{R}^7$. Here x and y represent the position of the contact point in the plane, θ_r, θ_p, and θ_y the Euler angles used to represent body orientation, and ϕ_1 and ϕ_2 the leg position.

The equations of motion for RRRobot on a plane, which can be derived using any method listed in Sect. 2, take the form

$$M(q_{\text{rr}})\ddot{q}_{\text{rr}} + C(q_{\text{rr}}, \dot{q}_{\text{rr}})\dot{q}_{\text{rr}} + G(q_{\text{rr}}) = \tau + \left(\lambda_1 \omega^1\right)^T + \left(\lambda_2 \omega^2\right)^T + \zeta_{\text{rr}}, \tag{1}$$

$$\omega^1 \dot{q}_{\text{rr}} = 0, \tag{2}$$

$$\omega^2 \dot{q}_{\text{rr}} = 0, \tag{3}$$

$$\omega^1 = (1, 0, -r\cos\theta_p \sin\theta_y, -r\cos\theta_y, 0, 0, 0), \tag{4}$$

$$\omega^2 = (0, 1, r\cos\theta_r \cos\theta_y, -r\sin\theta_y, 0, 0, 0), \tag{5}$$

where $M(q_{\text{rr}}) \in \mathbb{R}^{7\times 7}$ represents the positive-definite nondiagonal configuration-dependent mass matrix, $C(q_{\text{rr}}, \dot{q}_{\text{rr}})\dot{q}_{\text{rr}} \in \mathbb{R}^7$ the vector of Coriolis and centrifugal terms, $G(q_{\text{rr}}) \in \mathbb{R}^7$ the vector of gravitational terms, $\zeta_{\text{rr}} \in \mathbb{R}^7$ the energy loss, and $\tau = (0, 0, 0, 0, 0, \tau_1, \tau_2)^T$ the generalized force. The generalized force τ indicates that only the legs are actuated. The gravitational terms cause RRRobot to behave as a pendulum (for small amplitude oscillations),

and RRRobot's pitch and roll natural frequencies are governed by its mass distribution and the shell's curvature. Thus, when the legs are swung along oscillatory trajectories, the superimposition of the dynamic effect of leg swings, gravity, and contact losses cause RRRobot to behave as a forced damped oscillator [38].

The sphere-plane no-slip contact constraints [24] are defined by (2) and (3), and the interplay of oscillatory body rotations in legless locomotion and the contact kinematics have been discussed in [1]. Two points to keep in mind are: (1) out-of-phase pitch-yaw rotations cause RRRobot to translate in a straight line, and (2) when this motion is coupled with yaw drift, RRRobot translates in a curved path. The symbols $\lambda_1, \lambda_2 \in \mathbb{R}$ represent the magnitudes of the contact constraint forces. All energy losses due to the sphere rolling on the ground are bundled into a viscous damping term $\zeta_{\text{rr}} = \kappa \dot{q}$, where $\kappa \in \mathbb{R}^{7\times 7}$ depends on the surface. The resulting equations of motion are complex (over two hundred terms), and understanding the contribution of various elements like robot shape, mass distribution, and control choices to legless locomotion is difficult.

4 Simplified legless locomotion models

In this section, we present three types of simplifications that provide insight into the various dynamics and kinematics aspects of legless locomotion. The first two types of simplifications result in approximations of legless locomotion's mechanics and, to the knowledge of the author, similar simplifications have not been applied to other systems in prior literature. Legless locomotion's unique mechanics lends itself to such simplifications:

1. Study the system's rotational dynamics and nonholonomic contact kinematics separately and then recombining them. This is achieved by pivoting the robot body's geometric center at a spherical joint and studying the effect of leg motions on the body's rotational motion. The body rotations are then piped through the contact kinematics (2) and (3) to compute the robot's translation. This model is called the pivoting dynamics model (see Fig. 3 and Sect. 4.1).
2. Study the system's rolling dynamics along each rotational freedom of the body separately and then recombining the individual rotations using the contact kinematics. These models are called the single-axis models (see Sect. 4.2).
3. Explore kinematic reductions for the simplified models; that is, explore if the simplified models can be modeled as a drift-free system with velocity inputs rather than a dynamic system with drift and force/torque controls. Note that kinematic reductions do not exist for the complete RRRobot dynamics due to gravitational drift (see Sect. 4.2).

The first two types of simplification are approximations since they assume that the leg-body rotational dynamics and contact kinematics are decoupled. The second type of simplification further assumes that the body's dynamics along each rotational axis, namely pitch, roll, and yaw, are decoupled. We discuss the implications of each assumption in the following sections.

The third type of simplification, however, is an exact simplification and is based on techniques developed by Bullo, Lewis, and Lynch [5]. It involves identifying if the dynamic system with acceleration inputs and drift can be modeled as a drift-free kinematic system with velocity inputs. This is useful because control and planning for kinematic systems is easier. Section 4.2 explores kinematic reductions for RRRobot's simplified models.

The key motivation for decoupling legless locomotion's dynamics from its kinematics is that planning and control for the decoupled models becomes simpler. For example, considering just the sphere–plane contact kinematics and ignoring how the body rotations are produced, we notice that interleaved pitch–yaw body rotations produce net displacement (similar to parallel parking with a unicycle). Similarly, considering just the interplay between the dynamics of RRRobot's leg motions and body rotations while ignoring the robot's planar translation, we notice that swinging the legs with different phase relationships produces body pitch, roll, and yaw rotations. For example, swinging the legs in phase produces pitch oscillations, while swinging the legs 180 degrees out of phase induces yaw oscillations. A key result in this paper is that combining such decoupled dynamics and kinematics models provides a good approximation to the RRRobot's original fully integrated dynamics. Thus, we can exploit these simplified models to find the gaits that produce the required motions

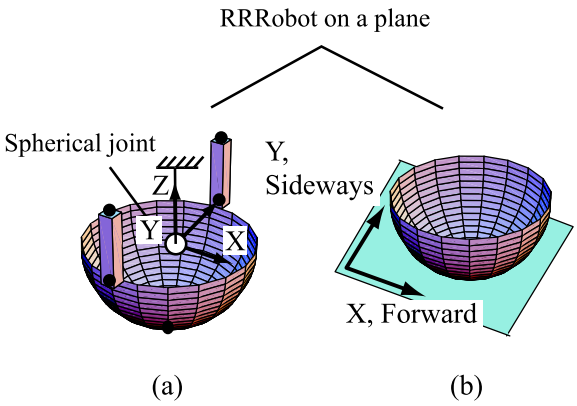


Fig. 3 The pivoting dynamics model simplifies the RRRobot-on-a-plane model (see Fig. 1) into two parts: (**a**) RRRobot pivoted at its geometric center on a spherical joint and (**b**) a sphere on a plane

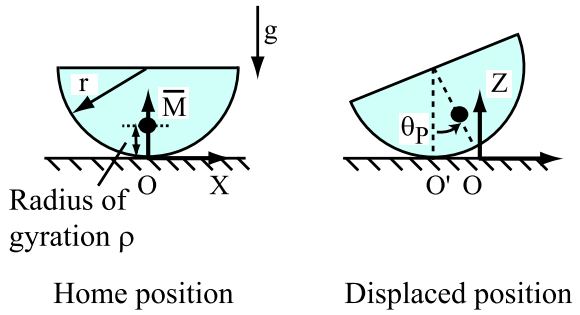


Fig. 4 A planar eccentric-mass wheel performs harmonic oscillations for small amplitude

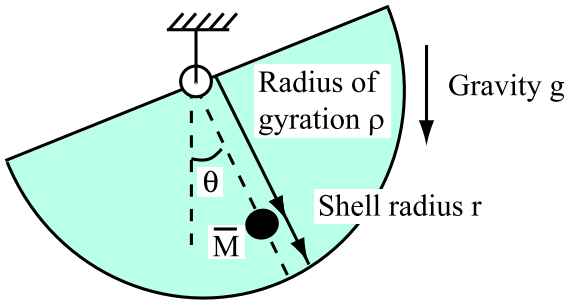


Fig. 5 The simple pendulum performs harmonic oscillations for small amplitude

in the decoupled models individually and then apply those same gaits in the full dynamics models.

We now discuss the three types of simplifications.

4.1 Pivoting dynamics model

The configuration q_{pd} of the pivoting dynamics model consists of the sphere's orientation $R(\theta_r, \theta_p, \theta_y)$ with respect to a inertial frame and the configuration of its legs (ϕ_1, ϕ_2); that is, $q_{\text{pd}} = (\theta_y, \theta_p, \theta_r, \phi_1, \phi_2)^T \in \mathbb{R}^5$.

The equations of motion for the pivoting dynamics model take the form

$$M_{\text{pd}}(q_{\text{pd}})\ddot{q}_{\text{pd}} + C(q_{\text{pd}}, \dot{q}_{\text{pd}})\dot{q}_{\text{pd}} + G(q_{\text{pd}}) = \tau_{\text{pd}} + \zeta_{\text{pd}}, \tag{6}$$

where $M_{\text{pd}}(q_{\text{pd}}) \in \mathbb{R}^{5\times5}$ represents the positive-definite nondiagonal variable mass matrix, $C(q_{\text{pd}}, \dot{q}_{\text{pd}})\dot{q}_{\text{pd}} \in \mathbb{R}^5$ represents the vector of Coriolis and centrifugal terms, $G(q_{\text{pd}}) \in \mathbb{R}^5$ represents the vector of gravitational terms, and $\tau_{\text{pd}} = (0, 0, 0, \tau_1, \tau_2)^T$ represents the generalized force. The generalized force τ_{pd} indicates that only the legs are actuated, and there are no external constraints on the system. Note that (6) does not include the influence of the contact kinematics and differs from RRRobot's dynamics modeled in (1). All energy losses are bundled into the viscous damping term $\zeta_{\text{rr}} = \kappa_{\text{pd}}\dot{q}_{\text{pd}}$, where $\kappa \in \mathbb{R}^{5\times5}$.

Once we compute the changes in body configuration for a certain leg trajectory, we use the kinematic contact equations in (2) and (3) to compute the velocity of the contact point in the plane, where $q_{\text{rr}} = (x, y, q_{\text{pd}}^T)^T$ represents robot configuration. We now can use the pivoting dynamics model to approximate RRRobot's motion.

The pivoting dynamics model only "approximates" the full RRRobot model because the dynamics has been decoupled from the contact kinematics. Also, the body's rotational axes is different in the two systems—the full dynamics model has a rolling contact while the pivoting dynamics model has a spherical joint (at the sphere center). As a result, RRRobot's center of mass is oscillating about a moving contact point, whereas the pivoting dynamics model's center of mass is oscillating about the sphere's fixed geometric center. Thus, if we consider just one axis of rotation for the body, RRRobot behaves like an rolling pendulum (see Fig. 4), and the pivoting dynamics model behaves like a simple pendulum (see Fig. 5). The different rotational axes result in different effective rotational inertias and, consequently, different natural frequencies and oscillation amplitudes. Specifically, the natural time period of oscillations for a rolling pendulum for

Table 3 Rotation time-periods for the RRRobot-on-a-plane model and the pivoting dynamics model

	Roll rotations (sec)	Pitch rotations (sec)
RRRobot-on-a-plane	1.29	1.07
Pivoting dynamics	1.19	0.96

small amplitudes is

$$T_{\text{ip}} = 2\pi\sqrt{\frac{\rho^2}{g(r-\rho)}}, \tag{7}$$

where ρ is the radius of gyration, and g is gravity, while the time period for oscillation for a simple pendulum is

$$T_{\text{sp}} = 2\pi\sqrt{\frac{\rho}{g}}. \tag{8}$$

Table 3 compares the roll and pitch rotational time periods for RRRobot and the pivoting dynamics model.

Translation predicted by the pivoting dynamics models Figures 6 and 7 compare RRRobot's motion in simulation with the motion predicted by the pivoting dynamics models (see [1] for details of experiments with the robot prototype). The RRRobot simulations use the damping parameters

$$\zeta_{\text{rr}} = \text{Diag}(0, 0, -0.01\dot{\theta}_r, -0.01\dot{\theta}_p, -0.01\dot{\theta}_y, -0.01\dot{\phi}_1, -0.01\dot{\phi}_2), \tag{9}$$

and the pivoting dynamics simulations use the damping parameters

$$\zeta_{\text{pd}} = \text{Diag}(-0.01\dot{\theta}_r, -0.01\dot{\theta}_p, 0, -0.01\dot{\phi}_1, -0.01\dot{\phi}_2). \tag{10}$$

We use different yaw damping values, because yaw damping nullifies any net yaw produced by leg motions in the pivoting dynamics model.

The translation produced in the pivoting dynamics model and in the RRRobot-on-a-plane model match qualitatively. While the linear translation of the pivoting dynamics model and the full dynamics model are almost identical, the pivoting dynamics model rotates significantly faster. This is because of the strong coupling between the body's rotational motions and the continuous transfer of energy between the pitch and yaw freedoms, which causes the robot's yaw configuration to increase rapidly.

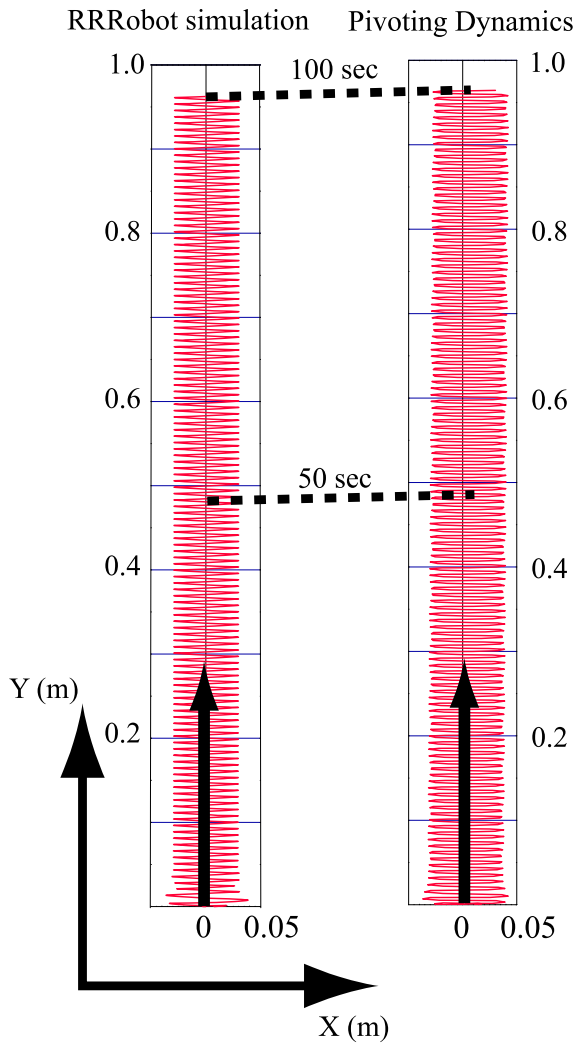


Fig. 6 Planar plots of contact-point time profile during sideways locomotion produced by Gait 1 in RRRobot-on-a-plane simulation and pivoting dynamics simulation. The *solid arrow* gives robot motion direction, and the *dotted lines* indicate the robot position at the specified time

The key insight from the pivoting dynamics model is that even if the system's dynamics and kinematics are decoupled, the mechanism's translation in the plane is still qualitatively similar to the full dynamics model. However, the body rotations and leg motions are still coupled in the pivoting dynamics model, making analysis complex.

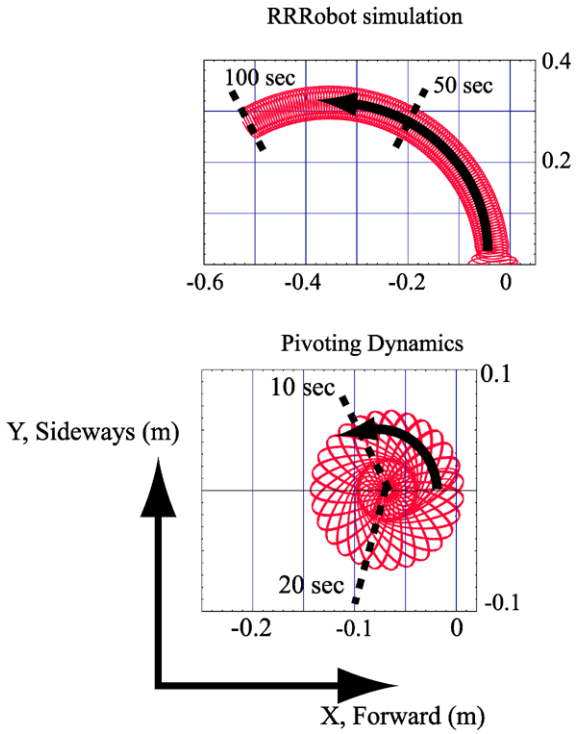
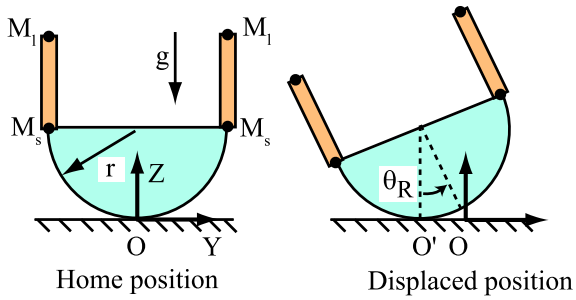


Fig. 7 Planar plots of contact-point time history during counter-clockwise circular locomotion produced by Gait 2 in RRRobot-on-a-plane simulation and pivoting dynamics simulation. The *solid arrow* gives robot motion direction, and the *dotted lines* indicate the robot position at the specified time

4.2 Single-axis-rotation models

The single-axis-rotation models assume that there is negligible coupling between the three rotational motions of the body (see Figs. 8, 9, and 10). Thus, the single-axis rotation models focus on each specific rotational freedom by disabling the remaining nonactuated freedoms. For example, in the pitch model the body's roll and yaw rotations are set to zero while allowing only pitch rotations. Similarly, we allow only roll rotations and yaw rotations in the roll and yaw dynamics models, respectively. Note that the roll and pitch models have a rolling contact, while the yaw model is pivoted. The body rotations that result from these dynamic models are piped into the contact kinematics equations.

The dynamics analysis in the single-axis-rotational models is similar to analyzing a satellite in space with three reaction wheels aligned with perpendicular axes, but restricting the satellite's roll and pitch rotational freedoms when studying the influence of the reaction



Shell radius	r	Gravity	g
Motor mass	M	Leg mass	m

Fig. 8 RRRobot's roll freedom (side view)

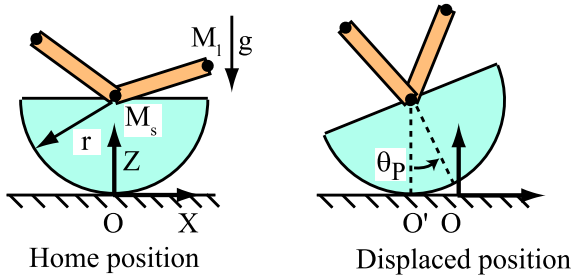


Fig. 9 RRRobot's pitch freedom (side view)

wheel motions on the yaw rotations. This allows us to understand the influence of controls on the various passive freedoms individually. The individual motions are then superposed.

The equations of motion for these fictitious single-axis-rotation models take the form

$$M(q_{\mathrm{sa}})\ddot{q}_{\mathrm{sa}} + C(q_{\mathrm{sa}}, \dot{q}_{\mathrm{sa}})\dot{q}_{\mathrm{sa}} + G(q_{\mathrm{sa}}) = \tau_{\mathrm{sa}} + \zeta_{\mathrm{sa}}, \tag{11}$$

where $q_{\mathrm{sa}} = \{\theta, \phi_1, \phi_2\} \in \mathbb{R}^3$. The last two elements of q_{sa} represent leg configuration, while the first element θ is the body roll, pitch, or yaw configuration depending on the model. The symbols $M(q_{\mathrm{sa}}) \in \mathbb{R}^{3\times3}$, $C(q_{\mathrm{sa}}, \dot{q_{\mathrm{sa}}})\dot{q_{\mathrm{sa}}} \in \mathbb{R}^3$, $G(q_{\mathrm{sa}}) \in \mathbb{R}^3$ represent standard mechanical-system terms, and $\tau_{\mathrm{sa}} = (0, \tau_1, \tau_2)^T$ is the generalized force. The input torques τ_1 and τ_2 are applied to the legs, while body rotation is not actuated. The resulting body-rotation trajectories are plugged into the sphere–plane contact kinematics given by (2) and (3) to compute translation. The symbol ζ_{sa} represents damping, and we use $\zeta_p = (0.01\dot{q}_p, 0, 0)^T$ in the pitch decoupled model, $\zeta_r = (0.01\dot{q}_r, 0, 0)^T$ in the

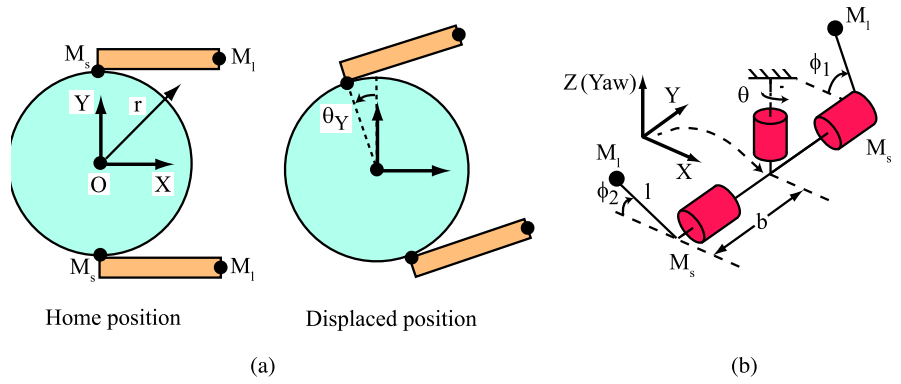


Fig. 10 The Yaw model is derived from RRRobot design by pivoting the body at its body center, allowing free rotation about the yaw axis only: (**a**) top view and (**b**) schematic view

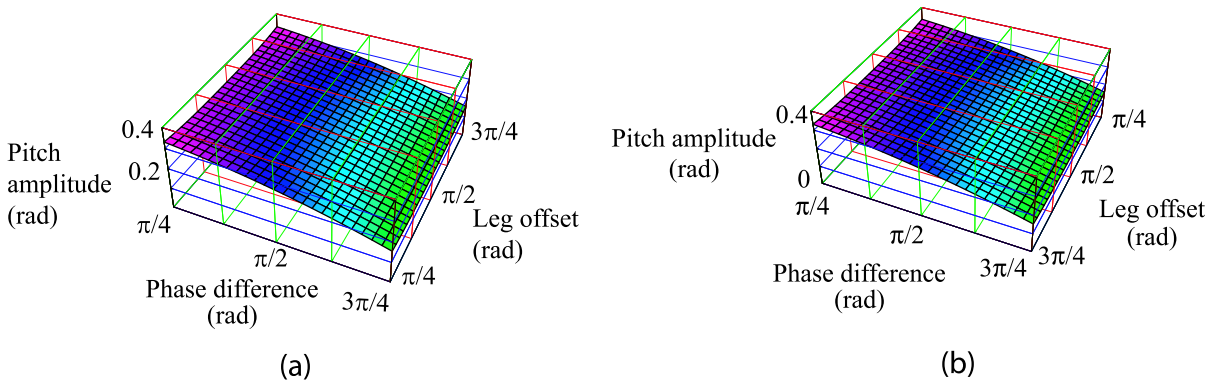


Fig. 11 Comparison between (**a**) RRRobot's pitch dynamics and (**b**) the single-axis pitch model: pitch amplitude as a function of leg motion phase difference and offset

roll decoupled model, and $\zeta_y = (0, 0, 0)^T$ in the yaw decoupled model.

We now discuss in detail the dynamics of the pitch and yaw single-axis models.

4.2.1 The single-axis pitch model

The single-axis pitch model is derived by restricting RRRobot's rotational freedoms to only the pitch freedom (see Fig. 9). The robot body oscillates about its vertical configuration depending on leg torques and the natural oscillatory dynamics (due to gravity) and settles into a limit cycle due to frictional damping. Thus, the pitch model's oscillations may be controlled through the leg oscillation amplitude, frequency, phase, and offset. The mean body pitch offset may be determined by static analysis. It was noticed that the single-axis pitch model's dynamics is predominantly linear for sinusoidal leg trajectories. As a result, there are several linear-control techniques to control the pitch oscillations for the robot [39]. For example, for the choice of the simulation parameters we have used in this paper, the body pitch oscillation frequency is only slightly different from leg frequency. Note that the pitch model does not have a kinematic reduction due to the configuration-dependent drift produced by gravity.

Figure 11b shows how body pitch oscillation amplitude varies as a function of the leg trajectory controls. Comparing with Fig. 11a, we note that the pitch model represents RRRobot's pitch-oscillation amplitudes well. The remaining parameter, pitch phase, is not important in an absolute sense; rather the pitch phase value relative to the yaw phase value is important, since the relative phase influences robot translation. We discuss the relation between pitch and yaw phase in the yaw-model subsection. Note that this pitch model better represents RRRobot's pitch motion than the pivoting dynamics models, since the rolling contact is retained.

4.2.2 The single-axis yaw model

The yaw model is derived by pivoting RRRobot at a revolute joint (aligned with the Z axis) placed at the sphere's geometric center (see Fig. 10), and its mechanics helps us understand RRRobot's yaw rotations.

The Yaw model body has two masses, each M_s, at the ends of a diameter. Each massless leg has an actuated hip joint and a point mass M_l at the distal end. The yaw model configuration is represented by $q_y = (\theta_y, \phi_1, \phi_2)^T \in \mathbb{R}^3$, where θ_y denotes the body configuration, ϕ_1 leg 1's joint configuration, and ϕ_2 leg 2's joint configuration.

The yaw model has no gravity, there are no joint limits, and torques u_1 and u_2 can be applied at leg joints 1 and 2. The mass matrix $M_y(q_y)$ associated with the Yaw model and describing the system kinetic energy is

$$M_y(q_y) = \begin{pmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{pmatrix}, \tag{12}$$

where

$$\begin{aligned} g_{11} &= 2(M_m + M_l)b^2 + M_l l^2 \\ &\quad + \frac{1}{2} M_l l^2 (\cos 2\phi_1 + \cos 2\phi_2), \\ g_{12} &= -M_l l r \sin\phi_1, \\ g_{13} &= M_l l r \sin\phi_2, \\ g_{21} &= -M_l l r \sin\phi_1, \\ g_{22} &= M_l l^2, \\ g_{23} &= 0, \\ g_{31} &= M_l l r \sin\phi_2, \\ g_{32} &= 0, \\ g_{33} &= M_l l^2. \end{aligned}$$

Note that the mass matrix $M_y(q_y)$ depends on leg configurations but is independent of yaw rotations; that is, $M_y(q_y)$ does not depend on θ_y. Such an invariance is called a symmetry in the Yaw model, implying the existence of a conserved quantity [40]. In the yaw model, this conserved quantity is the yaw angular momentum.

The Yaw model equations of motion [41] are given by

$$M_y(q_y)\ddot{q}_y + C_y(q_y, \dot{q}_y) = \tau, \tag{13}$$

where

$$\tau = \begin{pmatrix} 0 \\ \tau_1 \\ \tau_2 \end{pmatrix}$$

is the control. The control τ indicates that the Yaw model is underactuated. Also, if the system's initial velocity $\dot{q}_y$ is zero, then the body must be stationary when the legs are stationary.

In contrast to the linearity of the single-axis pitch model, the nonlinearity of the yaw model makes control and planning difficult. Also, a key difference between the single-axis yaw model and the full-dynamics model is that the yaw inertia in the full dynamics model is a function of body pitch (due to the rolling contact) and leg configuration, while yaw inertia in the single-axis yaw model is only a function of leg configuration (the yaw pivot prevents body pitch). This causes a larger yaw drift rate in the full-dynamics model. If we want to use the decoupled dynamics models to approximate RRRobot's motion, some adjustment is required to match the yaw drift between the yaw model and the full dynamics model. In our work, we vary leg amplitude as a function of leg offset for the decoupled yaw model to ensure that the yaw drift matches with RRRobot's dynamics.

Control for the single-axis yaw model Planning and control for the yaw model using (13) is difficult, because of the velocity-related terms and the torque inputs; that is, there is no systematic analytic procedure to find torque inputs to achieve a given goal trajectory. While it is clear from the principle of conservation of angular momentum that the body will rotate if a leg is moved, the key question with the single-axis yaw model is whether the body can reach arbitrary configuration using leg motions. When viewed in the context of RRRobot's locomotion, this question about the yaw model will help answer if RRRobot can reach an arbitrary position and orientation in the plane.

A key feature of the yaw model will help answer this question. Specifically, the invariance of the yaw model dynamics to yaw orientation permits a kinematic reduction for the system [42]. The yaw model's

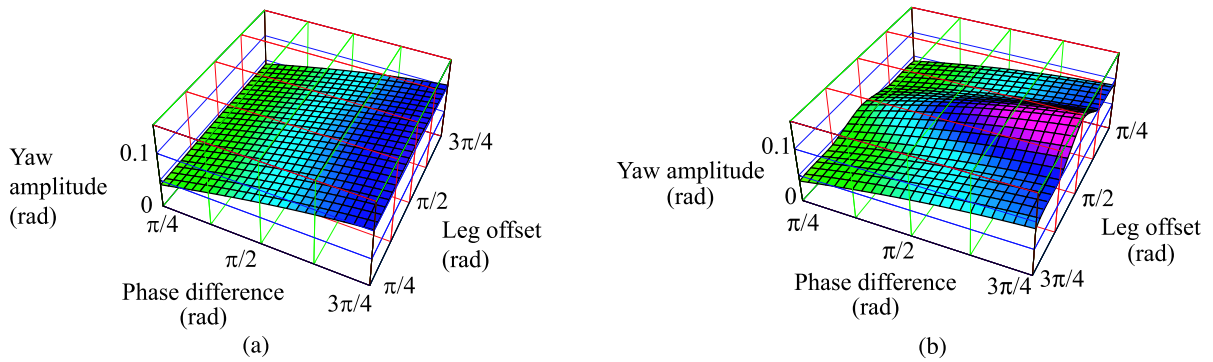


Fig. 12 Comparison between (**a**) RRRobot's yaw dynamics and (**b**) the single-axis yaw model: yaw oscillation amplitude as a function of leg motion phase difference and offset for sinusoidal leg motions

invariance effectively means that the dynamics in (13) can be integrated to provide a mapping between the leg velocities and the body yaw velocity as follows:

$$g_{11}\dot{\theta} + g_{12}\dot{\phi}_1 + g_{13}\dot{\phi}_2 = 0. \tag{14}$$

The kinematic reduction greatly simplifies planning and control for the yaw model since we can now use velocity inputs for leg motions rather than joint torques. Using techniques in [5], we find two gaits for the yaw model that allow the kinematic model configuration controllability (the ability to reach any configuration at rest), while ensuring that the trajectories can be tracked by the mechanical system. One gait involves moving one leg while keeping the other leg stationary and produces net yaw for acyclic leg motions. The second gait involves moving both legs in out-of-phase sinusoids. The correct phase relationship and leg offset produces net yaw. The key idea in the second gait is that the body inertia seen by the system is different during different segments of leg motions, and such trajectories when coupled with the angular momentum conservation principle results in net yaw. Thus, if we want to move the yaw model from one configuration to another, we apply the second gait followed by first gait to both legs.

Figure 12b shows how body yaw oscillation amplitude relates to leg oscillatory trajectories for the yaw model. This compares favorably with RRRobot's yaw-oscillation amplitudes shown in Fig. 12a. The main difference is the small hump near offset $\pi/2$ in the yaw model, while there is no spike in the full RRRobot model. This is attributed to the pitch–yaw coupling in the full dynamics model: as the leg offset shifts from the vertical ($\pi/2$), the robot pitches from the vertical. This causes the yaw inertia about the rolling contact to increase, since the battery mass is offset from the axis and, consequently, produces smaller yaw oscillations.

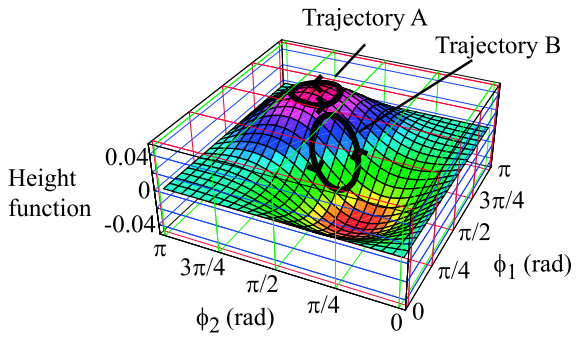


Fig. 13 Yaw model height function. Trajectory A (leg 1: $5\pi/8 + 0.15 + 0.3\sin(8t)$ and leg 2: $5\pi/8 + 0.15 + 0.3\cos(8t)$) produces net body yaw, while trajectory B (leg 1: $\pi/2 + 0.3\sin(8t)$ and leg 2: $\pi/2 + 0.3\cos(8t)$) does not produce net yaw

Furthermore, techniques developed by Shammas et al. [35] allow us to compute net body yaw for different leg trajectories (see Fig. 13). Note that these height functions are time-independent and purely depend on the paths in leg configuration space. As expected, cyclic leg motions about the vertical configuration produces zero net yaw, while cyclic leg motions about configurations offset from the vertical produces net yaw.

Even though we have found kinematic reductions for the yaw model, it does not extend directly to RRRobot, because of the effect of gravity and the coupling between the body pitch and yaw rotations.

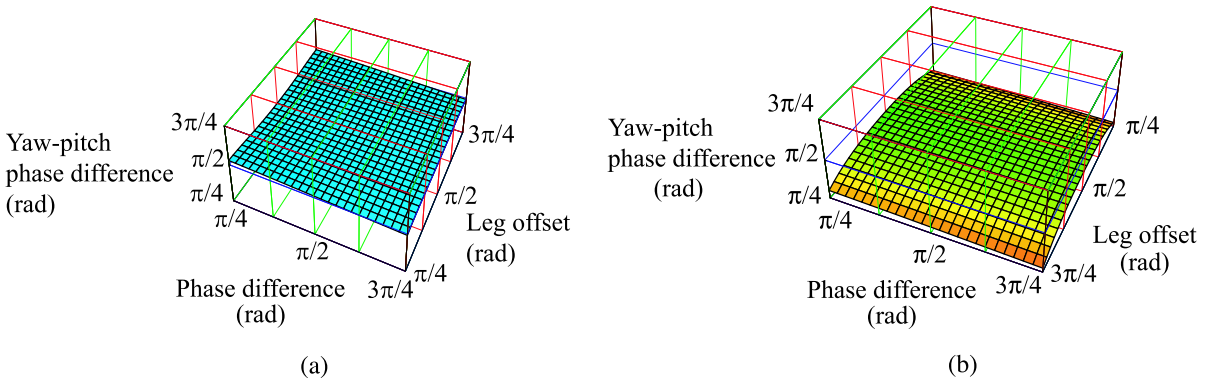


Fig. 14 Comparison between (**a**) RRRobot's yaw dynamics and (**b**) the single-axis yaw model: pitch–yaw phase difference as a function of leg motion phase difference and offset

Specifically, the leg cycles that produce maximum body yaw motion is different in the two systems since the yaw inertia varies with pitch configuration in the full dynamics model. But we can still use the kinematic reduction for the yaw model as an approximate model of RRRobot's yaw orientation. Making the leg amplitude a function of leg offset in the yaw model adjusts for the body pitch-yaw rotational coupling in the full dynamics model (see Sect. 5).

Also, while the phase difference between body pitch and yaw oscillations predicted by the single-axis models is different from the phase difference in the full dynamics model (see Fig. 14), the phase difference between pitch and yaw influences translation velocity alone and not curvature. Since RRRobot's velocity is small, this discrepancy does not impact control significantly if we focus only on the path traveled. A more detailed analysis of the factors that influence the phase relationship between the various oscillators in the full dynamics model is required [37]. Finally, while we have only discussed kinematic reduction results for the simple Yaw model, finding kinematic reductions for complex systems such as the legless locomoting RRRobot is an open problem.

4.2.3 Translation predicted by single-axis rotation models

Figure 15 shows one example of how closely translation predicted by the single-axis models match RRRobot's translation for small amplitude $\pi/2$ out-of-phase leg oscillations about the vertical. These leg motions produce body pitch and yaw oscillations about

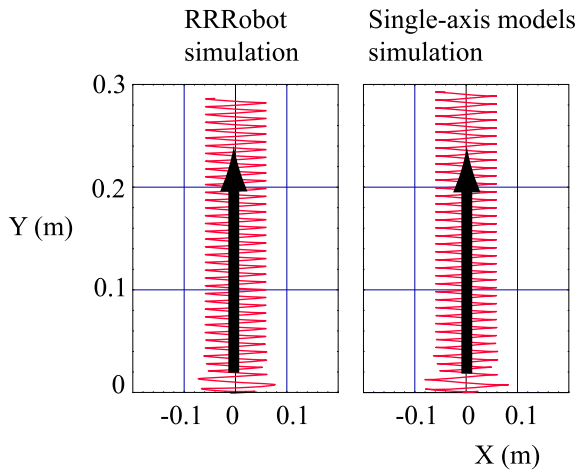


Fig. 15 The lateral translation gait: comparison of RRRobot's motion with motion predicted by the single-axis models over thirty seconds. Leg 1 trajectory: $\pi/2 + 0.3\sin(8t)$, and leg 2 trajectory: $\pi/2 + 0.3\cos(8t)$

zero, while roll rotation is negligible. There is a strong match between the body rotation trajectories for the decoupled models and the full dynamics models, indicating that pitch and yaw oscillations are decoupled.

Figure 16 shows one example of how closely translation predicted by the single-axis models match RRRobot's translation for small amplitude out-of-phase leg oscillations offset from the vertical. These leg motions produce pitch and yaw body oscillations primarily in addition to small roll oscillations. In addition, body pitch tilts from the vertical, and each leg cycle produces net body yaw in both the full-dynamics model and the decoupled dynamics model (see Sect. 5

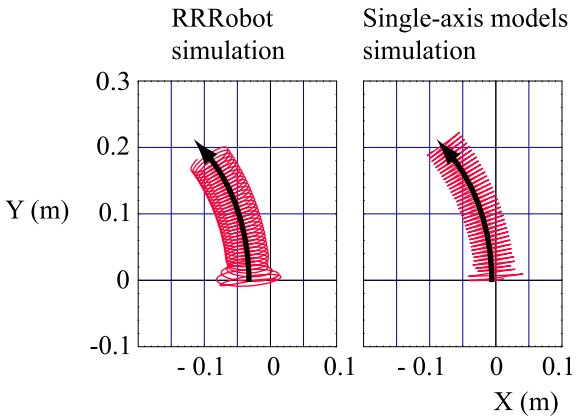


Fig. 16 The circular translation gait: comparison of RRRobot's motion with motion predicted by the single-axis models over thirty seconds. Leg 1 trajectory: $\pi/4 + 0.3\sin(8t)$, and leg 2 trajectory: $\pi/4 + 0.3\cos(8t)$

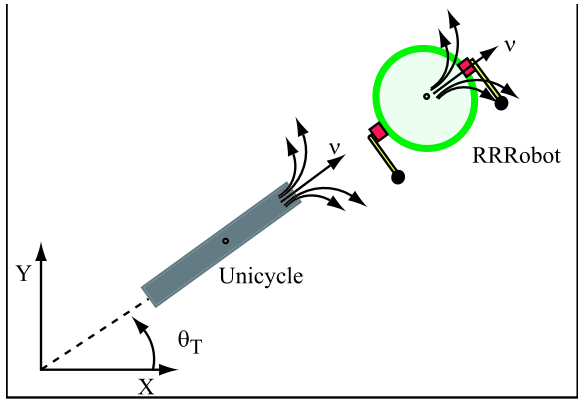


Fig. 17 Similarity in planar translation between a vertical unicycle and RRRobot (top view)

for a more detailed analysis of the ability of the simplified models to approximate the full-dynamics model).

5 Toward legless locomotion control

Control in robotics may be defined as finding a mapping from the desired robot motion to the inputs. In general, control is easier if there is an input to control each of the robot's freedoms, that is, the robot is fully actuated, and if the dynamics did not depend on configuration. Legless locomotion's properties, specifically underactuation, nonholonomic contact constraints, and drift due to gravity, particularly make the control problem complex.

In this paper, we focus only on RRRobot's net motion over a cycle, rather than RRRobot's instantaneous motion during the oscillatory cycle. Thus, the control problem we focus on is to find the leg motion trajectory that produces the net contact-point velocity over a cycle. In this section, we show that the control mapping is similar for the single-axis models and the full dynamics models. The key motivation is that developing control strategies for the single-axis models is simpler (as shown in Sects. 4.2.1 and 4.2.2), and then the same control strategies can be applied to the full RRRobot dynamics.

RRRobot's predominant translation mode is translation along the leg-rotation axis with bounded net curvature and bounded linear velocity, and this results from the limited body rotational dynamics that the leg motions can produce [1]. Such motion is similar to a unicycle with limited velocity and turning range (see Fig. 17). RRRobot's leg motions produce primarily pitch and yaw oscillations using different leg offsets and phase differences, while roll oscillations are neg-

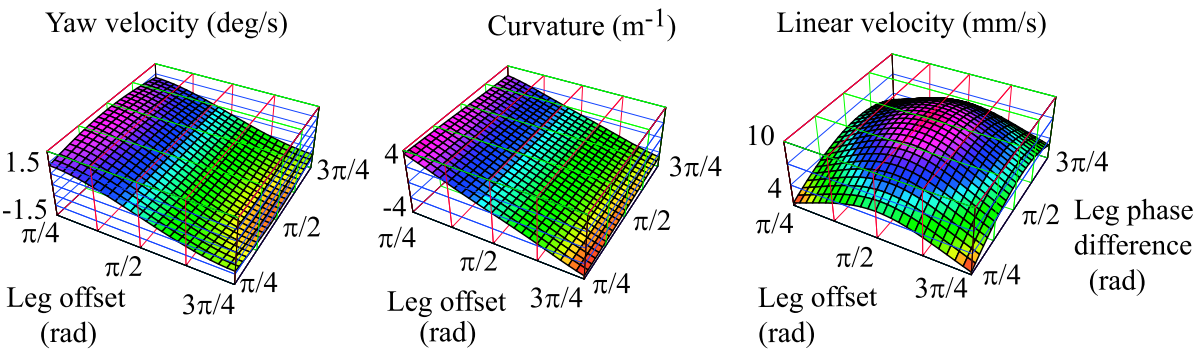


Fig. 18 RRRobot translation as a function of offset and leg phase difference

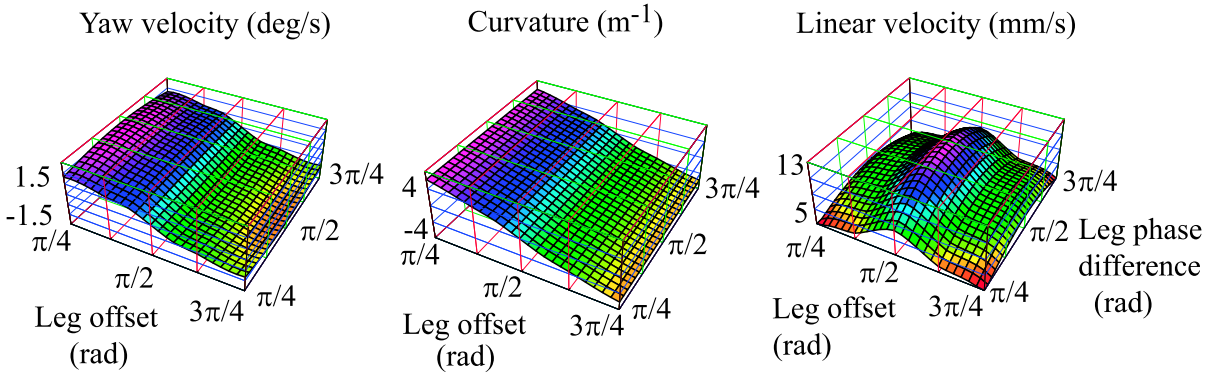


Fig. 19 RRRobot translation as a function of offset and leg phase difference as predicted using the decoupled models (compare with Fig. 18)

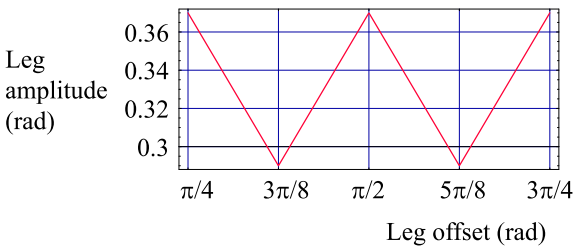


Fig. 20 Amplitude modulation used in the decoupled yaw model

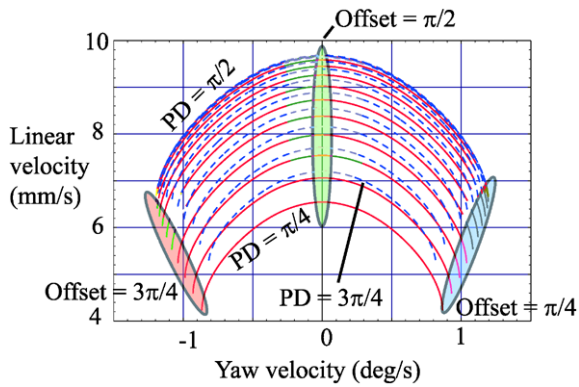


Fig. 21 Mapping between RRRobot linear velocity and yaw velocity and leg offset and phase difference (PD)

ligible. The pitch and yaw oscillations when coupled with the contact constraints produce translation. Inertial differences during out-of-phase leg motions produce yaw drift, which results in translation curvature.

Figures 18 and 19 provide a comparison between the full dynamics model and the single-axis models in terms of translation velocity v, yaw velocity α, and curvature $K = \alpha/v$ across the leg trajectory parameter space including leg offset and phase difference (using fixed leg amplitude 0.3 rad (see caveat below), angular frequency 8 rad/s, and measured at the mean of every periodic cycle). The magnitudes and structure of yaw velocity and curvature match well, but there is a structural difference in the linear-velocity mapping.

This structural difference in the linear-velocity mapping is because of yaw inertia differences between the RRRobot dynamics model and the yaw model (see Sect. 4.2). To overcome this difference, we define the leg amplitude in the yaw model as a function of leg offset to get a favorable comparison in curvature control for the full dynamics model and the single-axis dynamics models (Sect. 5 shows one implementation).

In the decoupled yaw model, the leg motion amplitude is not fixed at 0.3; rather it is defined as a function of leg offset (see Fig. 20). We can use these mappings to derive an inverse relationship for RRRobot control using the full dynamics model (see Fig. 21) and the simplified models (see Fig. 22). Again, there are some discrepancies in the linear velocity mapping; but if we track only path curvature (since RRRobot's linear velocity is small), then the single-axis models provide a good approximation to the full dynamics model. Table 4 provides a summary of how well the approximation techniques predict RRRobot's motion. Clearly, the single-axis models provide a significantly better approximation to the full-dynamics model when compared with the pivoting dynamics model.

This subsection provides a geometrical solution to RRRobot control by finding an approximate mapping between legless-locomotion translation and leg trajec-

Table 4 Comparison of legless locomotion performance as predicted by the approximation techniques with that of the full dynamics simulation (worst case scenarios)

	Rotational dynamics coupled?	Linear travel (compared with full dynamics simulation)	Travel curvature (compared with full dynamics simulation)
Pivoting dynamics	Yes	Overprediction (1.01×)	Overprediction (10×)
Single-axis models	No	Structural difference; overprediction (1.25×)	Underprediction (0.9×)

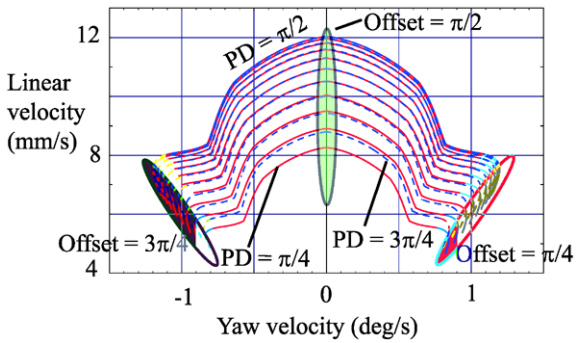


Fig. 22 Mapping between RRRobot linear velocity and yaw velocity and leg offset and phase difference (PD) as predicted by the decoupled models

tories at steady state using dynamics decoupling. Note that we resort to a numerical comparison between the full dynamics and the simplified dynamics, since the structure of dynamics and nonholonomic contact kinematics makes a symbolic comparison difficult.

6 Conclusion

Legless locomotion is a novel locomotion technique that is challenging to analyze using existing dynamics analysis techniques. This paper presents a novel, albeit approximate, dynamics analysis approach where the robot's rotational dynamics and contact kinematics are decoupled. This permits an isolated study of the robot's motion along each rotational axis, allowing us to develop control schemes for each axis separately. In addition, the decoupled models provide a good approximation of how the full dynamics system will evolve when the same control schemes are applied. While this paper explores legless locomotion's mechanics only at steady-state, more work is required to find a generic planning technique for arbitrary motions. It will also be interesting to explore the factors influencing synchronization between the body rotational oscillations in the full dynamics model and the applicability of the decoupled analysis technique to other systems.

Acknowledgements The authors thank Brendan Meeder, Devin Balkcom, Elie Shammas, and Klaus Schmidt from the Robotics Institute at Carnegie Mellon University for providing feedback in this research.

References

1. Balasubramanian, R., Rizzi, A.A., Mason, M.T.: Legless locomotion: A novel locomotion technique for legged robots. Int. J. Robot. Res. **27**, 575–594 (2008)
2. Craig, J.J.: Introduction to Robotics. Addison Wesley, Reading (1989)
3. Murray, J.J., Johnson, D.W.: The linearized dynamic robot model: Efficient computation and practical applications. In: Proceedings of the IEEE Conference on Decision and Control, Tampa, FL, pp. 1659–1664 (1989)
4. Jain, A., Rodriguez, G.: Linearization of manipulator dynamics using spatial operators. IEEE Trans. Syst. Man Cybern. **23**(1), 239–248 (1993)
5. Bullo, F., Lewis, A.D., Lynch, K.M.: Controllable kinematic reductions for mechanical systems: concepts, computational tools, and examples. In: Mathematical Theory of Networks and Systems, Notre Dame, IN, August 2002
6. Walsh, G., Sarti, A., Sastry, S.: Algorithms for steering on the group of rotations. In: The proceedings of the American Control Conference, San Francisco, CA, pp. 1312–1316 (1993)
7. Bullo, F., Lynch, Kevin M.: Kinematic controllability for decoupled trajectory planning in underactuated mechanical systems. IEEE Trans. Robot. Autom. **17**(4), 402–412 (2001)
8. Endo, G., Togawa, K.: S. Hirose. A self-contained and terrain-adaptive active cord mechanism. Adv. Robot. **13**(3), 243–244 (1999)
9. Lewis, A., Ostrowski, J., Murray, R., Burdick, J.: Nonholonomic mechanics and locomotion: the snakeboard example. In: Proceedings of the International Conference on Robotics and Automation, San Diego, CA, vol. 3, pp. 2391–2397 (1994)
10. Krishnaprasad, P.S., Tsakiris. Oscillations, Dimitris P.: SE(2)- snakes and motion control: a study of the roller racer. Technical report, Institute for Systems Research, University of Maryland (1998)

11. Papadopoulous, E., Dubowsky, S.: On the nature of control algorithms for free-floating space manipulators. IEEE Trans. Robot. Autom. **7**(6), 750–758 (1991)
12. Fernandes, C., Gurvits, L., Li, Z.: Near optimal nonholonomic motion planning for a system of coupled rigid bodies. IEEE Trans. Autom. Control **39**(3), 450–463 (1994)
13. Rui, C., Kolmanovsky, I.V., Mc-Clamroch, N.H.: Nonlinear attitude and shape control of space-craft with articulated appendages and reaction wheels. IEEE Trans. Autom. Control **45**(8), 1455–1469 (2000)
14. Chitta, S., Cheng, P., Frazzoli, E., Robotrikke, V. Kumar: A novel undulatory locomotion system. In: Proceedings of the International Conference on Robotics and Automation, Barcelona, Spain, pp. 1597–1602 (2005)
15. Shammas, E.: Generalized motion planning for underactuated mechanical systems. Ph.D. thesis, Carnegie Mellon University (2006)
16. Balasubramanian, R., Rizzi, A.A., Mason, M.T.: Legless locomotion: models and experimental demonstration. In: Proceedings of the IEEE International Conference on Robotics and Automation, New Orleans, LA, pp. 1803–1808 (2004)
17. Uicker, J.J.: Dynamic force analysis of spatial linkages. ASME Trans. J. Appl. Mech. **34**, 418–424 (1967)
18. Kahn, M.E., Roth, B.: The near minimumtime control of open-loop articulated kinematic chains. J. Dyn. Syst. Meas. Control **93**, 164–172 (1971)
19. Luh, J.Y.S., Walker, M.W., Paul, R.P.C.: On-line computational scheme for mechanical manipulators. Trans. ASME, J. Dyn. Syst., Meas. Control **102**(2), 69–76 (1980)
20. Kovecses, J., Piedboeuf, J.-C., Lange, C.: Dynamics modeling and simulation of constrained robotic systems. IEEE/ASME Trans. Mechatron. **8**(2), 165–177 (2003)
21. Lee, C.S.G., Lee, B.H., Nigam, R.: Development of the generalized d'Alembert equations of motion for mechanical manipulators. In: Proc. of the 22nd IEEE Conference on Decision and Control, San Antonio, TX, vol. 22, pp. 1205–1210 (1983). doi:10.1109/CDC.1983.269715
22. Song, S.-M., Lin, Y.-J.: An alternative method for manipulator kinetic analysis-the d'Alembert method. In: Proceedings of the International Conference on Robotics and Automation, pp. 1361–1366 (1988). doi:10.1109/ROBOT.1988.12257
23. Featherstone, D., Orin, D.E.: Robot dynamics: equations and algorithms. In: Proceedings of the IEEE Int. Conf. Robotics and Automation, San Francisco, CA, pp. 826–834 (2000)
24. Montana, D.J.: The kinematics of contact and grasp. Int. J. Robot. Res. **7**(3), 17–32 (1988)
25. Li, Z., Canny, J.: Motion of two rigid bodies with rolling constraint. IEEE Trans. Robot. Autom. **6**(1), 62–72 (1990)
26. Murray, R.M., Li, Z.X., Sastry, S.S.: A Mathematical Introduction to Robotic Manipulation. CRC Press, Boca Raton (1994)
27. Ostrowski, J.P.: The mechanics and control of undulatory robotic locomotion. Ph.D. thesis, California Institute of Technology (1996)
28. Camicia, C., Conticelli, F., Bicchi, A.: Nonholonomic kinematics and dynamics of the sphericle. In: Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Takamatsu, Japan, pp. 805–810 (2000)
29. Bhattacharya, S., Agrawal, S.K.: Spherical rolling robot: a design and motion planning studies. IEEE Trans. Robot. Autom. **16**(6), 835–839 (2000)
30. Laumond, J.P.: Robot Motion Planning and Control. Springer, Berlin (1998)
31. Agahi, D., Kreutz-Delgado, K.: Approximate dynamic decoupling of multilimbed robotic systems. In: Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Pittsburgh, PA, pp. 480–485 (1995)
32. Caccavale, F., Siciliano, B., Villani, L.: The tricept robot: Dynamics and impedance control. IEEE/ASME Trans. Mechatron. **8**(2), 263–268 (2003)
33. Chen, W., Yeo, S.H., Low, K.H.: Modular formulation for dynamics of multi-legged robots. In: Proc. of the International Conference on Advanced Robotics, Monterey, CA, pp. 279–284 (1997). doi:10.1109/ICAR.1997.620195
34. Bullo, F., Lewis, A.D.: Kinematic controllability and motion planning for the snakeboard. IEEE Trans. Robot. Autom. **19**(3), 494–498 (2003)
35. Shammas, E.A., Choset, H., Rizzi, A.A.: Towards a unified approach to motion planning for dynamic underactuated mechanical systems with non-holonomic constraints. Int. J. Robot. Res. **26**(10), 1075–1124 (2007)
36. Shammas, E.A., Choset, H., Rizzi, A.A.: Geometric motion planning analysis for two classes of underactuated mechanical systems. Int. J. Robot. Res. **26**(10), 1043–1073 (2007)
37. Pikovsky, A., Rosenblum, M., Kurths, J.: Synchronization: A universal concept in nonlinear sciences. Cambridge University Press, Cambridge (2001)
38. Jose, J.V., Saletan, E.J.: Classical Dynamics: A Contemporary Approach. Cambridge University Press, Cambridge (1998)
39. Driels, M.: Linear Control Systems Engineering. Mcgraw-Hill College, New York (1995)
40. Bullo, F., Lewis, A.D.: Geometric Control of Mechanical Systems Modeling, Analysis, and Design for Simple Mechanical Control Systems. Springer, New York (2004)
41. Abraham, R., Marsden, J.E.: Foundations of Mechanics. Addison-Wesley, Reading (1978)
42. Balasubramanian, R., Rizzi, A.A.: Kinematic re-duction and planning using symmetry for a variable inertia mechanical system. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, Japan, vol. 4, pp. 3829–3834 (2004)