Evolving Dynamic Maneuvers in a Quadruped Robot

DISSERTATION

Presented in Partial Fulfillment of the Requirements for

the Degree Doctor of Philosophy in the

Graduate School of The Ohio State University

By

Darren Paul Krasny, B.S.E.C.E., M.S.E.C.E.

* * * * *

The Ohio State University

2005

Dissertation Committee:

David E. Orin, Adviser

Stanley C. Ahalt

Kevin M. Passino

James P. Schmiedeler

Approved by

Adviser

Graduate Program in
Electrical and Computer
Engineering

ABSTRACT

In nature, quadrupedal mammals can run at high speeds over uneven terrain, turn sharply, jump obstacles, and stop suddenly. While these abilities are common features in biological locomotion, they represent remarkable feats from both an engineering and control perspective. Although some robots over the past several decades have been capable of dynamic running gaits, none have adequately demonstrated the gallop, the preferred gait of high-speed quadrupedal locomotion for most quadrupeds; and none have been capable of high-speed dynamic maneuvering, especially the turn.

There are several reasons for the lack of progress in high-speed locomotion and maneuvering. First, the concept of dynamic stability, where the system undergoes recurring periods of statically unstable motion, requires a different approach for control that must rely in part on the natural dynamics of the system, while affording only limited time for significant control actions. Furthermore, traditional analytical techniques used in control design typically require simplifying assumptions for both the dynamic model and the gait. To date, these approaches have failed to generate usable results for systems with more biologically motivated designs and for complex, asymmetric high-speed gaits like the gallop. Since high-speed gaits like the gallop remain out of reach, dynamic maneuvering has remained a distant goal, as well. Furthermore, dynamic maneuvers involve significant acceleration and, consequently, large peak power requirements, which appear to be largely beyond the capabilities

of standard actuators. Finally, studying dynamic maneuvers is difficult because they typically occur at high speeds and over highly varying terrain, conditions which are difficult to reproduce in a laboratory environment and equally difficult to manage when studying maneuvers in biological systems.

In this work, a practical approach is developed to study various high-speed locomotion behaviors such as galloping, turning, jumping, and stopping in a simulated quadrupedal model with biologically-based characteristics. A flexible control architecture comprised of low-level primitive functions for each leg is used to create the various behaviors by combining the functions sequentially. A multiobjective genetic algorithm (MOGA) is then used to search for parameter values for these functions, where the search space is minimized by efficient parameterization of each leg-level primitive function.

Results for the 3D gallop, high-speed turn, running jump, jump-start, and sudden stop are presented, which represent new results for a simulated system of this complexity. Furthermore, the resulting behaviors mimic biological motion, providing important data on the underlying mechanics required to achieve this level of locomotion. Dynamic characterization of each behavior promises to facilitate the future development of real-time controllers for galloping and maneuvering in a quadruped robot. A final demonstration of galloping and maneuvering is presented to illustrate the effectiveness of the approach.

To my family

ACKNOWLEDGMENTS

I would like to thank my adviser, Dr. David E. Orin, for his continued support and encouragement throughout my academic career. His quest to go beyond the traditional avenues employed for this type of problem have proved to be both effective and stimulating. Thanks also to Dr. Jim Schmiedeler for his insight and ideas, to Dr. Kevin Passino, whose course on intelligent control has been the foundation of this work, and to Dr. Stanley Ahalt, who, along with Prof.'s Orin, Schmiedeler, and Passino, have kindly agreed to serve on my committee. Thanks also to Dr. Kenneth Waldron for his suggestions and guidance throughout much of my undergraduate and graduate studies.

Thanks also to my cohorts, Luther Palmer, Simon Curran, Brian Knox, and to the students at the Stanford Loco Lab. Frequent discussions on quadrupedal locomotion, research in general, and the plight of the average graduate student have made the trip much more enjoyable.

Most of all, my appreciation goes out to my wife, my family, and my friends for their continued support of my academic pursuits.

Finally, I would like to acknowledge financial support from the National Defense Science and Engineering Graduate Fellowship, the Ohio State University Dean's Distinguished University Fellowship, and the National Science Foundation under Grant No. IIS-0208664.

VITA

October 25, 1972	Born – Marion, OH, U.S.A.
June 1995	B.S. Business Administration, The Ohio State University, Columbus, OH, *Summa cum laude*
September 1995 - June 1996	Staff auditor, Coopers and Lybrand, LLP
June 2000	B.S. Electrical and Computer Engineering, The Ohio State University, Columbus, OH, *Summa cum laude, with distinction*
June 2000 - August 2000	Dean's Distinguished University Fellow, The Ohio State University
September 2000 - August 2003	Department of Defense Fellow, The Ohio State University
June 2002	M. S. Electrical Engineering, The Ohio State University
September 2003 - August 2005	Dean's Distinguished University Fellow, The Ohio State University

PUBLICATIONS

"A 3D Galloping Quadruped Robot," *Proceedings of the 6th International Conference on Climbing and Walking Robots (CLAWAR)*, (London, England), 2005, pp. 467-474 (with D. E. Orin).

"Generating High-Speed Dynamic Running Gaits in a Quadruped Robot Using an Evolutionary Search," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 34, no. 4, pp. 1685-1696, August 2004 (with D. E. Orin).

"Using Hybrid Simulation for a Quadruped Standing Leap," *Proceedings of the 5th World Congress on Intelligent Control and Automation*, (Hangzhou, China), pp. 4603-4608, June 2004 (with W. Hu, D. W. Marhefka, and D. E. Orin).

"Achieving Periodic Leg Trajectories to Evolve a Quadruped Gallop," *Proceedings of IEEE International Conference on Robotics and Automation*, (Taipei, Taiwan), pp. 3842-3848, September 2003 (with D. E. Orin).

"Evolving a Quadruped Gallop," *Video Proceedings, Session II, IEEE International Conference on Robotics and Automation*, (Taipei, Taiwan), September 2003 (with D. E. Orin).

"An Analysis of High-Speed Running Gaits in a Quadruped Robot Using An Evolutionary Optimization Strategy," *Master's Thesis*, The Ohio State University, 2002.

"Establishing Twelve-Axis Control over the WAAV (Wheeled, Actively-Articulated Vehicle)," *Undergraduate Honors Thesis*, The Ohio State University, 2000.

FIELDS OF STUDY

Major Field: Electrical and Computer Engineering

Studies in:

Robotics	Professors D. E. Orin, C. H. Menq
Computer Engineering	Professors F. Özgüner, C. A. Klein, J. E. DeGroat, D. E. Orin, S. C. Ahalt
Control Engineering	Professors K. M. Passino, V. I. Utkin, S. Yurkovich
Dynamics and Kinematics	Professors R. Parker, B. Lilly

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Motivation

The ability to traverse uneven or varying terrain at high speeds, turn sharply, jump over obstacles, and start or stop suddenly are all ordinary aspects of legged locomotion for a variety of cursorial[1] mammals — yet each of these behaviors represents a remarkable feat from an engineering and control perspective. In general, these animals must have robust mechanical systems capable of absorbing significant impacts, storing energy, and generating large amounts of power while maneuvering or changing speeds. Furthermore, they must possess finely-tuned sensory and control systems that can stabilize their bodies while running or maneuvering, both of which can involve recurring periods of statically unstable motion.

While biological locomotion has inspired the creation of a number of legged vehicles over the past several decades, the development of an artificial system with the high-speed performance capability described above has remained elusive. Numerous legged robots have been built [2]-[20], although these machines employed statically

[1]Cursorial mammals stand and run with humerus and femur nearly vertical [1] and include such species as dogs, cats, and horses. Non-cursorial mammals include such animals as mice, squirrels, and ferrets.

stable walking gaits, which are relatively slow and require the vertical projection of the center of mass to remain within the support polygon of the feet at all times. Far fewer legged vehicles have been developed that are capable of dynamic[2] running [21]-[24], although they used lower-speed gaits like the trot and bound. Recently, Smith and Poulakakis [25] demonstrated what appears to be the first rotary footfall sequence reminiscent of the gallop, the preferred gait of high-speed locomotion in cursorial mammals [26]. However, their robot demonstrated no heading control, a fundamental feature of biological locomotion, as it moved in a tight circular trajectory. Consequently, the resulting gait appears to be of limited use in studying high-speed running and maneuvering.

The biological gallop, shown in Fig. 1.1, consists of four separate footfalls, and, almost always, contains at least one significant flight phase called "gathered flight." Occurring after the front legs touch down, the gathered flight phase is so-named because the legs appear gathered under the body. In Fig. 1.1, gathered flight appears as the last image in the bottom rows of (a) and (b). At higher speeds, an additional flight phase called extended flight (Fig. 1.1 (b), top row, rightmost image) may appear after the rear legs touch down. This phase is so-named because the legs are extended from the body. In both examples shown in the figure, the footfall pattern is transverse (e.g., LR-RR-LF-RF, where "LR" means "Left-Rear", etc.), as opposed to the alternate form of the gallop, which consists of a rotary footfall sequence (e.g., LR-RR-RF-LF).

Other predominant features of the biological gallop include minimal height excursion of the center of mass [28] and minimal pitching motion, both of which tend

[2] "Dynamic" refers to the type of stability, where the system may have recurring periods of static instability during which, if the motion were arbitrarily stopped, the robot would fall over.

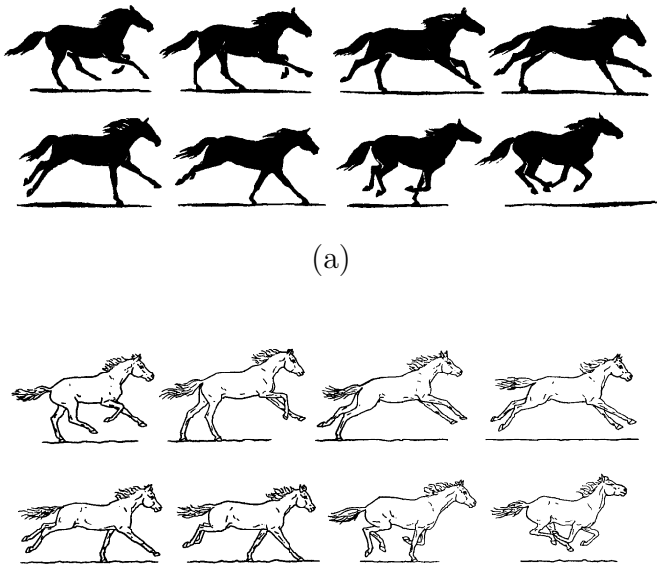


Figure 1.1: The gallop gait: (a) slow (no extended flight), (b) fast (extended flight). Both gallops demonstrate a transverse footfall pattern. Reproduced from [27].

to produce a relatively smooth gait in comparison to other gaits, like the trot or bound [29]. In addition, the gallop also exhibits early leg retraction, where the leg begins retracting (rotating rearward) even before the foot strikes the ground. Although it is not known for certain why early retraction is employed, one theory is that animals deliberately employ it using spatial cues to reduce the relative velocity of the foot with respect to the ground, thereby minimizing impact losses. On the other hand, early retraction could simply be a result of inter-leg phasing that is controlled temporally by some type of internal central pattern generator (CPG) [30]. The latter hypothesis is further supported by the fact that the total cycle time of the legs appears fairly constant across a wide range of galloping speeds [31].

Due to the asymmetric footfall pattern in the gallop, moments are induced about all three of the body's principal axes of rotation. Consequently, the gallop requires simultaneous control in all three planes of motion. In the sagittal plane (the plane of forward motion), height, pitch, and forward velocity must be stabilized. In the transverse and dorsal (horizontal) planes, roll and yaw must be stabilized, while lateral velocity occurs in both planes. Unfortunately, the dynamics of the gallop are also cross-coupled among the multiple planes of motion, which further complicates the required control approach. Furthermore, the lack of symmetry in the gait itself prevents employing traditional control techniques based on simple mechanics [32]. Finally, because the gallop is a dynamic running gait where the system has recurring periods of static instability, it necessarily requires reliance on the system's natural dynamics. However, the hybrid nature of the system, which arises from its continuous dynamics punctuated by discrete flight and stance phases, often yields both chaotic and nonlinear behavior [33]. The number of discrete stance phases for the gallop, as well as any model complexities like leg mass or asymmetric body mass distribution, have, to date, impeded the development of tractable, analytical models for control system design in more realistic systems.

As a result of these features, the gallop has remained largely beyond our capabilities in an actual legged machine. Even in simulation, which could facilitate the development of a hardware solution, a spatial gallop with all of the predominant features of the biological gait has not yet been demonstrated. Nanua and Waldron [34], Ringrose [35], Marhefka et al. [36], Herr and McMahon [30], and Krasny and Orin [37, 38] all investigated the gallop in simulation, although one or more simplifying assumptions were made in each case. Of these, only Ringrose's simulation

was 3D, although he utilized large circular feet for stability, a feature not found in cursorial animals. To date, Herr and McMahon appear to have the most realistic, albeit planar, simulation with the use of biological model parameters, early leg retraction, and phase-locked leg motion. Nanua and Waldron included the limiting assumptions of massless legs and a symmetric body. Marhefka et al. assumed only the latter, although he employed prismatic legs, as did Herr and McMahon. Krasny and Orin employed articulated legs, although the focus was on characterization of the planar gallop on a single-stride basis. Thus, a simulation of a biologically-inspired, 3D gallop using a system with non-trivial, biologically-based assumptions would represent a significant departure from the current state of the art.

Apart from high-speed dynamic running, quadrupedal dynamic maneuvers have received even less attention in the literature. Generally, a dynamic maneuver is a sudden change in trajectory or speed, including turning, dodging, jumping, or stopping suddenly. Dynamic maneuvers involve significant acceleration and are "dynamic" in the twofold sense that (1) the system is dynamically stable during the maneuver itself, and (2) the maneuver typically starts, stops, or interrupts high-speed dynamic locomotion.

Dynamic maneuvers are not well studied for several reasons. First, they represent extreme examples of dynamic stability, which, as stated above, remains a challenging area in its own right. The second reason that maneuvers are not widely studied is that they typically occur at high speeds and over highly varying terrain, conditions that are difficult to reproduce in a laboratory environment and equally difficult to manage when studying maneuvers in biological systems. Finally, maneuvers involve high acceleration rates, which result in significant power consumption. At present,

power is still a significant limitation in legged machines with respect to both actuators and the power sources themselves.

In the few studies on dynamic maneuvers, there is a noticeable lack of in-depth exploration of quadrupedal maneuvering, especially turning. Hodgins and Raibert [39] studied front somersaults with a biped, although the analog is not typically seen in quadrupedal maneuvers. Wong and Orin [40, 41] simulated quadrupedal standing jumps, and running jumps while bounding, although the bound is not typically used at top speeds, and their study was limited to the sagittal plane. Finally, Jindrich and Full [42] studied hexapod turns in cockroaches. However, the stability characteristics of hexapod gaits and turning maneuvers differ significantly from those in quadrupedal locomotion because hexapods can employ an alternating tripod gait that is statically stable even at higher speeds.

1.2 Objectives

In addition to their inherent complexity, both high-speed galloping and dynamic maneuvering remain unmet challenges in robotics largely because traditional analytical techniques developed for hybrid dynamical systems begin breaking down as gait and/or model complexity increase. Berkemeier [43] studied quadrupedal bounding and pronking[3], both of which involve symmetric footfalls. Furthermore, his model was symmetric with massless legs, both of which are unrealistic, yet necessary, assumptions required to derive tractable, analytical return-map equations.

[3]In his work, "bounding" assumed an extended flight phase, while the gait referred to as "pronking" was, according to [27], simply a bound without extended flight. A third gait, the "hop," corresponded to the pronk gait described in the biomechanics literature.

Even in simpler hybrid dynamical systems like the one-legged hopper, the resulting model can become unsolvable analytically when the assumptions are relaxed. For example, M'Closkey and Burdick [44] found that adding forward motion to a simple one-legged hopper with a massless leg precluded them from obtaining a closed-form solution in the return-map analysis. More complex systems, they concluded, would require an experimental approach. However, because development of untested control strategies on an actual hardware implementation is typically impractical, the main thrust of this dissertation is to find an alternative strategy for investigating high-speed galloping and maneuvering using a simulated model, but with biologically-based assumptions.

To achieve this goal, the first objective of this work is to develop a flexible control architecture that can be used to create a variety of complex behaviors, including, but not limited to, the following:

- Galloping,
- Turning,
- Starting,
- Stopping, and
- Jumping.

To avoid the overhead of having to hard-code each new behavior, the architecture is comprised of low-level functions that are implemented for each leg and represent the basic motions and thrusts required for running and maneuvering. The idea of composing complex motions using a library, or vocabulary, of simple primitives is an

appealing approach that avoids having to use specific trajectory planning algorithms for each behavior [45]. This scheme has biological motivation, as well, as evidence suggests that high-level motor patterns in a variety of biological systems are created by the sequencing of simpler motion primitives [46, 47].

Another reason for defining low-level leg control primitives is to achieve a reasonable and unifying method for parameterizing the various target behaviors. For instance, both the gallop and the turn involve similar movements of the legs, although at different angles within the body's transverse plane. Consequently, both of these behaviors can be created with the same general set of leg primitive functions, but with different parameters. Without a uniform set of primitives, however, important commonalities like this would not be revealed. Furthermore, the use of standard control primitives restricts the total number of parameters for each behavior by restricting the type of motions possible. In this way, a more efficient and effective parameterization scheme is created, which facilitates the search for parameter values by minimizing the dimensionality of the search space.

A second objective of this work, then, is to determine the various parameter values for the control primitives comprising each target behavior. Because there are generally too many parameters to tune by hand, and each target behavior has multiple criteria with which to evaluate it, a multiobjective genetic algorithm (MOGA), which uses a vector-valued fitness function, is used for optimization. Thus, one of the tasks for this objective will be to determine which criteria are important in evaluating and analyzing each target behavior. In making this determination, a more fundamental understanding of the underlying mechanics will be gained.

A third objective of this dissertation, which fulfills the overall thrust of this work, will be to analyze and characterize the solutions to each of the target behaviors described above. First, a control methodology will be developed for the gallop which lays the groundwork for more in-depth study in the future. Second, an analysis of the solutions for the remaining behaviors will quantify actuation and control requirements and provide, in several cases, what appear to be new results for quadrupedal dynamic maneuvering in the spatial domain. Hopefully, this will facilitate the future design of an actual quadrupedal robot capable of both high-speed galloping and maneuvering. As a final demonstration, an entire sequence of galloping and dynamic maneuvering will be evolved.

1.3 Organization

In Chapter 2, the dynamic model used throughout this work is presented first, followed by the controller architecture used to create the various target behaviors. A discussion of each of the leg-level primitive functions is then presented, followed by the coordinated functions, which consist of coordinated control among all legs. For both the leg-level and coordinated functions, the complete parameterization is provided, which serves as a reference for the material in later chapters.

In Chapter 3, the evolutionary algorithm is introduced. First, evolutionary multi-objective optimization (EMO) is explained. EMO forms the foundation of the MOGA and is used to optimize the population of vector-valued fitness functions. The MOGA is then described, starting with the non-dominated sorting procedure used to rank individuals according to their Pareto optimality. Following this, the niche-formation strategy employed in the MOGA is presented, which is used to encourage dispersion of

solutions across the Pareto front. The basic genetic operators of selection, crossover, and mutation are presented next, including how these operators are modified for inclusion in the MOGA. Finally, controlled multiobjective elitism is explained, which is used to preserve good solutions and facilitate the multiobjective evolutionary search.

Chapter 4 presents the development of the high-speed 3D gallop controller. The leg-level control primitives, which form a simple state machine, are discussed with respect to the control laws used to compute various parameters. Particular emphasis is placed on discussing the simple energy-based control strategy employed during stance. A complete list of evolved parameters for the gallop is presented, and the multi-stage evolutionary approach is also described. The fitness function criteria used to evaluate potential gallop solutions are discussed next. Finally, the results for the gallop are presented, including intermediate results from each evolution stage, as well as an analysis of the mechanics of the gallop at both the sub-stride and multiple-stride levels.

In Chapter 5, the high-speed turn is described. The turn utilizes the same leg control primitives and sequence as the gallop, although the control parameters are open-loop values that must be found by evolution. The complete list of these parameters is presented, followed by the fitness criteria used to evaluate solutions. Finally, the results across a range of turn angles are analyzed to reveal the basic mechanics of the turn, which are explained in terms of a simple physical model, the conical pendulum.

Chapter 6 presents the solution approaches for the jump-start, the sudden stop, and the running jump. The first and last of these behaviors are evolved in stages, and the parameters and fitness criteria for each stage are presented. The sudden stop

employs only three leg-level control primitives, the most important of which is used during the stance phase and is discussed in detail. Results for all three behaviors are presented, including the unanticipated sliding-style solution obtained for the sudden stop.

Finally, Chapter 7 summarizes the contributions of this work and proposes a list of related future research topics.

CHAPTER 2

CONTROLLER ARCHITECTURE

2.1 Introduction

In order to design high-level behaviors such as galloping or maneuvering, and to assemble these behaviors into a cohesive sequence, a modular, hierarchical control architecture is needed. This chapter describes such an architecture, where leg-level primitive control functions comprise the lowest-level building blocks of each of the target behaviors listed in Sect. 1.2 and represent the basic leg motions necessary for high-speed dynamic running and maneuvering.

The concept of decomposing high-level behaviors into low-level modules has been a popular research area in robotics for at least the past two decades. Brooks' subsumption architecture [48] is generally considered one of the earliest examples of a behavior-based controller. His architecture consisted of multiple layers of control, from simple strategies at the bottom to complex strategies at the top. Higher level control layers could subsume those at lower levels by suppressing their outputs. Although behavior-based controllers can generate stable motion in some legged applications [49], the results can also be unpredictable and not representative of biological

motion. In order to realize more complex biological locomotion, more task-specific motor primitives may be required [50].

Because the goal in this work is to realize biological-mode galloping and maneuvering, an alternative decomposition strategy is used instead of behavior-based control. Here, the basic building block of high-level behaviors is the motor primitive, which is a collection of simple movements that jointly accomplish a basic goal [51, 52]. Biological evidence suggests that complex motions are comprised of low-level motor primitives [46, 47], which has inspired their use in studying human behaviors with a humanoid robot [53, 54]. Instead of layering control primitives in a parallel fashion, however, complex sequences are constructed in this work by combining them sequentially for each leg (but in parallel across all four). This approach eliminates having to explicitly generate a trajectory for each new behavior. Furthermore, it effectively reduces the dimensionality of the parameter space by providing the necessary building blocks for a wide range of target behaviors. Given the dimensionality of the system and the complexity of the target behaviors under consideration, the reduction of parameters is a critical aspect of the control system design.

In the remainder of the chapter, the dynamic model to be controlled is presented first. Following this, the controller architecture is presented, starting with a description of each of the leg-level primitive functions. The coordination level, which exists above the primitive functions, is described next, and includes the STAND, CROUCH, and INIT-TOF functions. Finally, the sequence level of the controller is presented, where individual behaviors can be chained together to realize a series of actions.

m_b	$6.71\,\mathrm{kg}$	I_t *	$4.12\times10^{-4}\,\mathrm{kg\,m^2}$
m_a	$0.61\,\mathrm{kg}$	I_s *	$3.76\times10^{-4}\,\mathrm{kg\,m^2}$
m_t	$0.06\,\mathrm{kg}$	l_f	$0.14\,\mathrm{m}$
m_s	$0.06\,\mathrm{kg}$	l_r	$0.20\,\mathrm{m}$
I_b^{xx}	$0.14\,\mathrm{kg\,m^2}$	d	$0.14\,\mathrm{m}$
I_b^{yy}	$0.19\,\mathrm{kg\,m^2}$	l_t	$0.14\,\mathrm{m}$
I_b^{zz}	$0.16\,\mathrm{kg\,m^2}$	l_s	$0.14\,\mathrm{m}$
I_a^{xx}	$9.87\times10^{-4}\,\mathrm{kg\,m^2}$	l_0	$0.24\,\mathrm{m}$
I_a^{yy}	$9.82\times10^{-3}\,\mathrm{kg\,m^2}$	k_s	$38.00\,\mathrm{N\,m/rad}$
I_a^{zz}	$1.07\times10^{-2}\,\mathrm{kg\,m^2}$	b_s	$0.26\,\mathrm{N\,m\,s/rad}$

*Principal moment of inertia perpendicular to the axial direction of the thigh or shank link. Inertia along the axial direction is negligible.

Table 2.1: Parameters for the dynamic model in Fig. 2.1.

2.2 Dynamic Model

The quadrupedal model used throughout this work is largely based on the physical characteristics of a small dog[4]. The model and joint axes assignment are shown in Figs. 2.1 and 2.2, and the corresponding parameters are listed in Table 2.1. Unlike other studies, the model shown here has mass in each of the leg links, and the body's center of mass (COM) is *not* located at its geometric center. Articulated knee joints are used instead of prismatic joints to better model biological legs. For simplicity, however, the model lacks ankles, and the front and rear knee joints are oriented in the same direction. Because dynamic running gaits utilize compliance in the legs, passive torsional springs are included at the knees. During contact, the knee spring rest position is adjusted instantaneously to inject energy into the knee at maximum

[4]Due to the difficulty in finding data on small dogs, some of the inertial properties of the trunk are based on scaled values from horses [55].

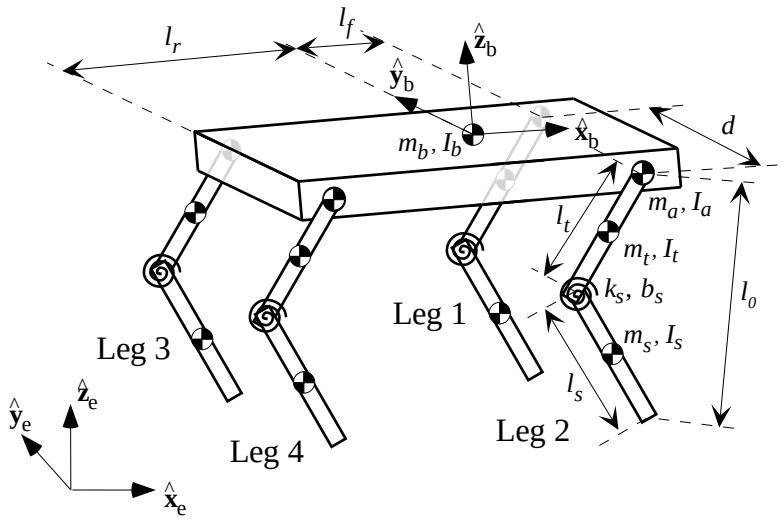


Figure 2.1: Dynamic model of the 3D quadruped with compliance at the knees. l_0 is the nominal, unloaded length of the leg. Parameters are listed in Table 2.1.

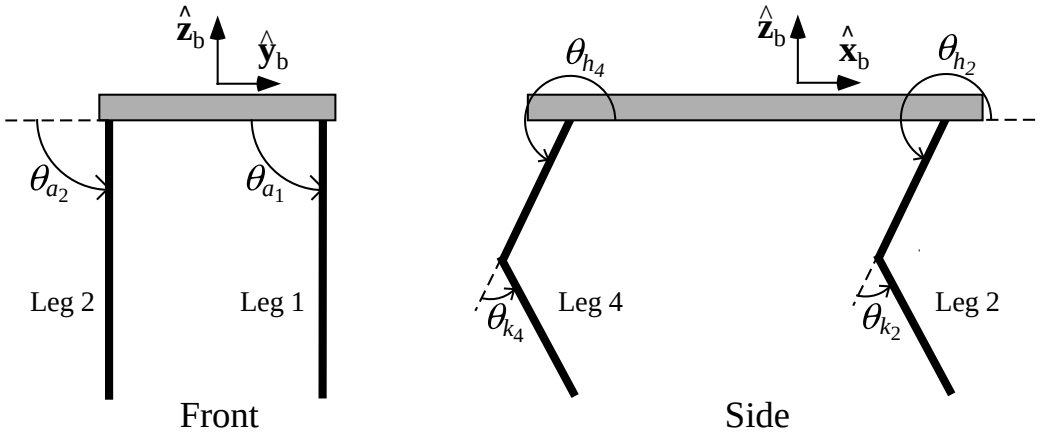


Figure 2.2: Joint axes assignment for the 3D quadrupedal model. Similar joint assignments are used for legs not shown in the diagram.

Joint	Range	
Ab/ad (θ_{a_i})	[0.52, 2.62]	rad
Hip (θ_{h_i})	[−3.14, 3.14]	rad
Knee (θ_{k_i})	[0.17, 2.97]	rad

Table 2.2: Joint limits for each articulated joint.

leg compression. The rest position is smoothly returned to its nominal value after the leg breaks contact.

A compliant contact model is used to compute contact forces acting on the feet. Linear springs (10.0 kN/m) and dampers (50.0 N/m/s) are used in both the vertical and horizontal directions. Slipping is computed by using static and kinetic friction coefficients of $\mu_s = 1.0$ and $\mu_k = 0.8$, respectively, to model the interaction of rubber-coated feet on concrete. Ideal actuators are modeled at each of the abductor/adductor (hereafter, "ab/ad"), hip fore-aft (hereafter, "hip" [5]), and knee[6] joints so that the results of this study are independent of assumptions regarding specific actuator models. Joint limits for each joint are given in Table 2.2, and ideal actuator torque limits are given in Table 2.3. Hip joint limits were set for a wide range of motion, since it was not known in advance what the requirements would be for the various behaviors. Dynamic simulation is implemented using a software package developed for tree-structured robots [56], and Runge-Kutta fourth-order integration is used with a 0.5 ms time step.

[5]Note that the coordinate axes for the ab/ad and hip joints are coincident, although their joint axes are orthogonal.

[6]The passive springs are modeled in parallel with the ideal knee actuators, although backdrive of the actuators is not modeled.

Joint	Range
Ab/ad (τ_{a_i})	[−20.0, 20.0] Nm
Hip (τ_{h_i})	[−20.0, 20.0] Nm
Knee (τ_{k_i})*	[−60.0, 60.0] Nm

*Knee torque limits are for the ideal actuator and exclusive of the knee spring torque.

Table 2.3: Ideal actuator torque limits for each joint.

Yaw, pitch, and roll are defined using the Z-Y-X Euler angle convention described in [57]. Starting from the inertial axes, yaw is defined as the rotation around $\hat{\boldsymbol{z}}_e$ (Fig. 2.1). Pitch is then defined about the rotated axis $\hat{\boldsymbol{y}}_b'$, and roll is defined about the twice rotated axis $\hat{\boldsymbol{x}}_b''$, which is coincident with $\hat{\boldsymbol{x}}_b$.

2.3 Controller

The controller architecture used to design the gallop and dynamic maneuvers is developed in this section. The controller has a hierarchical structure, as shown in Fig. 2.3. Leg primitive control functions exist at the lowest level, which can be combined in sequential fashion to form coordinated functions, shown in the middle level. Coordinated functions can then be combined sequentially to form behavior modules, which, in turn, can be combined to form high-level sequences. The following section discusses the leg primitive functions that comprise the bottom level of the controller.

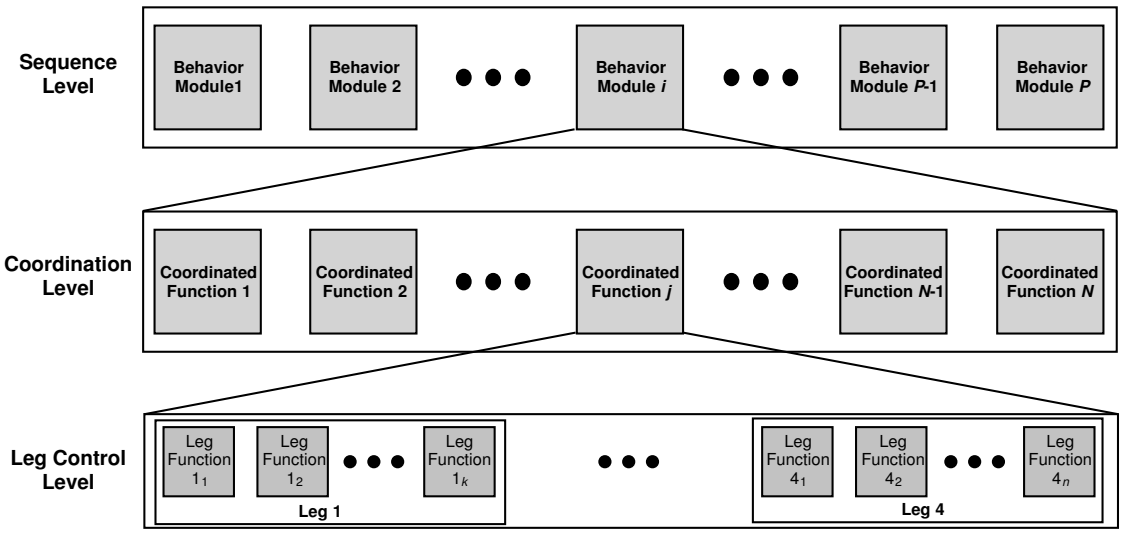


Figure 2.3: Controller architecture for the gallop and maneuvers. Coordinated functions, where all legs operate in a coordinated fashion, are comprised of a sequence of low-level, primitive control functions for each leg. One or more coordinated functions is sequenced to create a behavior module, which may be assembled into a high-level sequence.

2.3.1 Leg Primitive Functions

The leg primitive functions describe the basic motions required for high-speed dynamic running and maneuvering, including transfer, hold, early retraction, thrust, and stance control. Table 2.4 lists all of the leg primitive functions, which are discussed in the following sections.

2.3.2 FREE

The FREE function allows each of the three leg joints to rotate freely for period T. This function is primarily used during flight after the leg breaks contact to prevent excessive jerk on the system, which might occur if significant torque is applied to a joint as it becomes unloaded. The FREE function also contains two binary-valued flags, σ_{a_f} and σ_s. (Hereafter, the symbol σ will be used to denote a binary-valued variable.) The first flag determines whether the ab/ad joint is allowed to rotate freely and provides a mechanism to hold the ab/ad joint at its initial value. This option is useful in restricting the motion of the legs to the sagittal plane during flight.

The second flag, σ_s, is used to determine whether or not the knee spring will be reset during the FREE function. If σ_s is 1, then the knee spring rest position is gradually returned to its nominal position over time $T_r \cdot T$, where $T_r \in [0.0, 1.0]$. Resetting the knee spring rest position is necessary after such functions as GALLOP-STANCE-CONTROL and GALLOP-STOP-CONTROL, where the rest position is instantaneously adjusted at maximum compression to achieve a target energy value. This is discussed further in Sect.'s 4.2.5 and 6.3.1. The variables σ_s and T_r are included in most of the primitive functions that can be employed during flight, as resetting the leg spring rest position is most conveniently done during the flight phase. The

Function	Parameters	Description
FREE	σ_{a_f}, σ_s, T_r, T	• Allow hip and knee to rotate freely for time T. • If binary flag σ_{a_f} is 1, let the ab/ad joint rotate freely; otherwise, hold at initial position. • If σ_s is 1, return the knee spring rest position to its nominal value over time $T_r \cdot T$. • Expire at time T or if leg makes or breaks contact.
HOLD	σ_{a_e}, σ_{h_e}, σ_{k_f}, σ_s, T_r, $\boldsymbol{\theta}_{l_d}$, T	• Hold leg joints at desired positions $\boldsymbol{\theta}_{l_d}$ for time T. • If σ_{a_e} is 1, maintain the desired ab/ad angle with respect to the inertial frame (hereafter, "inertially"). • If σ_{h_e} is 1, maintain the desired hip angle inertially. • If σ_{k_f} is 1, allow the knee joint to be free (i.e., let the knee spring dominate). • If σ_s is 1, reset the knee spring over $T_r \cdot T$. • Expire at T or when leg makes or breaks contact.
TRANSFER	σ_{a_e}, σ_{h_e}, σ_{k_d}, σ_s, T_r, $\boldsymbol{\theta}_{l_f}$, T_t, T	• Transfer all leg joints to $\boldsymbol{\theta}_{l_f}$ over $T_t \cdot T$. • If σ_{a_e} is 1, control the ab/ad joint inertially. • If σ_{h_e} is 1, control the hip joint inertially. • If σ_{k_d} is 1, use a double-spline for the knee. • If σ_s is 1, reset the knee spring over $T_r \cdot T$. • Expire at T or when leg makes or breaks contact.
WAIT-FOR-TRIGGER	L, R, δ, h_t, σ_{a_e}, σ_{h_e}, σ_{k_f}, σ_s, T_r, $\boldsymbol{\theta}_{l_d}$, T	• HOLD using σ_{a_e}, σ_{h_e}, σ_{k_f}, σ_s, T_r, $\boldsymbol{\theta}_{l_d}$, and T. • Terminate if T has elapsed, if trigger R is detected in leg L, or if leg makes contact.

Table 2.4: Leg-level primitive control functions (cont'd on next page).

Function	Params.	Description
EARLY-RE-TRACTION	σ_{a_e}, θ_a, v_d, T	• Hold ab/ad joint at θ_a and knee joint at initial position. • Rotate hip to achieve tangential foot velocity v_d. • If σ_{a_e} is 1, maintain the ab/ad joint inertially. • Expire at T or when leg makes contact.
THRUST	τ_{h_1}, τ_{h_2}, t_{h_1}, t_{h_2}, τ_{k_1}, τ_{k_2}, t_{k_1}, t_{k_2}, T	• Hold ab/ad joint at initial position. • Execute hip torque profile using τ_{h_1}, τ_{h_2}, t_{h_1}, t_{h_2}. • Execute knee torque profile using τ_{k_1}, τ_{k_2}, t_{k_1}, t_{k_2}. • Expire at T or when leg breaks contact.
GALLOP-STANCE-CONTROL	v_d, v_b, E_d, σ_{slip}, T	• Maintain touchdown ab/ad angle using PD control. • Use hip velocity servo with desired tangential foot velocity $v = v_d + v_b$. • Adjust knee spring to achieve E_d energy at maximum compression. • If σ_{slip} is 1, bound ab/ad and hip torques to enforce no-slip condition. • Expire at T or when leg breaks contact.
GALLOP-STOP-CONTROL	k_{p_a}, k_{d_a}, k_{p_h}, k_{d_h}, k_{p_k}, k_{d_k}, E_d, $\boldsymbol{\theta}_{l_d}$, σ_{a_e}, σ_{h_e}, T_s, T_t, T	• If $t < T_s \cdot T$, use PD control for all joints with gains k_{p_a}, k_{d_a}, k_{p_h}, k_{d_h}, k_{p_k}, and k_{d_k}. If $t < 0.01$ or leg is decompressing, disable knee damping. • At maximum compression and if $t < T_s \cdot T$, adjust knee spring to achieve E_d. • If $t \geq T_s \cdot T$, execute TRANSFER using $\boldsymbol{\theta}_{l_d}$, σ_{a_e}, σ_{h_e}, T_t, and T (with TRANSFER parameters $\sigma_{k_d} = \sigma_s = 0$ and $T_r = 0.0$). • Expire at T or when leg breaks contact.

Table 2.4: Leg-level primitive control functions (cont'd).

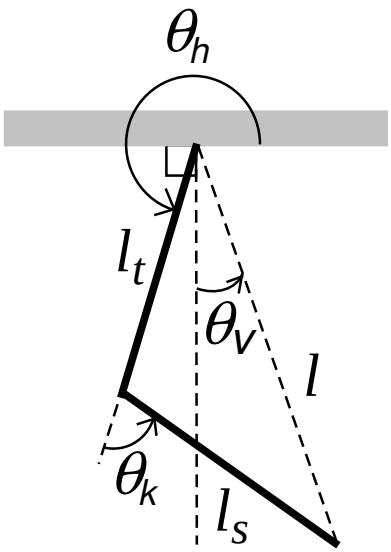


Figure 2.4: The virtual leg control convention. θ_v is the virtual leg angle, measured from the body's normal to the imaginary line connecting the hip to the foot, and l is the virtual length, which is the hip-to-foot distance. l_t and l_s are the thigh and shank lengths in Fig. 2.1, and θ_h and θ_k are the articulated hip and knee joint angles given in Fig. 2.2.

FREE function expires after T or when there is a change in the leg's contact state. This dual-mode termination is also typical of the other functions.

2.3.3 HOLD

During HOLD, each of the leg joints is held at the desired leg position $\boldsymbol{\theta}_{l_d}$, where $\boldsymbol{\theta}_l$ describes the actual leg position as follows:

$$\boldsymbol{\theta}_l = \begin{bmatrix} \theta_a \\ \theta_v \\ l \end{bmatrix}, \tag{2.1}$$

where θ_a is the ab/ad joint angle, θ_v is the *virtual leg angle*, and l is the *virtual leg length*, which are shown with respect to the articulated hip and knee angles θ_h and θ_k in Fig. 2.4. The virtual leg angle is measured from the body's normal to

the imaginary line connecting the hip to the foot, while the virtual leg length is the hip-to-foot distance. The virtual leg length is computed from the articulated knee angle as follows:

$$l = \sqrt{(l_t)^2 + (l_s)^2 + 2l_t l_s \cos\theta_k}\,. \tag{2.2}$$

The virtual leg angle is computed using both the articulated knee and hip angles as follows:

$$\theta_v = \arctan\left(\frac{l_s \sin\theta_k}{l_t + l_s \cos\theta_k}\right) + \left(\theta_h + \frac{\pi}{2}\right)\,, \tag{2.3}$$

where $\theta_h + \frac{\pi}{2}$ biases the virtual leg angle perpendicular to the body's normal axis $\hat{\boldsymbol{z}}_b$ (Fig. 2.1). The joint limits given in Table 2.2 keep the knee angle bounded such that Eq.'s (2.2) and (2.3) remain well-behaved. The virtual leg convention described here effectively transforms an articulated leg to a prismatic leg, which is more convenient for control purposes.

Like the FREE function, the HOLD function also contains several options. The binary-valued variables σ_{a_e} and σ_{h_e} determine whether the ab/ad and hip[7] joints will be maintained at their initial angles with respect to the inertial frame (hereafter, "inertially"). If each variable is zero, then the associated joint is held at the initial angle with respect to the body. These variables provide a convenient means of controlling the leg orientation inertially, which can be used to maintain a certain touchdown orientation with respect to the ground despite any roll or pitch motion of the body. This is an especially useful approach for controlling the legs as the quadruped prepares to land after jumping.

[7]For convenience, and unless otherwise noted, the terms "hip" and "knee" will also refer to the virtual leg angle and length, respectively, as both quantities are related to the hip and knee joints. The virtual leg length is a function of only the knee angle θ_k (Eq. (2.2)), although the virtual leg angle is a function of both hip (θ_h) and knee angles (Eq. (2.3)). For a constant knee angle, however, the virtual leg angle is purely a function of the hip angle.

Parameter	Value	Units	Description
$k^0_{p_a}$	215.5	Nm/rad	Proportional gain for the ab/ad joint.
$k^0_{d_a}$	0.86	Nm/(rad/s)	Derivative gain for the ab/ad joint.
$k^0_{p_h}$	202.3	Nm/rad	Proportional gain for the hip joint.
$k^0_{d_h}$	0.81	Nm/(rad/s)	Derivative gain for the hip joint.
$k^0_{p_k}$	76.0	Nm/rad	Proportional gain for the knee joint.
$k^0_{d_k}$	0.18	Nm/(rad/s)	Derivative gain for the knee joint.

Table 2.5: PD control gains used for each joint.

The binary variable σ_{k_f} determines whether the knee is actively held at its initial position with respect to the thigh or allowed to rotate under the influence of the passive knee spring. This option is useful for maintaining a passive-knee stance, which is more energy-efficient than actively controlling the knee. Like the FREE function, the variables σ_s and T_r can be used to reset the knee spring rest position. Finally, hand-tuned PD gains are used for position control of each joint (after converting from virtual leg coordinates) for the HOLD function, as well as the other leg functions, unless otherwise specified. The PD gains are given in Table 2.5. The parameters were hand-tuned to provide the desired response for each joint and are used throughout the remainder of this work (unless otherwise specified). To compensate for the parallel spring, a feedforward torque for the spring is added to the PD control for the knee joints, as follows:

$$\tau_k = k^0_{p_k}\left(\theta_{k_d} - \theta_k\right) + k^0_{d_k}\left(\dot{\theta}_{k_d} - \dot{\theta}_k\right) + \tau_{\text{ffd}}\,, \tag{2.4}$$

where θ_k and $\dot{\theta}_k$ are the actual knee angle and rate, θ_{k_d} and $\dot{\theta}_{k_d}$ are the desired knee angle and rate, and $\tau_{\text{ffd}} = -\tau_{k_s}$, where τ_{k_s} is the torque generated by the knee spring.

The feedforward term τ_{ffd} is included in all functions where PD position control is used for the knee joints.

2.3.4 TRANSFER

During the TRANSFER function, the leg joints are transferred from their initial positions to the desired position $\boldsymbol{\theta}_{l_f}$ over time $T_t \cdot T$, where $T_t \in [0.0, 1.0]$ specifies the actual proportion of the period used for transfer. Each joint is held at its final value for $T_t \cdot T < t < T$. The transfer is accomplished using a cubic-spline trajectory for each joint (Fig. 2.5). Initial and final rates are assumed to be zero to facilitate modifying trajectories on-the-fly, as is required during galloping. This is discussed further in Sect. 4.2.1. In addition, setting both initial and final rates to zero smooths transitions from one coordinated behavior to the next, which most often occurs when each leg is in the TRANSFER phase. During such a transition, two cubic splines must be connected together seamlessly. Allowing nonzero initial and final rates could lead to unexpected "humps" in the joint trajectory in order to equate the final rate of the previous spline to the initial rate of the next one. These humps could then cause unexpected leg motion and unwanted results. Consequently, all initial and final spline rates are assumed to be zero for more robust chaining and on-the-fly modifications.

As in the HOLD function, the flags σ_{a_e} and σ_{h_e} allow for inertial control of the ab/ad and hip joints. The binary flag σ_{k_d} is used to create a double-spline for the knee joint (Fig. 2.6), which is used to avoid ground interference of the foot during various gaits like the gallop. When the double-spline option is employed ($\sigma_{k_d} = 1$), the target value l_f in $\boldsymbol{\theta}_{l_f}$ (Table 2.4) indicates the *mid-point value* of the spline, while the initial length is also used as the final target length. This approach generates the

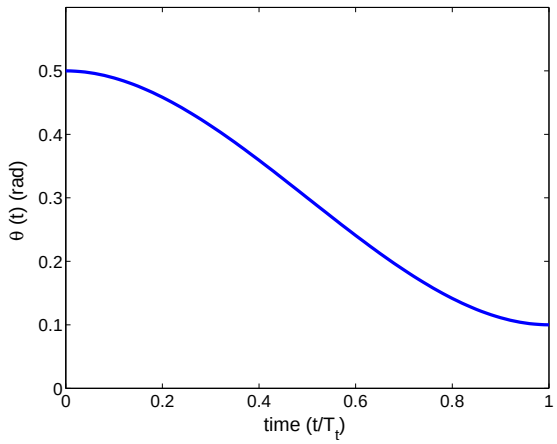


Figure 2.5: Simple cubic spline trajectory used during TRANSFER. Initial and final rates are assumed to be zero. Time is normalized with respect to the transfer time parameter T_t.

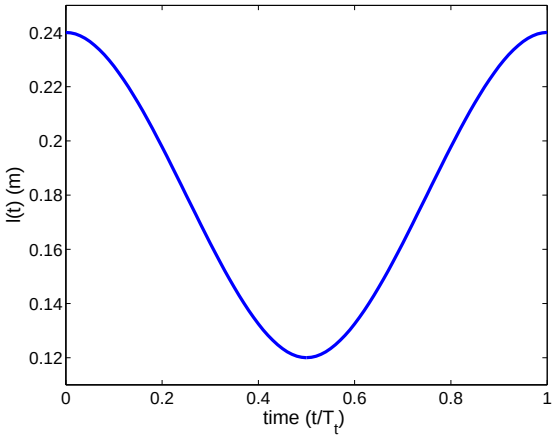


Figure 2.6: Double-spline option used for knee TRANSFER to avoid ground interference of foot during galloping. Time is normalized with respect to the transfer time parameter T_t.

symmetric double-spline function shown in Fig. 2.6. Note that both the hip and the knee splines are specified as virtual leg angle and virtual leg length, respectively. This approach provides a better indication of foot position during transfer as opposed to specifying splines for each of the articulated joints. Finally, σ_s and T_r can be used to reset the knee spring rest position.

2.3.5 WAIT-FOR-TRIGGER

The WAIT-FOR-TRIGGER function is used to synchronize the motion of the legs for galloping and is inspired by Herr and McMahon's approach [30]. Each leg executes a HOLD using parameters σ_{a_e}, σ_{h_e}, σ_{k_f}, σ_s, T_r, and $\boldsymbol{\theta}_{l_d}$ for a maximum time T, at which point the function expires. If the trigger event R is detected in trigger leg L at some $t^* < T$, or if the leg unexpectedly makes contact, the function expires

early, forcing the leg to begin the next control primitive in the sequence. In this work, EARLY-RETRACTION (Sect. 2.3.6) follows WAIT-FOR-TRIGGER[8], which means that the leg begins rotating rearward when the trigger event R is detected.

The trigger event R is a discretely-valued variable taking on one of several types of temporal or spatial triggers, as listed in Table 2.6. Generally, the trigger events can be classified as spatial, temporal, or hybrid. Spatial triggers occur when the trigger leg's foot or hip height falls beneath the threshold height h_t, while temporal triggers occur after a delay δ expires following the touchdown, beginning of retraction, or maximum compression of the trigger leg L.

Hybrid triggers employ temporal trigger events with a spatial safety mechanism. For example, the ER-FOOT-HEIGHT trigger occurs only after the delay δ has expired following the beginning of retraction of trigger leg L *and* the foot height of the *current* leg is within h_t height of the ground. This prevents the leg from triggering if the quadruped's body is pitched or rolled, such that retracting the leg would result in missing the ground or touching down at an excessive angle. More is said about this type of trigger in Sect. 4.2.3.

2.3.6 EARLY-RETRACTION

During EARLY-RETRACTION, the leg is rotated rearward at desired tangential velocity v_d for maximum period T, where v_d is typically set to the desired running velocity. The leg length is kept constant at the nominal length l_0 given in Table 2.1. EARLY-RETRACTION is used to reduce impact losses by minimizing the relative

[8]In the event of an unexpected contact, WAIT-FOR-TRIGGER transitions to EARLY-RETRACTION, which will also immediately transition to the next function, which is typically a stance-phase function.

Trigger Type	Value	Description
FOOT-HEIGHT	0	Trigger occurs when the foot height of trigger leg L^* in inertial coordinates falls beneath the specified threshold height h_t.
TD-DELAY	1	Trigger occurs delay δ after trigger leg L touches down.
MAX-COMPRESS-DELAY	2	Trigger occurs delay δ after trigger leg L reaches maximum compression.
ER-DELAY	3	Trigger occurs delay δ after trigger leg L begins EARLY-RETRACTION.
HIP-HEIGHT	4	Trigger occurs when trigger leg L's hip height in inertial coordinates falls beneath the specified threshold height h_t.
ER-FOOT-HEIGHT	5	Trigger occurs when the latter of two events occur: 1. Delay δ has expired since trigger leg L has begun retracting, or 2. The current leg's foot height has fallen beneath the threshold height h_t.
ER-HIP-HEIGHT	6	Same as ER-FOOT-HEIGHT, except that the hip height is compared to the threshold height h_t.
TD-FOOT-HEIGHT	7	Trigger occurs when the latter of two events occur: 1. Delay δ has expired since trigger leg L has touched down, or 2. The current leg's foot height has fallen beneath the threshold height h_t.
TD-HIP-HEIGHT	8	Same as TD-FOOT-HEIGHT, except that the hip height is compared to the threshold height h_t.

*Note that L can be the current leg.

Table 2.6: Enumerated trigger types R used in the WAIT-FOR-TRIGGER function.

tangential velocity of the foot with respect to the ground. In addition, EARLY-RETRACTION is responsible for maintaining phase offsets between the legs during galloping, which is discussed in Sect. 4.2.4.

The flag σ_{a_e} determines whether or not the ab/ad joint is maintained inertially or with respect to the body. In either event, θ_a specifies the target angle in the specified coordinate frame. The EARLY-RETRACTION function expires when the period T has elapsed or the leg has made contact with the ground.

2.3.7 THRUST

While the previous leg functions are typically used when the leg is in flight, the THRUST function is used primarily during contact. This function executes separate open-loop torque profiles for the hip, using τ_{h_1}, τ_{h_2}, t_{h_1}, and t_{h_2}, and knee, using τ_{k_1}, τ_{k_2}, t_{k_1}, and t_{k_2}. The ab/ad joint is held at its initial position thoughout the profile. An example of a torque profile for one joint is given in Fig. 2.7. The parameter τ_0, which is not a configurable parameter, is equal to the torque that was commanded at the joint just prior to calling the THRUST function. This provides a simple method of holding the joint from the start of the THRUST function until time t_1. This is especially useful for the knee joints, where the ideal actuator is in parallel with the spring, and larger torques are necessary to maintain compression of the spring. During a crouching leap (Sect. 6.2.1), for example, the t_{k_1} parameter for each of the knee joints is typically nonzero so that the knee springs remain compressed for a short time before being released, which is accomplished by nulling the torque or reversing its direction.

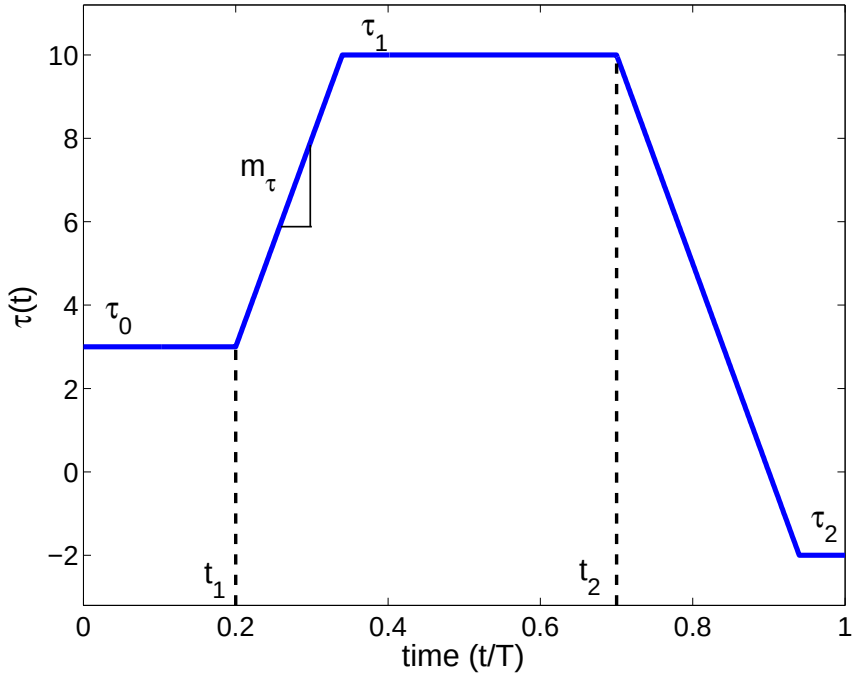


Figure 2.7: Sample torque profile for one joint for the THRUST function. τ_0 is the most recent torque commanded at the joint prior to the THRUST function. τ_1, τ_2, t_1, and t_2 must be specified, where t_1 and t_2 are normalized with respect to the period T and must be in the range $[0.0, 1.0]$. The torque-slope m_τ is a constant parameter that determines how quickly torque levels can change.

Because the profile for each joint has four DOF (degrees of freedom), there are numerous possibilities. By setting $t_1 = 0.0$ and $t_2 = 1.0$, for example, a constant torque value is commanded for the entire period T. On the other hand, it is possible to realize a pulse by setting t_2 close to the value of t_1[9]. The length of the pulse, as well as the time required to change torque levels, is governed by the torque-slope constant m_τ, which is set to 500.0 Nm/s. The flexibility offered by the THRUST function allows for the design of many different types of leaps and landings. The THRUST function is terminated when period T expires or when the leg breaks contact.

2.3.8 GALLOP-STANCE-CONTROL and GALLOP-STOP-CONTROL

Both GALLOP-STANCE-CONTROL and GALLOP-STOP-CONTROL are specialized functions employed during the stance phases of galloping and the sudden stop, respectively. Both employ PD control for the ab/ad and hip joints, and both involve adjusting the knee spring rest position at maximum compression to achieve a specified target energy value. Furthermore, both functions can also be terminated when the leg breaks contact, similar to the THRUST function. Beyond these common elements, each function is specialized for stance-phase control during their respective behavior modules. Consequently, a more detailed discussion is deferred until Sect.'s 4.2.5 and 6.3.1, respectively.

2.4 Coordination Level

The leg-level control primitives discussed in the previous section can be assembled sequentially for each leg, and in parallel across all four legs, to form a higher

[9]Note that the condition $t_1 < t_2$ is enforced by the THRUST function by sorting them in ascending order for each joint.

level coordinated function. Coordinated functions can then be added to the library, or vocabulary, of available functions and used to design even higher level behavior modules. Typically these functions are hard-coded in more compact form once the initial development is completed using primitive functions. Throughout this work, coordinated functions and behavior modules are equivalent, since only a single coordinated function is currently used in each of the behavior modules investigated. However, the design strategy of incorporating multiple coordinated functions into a behavior module should provide greater flexibility when creating high-level behaviors. The list of coordinated functions is given in Table 2.7. In the following sections, the coordinated functions in Table 2.7 are presented. Although the gallop, turn, jump-start, running jump, and sudden stop were coded as coordinated functions after their development, they are discussed in later chapters in terms of the low-level primitives they employ, which provides greater insight about their operation.

2.4.1 STAND

The STAND function is used to maintain a nominal, passive-knee stance using desired ab/ad and virtual leg angles given in ${}^{e}\boldsymbol{\theta}_{a}$ and ${}^{e}\boldsymbol{\theta}_{v}$, both of which are given with respect to the inertial frame. Each leg executes a HOLD function to maintain the specified angles for the ab/ad and hip joints, which means that the user is responsible for ensuring that the initial configuration and the specified angles do not create conflicts among the joints. In the nominal configuration, with the feet directly under each of the hip joints, the STAND function performs as expected on level terrain. However, using asymmetric hip and/or ab/ad angles has not been tested, as

Function	Parameters	Description
STAND	${}^e\boldsymbol{\theta}_a$, ${}^e\boldsymbol{\theta}_v$, T	• Use PD control on the ab/ad joints using desired inertial positions in ${}^e\boldsymbol{\theta}_a$ for time T. • Use PD control to maintain the desired inertial virtual leg angles in ${}^e\boldsymbol{\theta}_v$ for time T. • Use no actuator torque at the knee joints for a passive-knee stance.
CROUCH1	l_f, l_r, ${}^e\theta_{v_f}$, T_c, T	• Execute a planar crouch from the initial stance position over time $T_c \cdot T$ using the TRANSFER function for each leg to achieve virtual leg length l_f in the front legs and l_r in the rear. ${}^e\theta_{v_f}$ is the target inertial virtual leg angle for the front legs, while that of the rear legs is left unspecified by zeroing the rear hip torque throughout the function. • Use PD control to maintain initial ab/ad joint angles. • For $t > T_c \cdot T$, hold all leg joints (except the rear hip joints) at their target values.
CROUCH2	l_f, l_r, ${}^e\theta_{v_f}$, ${}^e\theta_{v_r}$, T	• Initialize the quadruped in a planar crouch with front leg length l_f, rear leg length l_r, front inertial virtual leg angle of ${}^e\theta_{v_f}$, and target rear inertial virtual leg angle ${}^e\theta_{v_r}$. • Use static analysis to initialize the quadruped such that the feet are penetrating the ground at the appropriate depth based on the kinematic configuration of the quadruped and the known ground compliance values.

Table 2.7: Coordinated control functions (cont'd on next page).

Function	Params.	Description
INIT-TOF	σ_p, σ_{a_e}, σ_{h_e}, σ_{k_d}, σ_{l_a}, $\boldsymbol{\theta}_b$, $\boldsymbol{\delta}_b$, $\boldsymbol{\phi}_l$, $\boldsymbol{\theta}_l$, $\boldsymbol{t}^*$, T_t	• Initialize the quadruped at the top-of-flight (TOF) point of its flight phase using body state $\boldsymbol{\theta}_b$. • If σ_p is 1, perturb each element of the body state randomly using the ranges specified in $\boldsymbol{\delta}_b$ such that $\boldsymbol{\theta}'_b = \boldsymbol{\theta}_b + \boldsymbol{\Psi}\boldsymbol{\delta}_b$, where $\boldsymbol{\Psi}$ is a diagonal matrix containing random values on the interval $[-1.0, 1.0]$. • Compute cubic transfer splines for each leg using $\boldsymbol{\phi}_l$, which contains the starting and final positions for all joints (initial and final rates are assumed to be zero, as stated in Sect. 2.3.4). • If σ_{l_a} is 1, initialize each joint using the joint states specified in $\boldsymbol{\theta}_l$. Otherwise, initialize each joint by computing its position and velocity along the spline using each leg's spline completion time given by $\boldsymbol{t}^* = [t_1^*, t_2^*, t_3^*, t_4^*]^T$, where each time is normalized with respect to the transfer period T_t. • Compute the ab/ad joint's transfer spline with respect to the inertial frame if σ_{a_e} is 1, and compute the hip spline inertially if σ_{h_e} is 1. • If σ_{k_d} is 1, use a double-spline for the knees.

Table 2.7: Coordinated control functions (cont'd).

the STAND was developed primarily as a nominal starting configuration for various behaviors where the quadruped is initialized on the ground.

Although the STAND function is designed such that the quadruped starts in contact with the ground, the dynamic simulation software, DynaMechs [56], assumes that links do not start in contact. If any of the links does start in contact, the result is a significant step-change in ground contact forces acting on the link during the first integration step of the simulation. This usually causes stability problems with the STAND function, so the quadruped is normally initialized with its feet just slightly above the terrain so that it actually makes contact shortly after the simulation is started. Under the nominal configuration and using properly tuned gains for the ab/ad and hip joints, the STAND function is able to stabilize the system into a nominal stance position in spite of this initialization method. Because of the knee compliance, however, the system may take several seconds to quit oscillating, which can lead to excessive simulation time when performing the repetitive testing required to evolve behaviors with ground-based initial configurations. To address this problem, the CROUCH2 function was developed, as described in Sect. 2.4.3, below.

2.4.2 CROUCH1

The CROUCH1 function was developed to follow the STAND function described above. During CROUCH1, which is basically a planar maneuver with ab/ad stabilization, the quadruped physically crouches to achieve a front virtual leg length of l_f and a rear length of l_r. The angle ${}^e\theta_{v_f}$ is the target front virtual leg angle with respect to the inertial frame. The rear angle is not specified to avoid the closed-loop kinematic computations, and the rear hip joint torques are zeroed throughout the

function to prevent them from fighting other joints. The crouch occurs during the first T_c proportion of the function using the TRANSFER function. At $t > T_c \cdot T$, each joint (except the rear hip joints, which are free) is held in its final position using the HOLD. The CROUCH1 function was developed primarily to provide a good configuration from which to execute power strokes for leaping using the THRUST function. However, due to the time required to execute the CROUCH1 and to stabilize the STAND function, a faster method of initializing the quadruped was developed, as explained next.

2.4.3 CROUCH2

The CROUCH2 function is a planar method that initializes the quadruped in the desired crouching configuration using front and rear virtual leg lengths l_f and l_r and front and rear virtual leg angles ${}^e\theta_{v_f}$ and ${}^e\theta_{v_r}$, where both angles are given with respect to the inertial frame. Unlike the STAND and CROUCH1, however, the CROUCH2 uses kinematic and static analysis to initialize the quadruped in contact with the ground assuming steady-state conditions. As a result, it is possible to initialize and hold the quadruped in the desired configuration without expending excess time waiting for the system to stabilize or to execute the crouch itself.

For the repetitive testing required to evolve ground-based leaping maneuvers, this time savings can be significant. For example, if it takes 2.5 s of simulation time for the quadruped to stabilize during the STAND function, then execute a CROUCH, evolving 32 individuals over 100 generations and 20 trials, assuming a three-to-one ratio between simulation vs. computation time[10], would take 14.8 hours — all of which could be eliminated by using the CROUCH2 function.

[10]On a Pentium IV, 2.4 GHz, the ratio appears to be about 3.25 : 1.0.

Although CROUCH2 is more efficient with respect to computation time than both STAND and CROUCH1, the latter functions provide a means of investigating realistic motion profiles for standing and crouching that would not otherwise be possible using the abbreviated CROUCH2 function. Furthermore, both STAND and CROUCH1 offer the potential to investigate both behaviors using uneven terrain, which the CROUCH2 function is not designed to handle. Consequently, the STAND and the CROUCH1 functions remain important parts of the library for future use.

2.4.4 INIT-TOF

The INIT-TOF coordinated function is a special function because it is not actually composed of leg-primitive functions. Instead, this function was designed as a convenient means of initializing the quadruped at a known configuration at the apex of the body's flight phase, referred to as the top-of-flight (TOF[11]). The vector $\boldsymbol{\theta}_b$ specifies the body state position, orientation, and their rates as follows:

$$\boldsymbol{\theta}_b = \left[\alpha,\, \beta,\, \gamma,\, {}^{e}x_b,\, {}^{e}y_b,\, {}^{e}z_b,\, \dot{\alpha},\, \dot{\beta},\, \dot{\gamma},\, {}^{e'}\dot{x}_b,\, {}^{e'}\dot{y}_b,\, {}^{e}\dot{z}_b\right]^T , \tag{2.5}$$

where α, β, and γ are the yaw, pitch, and roll, respectively, which are defined using the Z-Y-X Euler angle convention discussed in Sect. 2.2, ${}^{e}x_b$, ${}^{e}y_b$, and ${}^{e}z_b$ define the position of the body's COM in earth coordinates, $\dot{\alpha}$, $\dot{\beta}$, and $\dot{\gamma}$ are the yaw rate, pitch rate, and roll rate, respectively, ${}^{e'}\dot{x}_b$ and ${}^{e'}\dot{y}_b$ are the forward and lateral velocities, both defined in the frame $\{\boldsymbol{e}'\}$, which is obtained by rotating the inertial frame $\{\boldsymbol{e}\}$ by α about $\hat{\boldsymbol{z}}_e$ (Fig. 2.1), and ${}^{e}\dot{z}_b$ is the vertical velocity in earth coordinates. Note that ${}^{e}\dot{z}_b$ is typically set to zero, since the goal is to initialize the quadruped at the apex of its flight phase. However, this value may also be nonzero for more accurate

[11]Later, the top-of-flight point is qualified with respect to leg contacts such that TOF refers to the top of gathered flight. For now, however, TOF will refer to the apex of any general flight phase.

initialization of simulated TOF conditions, where numerical integration and machine precision generally preclude achieving precisely zero vertical velocity at the detected TOF.

If σ_p is 1, then the body state variables are perturbed randomly using the following equation:

$$\boldsymbol{\theta}_b' = \boldsymbol{\theta}_b + \boldsymbol{\Psi}\boldsymbol{\delta}_b\,, \tag{2.6}$$

where $\boldsymbol{\delta}_b$ contains the perturbation range for each variable in $\boldsymbol{\theta}_b$ and $\boldsymbol{\Psi}$ is a diagonal matrix with random numbers drawn from $[-1.0, 1.0]$ on the diagonal. This feature allows for random initialization of the body and is used to evolve the sudden-stop maneuver described in Sect. 6.3.

Because the legs are almost always in mid-TRANSFER when the body is at TOF, the INIT-TOF function provides a mechanism to initialize the legs at either arbitrary values or at specified points along their cubic transfer splines. The transfer spline parameters are specified in the vector $\boldsymbol{\phi}_l$, which contains the initial and final positions for all twelve leg joints in the system. This is sufficient to compute the associated cubic transfer splines since initial and final joint rates are always assumed to be zero (Sect. 2.3.4). If σ_{l_a} is 1, then the leg joints are initialized using the angles and rates provided in $\boldsymbol{\theta}_l$. Otherwise, the leg joint angles and rates are computed from their respective cubic splines using the spline completion times given in $\boldsymbol{t}^*$, which are normalized with respect to T_t. Note that $\boldsymbol{t}^*$ contains four normalized times, since leg joints are coordinated at the leg level during TRANSFER. The binary flags σ_{a_e} and σ_{h_e} determine whether the ab/ad and hip joint angles, respectively, are computed inertially. Finally, the variable σ_{k_d} determines whether the knee joint transfer splines are double-splines or standard single-splines.

Like the CROUCH2 function, the INIT-TOF function is designed to provide a fast and efficient way to initialize the quadruped. Unlike CROUCH2, however, INIT-TOF is used to initialize target behaviors that originate during *flight*. INIT-TOF is used to initialize the gallop, turn, running jump, and sudden stop, since all of these behaviors are most conveniently initialized at TOF points. As the objective of the INIT-TOF function is purely initialization, the period is defined to be zero, and, consequently, INIT-TOF is executed in zero simulation time, unlike the other coordinated functions. INIT-TOF is discussed further in Sect. 4.2.9.

2.5 Sequence Level

The top level of the controller architecture (Fig. 2.3) is the sequence level. At this level, previously evolved behavior modules can be assembled sequentially to form a complete sequence. For example, as a final demonstration of the power of the approach described in this dissertation, a complete sequence of galloping and maneuvering is evolved, reminiscent of an equestrian trial. The results of this sequence are presented in Sect. 6.5. Significant expansion possibilities exist for this layer, as well. For example, a high-level supervisory controller could be developed to select various behavior modules in real time depending on stimuli from the environment.

2.6 Summary and Conclusions

In this chapter, the controller architecture used to develop the gallop and the dynamic maneuvers was presented. The architecture is comprised of leg primitive control functions at the bottom level, which can be assembled in sequential fashion to achieve higher level behaviors. Each of the leg primitive functions was presented,

and collectively, they represent the basic motions and thrusts found in high-speed dynamic running and maneuvering, as well as more general behaviors.

The use of primitive motor functions to define a library of behaviors is appealing in several ways. First, this scheme provides the basic building blocks to facilitate the design of numerous behaviors. Second, it reduces the size of the parameter space, which facilitates the use of optimization methods to locate parameters that cannot be tuned by hand. Third, using motor primitives provides for a higher degree of reuse, obviating the need to re-code each new behavior from scratch or implement complex trajectory-generation schemes.

Leg primitive functions can be assembled sequentially to form coordinated functions, where the motion of all legs is controlled simultaneously. The main coordinated functions were explained in this chapter, where the CROUCH2 and INIT-TOF functions were described as being highly optimized for efficient ground- and flight-based initialization, respectively, which facilitates the type of repetitive testing required to evolve the various target behavior parameters. The genetic algorithm used to find parameter values for each of these target behaviors is presented in the next chapter.

CHAPTER 3

THE EVOLUTIONARY ALGORITHM

3.1 Introduction

The genetic algorithm (GA) is a powerful optimization approach that can perform well in large-dimensional, nonlinear, and, often, largely unknown, parameter spaces. A direct-search optimization approach, the GA is based on Darwinian evolution and makes use of the genetic operators of selection, crossover, and mutation to modify a population of individuals over several generations to maximum their fitness with respect to a given objective function [58].

The selection operator rewards individuals with higher fitness values by allowing them to enter a mating pool, where genetic material from each pair of individuals is combined, via crossover, to produce a pair of offspring, much like biological reproduction. During the mating process, the genetic material of each offspring is also subject to random variation through the mutation operator. As a result of these processes, each successive generation is altered to optimize performance with respect to the fitness function.

In the field of robotics, evolutionary algorithms, a broader class of algorithms that employ a subset of the evolutionary operators [58], have become a popular tool for

developing complex behavior. In fact, a new subfield within robotics has emerged, called evolutionary robotics. However, much of the work in this area has focused on evolving complex behaviors for relatively simple, statically stable, wheeled robotic vehicles [59]-[61]. Much of this research also falls under the category of artificial life, where evolutionary methods are used on simulated or robotic agents to develop complex adaptive behaviors in an artificial environment to study the behavior of biological systems [62].

Although evolutionary robotics studies are useful because they examine the development of complex, adaptive behavior, they have not been used extensively for control development in legged systems, especially those that are dynamically stable. In dynamically stable legged systems, there appears to be a significant bootstrapping problem, where only a small portion of the parameter space may actually lead to stable behaviors. As such, the majority of the effort is geared to simply locating this part of the space. In most evolutionary robotics studies, however, physical stability of the system is guaranteed, which leaves open the possibility to explore a wide range of complex, often emergent, behaviors using generic control structures that are tuned using on-line adaptation. While such an approach might be adapted for dynamically stable legged systems, there have been no known studies that have successfully done so.

Instead, evolutionary algorithms have been applied as a basic optimization strategy to develop stable control strategies for legged robots, but using specially designed control architectures. Still, most of these studies have focused on developing statically stable walking gaits. For example, Gallagher et al. [63] evolved parameters for a neural oscillator network to develop a statically-stable hexapod walking gait. Gomi

and Ide [64] evolved walking gaits for an octopod robot using a specially designed genome that encoded the various motions, state, and timing of each leg. Finally, Hornby et al. [65] evolved various walking gaits with the Sony quadruped AIBO using parameterized modules capable of producing different gaits. In each of these cases, significant effort was made in designing control architectures that were well-suited to generate usable, statically-stable locomotion.

Using an evolutionary algorithm for a dynamically stable legged robot seems to be a relatively new, but logical, extension to existing techniques. Furthermore, using evolutionary optimization with low-level motor primitives addresses the problems of requiring a specific control architecture for locomotion, the need for flexibility in defining numerous behaviors, and the necessity of minimizing the number of tunable parameters. Finally, the use of evolutionary multiobjective optimization (EMO), where fitness is represented as several, possibly competing objectives, addresses the inherent difficulty in trying to design a single, scalar-valued objective function that adequately represents the characteristics of the target behavior.

In the sections that follow, multiobjective optimization is presented first, to provide necessary background for understanding the multiobjective genetic algorithm (MOGA), which is introduced next. The non-dominating sorting procedure used by the MOGA to rank individuals is described after this, followed by the niche-formation method, which encourages dispersion of solutions across the Pareto front. A discussion of the genetic operators of selection, crossover, and mutation follows. Controlled multiobjective elitism, which preserves good solutions and improves the performance of the algorithm, is described next. Lastly, a summary of the chapter is given.

3.2 Multiobjective Optimization

Multiobjective, or Pareto, optimization involves the use of a vector-valued objective variable, as opposed to a scalar one. Such an approach addresses the difficulty in many problems of deriving a single value to determine the quality of solutions. In particular, the use of a single value typically requires a weighting scheme with multiple factors. Unfortunately, optimization approaches are generally very sensitive to the choice of weights, making it difficult to design an adequate fitness function [66].

In multiobjective optimization, each element of the vector is considered equally important, and determining optimality involves locating those solutions that are *non-dominated* [67]. The concept of domination can be best explained by a simple example. Consider two vectors with elements $[1, 3, 5]^T$ and $[1, 2, 5]^T$. The first is said to dominate the second, since it is at least as good as the second in every element and better in at least one. However, if a third vector $[2, 1, 5]^T$ is introduced, then the first and third vectors are non-dominated with respect to each other, as are the second and third vectors. Note, however, that the first and second vectors can both be non-dominated with respect to a third vector, yet *not* be non-dominated with respect to each other. Thus, there is no transitive property for non-domination. This means that finding all non-dominated solutions requires a comparison between every individual in the set, which is discussed further in Sect. 3.3.3.

In a set of solutions, those solutions that are non-dominated represent the trade-off surface, or Pareto front [67, 68]. In two or three dimensions this surface can be thought of as the outer-most extents in fitness space, where the objective is to maximize each element of the fitness vector. An example is shown in Fig. 3.1. Each individual on the Pareto front is non-dominated with respect to all other individuals.

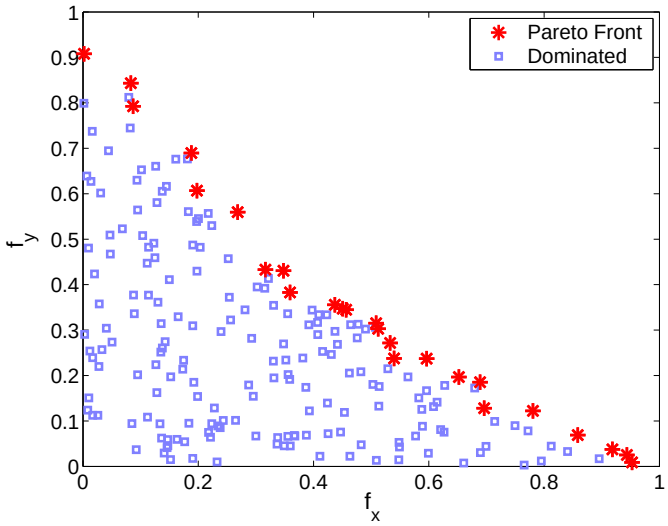


Figure 3.1: An example of a Pareto front in two-dimensional fitness space.

Starting at the x-axis intercept and moving along the front, the fitness f_x generally decreases as the fitness f_y increases, illustrating the trade-off between f_x and f_y.

3.3 Multiobjective Genetic Algorithm

The multiobjective genetic algorithm (MOGA) is based on the standard GA with the exception that the computation of fitness must be modified to handle vectors instead of scalars. In particular, vector fitness values generate two immediate problems in evolutionary multiobjective optimization (EMO). First, a special sorting algorithm must be used to assign some type of rank to each individual. The method employed here is called non-dominated sorting, which is discussed in the next section. Second, since multiple solutions will be found, it would be optimal if those solutions were evenly spread across the Pareto front. Unfortunately, the MOGA is prone to a condition called genetic drift, where solutions tend to congregate, or cluster, around

individual local optima [69]. Niche formation methods, which encourage the formation of multiple optima, are used to minimize this phenomenon. Details of the niche-formation strategy used in this work are presented in Sect. 3.3.4. To preserve good solutions, controlled, multiobjective elitism is employed, which is discussed in Sect. 3.3.8.

Before discussing the MOGA, a few terms must be introduced. First, an "individual" refers to a single solution $\boldsymbol{\phi}^j$ which is comprised of a vector of p parameters, or genes:

$$\boldsymbol{\phi}^j = \left[\phi_1^j,\, \phi_2^j,\, \ldots,\, \phi_p^j\right]^T \quad . \tag{3.1}$$

Each parameter ϕ_i^j is normalized over its associated range $\boldsymbol{\Omega}_i$ and encoded as a 32-bit, unsigned integer. This encoding process ensures that all parameters stay bounded within their appropriate ranges. The p binary-encoded genes represent a binary string called the chromosome. The set of S individuals is referred to as the current working population $W(k)$, where k denotes the current generation. Finally, the Pareto archive $A(k)$ (Sect. 3.3.8) is used to store all non-dominated solutions discovered since the beginning of the evolution.

3.3.1 Initialization

The basic algorithm for the MOGA is given in Table 3.1. The first step is to select the general parameters for the algorithm, including the size S of the working population $W(k)$, number of generations N, crossover rate p_c, mutation rate p_m, fitness criteria $\boldsymbol{f}_{L\times 1}$, fitness sharing parameters $\boldsymbol{\rho}_{\mathrm{sh},\,L\times 1}$, dummy fitness scaling factor ϵ (Sect. 3.3.3), initial archive selection factor γ_{a_0}, minimum archive selection factor γ_{a_m}, archive selection decrement factor $\delta\gamma_a$, and maximum archive size factor K_a. The

1. **Initialization:** Choose S, the size of the working population $W(k)$, number of generations N, crossover rate p_c, mutation rate p_m, fitness criteria $\boldsymbol{f}$, fitness sharing parameters $\boldsymbol{\rho}_{\text{sh}}$, dummy fitness scaling factor ϵ, initial archive selection factor γ_{a_0}, minimum archive selection factor γ_{a_m}, archive selection decrement factor δ_{γ_a}, and maximum archive size factor K_a. Choose parameter domain $\boldsymbol{\Omega}$. Initialize each individual $\boldsymbol{\phi}^j$, $j = 1, \ldots, S$, randomly. Initialize archive selection factor: $\gamma_a(k) = \gamma_{a_0}$.

2. For $k = 1, \ldots, N$:

   (a) Evaluate each individual's fitness $\boldsymbol{J}(\boldsymbol{\phi}^j)$, $j = 1, \ldots, S$.

   (b) **Compute dummy fitness:**

      i. **Domination count:** Compute the domination count of each individual D^j, $\forall j \in W(k) \cup A(k)$, where $W(k)$ is the current working population, and $A(k)$ is the current Pareto archive. Organize individuals in $W(k)$ into separate fronts M^1 to M^P with ascending domination counts, where M^1 contains non-dominated individuals, M^P contains individuals with maximum domination count, and P is the total number of fronts found.

      ii. **Pareto-Archive:** Update the Pareto archive $A(k)$ by adding new non-dominated individuals from M^1 and eliminating individuals that have become dominated. *Replace* M^1 with $A(k)$, since non-dominated solutions will be selected from the archive.

      iii. **Non-dominated sorting:** For $n = 1, \ldots, P$:

         A. If $n = 1$, choose an arbitrary maximum fitness value J^1_{max}. For $n > 1$, choose a maximum scalar fitness value J^n_{max} for M^n such that $J^n_{max} < J^{n-1}_{min}$.

         B. Set dummy scalar fitness value $v^q = J^n_{max} \forall q \in M^n$.

         C. **Fitness-sharing:** Use fitness sharing to degrade the dummy fitness v^q of each individual q in M^n using sharing parameter $\boldsymbol{\rho}_{\text{sh}}$.

   (c) **Selection:** Create next generation $W(k+1)$ by selecting $\gamma_a(k) \cdot S$ individuals from M^1 and $(1 - \gamma_a(k)) \cdot S$ individuals from the set of dominated fronts $W'(k) = M^2 \cup M^3 \cup \ldots \cup M^P$. In both cases, use fitness-proportionate selection based on the dummy scalar fitness values computed above.

      i. Update archive selection factor: $\gamma_a(k+1) = \max(\gamma_{a_m}(k), (1 - \delta_{\gamma_a})\gamma_a(k))$.

   (d) **Crossover:** Select each individual in $W(k+1)$ for crossover using probability p_c. Randomly locate another individual, distinct from the first, for mating. Cross the individuals using standard single-point crossover.

   (e) **Mutation:** For each individual in $W(k+1)$, mutate each bit position of the chromosome using probability p_m.

Table 3.1: The Multiobjective Genetic Algorithm (MOGA).

archive selection factor $\gamma_a(k)$ is initialized to γ_{a_0} and is explained in Sect. 3.3.8, and the maximum archive size factor determines the maximum size of the Pareto archive $A(k)$, which is $Q_{\max} = K_a S$. Next, the parameter domain $\mathbf{\Omega}$ must be chosen, where $\mathbf{\Omega}$ contains $\mathbf{\Omega}^i$, $i = 1, \ldots, p$, where p is the number of parameters from Eq. (3.1). Finally, each individual $\boldsymbol{\phi}^j$, $j = 1, \ldots, S$, must be initialized randomly over the 32-bit range. After initialization, each individual is simulated, then evaluated using the vector fitness function $\boldsymbol{J}(\boldsymbol{\phi})$.

3.3.2 Domination Count

The next step in the MOGA involves determining the domination count for each individual in the working population $W(k)$ and the Pareto archive $A(k)$. The domination count D^j is equal to the number of solutions dominating individual j [70]. After initializing the domination count of each individual in $W(k)$ and $A(k)$ to zero, every individual in the population $W(k)$ is compared with every other individual in $W(k)$ and $A(k)$, requiring up to $S \cdot Q + \frac{S(S-1)}{2}$ comparisons, where S is the population size (Sect. 3.3.1) and Q is the current size of the Pareto archive $A(k)$. During each comparison, the domination count for *both* individuals are updated appropriately based on which individual is dominated. For example, if individual j is compared with individual k, and j dominates k, then D^k is incremented, while D^j remains unchanged. If j and k are non-dominated with respect to each other, neither D^j nor D^k is incremented. Those individuals having zero domination counts after all comparisons are complete will be part of the current estimate of the Pareto front.

3.3.3 Non-dominated Sorting

The non-dominated sorting routine is based on [71] and computes dummy scalar fitness values for each individual based on the "non-domination front"[12] in which it resides, where each front is comprised of individuals with the same domination count. After the domination counts for the current working population $W(k)$ and the Pareto archive $A(k)$ are computed, individuals from the current working population $W(k)$ are grouped into separate fronts M^1 to M^P in ascending order of domination, where P is the number of different domination counts for all individuals in $W(k)$. The *lowest-numbered* front, M^1, consists of non-dominated individuals ($D^q = 0 \,\forall\, q \in M^1$), making it, along with those individuals in $A(k)$ that have zero domination count, the current estimate of the Pareto front. On the other hand, the highest-numbered front, M^P, contains individuals with the maximum number of solutions dominating them. After M^1 is determined, the current Pareto archive $A(k)$ is updated, as described in Sect. 3.3.8. At this point, M^1 is *replaced* with $A(k)$, since all non-dominated solutions will ultimately be selected from this set (Sect. 3.3.5).

Once individuals are grouped in separate fronts, a dummy scalar fitness value can be computed for each one. An arbitrary maximum scalar fitness value J^1_{max} is assigned to the first front, M^1, with scalar fitness value J^n_{max} assigned to each successive front M^n such that $J^n_{max} < J^{n-1}_{min}$, $n = 2, \ldots, P$. Various approaches are possible in assigning fitness values, including both linear and nonlinear methods. Nonlinear methods may result in higher selection pressure such that individuals in lower-numbered fronts have a disproportionate chance of being selected. In this case,

[12]Only the outermost front where the domination count is zero is technically non-dominated. However, the term "non-domination front" refers to the fact that removing each front, starting with the Pareto front, would make the next one beneath it the new non-dominated front.

a nonlinear method was selected such that the maximum fitness for each front M^n is computed using the following equation:

$$J_{max}^{n+1} = (1 - \epsilon)\, J_{min}^{n} \quad \text{for} \quad n = 1, \ldots, P - 1, \tag{3.2}$$

where ϵ is the dummy fitness scaling factor, typically set to 0.75, and J_{min}^{n} is the minimum fitness value across all individuals in M^n after fitness sharing, discussed below. In each front M^n, the dummy scalar fitness value v^q for each individual q is initially assigned the maximum scalar fitness value for that front, J_{max}^{n}. The dummy fitness value v^q is then "shared" using the niche-formation strategy discussed next.

3.3.4 Niche-Formation

Niche-formation techniques are used to encourage the dispersion of solutions on the Pareto front. Ideally, solutions should be evenly dispersed over the Pareto front so that maximum flexibility is achieved when picking final solutions. However, the MOGA is prone to genetic drift, where solutions tend to cluster around a single local optimum. To combat this, fitness is shared, or degraded, between two individuals in the same front that are too close together in fitness space. The algorithm used here is based on Srinivas and Deb [71], with several modifications, discussed below. First, the niche factor N_c^i for each individual i is computed as follows:

$$N_c^i = \sum_{\substack{j \in M^n \\ j \neq i}} \max\left(0, 1 - (d_{ij})^2\right), \tag{3.3}$$

where i and j are in the M^n front, and $(d_{ij})^2$ represents the squared ellipsoidal distance between i and j in fitness space:

$$(d_{ij})^2 = \sum_{k=1}^{L} \frac{\left(f_k^i - f_k^j\right)^2}{(\rho_k)^2}, \tag{3.4}$$

where ρ_k is the k-th sharing parameter of $\boldsymbol{\rho}_{\text{sh}}$, f_k^i is the k-th element of the fitness vector $\boldsymbol{f}^i$ for individual i, and L is the number of fitness criteria (Sect. 3.3.1). The use of a vector-valued sharing parameter $\boldsymbol{\rho}_{\text{sh}}$ is a modification of Srinivas and Deb's algorithm and determines whether or not i and j lie within the same ellipsoidal volume. If so, then $(d_{ij})^2 < 1$, and the niche factor is increased in Eq. (3.3). The niche factor N_c^i computes a penalty based on the degree of crowding for individual i. The niche factor is then used to compute the shared fitness for individual i:

$$v^i = \frac{J_{max}^n}{1 + N_c^i}\,, \tag{3.5}$$

where J_{max}^n is the maximum fitness assigned to front M^n (Sect. 3.3.3). The inclusion of a 1 in the denominator of Eq. (3.5) is another modification to Srinivas and Deb's method and addresses the case where N_c^i is zero.

3.3.5 Selection

Once the non-dominated sorting and fitness sharing procedures are complete, each individual in fronts M^1 through M^P will have a dummy scalar fitness value v^j based on the front to which it belongs and the degree of crowding with respect to its neighbors. Individuals are then selected for the next generation $W(k+1)$ from two sets, the Pareto archive, represented by M^1, and the set of remaining dominated fronts, $W'(k) = M^2 \cup M^3 \cup \ldots \cup M^P$. The parameter $\gamma_a(k)$ governs what percentage of the population comes from M^1, while $1 - \gamma_a(k)$ determines the percentage selected from the dominated fronts $W'(k)$ (Sect. 3.3.8). Fitness-proportionate selection is used to select individuals from each set. Fitness-proportionate selection is a simple process where each individual has a chance of being selected based on its relative fitness [72].

The relative fitness r^j for individual j is computed by dividing its dummy scalar fitness by the total scalar fitness of the appropriate set:

$$r^j = \frac{v^j}{\sum_{k \in D} v^k}, j \in D, \tag{3.6}$$

where $D = W'(k)$ for an individual in the dominated fronts, and $D = A(k) = M^1$ for an individual in the Pareto archive $A(k)$. Thus, from each selection set $A(k)$ or $W'(k)$, an individual will have, on average, r^j probability of being selected.

Another way to understand fitness-proportionate selection is through the roulette wheel analogy. The total fitness of the population is represented by a roulette wheel, where each individual is given a pie-shaped segment proportional to its relative fitness r^j. If the wheel is spun and a dart is thrown at the wheel, then those individuals with larger pie-shaped areas should have a greater probability of being hit by the dart, which signifies selection. Furthermore, some individuals could be selected multiple times. Individuals with small fitness will have a relatively low probability of being selected. The selection operator behaves similarly to Darwin's natural selection, where the fittest individuals survive to reproduce and continue their genetic line [73].

3.3.6 Crossover

Once individuals are selected, they are mated with another individual with probability p_c. Individuals selected for mating are not mated with identical copies of themselves. During mating, each individual's chromosome is crossed with a partner's using standard, single-point crossover, as shown in Fig. 3.2. A random number between zero and one is generated to determine where the crossover point will be placed. The bits before the crossover site in each parent are maintained, while those beyond

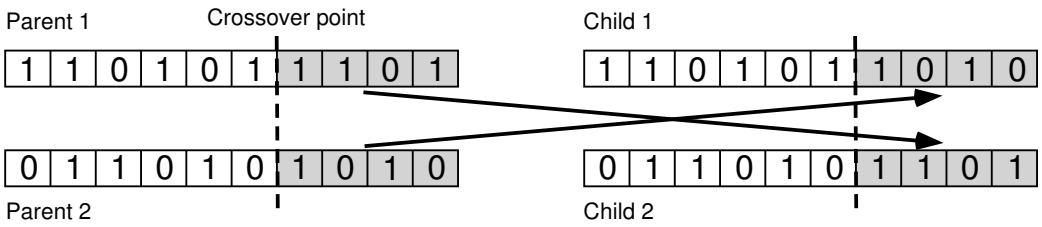


Figure 3.2: Example of single-point crossover. Reproduced from [72].

the crossover point are swapped. Crossover creates new genetic material by mixing that of each parent, although it cannot introduce random variation. To address this, the mutation operator is employed.

3.3.7 Mutation

After crossover is complete, each bit position of the individual's binary-encoded chromosome is mutated with probability p_m. In evolutionary algorithms, mutation is generally considered secondary to crossover in terms of importance [74], which is why the mutation rate p_m is typically much smaller than the crossover rate p_c. However, mutation remains an important mechanism in evolutionary algorithms for facilitating an effective stochastic search by the introduction of new genetic material. This prevents solutions from becoming trapped in local optima by ensuring some degree of random variation with which to explore new areas of the parameter space.

Mutation can be extremely effective in certain applications. In fact, some evolutionary algorithms have used mutation as the only stochastic operator with some success [37, 75, 76]. However, it has been found that using excessive mutation rates can lead to a "random walk" through the parameter space [69] and the loss of good

solutions. To guard against such losses and facilitate convergence, elitism can be employed, as discussed next.

3.3.8 Controlled Multiobjective Elitism

Elitism refers to the preservation of good solutions from one generation to the next. In single-objective GA's, this would mean copying the best solution in the current generation to the next without altering it through crossover or mutation. In multiobjective GA's, elitism involves the entire current estimate of the Pareto front, since all of these individuals are considered "the best." However, in EMO, ensuring that all individuals in the emerging Pareto front are preserved, which is called uncontrolled elitism, can saturate the current working population and disable the search for new solutions [77]. Thus, a method of controlled elitism is required.

There are a number of different methods of controlled multiobjective elitism, although a common strategy is to maintain a separate archive of non-dominated solutions that must be updated during each new generation [77, 78]. While some strategies use this "Pareto archive" as a pool from which to select new individuals, others do not. In this work, the Pareto archive $A(k)$ is used as a selection pool to control the number of non-dominated individuals in the current working population $W(k)$ using the archive selection factor $\gamma_a(k)$. The factor $\gamma_a(k)$ is initialized to γ_{a_0} and decremented during each generation such that

$$\gamma_a\left(k+1\right) = \max(\gamma_{a_m}, \left(1-\delta_{\gamma_a}\right)\gamma_a\left(k\right)) \ , \tag{3.7}$$

where δ_{γ_a} is the archive selection factor decrement ratio, and γ_{a_m} is the minimum archive selection factor. During each generation, $\gamma_a(k) \cdot S$ individuals are selected from the Pareto archive, and $(1-\gamma_a(k)) \cdot S$ are selected from the set of dominated

fronts $W'(k)$. In both cases, the shared fitness values of the individuals in each set are used to make the selection, as described above.

Controlling the number of non-dominated solutions in the working population is an attempt to balance exploration vs. exploitation [69]. Exploitation refers to using the genetic material from non-dominated solutions and combining it with other solutions in an effort to promote diversity across the current estimate of the Pareto front. However, using too much exploitation can lead to the degradation of the exploration mechanism. Exploration encourages the search through new parts of the parameter space using the prescribed mutation rate, which can typically be set significantly higher when elitism is employed, since the risk of losing good solutions is minimal.

To ensure that the Pareto archive contains only non-dominated solutions, the domination count of each of its members must be updated during each generation by comparing each member of the archive $A(k)$ with each member of the working population $W(k)$. New non-dominated solutions are added to the archive while newly dominated solutions in the archive are eliminated and no longer considered during subsequent processing. Finally, to prevent the archive from filling up, the archive size is set to a sufficiently large integer factor of the population, $K_a \cdot S$, where K_a is determined experimentally.

3.3.9 MOGA Summary

A set of typical parameters used in the MOGA is listed in Table 3.2. Parameters such as $\mathbf{\Omega}$, $\boldsymbol{\rho}_{\text{sh}}$, and L are dependent on the specific optimization problem and discussed in later chapters.

Parameter	Value	Description
S	32	Population size.
N	250	Number of generations.
p_c	0.60	Crossover rate.
p_m	0.05	Mutation rate.
ϵ	0.75	Dummy fitness scaling factor.
γ_{a_0}	0.50	Initial archive selection factor.
γ_{a_m}	0.125	Minimum archive selection factor.
δ_{γ_a}	5.53×10^{-3}	Archive selection decrement factor*.
K_a	10	Archive size factor, in multiples of S.

*The value was computed to cause $\gamma_a(k)$ to decrease from γ_{a_0} to γ_{a_m} over $N = 250$ generations.

Table 3.2: Typical parameters used for the MOGA.

3.4 Summary and Conclusions

In this chapter the multiobjective genetic algorithm was presented. Because the MOGA, as well as other EMO algorithms, utilizes a vector-valued fitness function, special methods are required to generate a usable and convenient measure of preference among the solutions. One popular technique presented here is non-dominated sorting, which groups individuals into distinct fronts according to their domination count. Dummy scalar fitness values are then assigned to each front and shared among the individuals based on their degree of crowding. A separate archive is maintained of all non-dominated individuals discovered since the start of the evolution. Using controlled elitism, a specified percentage of the population is selected from this archive, while the remainder comes from the set of dominated fronts. This approach attempts

to balance the exploitation of good solutions with the exploration of new parts of the space. Crossover and mutation are then employed using the standard approach, although a higher mutation rate can be tolerated because the risk of losing good solutions has been minimized through the controlled elitist strategy.

The use of multiobjective optimization addresses the difficulty of reducing the optimization problem to a single scalar value, where the performance of the search algorithm can be highly sensitive to the relative weighting of multiple criteria. As the goals for this work involve investigating complex dynamic behaviors whose properties are not well known, the use of evolutionary multiobjective optimization is appropriate since a single, scalar fitness value could be quite difficult to define. Furthermore, the use of a controlled, elitist strategy that has performed well in a number of benchmark problems should facilitate finding solutions in a parameter space where the bootstrapping problem is a significant issue.

CHAPTER 4

EVOLUTION OF 3D GALLOPING

4.1 Introduction

The allure of high-speed dynamic locomotion has motivated the design of many legged quadrupedal robots over the past several decades. Although some were capable of dynamic running [21]-[23], the achievement of a well-controlled, biological gallop has remained elusive. As stated in Sect. 1.1, the gallop is the preferred high-speed gait of most cursorial mammals [26]. The gallop typically occurs at higher running speeds, while trotting is used at lower speeds. Although no one has satisfactorily explained why animals prefer to gallop at higher speeds, there are several prevailing theories. Some believe that galloping is simply more energy efficient at higher speeds [79]-[80], while others believe that the characteristic smoothness of the gait results in lower peak forces on the legs [81]-[83]. Schmiedeler [84] reasoned that a combination of the two factors could explain the gait's attractiveness in terms of energy efficiency. No matter what the true reason, galloping is obviously an efficient, effective, and robust means of high-speed locomotion for many biological quadrupeds — yet it remains an elusive goal in the field of legged robotics.

The focus of this work extends beyond studying the gallop itself, however. Because the gallop is used at high speeds, it is the natural gait choice for studying high-speed dynamic maneuvers, which are the main thrust of this work. Consequently, the gallop is necessarily the starting point of this study. Furthermore, characterizing "biological-mode" galloping and maneuvering is desired, which emphasizes the importance of achieving a gait with biological characteristics. Such defining characteristics as early leg retraction, a prominent gathered flight phase, the phase-locked nature of the leg motion, and the smoothness of the gait with respect to minimal excursion of the body's center of mass and minimal pitch motion are considered here to be imperative in the development of the artificial gallop.

The main challenge in achieving a stable 3D gallop is that there currently exist no adequate models or control methods, as there have been no informative examples of the gait demonstrated in either hardware or simulation. The rotary gallop achieved by Smith and Poulakakis [25] failed to demonstrate any control over trajectory and lacked other defining features of the gait, including early retraction and phase-locked leg motion. Simulation studies [30, 34, 35, 36, 76] have also fallen short, as each one made one or more of the following simplifying assumptions: a planar model, massless legs, symmetric body, or contrived stability (e.g., the large, circular feet in [35]). To date, however, Herr and McMahon's [30] planar simulation has been the most realistic with the use of biological model parameters, early leg retraction, and phase-locked leg motion.

Due to the lack of an analytical model or established control methodology, the design approach for the 3D gallop controller relies on an understanding of the basic

dynamics of the gait and the evolutionary method described in Chap. 3. The results of this study will present an empirical characterization of the 3D gallop that, heretofore, has only been presented for actual biological systems. The results provide significant data on the underlying dynamics of the 3D gallop and how it can be stabilized in an artificial system. Furthermore, the design approach promises to facilitate further study of the gallop through the exploration of different control strategies and modifications to the system architecture.

In the sections that follow, a detailed description of the 3D gallop controller is provided, including an explanation of the control parameters required for each of the leg primitive functions that are employed. Next, the evolutionary optimization problem is explained, including a description of the three fitness criteria used by the multiobjective GA. Following that, the results of the gallop are presented, including an analysis of the stride-level and sub-stride mechanics and the torque and power requirements for each joint. Finally, a summary is provided at the end.

4.2 Description of the 3D Gallop Controller

The gallop controller is comprised of a state machine that uses the following leg control primitives: TRANSFER, WAIT-FOR-TRIGGER, EARLY-RETRACTION, GALLOP-STANCE-CONTROL, and FREE. Each primitive is summarized in Table 2.4, and the state diagram for the controller is shown in Fig. 4.1. Each leg executes the leg primitive functions independently, although WAIT-FOR-TRIGGER (Sect. 4.2.3) is used to synchronize the leg motion. Each of the leg primitive functions is explained in terms of the gallop controller in the following sections.

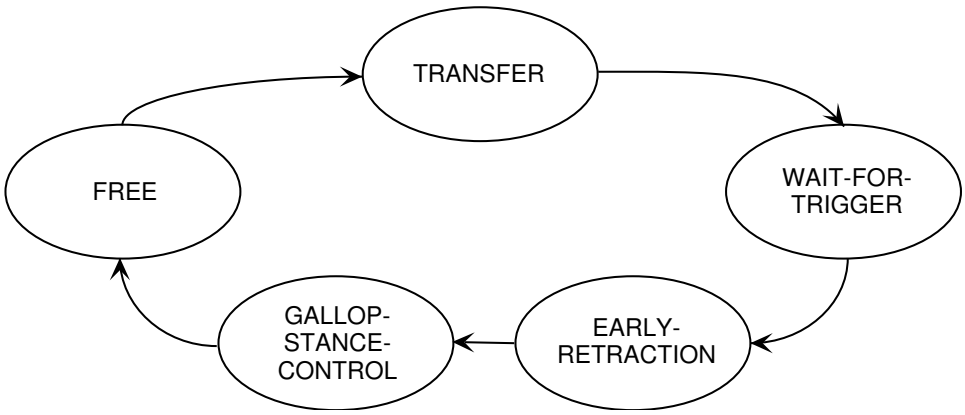


Figure 4.1: State diagram for each leg for the 3D gallop controller. Leg synchronization is performed by the WAIT-FOR-TRIGGER function. Leg control primitives are summarized in Table 2.4.

4.2.1 TRANSFER

The TRANSFER function is employed while each leg is in flight. During TRANSFER, the ab/ad, hip, and knee joints are rotated to desired angles using a cubic spline trajectory with PD control at each joint. The PD parameters employed for each joint are listed in Table 2.5. During TRANSFER, the double-spline option is employed for the knee joints ($\sigma_{k_d} = 1$ in Table 2.4) so that the virtual leg length l (Fig. 2.4) is decreased temporarily during mid-transfer to provide adequate foot clearance from the ground. The period T of the TRANSFER function was hand-tuned to $0.1\,s$ so that each leg reaches its destination position in an adequate amount of time.

At some point during TRANSFER, the system usually reaches the top of the gathered flight phase, which will be referred to as "TOF." At this point, the ab/ad touchdown angle (about the $\hat{\boldsymbol{x}}_b$ axis, Fig. 2.1) is computed with respect to the body

using the following control law:

$$\begin{aligned}\theta_{a_i} &= \theta_{a_0} + w_{\alpha_i}\text{sat}\left(s_{\alpha_i}\left(k_{p_\alpha}\alpha_e + k_{d_\alpha}\dot{\alpha}_e\right)\right) - w_{\gamma_i}f_\gamma\text{sat}\left(-w_{\gamma_i}\left(k_{p_\gamma}\gamma_e + k_{d_\gamma}\dot{\gamma}_e\right)\right) \\ &\quad - w_{\gamma_i}\text{sat}\left(w_{\gamma_i}\left(k_{p_\gamma}\gamma_e + k_{d_\gamma}\dot{\gamma}_e\right)\right) \ , \end{aligned} \tag{4.1}$$

where $\theta_{a_0} = \pi/2$, which biases the i-th ab/ad angle perpendicular to the body, α_e and $\dot{\alpha}_e$ are the yaw and yaw rate errors, γ_e and $\dot{\gamma}_e$ are the roll and roll rate errors,"sat" is the saturation function given by

$$\text{sat}\left(x\right) = \begin{cases} x & \text{if} \quad x \geq 0 \\ 0 & \text{otherwise} \end{cases} \ , \tag{4.2}$$

w_{α_i} is given by

$$w_{\alpha_i} = \begin{cases} 1 & \text{if} \quad i = 1 \\ -1 & \text{if} \quad i = 2 \\ f_\alpha & \text{if} \quad i = 3 \\ -f_\alpha & \text{if} \quad i = 4 \end{cases} \ , \tag{4.3}$$

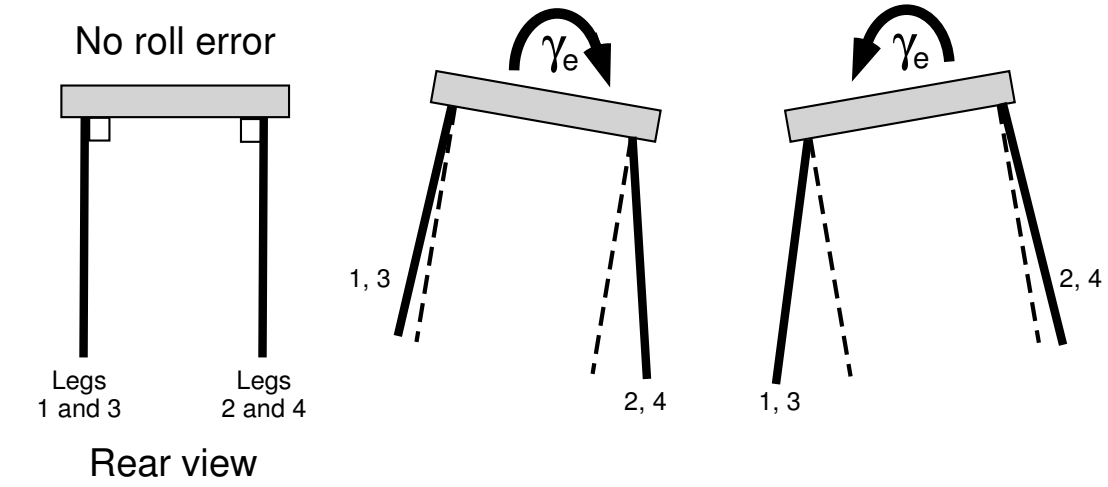
s_{α_i} is given by

$$s_{\alpha_i} = \begin{cases} 1 & \text{if} \quad i = 1 \quad \text{or} \quad i = 4 \\ -1 & \text{otherwise} \end{cases} \ , \tag{4.4}$$

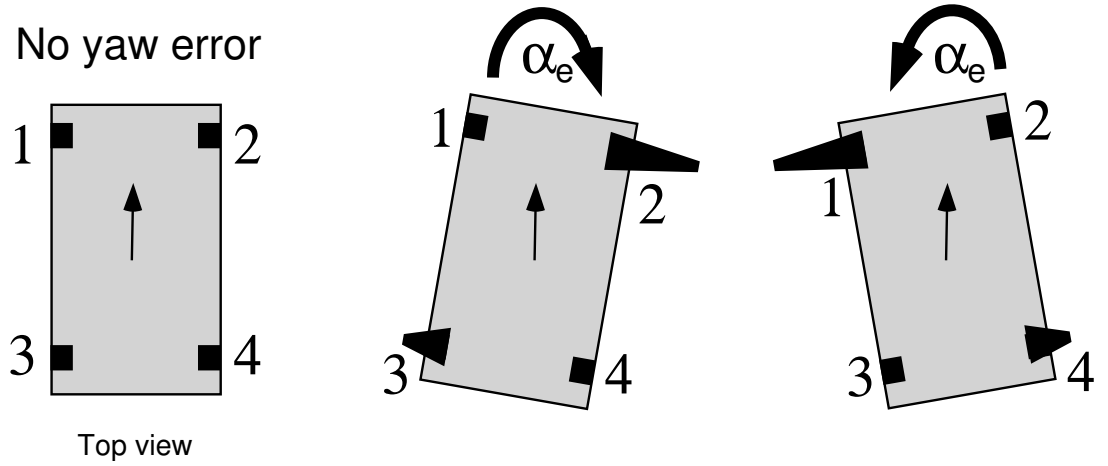
and w_{γ_i} is given by

$$w_{\gamma_i} = \begin{cases} -1 & \text{if} \quad i = 1 \quad \text{or} \quad 3 \\ 1 & \text{otherwise} \end{cases} \ . \tag{4.5}$$

The control law in Eq. (4.1) computes the ab/ad touchdown angle for each leg based on an "outward leg-rotation" control scheme (Fig. 4.2). For example, when the body is rolling onto its right side (Fig. 4.2 (a), middle), the ab/ad joints for the legs on this side (legs 2 and 4), which are considered the primary legs, are rotated outward from the body. At the same time, the ab/ad joints for the left-side legs (legs 1 and 3), which are the secondary legs, are also rotated outward, but by a small, positive scaling factor f_γ. The legs tend to sprawl outward asymmetrically to correct roll,



(a)



(b)

Figure 4.2: Outward leg control scheme for (a) roll, and (b) yaw.

which produces a wider, more stable stance. Note that the terms "primary" and "secondary" are dependent on the sign of the roll error. When the body rolls over onto its left side (Fig. 4.2 (a), right), legs 1 and 3 are considered primary, while legs 2 and 4 are considered secondary.

The yaw control term in Eq. (4.1) implements a similar scheme, although only diagonal leg pairs are affected. For example, if the body is yawing counterclockwise (CCW) about $\hat{\boldsymbol{z}}_e$ (Fig. 2.1), both θ_{a_1} and θ_{a_4} are adjusted outward from the body (Fig. 4.2 (b), right), although θ_{a_4} is adjusted by a small[13], positive scaling factor f_α. This control scheme differs from the traditional approach employed by Raibert [21], where the front pair of ab/ad angles are offset in equal and opposite directions from the rear pair. While this method was attempted initially, it was found to be unstable for high-speed galloping because of the narrow stance. The outward leg-rotation method, on the other hand, accomplishes yaw control primarily by generating a net moment along both the yaw and roll axes, which produces a bank. It seems that this coupled method of turning is a more stable approach for high-speed gaits like the gallop because of the wider stance configuration.

After recomputing ab/ad touchdown angles using Eq. (4.1), the transfer spline for the leg is modified on-the-fly to reach the new ending conditions. Although the TOF point can occasionally occur after the TRANSFER function for a particular leg has expired, the predictable dynamics of the system usually allow the adjustment

[13]It was determined through preliminary experiments that using larger scaling factors for f_α caused excessive disturbances in pitch and roll.

of the ab/ad touchdown angle to be made smoothly during mid-transfer. Otherwise, the ab/ad angle is adjusted via a step-change in desired angle during WAIT-FOR-TRIGGER or EARLY-RETRACTION, depending on when the TOF occurs. Recalculation of the transfer spline is discussed in the next section.

One final detail worthy of mention involves what is *not* being done during TRANSFER. While many dynamic running controllers adjust the fore-aft leg angle (here, the virtual leg angle) to control velocity based on Raibert's early control scheme [21, 32], the controller described here enforces a fixed touchdown leg angle from stride to stride. The reason for this is because a stance phase velocity servo is employed during GALLOP-STANCE-CONTROL (Sect. 4.2.5) based on [30]. This mechanism controls the velocity of the body during stance using shoulder and hip torque, making it unnecessary to use variations in touchdown angle. Touchdown virtual leg angles for the front and rear legs were found by evolution such that the quadruped could achieve a reasonable number of stable galloping strides over a wide range of control parameters. For the front legs, $\theta_{v_f} = 0.69\,\text{rad}$, and for the rear, $\theta_{v_r} = 0.65\,\text{rad}$. Both angles are defined with respect to the body's normal (Fig. 2.4).

4.2.2 Transfer Spline Recalculation

Transfer splines must often be recalculated on-the-fly to account for TOF control adjustments, such as the one for the ab/ad angle calculation in Eq. (4.1). In addition, since in-flight transitions between higher level behavior modules also occur at TOF, transfer spline parameters must be recalculated for this case, as well, since each leg will typically be in mid-TRANSFER at TOF[14]. Thus, a method is needed to guarantee

[14]Most of the behavior modules investigated in this work use the same leg functions and sequence as the gallop. Therefore, at the TOF transition, each leg function is initialized in the new behavior

that the recalculation of transfer spline parameters generates a reasonably smooth change in the trajectory of each joint.

The method described here is a heuristic one that has demonstrated adequate performance during galloping, and during transitions to and from galloping. The first step in the method is to determine whether the required change in the transfer spline is significant, which will be indicated by $\sigma_{\delta_T} = 1$. Consider a cubic spline function $\theta(t)$ for a single joint as follows:

$$\theta(t) = A + Bt + Ct^2 + Dt^3, \tag{4.6}$$

with starting angle θ_0, final angle θ_f, starting time t_0, transfer time T_t, and initial and final rates $\dot{\theta}_0$ and $\dot{\theta}_f$ both zero, as stated in Sect. 2.3.4. Thus there are four equations with four unknowns, and the spline parameters A, B, C, and D can be determined by solving the following equation:

$$\begin{bmatrix} 1 & t_0 & (t_0)^2 & (t_0)^3 \\ 1 & t_f & (t_f)^2 & (t_f)^3 \\ 0 & 1 & 2t_0 & 3(t_0)^2 \\ 0 & 1 & 2t_f & 3(t_f)^2 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} \theta_0 \\ \theta_f \\ 0 \\ 0 \end{bmatrix}, \tag{4.7}$$

where $t_f = t_0 + T_t$.

Now, consider a new set of parameters θ'_f and T'_t, which are determined at time t^* such that $t_0 < t^* < t_0 + \min(T_t, T'_t)$ [15] and $\theta(t^*) = \theta^*$. The variable σ_{δ_T} is then determined as follows:

$$\sigma_{\delta_T} = \begin{cases} 0 & \text{if } T'_t = T_t \text{ and } \frac{|\theta'_f - \theta_f|}{|\theta_f - \theta_0|} \leq \delta_T \\ 1 & \text{otherwise,} \end{cases}, \tag{4.8}$$

---

module using the same function that it ended on during the last module. For example, if leg 1 was in WAIT-FOR-TRIGGER at the TOF transition between the gallop and the turn, then it is initialized in the WAIT-FOR-TRIGGER function for the turn. Typically, however, most legs are in TRANSFER at the TOF point, requiring a spline recalculation.

[15] If $t^* > t_0 + \min(T_t, T'_t)$, then the transfer period has ended and a step change adjustment is made to set $\theta = \theta'_f$.

where $\delta_T = 0.1$. The change is considered significant if the new transfer time is not the same as the old transfer time, or the change in the final angle θ_f is greater than 10% of the original difference between θ_f and θ_0. In either case, the cubic spline must be recalculated such that $\theta(t^*) = \theta^*$, $\theta(t'_f) = \theta'_f$, $\dot{\theta}(t^*) = 0$, and $\dot{\theta}(t'_f) = 0$, where $t'_f = t_0 + T'_t$. These four constraints are sufficient to compute new cubic spline parameters A', B', C', and D' by solving the following equation:

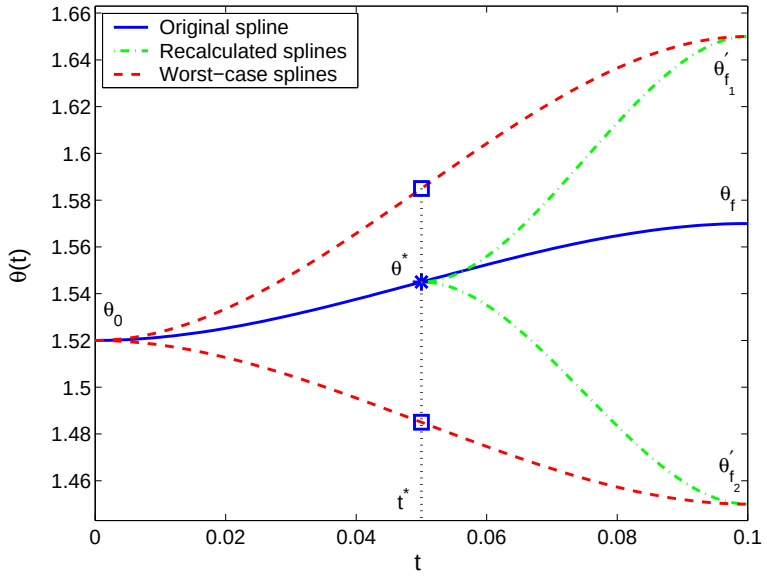
$$\begin{bmatrix} 1 & t^* & (t^*)^2 & (t^*)^3 \\ 1 & t'_f & \left(t'_f\right)^2 & \left(t'_f\right)^3 \\ 0 & 1 & 2t^* & 3\,(t^*)^2 \\ 0 & 1 & 2t'_f & 3\left(t'_f\right)^2 \end{bmatrix} \begin{bmatrix} A' \\ B' \\ C' \\ D' \end{bmatrix} = \begin{bmatrix} \theta^* \\ \theta'_f \\ 0 \\ 0 \end{bmatrix}. \tag{4.9}$$

If $\sigma_{\delta_T} = 0$, then the required change in the cubic spline is not considered significant, and the spline parameters are recalculated by solving the following equation with $\theta(t_0) = \theta_0$, $\theta(t_f) = \theta'_f$, and initial and final rates both zero:

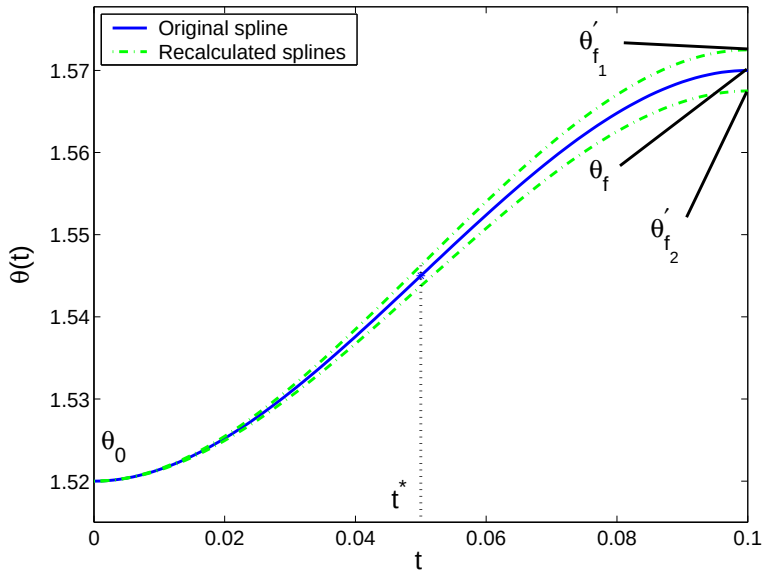
$$\begin{bmatrix} 1 & t_0 & (t_0)^2 & (t_0)^3 \\ 1 & t_f & (t_f)^2 & (t_f)^3 \\ 0 & 1 & 2t_0 & 3\,(t_0)^2 \\ 0 & 1 & 2t_f & 3\,(t_f)^2 \end{bmatrix} \begin{bmatrix} A' \\ B' \\ C' \\ D' \end{bmatrix} = \begin{bmatrix} \theta_0 \\ \theta'_f \\ 0 \\ 0 \end{bmatrix}, \tag{4.10}$$

where $t_f = t_0 + T_t$.

Figure 4.3 illustrates the cases when $\sigma_{\delta_T} = 1$ and $\sigma_{\delta_T} = 0$. The rationale behind this method is that minor changes in the cubic spline trajectory (Fig. 4.3 (b)) can be handled by simply recalculating the spline with a new endpoint, since the difference between the new and old trajectories will be small enough to result in minor changes in commanded position and velocity at t^*. For more significant changes (Fig. 4.3 (a)), however, a new spline should be computed starting at $\theta(t^*)$ with zero initial rate. This approach avoids commanding a large step change in trajectory at t^*, which is indicated by the squares plotted on the worst-case splines in Fig. 4.3 (a).



(a)



(b)

Figure 4.3: Recalculation of cubic splines during TRANSFER for (a) significant change ($\sigma_{\delta_T} = 1$), and (b) minor change ($\sigma_{\delta_T} = 0$). θ'_{f_1} and θ'_{f_2} represent new ending conditions that are farther from, and closer to, the original condition θ_0, respectively. In (a), the spline must be restarted at t^* with zero initial rate to avoid the large step change in trajectory, which is indicated by the squares. This large trajectory change would occur if the spline is recalculated from its initial point at t_0, but using either of the new final conditions. In (b), however, recalculating the spline from its initial point is permissible, since the change in final conditions is small.

Leg	Trigger Leg	Trigger Type	Description	Parameters
1	2	TD-DELAY	Delay δ_1 after trigger leg's touchdown.	$\delta_1 = 12.5\,\text{ms}$.
2	3	ER-HIP-HEIGHT	Trigger when the latter of two events occur: 1. δ_2 has expired since the trigger leg began retraction. 2. The current leg's hip height is less than h_t.	$\delta_2 = 15.9\,\text{ms}$, $h_t = 0.24\,\text{m}$.
3	4	TD-DELAY	Delay δ_3 after trigger leg's touchdown.	$\delta_3 = 7.0\,\text{ms}$.
4	4	HIP-HEIGHT	Delay δ_4 after trigger leg's hip height falls beneath h_t.	$\delta_4 = 0.0\,\text{ms}$, $h_t = 0.24\,\text{m}$.

Table 4.1: Trigger events for each leg for the gallop. See Table 2.6 for a description of triggering options for WAIT-FOR-TRIGGER.

4.2.3 WAIT-FOR-TRIGGER

Following TRANSFER, the leg enters WAIT-FOR-TRIGGER, where it is held at the appropriate virtual leg touchdown angle θ_{v_f} or θ_{v_r}, given in Sect. 4.2.1, until a triggering event occurs. When the trigger is detected, the leg automatically transitions to EARLY-RETRACTION. The period T of the function is set to a relatively large value of 1.0 s to prevent the leg from transitioning to EARLY-RETRACTION before detecting the appropriate trigger. Table 4.1 lists the trigger events for each leg. For leg 4 (Fig. 2.1), the trigger event is the spatial condition that its own hip height goes below a threshold h_t, which was set to the nominal leg length of $0.24\,m$ (Table 2.1). Thus, leg 4 is "self-triggering." For legs 1 and 3, temporal triggers are used based on

the touchdown times of the preceding legs in the transverse footfall sequence (4-3-2-1). Leg 3 is triggered δ_3 after leg 4 touches down, and leg 1 is triggered δ_1 after leg 2 touches down. Leg 2 employs a hybrid trigger with both temporal and spatial cues. The temporal cue is based on the retraction time of leg 3 plus a delay of δ_2. The spatial cue, which is employed as a safety feature, requires that leg 2's own hip height must be below h_t in order to begin early retraction. The delay values listed in Table 4.1 were evolved to produce good results at the nominal running speed of 4.15 m/s, the preferred galloping speed of an animal with the same mass as the artificial system [31].

The triggering scheme described here is based in part on Herr and McMahon's approach [30], although their method has several important differences. First, Herr and McMahon trigger leg 4 after a fixed delay from leg 2's retraction *during the previous stride*. This method is problematic for the 3D gallop, however, since the system's TOF conditions are not stable enough in the beginning to completely phase-lock the leg motion. Using a height sensor on both legs 2 and 4 makes a dramatic improvement to the robustness of the 3D gallop controller, as it compensates appropriately for variations in height and pitch before the system reaches steady-state.

The second difference between Herr and McMahon's triggering scheme and this one is that they trigger leg 2 using a temporal delay from the beginning of leg 4's retraction. This approach, however, does not guarantee the proper transverse leg phasing should the system have an excessively positive roll angle. In such a case, leg 2 could touch down before leg 3 because the former would have a closer proximity to the ground.

The third difference is that Herr and McMahon also lengthen the legs during retraction. However, it was found in this work that doing so increases the impact losses at touchdown. Increasing the leg length requires rotating the shank downward toward the ground. This rotation increases the vertical velocity of the foot relative to the ground, resulting in larger impact losses. Consequently, the nominal leg length l_0 (Table 2.1) is maintained throughout the function.

Finally, Herr and McMahon use different touchdown angles between each leg in the front and rear leg pairs. Thus, even without early retraction, each leg in the pair would still contact the ground at different times. Conversely, in this method the same angle is used for each leg in the front and rear pairs to reduce the parameter set and make the footfall sequence dependent on the triggering mechanism. Furthermore, a visual analysis of biological gallops reveals that each leg in the front and rear leg pair reaches approximately the same angle before retracting [85], suggesting that the footfall sequence is governed by a phase-locked temporal triggering mechanism as opposed to a fixed leg spread.

4.2.4 EARLY-RETRACTION

Once a leg is triggered, it switches from WAIT-FOR-TRIGGER to EARLY-RETRACTION. As discussed in Sect. 2.3.6, during EARLY-RETRACTION, the leg is retracted, or rotated rearward, by matching the tangential velocity of the foot with the desired ground speed. The ab/ad and knee angles are both held constant so that the retraction occurs only about the hip fore-aft rotation axis. One apparent advantage of early retraction is that the feet touch down with lower velocities relative to the ground, which reduces impact losses. For our system, the early retraction mechanism

also serves another very important purpose: it is used to enforce the transverse[16] leg phasing of the 3D gallop. Without early retraction, each pair of legs in the front and rear would touch down nearly simultaneously (assuming negligible roll), unless a fixed spread between touchdown leg angles was employed for each set of legs. However, biological evidence suggests that animals do not use a fixed leg spread to generate the leg phasing differences seen in galloping.

Note that because of early retraction, the nominal virtual leg angles θ_{v_f} and θ_{v_r} computed during TRANSFER (Sect. 4.2.1) are *not* the actual touchdown angles. The actual touchdown angles are smaller relative to $\hat{\boldsymbol{z}}_b$ (Fig. 2.1) than the nominal angles because of the rearward rotation of the joint. At steady-state, however, the actual virtual leg touchdown angles are related to the nominal angles by predictable offsets. Finally, the period T was set to a relatively large value of 1.0 s to prevent the function from expiring prematurely, effectively forcing the function to terminate upon ground contact.

4.2.5 GALLOP-STANCE-CONTROL

When the leg makes contact, the controller switches to GALLOP-STANCE-CONTROL, where several critical control functions are performed. First, to control the body's forward velocity, PI servo control on velocity is used to compute the hip torque:

$$\tau_{h_i} = k^0_{p_h} \int_{t_{\mathrm{td}_i}}^{t} \left(\frac{v_{x_d} + v_{b_i}}{l_i\,(t)} - \dot{\theta}_{v_i}\,(t) \right) \mathrm{d}t + k^0_{d_h} \left(\frac{v_{x_d} + v_{b_i}}{l_i} - \dot{\theta}_{v_i} \right), \tag{4.11}$$

where τ_{h_i} is the fore-aft torque at hip i, $\dot{\theta}_{v_i}$ and l_i are the angular velocity and length, respectively, for virtual leg i (Fig. 2.4), v_{x_d} is the desired forward velocity of 4.15 m/s

[16]The transverse gallop was selected over the rotary gallop because of its improved stability characteristics [86] and minimal leg interference [87].

(Sect. 4.2.3), t_{td_i} is the touchdown time for leg i, $k^0_{p_h}$ and $k^0_{d_h}$ are the PD hip position control gains from Table 2.5, and v_{b_i} is a velocity bias given by the following:

$$v_{b_i} = \begin{cases} v_{b_f} & \text{if} \quad i \leq 2 \\ v_{b_r} & \text{otherwise} \end{cases}, \tag{4.12}$$

where v_{b_f} and v_{b_r} are evolved front and rear velocity biases. At each time instant, a target angular velocity for virtual leg i is computed based on the desired running speed, plus a bias, divided by the current leg length. The integral of this quantity is then used to determine the error in position. Note that this approach differs slightly from that given in [30], where only proportional control over velocity was used. Using evolution, it was found that a negative bias in front and a positive bias in the rear produced the best results. This generates a "shoulder-braking" and "hip-thrusting" effect, which stabilizes the body's pitch during stance [30].

Due to limited surface friction (Sect. 2.2), the hip torque computed in Eq. (4.11) (and, to a lesser extent, the ab/ad torque) could lead to slipping of the foot contact during stance. When $\sigma_{\mathrm{slip}} = 1$, an algorithm is employed to reduce the slipping of the foot by bounding both the ab/ad and hip joint torques. This approach, which is necessary for stable galloping, is described in the next section.

The second control function performed during GALLOP-STANCE-CONTROL is the injection of energy into the knee springs to account for energy losses at impact and throughout the gait cycle. The target energy amount for each leg i is computed using the following discrete-time equation:

$$E_i[n] = \begin{cases} E_0\, d_f[n]\,(1 - d_l[n]) & \text{if } i = 1 \\ E_0\, d_f[n] d_l[n] & \text{if } i = 2 \\ E_0\,(1 - d_f[n])\,(1 - d_l[n]) & \text{if } i = 3 \\ E_0\,(1 - d_f[n])\, d_l[n] & \text{if } i = 4 \end{cases} \tag{4.13}$$

where E_0 is the total set point energy, and n is the n-th TOF. E_0 was set at $11.25\,J$ after determining experimentally that this value generated sufficient height to avoid

ground interference during leg transfer but resulted in minimal vertical excursion of the body's center of mass. The parameter $d_f[n]$ is a fore-aft energy distribution factor computed as follows:

$$d_f[n] = d_{f_0} + k_{p_\beta}\beta_e[n] + k_{d_\beta}\dot{\beta}_e[n] + k'_{p_\beta}\beta_e[n-1] + k'_{d_\beta}\dot{\beta}_e[n-1]\,, \tag{4.14}$$

where d_{f_0} is a bias fore-aft distribution factor, $\beta_e[n]$ and $\dot{\beta}_e[n]$ are pitch and pitch rate errors at TOF n, and $\beta_e[n-1]$ and $\dot{\beta}_e[n-1]$ are pitch and pitch rate errors at TOF $n-1$. The bias fore-aft energy distribution factor d_{f_0} and gains k_{p_β}, k_{d_β}, k'_{p_β}, and k'_{d_β} are found by evolution.

The parameter $d_l[n]$ in Eq. (4.13) is a lateral energy distribution factor and is computed as follows:

$$d_l[n] = d_{l_0} + k_{p_l}\gamma_e[n] + k_{d_l}\dot{\gamma}_e[n]\,, \tag{4.15}$$

where $d_{l_0} = 0.5$, $\gamma_e[n]$ and $\dot{\gamma}_e[n]$ are the roll and roll rate errors at TOF n, and gains k_{p_l} and k_{d_l} are found by evolution.

The control law in Eq. (4.14) shifts energy between front and rear legs based on pitch and pitch rate errors, while Eq. (4.15) shifts energy between right and left legs based on roll and roll rate errors. Equations (4.13) - (4.15) guarantee that the total energy is constant from cycle to cycle, which is critical for stabilizing the sagittal plane dynamics.

During the time from touchdown until maximum compression of the knee spring, the knee joint remains passive such that the natural response of the spring is the only torque acting at the joint. At maximum compression, the spring rest position is

instantaneously adjusted such that the total energy in the spring matches the target energy using the following relation:

$$E_i[n] = \frac{1}{2}k\left(\theta_{k_i} - \theta_{k_{0_i}}\right)^2 \ , \tag{4.16}$$

where $E_i[n]$ was computed in Eq. (4.13), k is the spring constant given in Table 2.1, θ_{k_i} is the current knee angle for leg i, and $\theta_{k_{0_i}}$ is the spring rest position for leg i. Energy is injected or withdrawn based on the spring state at maximum compression.

The last major control function performed during GALLOP-STANCE-CONTROL is maintaining the touchdown ab/ad angle relative to the body using PD control. Although more involved servos on body roll and yaw angles were investigated, it was found that simple position control on the ab/ad joint was sufficient to stabilize the system. The stance-phase ab/ad PD gains were hand-tuned for the desired performance and are given as follows: $k^s_{p_a} = 1077.7$ Nm/rad, and $k^s_{d_a} = 4.3176$ Nm/(rad/s). Like WAIT-FOR-TRIGGER and EARLY-RETRACTION, the period T is set to 1.0 s to prevent premature transitions, effectively forcing the function to terminate only when the foot breaks contact.

4.2.6 Enforcing the No-slip Condition

When $\sigma_{\text{slip}} = 1$, the ab/ad and hip torques are bounded such that the approximated resultant foot force remains within the friction *pyramid* to reduce slipping. A pyramid is employed instead of the standard friction cone to simplify the computation by decoupling each component of the foot force (Fig. 4.4). This avoids having to solve a coupled equation and dramatically simplifies the calculations. The foot force computations are done in the rotated frame $\{\boldsymbol{e}'\}$ (Sect. 2.4.4), which is obtained by rotating the inertial frame $\{\boldsymbol{e}\}$ by the yaw angle α about $\hat{\boldsymbol{z}}_e$ (Fig. 2.1). This resolves

the horizontal components of the foot reaction force in the body's forward and lateral heading directions. Using the pyramid approach, the $\hat{\boldsymbol{x}}_{e'}$ and $\hat{\boldsymbol{y}}_{e'}$ force components must each be less than $\mu F_z/\sqrt{2}$ such that the resultant force lies within the square region inscribed in Fig. 4.4.

The algorithm uses a simplified approach where a massless, virtual leg is used to approximate the induced contact force $\boldsymbol{F}_l$ acting on the ground due to the ab/ad and hip joint torques (Fig. 4.5). The knee spring is not included in the calculation, although the foot force from the compliant contact model (Sect. 2.2) $\boldsymbol{F}_c$ is measured in the rotated frame described above. The measured foot force accounts for the compression of the spring, which precludes having to compute the articulated-leg Jacobian to arrive at the resultant force. The $\hat{\boldsymbol{x}}_{e'}$ and $\hat{\boldsymbol{y}}_{e'}$ components of the total contact force must lie within the friction pyramid shown in Fig. 4.4. The total contact force is

$$\boldsymbol{F}_t = \boldsymbol{F}_c - \boldsymbol{F}_l \,, \tag{4.17}$$

where $\boldsymbol{F}_t = [F_t^x,\, F_t^y,\, F_t^z]^T$, $\boldsymbol{F}_c = [F_c^x,\, F_c^y,\, F_c^z]^T$, the measured reaction force acting on the foot, and $\boldsymbol{F}_l = [F_l^x,\, F_l^y,\, F_l^z]^T$, the resultant force acting on the ground induced by the ab/ad and hip torques τ_a and τ_h. Note that all three forces are resolved in the $\{\boldsymbol{e}'\}$ coordinate system (e.g., ${}^{e'}\boldsymbol{F}_t$), although the leading superscripts have been dropped for brevity. To further simplify the problem, the body's roll and pitch angles γ and β are ignored, and θ_a is assumed to be 90 deg.

The bounded torque for the hip is solved for first, starting with the following constraint:

$$|F_t^x| \leq c\frac{\mu}{\sqrt{2}}F_t^z \,, \tag{4.18}$$

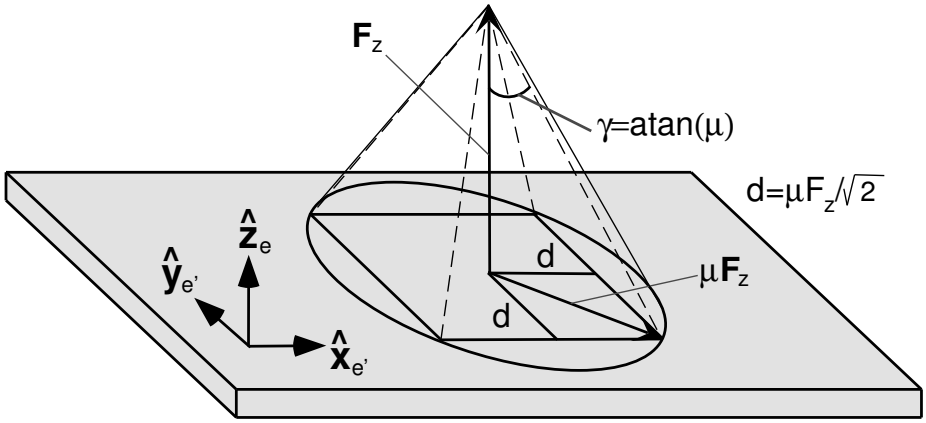


Figure 4.4: The friction pyramid approach for computing no-slip bounds for the ab/ad and hip torques. The axes are orientated such that the x and y directions are in the rotated frame $\{\boldsymbol{e'}\}$ obtained by rotating the inertial frame $\{\boldsymbol{e}\}$ about $\hat{\boldsymbol{z}}_e$ by the body's yaw angle α (Fig. 2.1). The pyramid is inscribed within the friction cone such that the $\hat{\boldsymbol{x}}_{e'}$ and $\hat{\boldsymbol{y}}_{e'}$ force components must individually be less than $\mu F_z/\sqrt{2}$.

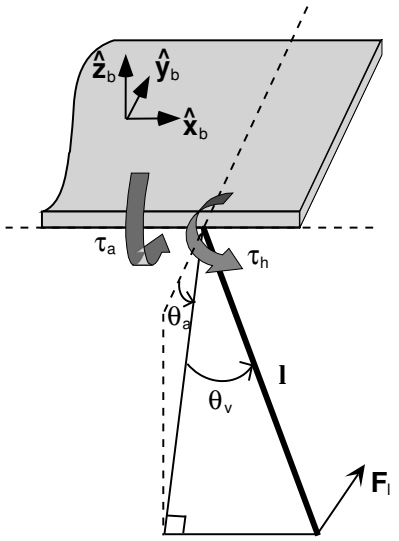


Figure 4.5: The simplified, virtual leg model for computing the approximate induced foot force from the ab/ad torque (τ_a) and hip torque (τ_h). The articulated leg is transformed to the virtual leg (Fig. 2.4) with length l and virtual leg angle θ_v, and the ab/ad angle is θ_a. The leg is considered massless, and static analysis is used to approximate the resultant foot force $\boldsymbol{F}_l$.

where c is an adjustment factor that accounts for pitch variations in the body during stance. The value of c was tuned experimentally to 0.7. The parameter μ is the friction coefficient, which is set to μ_s if the leg is not slipping, and μ_k when the leg is slipping, where $\mu_s = 1.0$ and $\mu_k = 0.8$ (Sect. 2.2). The $\hat{\boldsymbol{x}}$ component of the total contact force F_t^x is given as follows:

$$F_t^x = F_c^x - \frac{\tau_h}{l}\cos\left(\theta_v\right) , \tag{4.19}$$

while the $\hat{\boldsymbol{z}}$ component is given by

$$F_t^z = F_c^z - \frac{\tau_h}{l}\sin\left(\theta_v\right) , \tag{4.20}$$

where θ_v and l are the virtual leg angle and length shown in Fig. 4.5, and F_c^z is guaranteed to be non-negative by the contact model[17]. Solving Eq. (4.18) for τ_h, the following result is obtained:

$$\left(\frac{l}{c_v - c' s_v}\right)\left(-c' F_c^z + F_c^x\right) \leq \tau_h \leq \left(\frac{l}{c_v + c' s_v}\right)\left(c' F_c^z + F_c^x\right) , \tag{4.21}$$

where $s_v = \sin(\theta_v)$ and $c_v = \cos(\theta_v)$ (hereafter), and $c' = \frac{c\mu}{\sqrt{2}}$. Equation (4.21) is used to compute the minimum and maximum allowable hip torques $\tau_{h_{\min}}$ and $\tau_{h_{\max}}$, which are then used to determine the bounded hip torque τ_h^* as follows:

$$\tau_h^* = \begin{cases} \tau_{h_{\min}} & \text{if } \tau_{h_d} < \tau_{h_{\min}} , \\ \tau_{h_{\max}} & \text{if } \tau_{h_d} > \tau_{h_{\max}} , \\ \tau_{h_d} & \text{otherwise,} \end{cases} \tag{4.22}$$

where τ_{h_d} is the desired hip torque computed using Eq. (4.11).

[17]The compliant contact model uses a linear damper in the normal and lateral directions based on the velocity of the contact position. However, the normal damping coefficient can cause the total normal contact force to become negative as the normal velocity of the contact point becomes increasingly positive. This creates a "sticky" force, which is not a realistic condition. Consequently, the normal contact force is bounded at zero to prevent this condition.

Once the bounded hip torque τ_h^* has been computed, the bounded ab/ad torque is computed. The following constraint is the starting point for this calculation:

$$|F_t^y| \leq \frac{\mu\,(F_t^z)'}{\sqrt{2}}\,, \tag{4.23}$$

where $(F_t^z)'$ is the new estimate of the total normal contact force computed as follows:

$$(F_t^z)' = F_c^z - \frac{\tau_h^*}{l} s_v \quad . \tag{4.24}$$

The $\hat{\boldsymbol{y}}$ component of the total contact force is given by

$$F_t^y = F_c^y - \frac{\tau_a}{l\,c_v}\,, \tag{4.25}$$

and is substituted into Eq. (4.23) to yield the following:

$$l\,c_v \left(F_c^y - \frac{\mu}{\sqrt{2}}\,(F_t^z)' \right) \leq \tau_a \leq l\,c_v \left(F_c^y + \frac{\mu}{\sqrt{2}}\,(F_t^z)' \right) \quad . \tag{4.26}$$

Equation (4.26) is used to compute the minimum and maximum ab/ad torques $\tau_{a_{\min}}$ and $\tau_{a_{\max}}$, which are used to find the bounded ab/ad torque τ_a^*:

$$\tau_a^* = \begin{cases} \tau_{a_{\min}} & \text{if } \tau_{a_d} < \tau_{a_{\min}}\,, \\ \tau_{a_{\max}} & \text{if } \tau_{a_d} > \tau_{a_{\max}}\,, \\ \tau_{a_d} & \text{otherwise,} \end{cases} \tag{4.27}$$

where τ_{a_d} is the desired ab/ad torque computed using the PD parameters $k_{p_a}^s$ and $k_{d_a}^s$ given in the previous section.

While the method described above is simple, it is effective in reducing the slipping of the foot contact during stance during all but the first or last few milliseconds of contact. During these times, the normal component of the foot force is quite small, which causes the algorithm to compute small bounding values. However, these values are not sufficiently small to prevent the foot from slipping. A simple heuristic approach that would correct this problem would be to assume that the normal force

is zero if its value falls beneath some threshold ϵ_{F_z}, which could be determined experimentally. However, the slipping that does occur does not appear to cause any significant problems; consequently, no corrective action was deemed necessary.

4.2.7 FREE

After the leg breaks contact with the ground, the leg enters the FREE function, where the hip joint is allowed to rotate freely, which reduces jerk in the system after the leg lifts off. The variable σ_{a_f} (Table 2.4) is set to 0 so that the ab/ad joint is held in its initial position. Furthermore, the spring rest position, which was moved during GALLOP-STANCE-CONTROL, is now returned to its nominal value by setting $\sigma_s = 1$. This is done smoothly over 75% ($T_r = 0.75$) of the period T, which was experimentally tuned to $0.025\,s$. After the FREE function expires, the leg transitions back to the TRANSFER function with the spring rest position at its nominal value.

4.2.8 Gallop Control Summary

A summary of the parameters for each of the leg primitive functions employed for the gallop controller is given in Table 4.2. The evolved parameters listed in the table are found using the multiobjective genetic algorithm described in Chap. 3 and are discussed in Sect. 4.3.

4.2.9 Gallop Initialization

The INIT-TOF function, described in Sect. 2.4.4 is used to initialize the gallop. A summary of the INIT-TOF parameters is given in Table 4.3. The binary parameter σ_p is set to zero to indicate that the body parameters are not to be perturbed randomly.

Function	Parameters	Description
TRANSFER	$\sigma_{a_e} = 0,$ $\sigma_{h_e} = 0,$ $\sigma_{k_d} = 1,$ $\sigma_s = 0,$ $T_r = 0.0,$ $\boldsymbol{\theta}_{l_f} = [\theta_{a_i}, \theta_{v_i}, \frac{l_0}{2}]^T,$ $T_t = 1.0,$ $T = 0.1\,\mathrm{s}$	Transfer using target ab/ad angle θ_{a_i} from Eq. (4.1), and virtual leg angle θ_{v_i} given by $\theta_{v_i} = \begin{cases} 0.69 \text{ rad} & \text{if } i \leq 2 \\ 0.65 \text{ rad} & \text{otherwise,} \end{cases}$ where $i = 1, \ldots, 4$. Length l_0 is given in Table 2.1.
WAIT-FOR-TRIGGER	$L = L_i, \quad R = R_i,$ $\delta = \delta_i, \quad h_t = h_{t_i},$ $\sigma_{a_e} = 0, \quad \sigma_{h_e} = 0,$ $\sigma_{k_f} = 0, \quad \sigma_s = 0,$ $T_r = 0.0, \quad \boldsymbol{\theta}_{l_d} = [\theta_{a_i}, \theta_{v_i}, l_0]^T,$ $T = 1.0\,\mathrm{s}$	Hold using θ_{a_i} and θ_{v_i} from TRANSFER (above) using trigger parameters L_i, R_i, δ_i, and h_{t_i} given in Table 4.1 for leg i, $i = 1, \ldots, 4$.
EARLY-RE-TRACTION	$\sigma_{a_e} = 0,$ $\theta_a = \theta_{a_i},$ $v_d = 4.15\,\mathrm{m/s},$ $T = 1.0\,\mathrm{s}$	Hold ab/ad joint at desired position θ_{a_i} from Eq. (4.1) and rotate hip to achieve tangential foot velocity v_d.
GALLOP-STANCE-CONTROL	$v_d = 4.15\,\mathrm{m/s},$ $v_b = \begin{cases} v_{b_f} \text{ if } i \leq 2, \\ v_{b_r} \text{ otherwise,} \end{cases}$ $E_d = E_i,$ $\sigma_{\text{slip}} = 1,$ $T = 1.0\,\mathrm{s}$	Use velocity control with evolved velocity biases v_{b_f} and v_{b_r} for the front and rear legs, respectively. Use target energy value E_i computed from Eq. (4.13) for leg i, $i = 1, \ldots, 4$, and use slip control.
FREE	$\sigma_{a_f} = 0,$ $\sigma_s = 1,$ $T_r = 0.75,$ $T = 0.025\,\mathrm{s}$	Reset knee spring to the nominal position over 75% of period T while holding the ab/ad angle and allowing the hip to rotate freely.

Table 4.2: Summary of the leg function parameters for the gallop controller.

Parameter	Value
σ_p	0
σ_{a_e}	0
σ_{h_e}	0
σ_{k_d}	1
σ_{l_a}	0
$\boldsymbol{\theta}_b = [\ \alpha,\ \beta,\ \gamma,$ ${}^{e}x_b, {}^{e}y_b, {}^{e}z_b,$ $\dot{\alpha},\ \dot{\beta},\ \dot{\gamma},$ ${}^{e'}\dot{x}_b, {}^{e'}\dot{y}_b, {}^{e}\dot{z}_b\ ]^T$	$[\ 0.0\,\text{rad},\ 0.0\,\text{rad},\ 0.0\,\text{rad},$ $0.0\,\text{m},\ 0.0\,\text{m},\ 0.24\,\text{m},$ $0.0\,\text{rad/s},\ \dot{\beta}_d,\ 0.0\,\text{rad/s},$ $4.15\,\text{m/s},\ 0.0\,\text{m/s},\ 0.0\,\text{m/s}\ ]^T$
$\boldsymbol{\delta}_b$	$\mathbf{0}_{12\times 1}$
$\boldsymbol{\phi}_l = [\ \theta^1_{a_0},\ \ldots,\ \theta^4_{a_0},$ $\theta^1_{v_0},\ \ldots,\ \theta^4_{v_0},$ $d^1_0,\ \ldots,\ d^4_0,$ $\theta^1_{a_f},\ \ldots,\ \theta^4_{a_f},$ $\theta^1_{v_f},\ \ldots,\ \theta^4_{v_f},$ $d^1_f,\ \ldots,\ d^4_f\ ]^T$	$[\ \pi/2\,\text{rad},\ \ldots,\ \pi/2\,\text{rad},$ $-0.35\,\text{rad},\ \ldots,\ -0.35\,\text{rad},$ $0.24\,\text{m},\ \ldots,\ 0.24\,\text{m},$ $\pi/2\,\text{rad},\ \ldots,\ \pi/2\,\text{rad},$ $0.69\,\text{rad},\ 0.69\,\text{rad},\ 0.65\,\text{rad},\ 0.65\,\text{rad},$ $0.12\,\text{m},\ \ldots,\ 0.12\,\text{m}\ ]^T$
$\boldsymbol{\theta}_l$	$\mathbf{0}_{24\times 1}$
$\boldsymbol{t}^* = [t^*_1,\ t^*_2,\ t^*_3,\ t^*_4]^T$	$[0.3,\ 0.4,\ 0.6,\ 0.7]^T$
T_t	$0.1\,\text{s}$

Table 4.3: Summary of the INIT-TOF parameters for initializing the gallop. The symbol $\mathbf{0}_{m\times p}$ indicates an $m \times p$ array of zeros.

Consequently, $\boldsymbol{\delta}_b$ can be set to all zeros, as it will not be used by the INIT-TOF function. Parameters σ_{a_e} and σ_{h_e} are set to zero to indicate that spline computations for both the ab/ad and hip joints will not be computed with respect to the inertial frame. However, σ_{k_d} is set to 1 to indicate that a double-spline will be used for the knee joints. The transfer time for all splines, T_t, is set to 0.1 s to be consistent with the transfer time used during TRANSFER (Sect. 4.2.1). The parameter σ_{l_a} is set to zero, which indicates that the initial leg positions will be computed using the spline parameters in $\boldsymbol{\phi}_l$ instead of using the leg states specified in $\boldsymbol{\theta}_l$, the latter of which can be set to all zeros. The spline parameters in $\boldsymbol{\phi}_l$ are given as follows:

$$\begin{aligned} \boldsymbol{\phi}_l \;=\; & \Big[\theta^1_{a_0},\, \ldots,\, \theta^4_{a_0},\, \theta^1_{v_0},\, \ldots,\, \theta^4_{v_0},\, d^1_0,\, \ldots,\, d^4_0, \\ & \theta^1_{a_f},\, \ldots,\, \theta^4_{a_f},\, \theta^1_{v_f},\, \ldots,\, \theta^4_{v_f},\, d^1_f,\, \ldots,\, d^4_f\Big]^T \;, \end{aligned} \tag{4.28}$$

where $\theta^i_{a_0}$ and $\theta^i_{a_f}$ are the i-th initial and final ab/ad angles, $\theta^i_{v_0}$ and $\theta^i_{v_f}$ are the i-th initial and final virtual leg angles, and d^i_0 and d^i_f are the i-th initial and final virtual leg lengths. $\theta^i_{a_0}$ and $\theta^i_{a_f}$ were both set to nominal values of $\frac{\pi}{2}$, while $\theta^i_{v_0}$ was set to -0.35 rad after some preliminary trials. The final virtual leg angle $\theta^i_{v_f}$ was set to 0.69 rad for the front legs and 0.65 rad for the rear after preliminary evolution (Sect. 4.2.1). The initial and final virtual leg lengths d^i_0 and d^i_f were set to l_0 (Table 2.1) and $l_0/2$, respectively, to specify a double-spline ($\sigma_{k_d} = 1$) with length $l_0/2$ at the midpoint, $t = T_t/2$.

Because $\sigma_{l_a} = 0$, the actual starting joint angles and rates are computed using the specified transfer splines and the times given in $\boldsymbol{t}^*$:

$$\boldsymbol{t}^* = [t^*_1,\, t^*_2,\, t^*_3,\, t^*_4]^T \;, \tag{4.29}$$

where $t_i^* \in [0.0, 1.0]$, for $i = 1, \ldots, 4$, and represents an estimate of the average degree of completion of the transfer spline for the i-th leg at the TOF. After some preliminary experimentation, the following values were determined: $t_1^* = 0.3$, $t_2^* = 0.4$, $t_3^* = 0.6$, and $t_4^* = 0.7$.

Finally, the body state at TOF $\boldsymbol{\theta}_b$ must be specified, where $\boldsymbol{\theta}_b$ is given in Eq. (2.5). The orientation of the quadruped is initialized such that the yaw α, pitch β, and roll γ are all zero, as are the yaw rate $\dot{\alpha}$ and roll rate $\dot{\gamma}$. However, the pitch rate $\dot{\beta}$ is set to the desired rate $\dot{\beta}_d$, which is found from evolution. The position of the COM in the horizontal plane is arbitrary and typically set to ${}^e x_b = {}^e y_b = 0.0$, while the height of the COM ${}^e z_b$ is set to the nominal running height of 0.24 m (l_0, Table 2.1). The lateral velocity of the COM ${}^{e'}\dot{y}_b$ and the vertical velocity in earth coordinates ${}^e\dot{z}_b$ are both set to zero, while the forward velocity ${}^{e'}\dot{x}_b$ is set to the desired running speed of 4.15 m/s (Sect. 4.2.3).

4.3 The Evolutionary Optimization Problem

The control parameters and initial pitch rate described in the previous sections are found using the multiobjective genetic algorithm described in Chap. 3. The evolvable parameters are given as follows:

$$\boldsymbol{\phi} = \left[d_{f_0},\, v_{b_f},\, v_{b_r},\, \dot{\beta}_d,\, k_{p_\gamma},\, k_{d_\gamma},\, f_\gamma,\, k_{p_l},\, k_{d_l},\, k_{p_\alpha},\, k_{d_\alpha},\, f_\alpha,\, k_{p_\beta},\, k_{d_\beta},\, k'_{p_\beta},\, k'_{d_\beta} \right]^T, \quad (4.30)$$

where d_{f_0} is the bias fore-aft energy distribution factor referenced in Eq. (4.14), v_{b_f} and v_{b_r} are the forward and rear velocity biases referenced in Eq. (4.12), $\dot{\beta}_d$ (Table 4.3) is the desired pitch rate of the body used to compute pitch rate errors in Eq. (4.14), k_{p_γ}, k_{d_γ}, f_γ, k_{p_α}, k_{d_α}, and f_α are the ab/ad touchdown angle gains for roll and yaw errors referenced in Eqs. (4.1) and (4.3), k_{p_l} and k_{d_l} are the lateral energy distribution

Parameter	Range		Parameter	Range	
d_{f_0}	[0.25, 0.40]		k_{d_l}	[0.0, 1.5]	s/rad
v_{b_f}	[−0.30, −0.20]	m/s	k_{p_α}	[0.0, 1.0]	rad/rad
v_{b_r}	[0.0, 0.75]	m/s	k_{d_α}	[0.0, 0.5]	rad/(rad/s)
$\dot{\beta}_d$	[−0.50, 0.0]	rad/s	f_α	[0.0, 0.25]	
k_{p_γ}	[0.0, 3.0]	rad/rad	k_{p_β}	[−5.0, 0.0]	rad^{-1}
k_{d_γ}	[0.0, 1.5]	rad/(rad/s)	k_{d_β}	[−0.50, 0.50]	s/rad
f_γ	[0.0, 1.0]		k'_{p_β}	[0.0, 5.0]	rad^{-1}
k_{p_l}	[0.0, 3.0]	rad^{-1}	k'_{d_β}	[−0.50, 0.50]	s/rad

Table 4.4: Ranges for each evolved parameter for the gallop.

gains for roll error referenced in Eq. (4.15), and k_{p_β}, k_{d_β}, k'_{p_β}, and k'_{d_β} are the fore-aft energy distribution gains for pitch error referenced in Eq. (4.14). The search range for each parameter is given in Table 4.4 and was determined experimentally. In particular, the ranges for the first four parameters (d_{f_0}, v_{b_f}, v_{b_r}, and $\dot{\beta}_d$), which affect the sagittal plane dynamics, were restricted because values outside of these ranges produced poor results during early trials.

Although all the parameters in Eq. (4.30) could be found simultaneously using the MOGA, it was observed that a staged approach worked best. In the first stage, the first four parameters in Eq. (4.30), d_{f_0}, v_{b_f}, v_{b_r}, and $\dot{\beta}_d$, are evolved. These parameters all affect the sagittal plane dynamics of the system. After these parameters are evolved, the roll control parameters, k_{p_γ}, k_{d_γ}, f_γ, k_{p_l}, and k_{d_l}, are evolved, followed by the yaw control parameters, k_{p_α}, k_{d_α}, and f_α. Finally, the pitch control parameters, k_{p_β}, k_{d_β}, k'_{p_β}, and k'_{d_β}, are evolved to further stabilize the sagittal plane dynamics.

The motivation for partitioning the parameter space is to facilitate the search for good solutions by reducing the dimensionality of the space. Although this approach is not applicable to all problems, the partitioning is implemented by dividing the 3D gallop controller into smaller sub-controllers operating in approximately orthogonal subspaces. The degree of coupling between each sub-controller is small enough to facilitate a decoupled partitioning of the parameter space. However, intermediate solutions are stable for only a limited time, and the final stage of evolution for the pitch control parameters is necessary to stabilize the sagittal plane from the disturbances induced by the roll and yaw sub-controllers.

The final element of the evolutionary optimization problem is the development of the fitness function. Three basic criteria are used to evaluate gallop solutions: accuracy, stability, and correctness. The fitness vector is given by

$$\boldsymbol{f} = [f_a,\, f_s,\, f_c]^T \ , \tag{4.31}$$

where f_a, f_s, and f_c are the accuracy, stability, and correctness components.

The accuracy component refers to the Euclidean distance between the average TOF state vector and the desired state. The fitness score for accuracy is computed as follows:

$$f_a = \frac{w_a}{1 + \| \boldsymbol{\Lambda} \left(\bar{\boldsymbol{\theta}}_b - \boldsymbol{\theta}_b \right) \|^2} \tag{4.32}$$

where $\bar{\boldsymbol{\theta}}_b$ is the TOF state of the body averaged across all strides, $\boldsymbol{\theta}_b$ is the nominal TOF state given in Table 4.3, and w_a is a weight for accuracy, set to 100.0 to provide adequate fitness scaling for this component based on the fitness sharing parameters

Parameter	Value	Units	Parameter	Value	Units
λ_α	28.6	rad^{-1}	$\lambda_{\dot{\alpha}}$	5.7	$(\text{rad/s})^{-1}$
λ_β	11.5	rad^{-1}	$\lambda_{\dot{\beta}}$	5.7	$(\text{rad/s})^{-1}$
λ_γ	28.6	rad^{-1}	$\lambda_{\dot{\gamma}}$	5.7	$(\text{rad/s})^{-1}$
λ_x	0.0	m^{-1}	$\lambda_{\dot{x}}$	10.0	$(\text{m/s})^{-1}$
λ_y	0.0	m^{-1}	$\lambda_{\dot{y}}$	10.0	$(\text{m/s})^{-1}$
λ_z	200.0	m^{-1}	$\lambda_{\dot{z}}$	0.0	$(\text{m/s})^{-1}$

Table 4.5: Scaling factors used to compute the accuracy fitness component for the gallop. Scaling factors of zero indicate "don't-care" conditions.

discussed below. Finally, $\mathbf{\Lambda}$ is a 12×12 diagonal matrix of scaling factors, where the scaling factors are given in $\boldsymbol{\lambda}_d$ as follows:

$$\boldsymbol{\lambda}_d = \left[\lambda_\alpha,\, \lambda_\beta,\, \lambda_\gamma,\, \lambda_x,\, \lambda_y,\, \lambda_z,\, \lambda_{\dot{\alpha}},\, \lambda_{\dot{\beta}},\, \lambda_{\dot{\gamma}},\, \lambda_{\dot{x}},\, \lambda_{\dot{y}},\, \lambda_{\dot{z}}\right]^T \quad . \tag{4.33}$$

The elements of $\boldsymbol{\lambda}_d$ are given in Table 4.5. Each element represents the inverse of what is considered an acceptable tolerance for each variable. For example, the tolerance for λ_α is $1.0/28.6 = 0.035\,\text{rad}$, or about $2.0\,\text{deg}$. Several variables are considered "don't-care" conditions, including ${}^e x_b$, ${}^e y_b$, and ${}^e \dot{z}_b$. The first two specify the quadruped's position on the horizontal plane, while the last is the vertical velocity, which will always be approximately zero at TOF. Thus, the scaling factors for these variables are set to zero.

For the stability component, a stability metric is computed that penalizes each solution based on two things: (1) the dispersion of each TOF state variable from

its linear regression line, and (2) the magnitude of the regression line's slope. The stability score is computed as follows:

$$f_s = \frac{w_s N_{s_c}}{1 + \sum_{j=1}^{12} \lambda_j \left(w_m m_j + w_{\bar{\epsilon}} \bar{\epsilon}_j \right)} \ , \tag{4.34}$$

where $w_s = 5.0$, N_{s_c} is the number of error-free strides completed (error conditions are discussed below), m_j is the linear regression slope (i.e., the drift error) of the j-th element of $\boldsymbol{\theta}_b$ in Eq. (2.5), w_m is the drift error weighting factor, $\bar{\epsilon}_j$ is the root mean square (RMS) error for the j-th element of $\boldsymbol{\theta}_b$, $w_{\bar{\epsilon}}$ is the dispersion error weighting factor, and λ_j is the j-th element in $\boldsymbol{\lambda}_d$ in Eq. (4.33). The scaling factor w_s was tuned to produce adequate scaling along this dimension of the fitness space, while the weights w_m and $w_{\bar{\epsilon}}$ were tuned to 0.46 and 0.15, respectively, to balance the contributions of the drift error and the RMS error.

The correctness factor is computed as follows:

$$f_c = \left(w_c \sum_{j=1}^{N_s} \left(\sum_{k=1}^{4} t_{c_{k_j}} - t^*_{e_j} \right) \right) / N_s \ , \tag{4.35}$$

where w_c is a weight for correctness, N_s is the total number of strides (correct and otherwise), $t_{c_{k_j}}$ is the time elapsed before an error condition was detected for leg k during stride j (e.g., untriggered touchdown, out-of-order footfall sequence, etc.), and $t^*_{e_j}$ is the time from the first detected error among all legs until the end of stride j. (Note that the timer for each stride is reset at TOF.) The weight factor w_c was set to 1.0 to provide adequate fitness scaling. In general, the correctness factor rewards solutions with more error-free strides.

Finally, the fitness sharing parameters $\boldsymbol{\rho}_{\text{sh}}$ are given as follows:

$$\boldsymbol{\rho}_{\text{sh}} = \left[\rho_a, \, \rho_s, \, \rho_c \right]^T \ , \tag{4.36}$$

where ρ_a, ρ_s, and ρ_c are the accuracy, stability, and correctness sharing parameters, respectively. All sharing parameters were set to 6.25 to produce approximately 16 clusters if the maximum fitness in each dimension is approximately 100.0. The weighting values w_a, w_s, and w_c, described above, were set to provide sufficient scaling of each of the fitness dimensions, although it is difficult to predict maximum fitness values that might be encountered during an evolutionary trial. To reduce the number of tunable parameters, the fitness sharing values are generally kept constant at the values above, while the scaling factors for the fitness criteria are adjusted during preliminary trials in order to produce sufficient scaling for each axis.

4.4 Results

The results of evolving the control parameters listed in Eq. (4.30) are presented in this section. A population size of 32 individuals and 250 generations were selected for the evolution. Multiple trials were run for each stage, with some stages requiring up to twenty trials before suitable solutions were found. The final evolved values are listed in Table 4.6. This solution represents the best stability fitness achieved across all trials[18]. The following sections present both high-level and detailed analysis of the dynamics of the resulting gait.

4.4.1 Stride-Level Analysis

Figure 4.6 shows the TOF state variables for the first 60 seconds of 3D galloping. In Fig. 4.6 (a), the yaw error reaches a maximum value of approximately 0.125 rad,

[18]The Pareto front for the gallop was fractured and discontinuous, yielding only a few individuals from which to choose. Additional adjustment of the fitness function and sharing parameters would be necessary to generate a more continuous front. However, since the selected solution had good stability, further evolution was not required.

Parameter	Value	Parameter	Value
d_{f_0}	0.33	k_{d_l}	0.45 s/rad
v_{b_f}	-0.23 m/s	k_{p_α}	0.05 rad/rad
v_{b_r}	0.52 m/s	k_{d_α}	0.04 rad/(rad/s)
$\dot{\beta}_d$	-0.47 rad/s	f_α	0.23
k_{p_γ}	1.21 rad/rad	k_{p_β}	-1.75 rad^{-1}
k_{d_γ}	0.01 rad/(rad/s)	k_{d_β}	0.09 s/rad
f_γ	0.17	k'_{p_β}	3.10 rad^{-1}
k_{p_l}	1.45 rad^{-1}	k'_{d_β}	0.05 s/rad

Table 4.6: Evolved parameters for the gallop.

which leads to an initial error in heading in (d). After the system settles to steady-state, however, the quadruped gallops in a straight line with no further change in heading. Like yaw, Fig. 4.6 (c) shows that there is also a small roll error at TOF. The positive roll rate in (h), however, causes the body to touch down with a consistently small roll angle.

The PD roll and yaw gains (k_{p_γ}, k_{d_γ}, f_γ, k_{p_l}, k_{d_l}, k_{p_α}, k_{d_α}, and f_α) listed in Table 4.6 are largely responsible for stabilizing both of these variables. In fact, experiments have shown that these variables cannot be stabilized without some form of active roll and yaw control. It was also discovered that the outward leg-rotation roll strategy seems to yield more stable results than simpler strategies where all ab/ad angles are adjusted the same way for a given error. As shown in Eq. (4.1), the secondary roll compensation angle is computed by multiplying f_γ by the primary leg angle. Initially, the range for f_γ was set to $[-1.0, 1.0]$, which generated a variety of different roll control strategies, including the case where both primary and secondary ab/ad

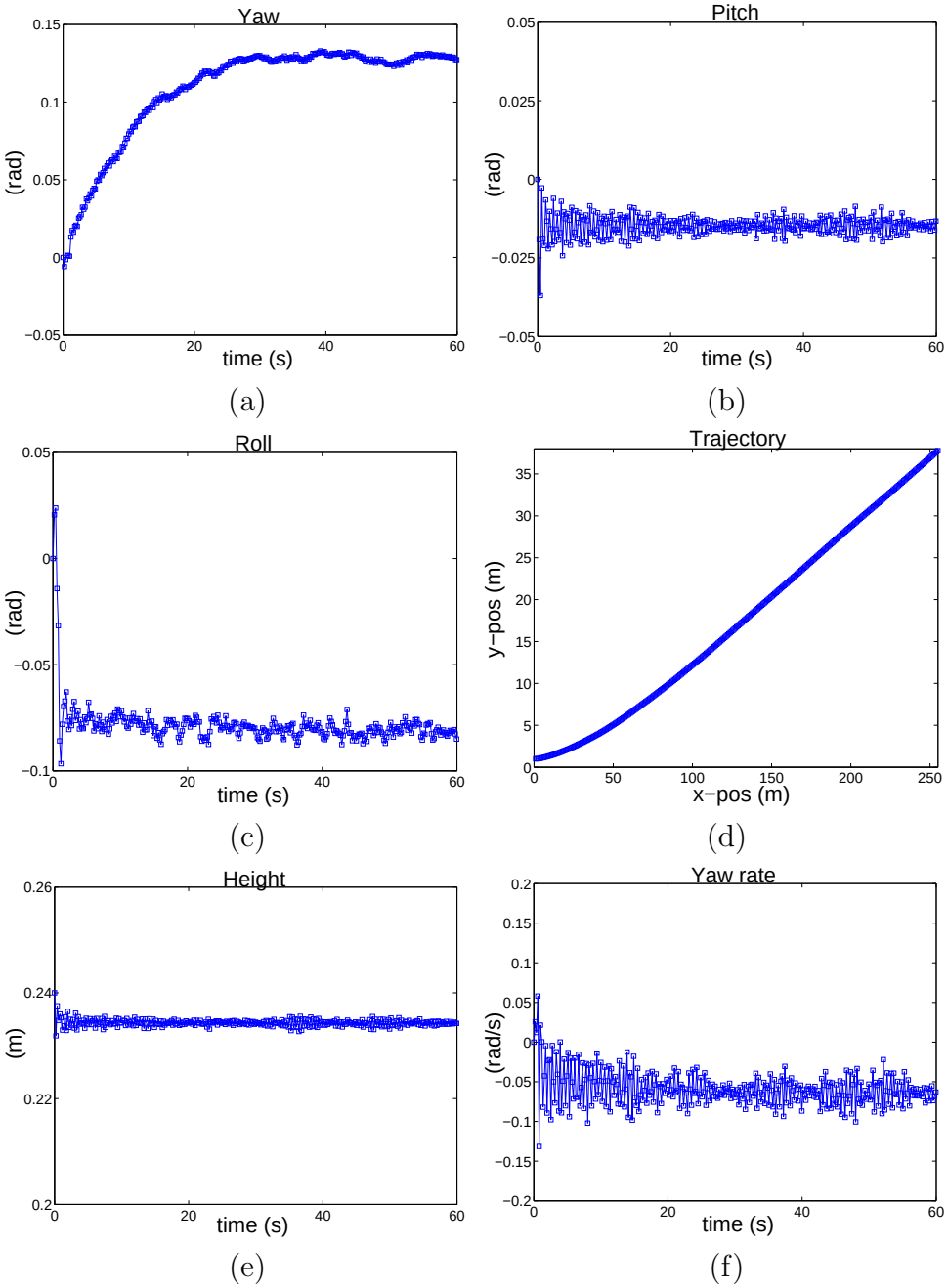


Figure 4.6: Top-of-flight state variables for the 3D gallop for the first 60 sec. (cont'd on next page).

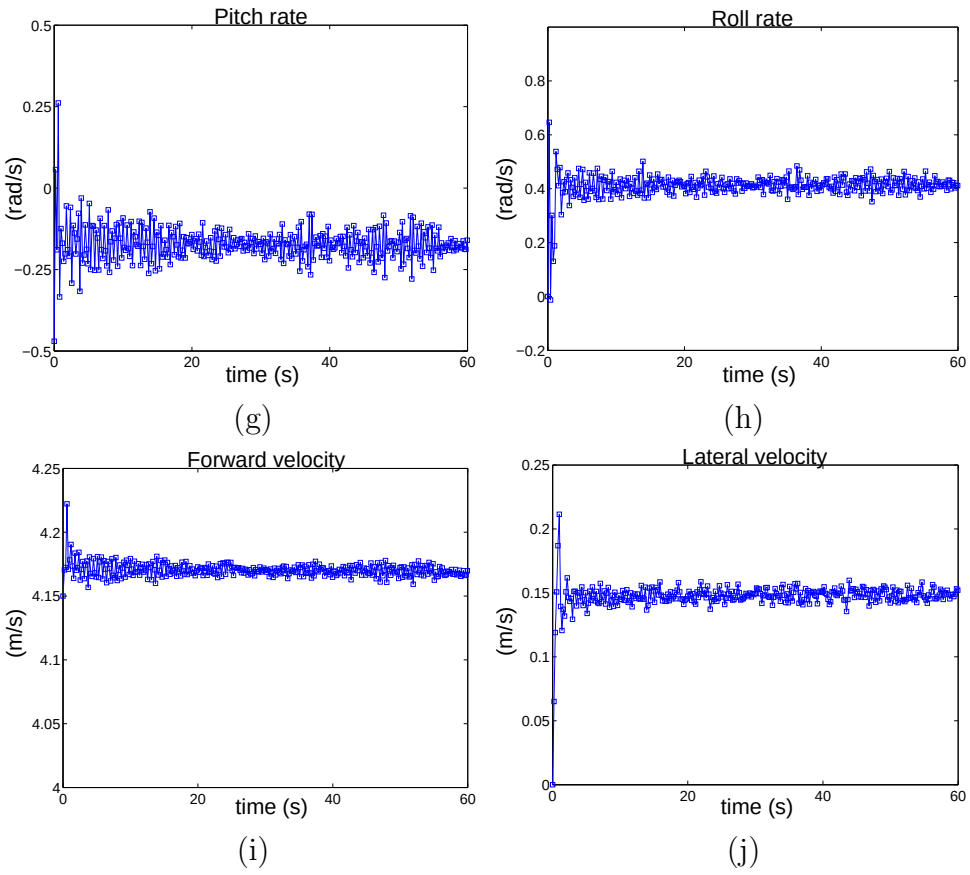


Figure 4.6: Top-of-flight state variables for the 3D gallop for the first 60 sec. (cont'd from previous page).

angles have the same sign. After a number of trials, however, it was discovered that the best solutions involved positive values for f_γ, which produces a sprawled posture to compensate for roll errors. The same sprawled posture result was found for f_α, which scales the rear leg ab/ad angle for yaw control.

In the sagittal plane, Fig. 4.6 (e) shows that the system settles out to a steady-state running height of approximately 0.235 m. As stated in Sect. 4.2.5, it was found that using a total energy of $E_0 = 11.25$ J consistently produced adequate height for ground clearance during leg transfer, yet it generated a smooth gait with minimal excursion

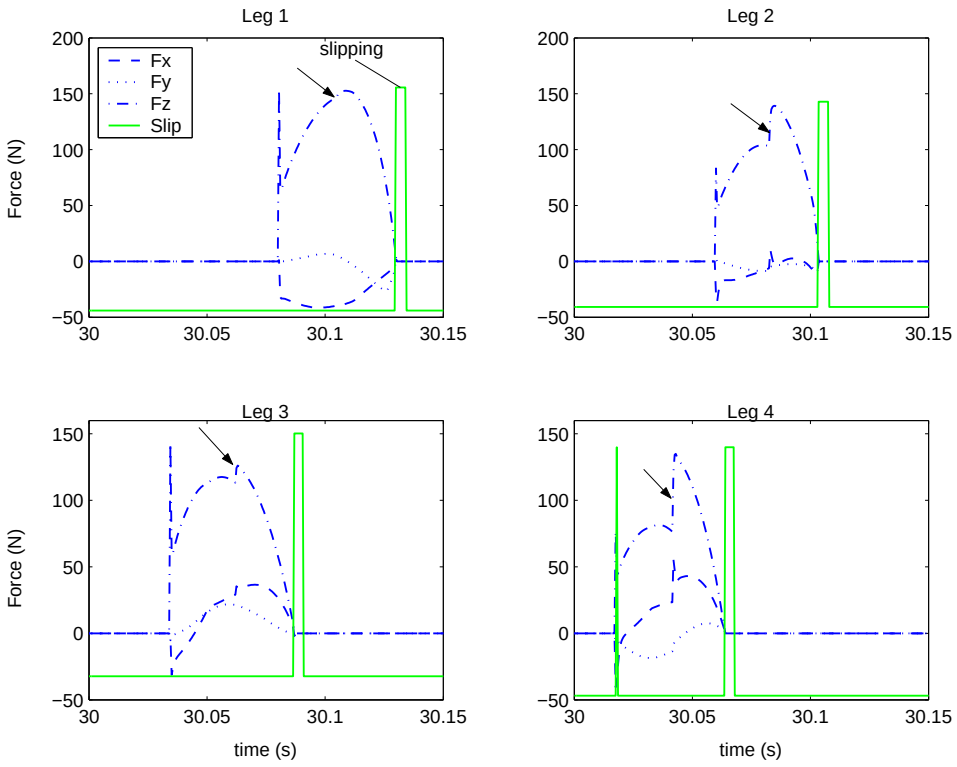


Figure 4.7: Foot forces for the 3D gallop. Arrows indicate injection of energy, and the square waveform indicates slipping when it goes high. The no-slip algorithm limits slipping to all but the first or last few milliseconds of contact.

of the center of mass. Larger energy values generated larger height excursions. At $E_0 = 11.25\,\mathrm{J}$, the peak vertical foot forces were limited to $150.0\,\mathrm{N}$, or about 1.6 times the total weight, as shown in Fig. 4.7. This force level seems to agree well with biological data, as Herr et al. [88] computed a peak vertical ground reaction force for a similarly sized galloping animal of approximately twice the weight.

In Table 4.6, the evolved value for d_{f_0} is $33\,\%$, which indicates that most of the energy in E_0 is biased toward the rear legs. After performing a number of evolutionary searches for the open-loop sagittal plane parameters, it was discovered that the d_{f_0} parameter must achieve a balance between front and rear energy injection based on

several factors: (1) the asymmetry of the body, (2) the front and rear velocity biases, and (3) the average TOF running height. The first factor is obvious, and the second is explained below. The third factor, however, indicates that there appears to be a fixed and predictable relationship between running height and pitch rate. Indeed, larger running heights required larger pitch rates, given the same leg compliance value. This would also suggest that very bouncy gallops, where the excursion of the center of mass is larger, also require larger TOF pitch rates.

As shown in Fig. 4.6, the average TOF pitch rate for the gallop is $-0.15\,\mathrm{rad/s}$, which means that the quadruped is rotating in a nose-up direction after the support phase. This value is considerably smaller in magnitude than pitch rates found in other studies, however. Marhefka [89] was able to stabilize a planar gallop with an average pitch rate of about $2.18\,\mathrm{rad/s}$, while Berkemeier [43] found stability regions for the bound that started at $1.25\,\mathrm{rad/s}$ and increased as the pitch moment of inertia decreased. In both of these cases, however, the dimensionless pitch moment of inertia[19] was less than 1.0. In this study, the dimensionless inertia of the body is greater than 1.0 if one considers the mass of the ab/ad joints[20]. In addition, both Berkemeier and Marhefka used gaits that contained significant extended flight phases. The extended flight mode was not selected for this study, however, as early experiments showed that both average pitch rate and galloping height were considerably larger, resulting in a gait that lacked the smoothness of biological gallops.

Finally, Fig. 4.6 (i) shows that the average forward running velocity is very close to the desired value of $4.15\,\mathrm{m/s}$. The front and rear velocity servos are responsible

[19]The dimensionless pitch inertia is $\bar{I} = I/ml^2$, where l is half of the shoulder-hip distance, I is the computed pitch inertia of the trunk, m is the mass of the trunk, and leg mass is neglected.

[20]The dimensionless pitch inertia for this model is 1.18, using a composite-rigid-body approach for the ab/ad joints and neglecting leg mass.

for stabilizing forward velocity. As stated in [30], the use of shoulder-braking and hip-thrusting, which occurs when the front and rear biases are negative and positive, respectively ($v_{b_f} = -0.23$ and $v_{b_r} = 0.52\,\mathrm{m/s}$ in Table 4.6), also serves to stabilize the pitch of the body during stance. Vertical ground reaction forces are largely responsible for generating the pitch moments during the support phase. More is said about this in Sect. 4.4.2. The hip thrust torque retards the nose-down pitching motion of the body during rear leg support, while the shoulder-braking torque retards the nose-up pitching motion during front leg support. Although the ranges for both velocity biases were initially bilateral, they were later restricted since the best solutions almost always involved shoulder-braking and hip-thrusting to better stabilize pitch.

4.4.2 Sub-Stride Dynamics

For a better understanding of the mechanics of the gallop, it is necessary to consider the dynamics in relation to the contact phases for each leg. Figure 4.8 shows waveforms for several body state variables over two strides with each of the leg functions superimposed as square waveforms. The heavy horizontal bars where the leg waveforms overlap indicate periods of single- and multiple-leg support. Analyzing the sagittal plane dynamics in terms of the individual leg contact phases is fairly straightforward. When the leading hind leg touches down ("TD 4" in Fig. 4.8 (a), $t = 30.025\,\mathrm{s}$), the pitch rate in (f) increases because of the moment induced about the body by the rear leg contact forces. The maximum pitch rate occurs just after leg 2 touches down ("TD 2" in plot (a), $t = 30.06\,\mathrm{s}$). At this point, the body begins pitching in the nose-up direction, indicated by a decrease in pitch rate. When all legs have broken contact ($t = 30.125\,\mathrm{s}$), the pitch rate increases again, reaching a local

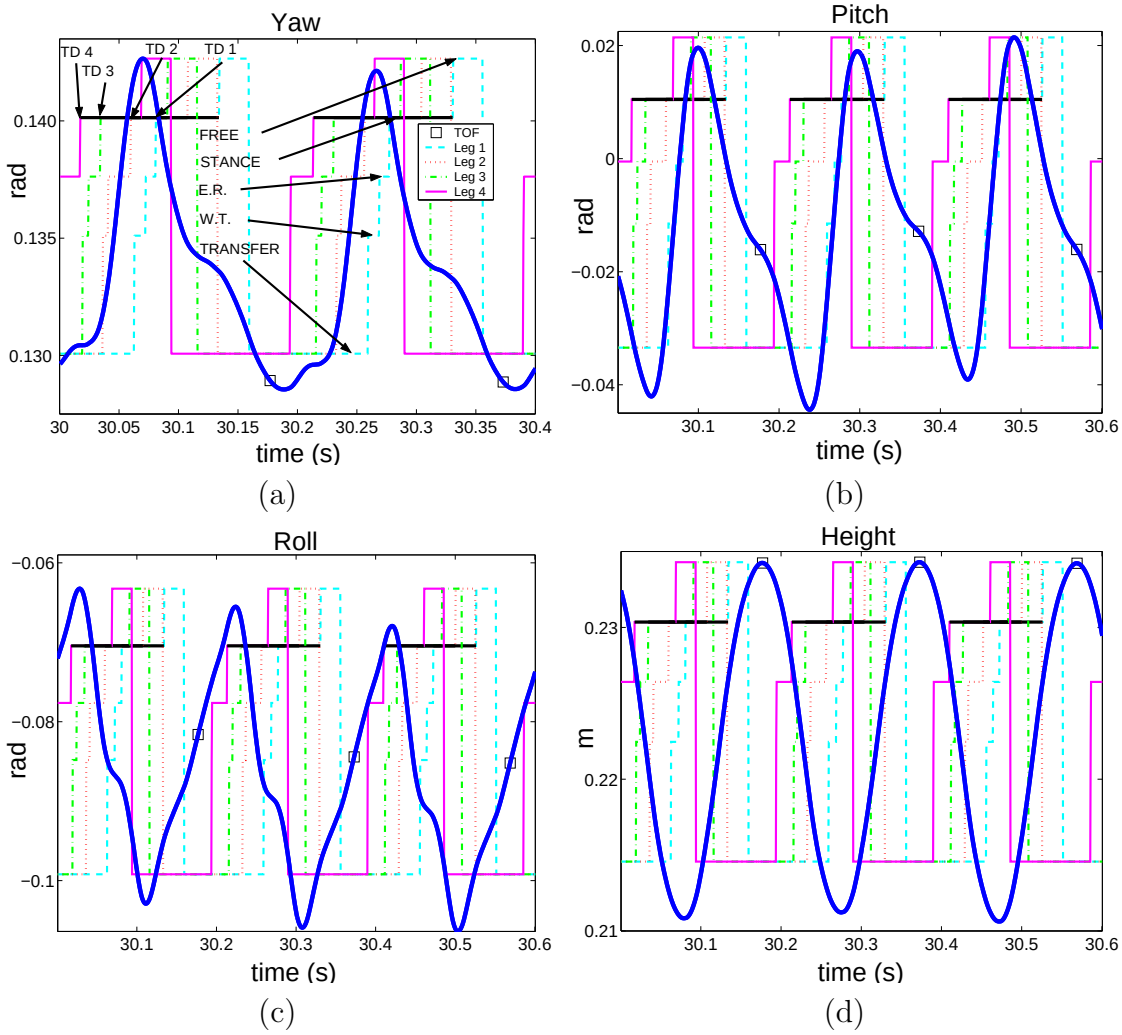


Figure 4.8: Results for the 3D gallop over three strides. Leg functions are superimposed as square waveforms with different levels for each of the different leg functions (Fig. 4.1). The five functions are labeled for leg 1 in (a), with the heavy horizontal bars, where multiple waveforms overlap, indicating contact phases. Note that "E.R.", "W.T.", and "STANCE" refer to EARLY-RETRACTION, WAIT-FOR-TRIGGER, and GALLOP-STANCE-CONTROL, respectively. Touchdown times for each leg (TD1 - TD4) also appear in (a). (Cont'd on next page).

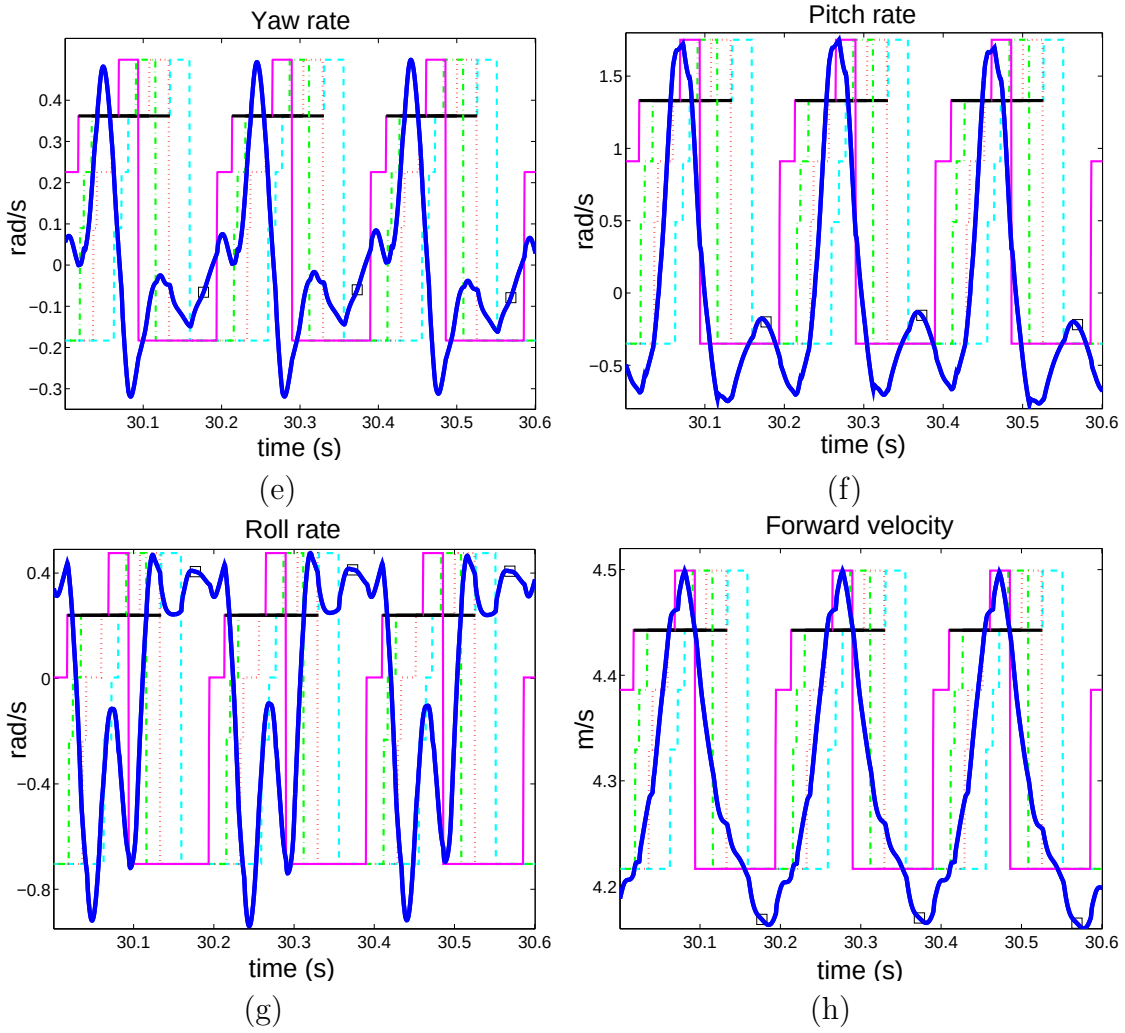


Figure 4.8: Results for the 3D gallop over three strides (cont'd from previous page).

maximum point at TOF. At this time, all of the legs are swinging forward, causing the body to pitch nose-down due to conservation of angular momentum. After TOF, the pitch rate begins to decline again, especially after the leading hind leg begins its early retraction ($t = 30.19\,\mathrm{s}$). The trajectory of the pitch rate curve causes a minimum (nose-up) pitch angle (Fig. 4.8 (b)) shortly after the leading hind leg touches down ($t = 30.24\,\mathrm{s}$) and a maximum (nose-down) angle about halfway through the double-support phase of the front legs ($t = 30.30\,\mathrm{s}$).

The remaining sagittal plane dynamics are described by the vertical fluctuation of the body's center of mass and by its forward velocity (Fig. 4.8 (d) and (h), respectively). The height fluctuates smoothly from the maximum at TOF to the minimum about midway through the stance period ($t = 30.08\,\mathrm{s}$). The excursion of the center of mass is less than $0.025\,\mathrm{m}$, which is about $10\,\%$ of the nominal standing height. This value seems to agree with data on galloping horses, where the excursion of the center of mass was about $0.15\,\mathrm{m}$, or $9\,\%$, of the shoulder height of $1.65\,\mathrm{m}$, according to Minetti et al. [28].

Unlike height, the forward velocity reaches a maximum during stance shortly after the leading front leg touches down ($t = 30.08\,\mathrm{s}$). Again, this seems to agree with biological data, as Minetti et al. also observed that the forward kinetic energy for galloping horses peaked just after the leading front leg (leg 2) touched down. The gravitational potential energy in biological gallops appears largely out of phase with the forward kinetic energy, which also holds true for this model. On a more fundamental level, one can see why the velocity peaks after the rear legs touch down by considering the effects of the velocity servos. During the rear leg support phase, the velocity ramps up quickly because of the positive velocity bias of $0.52\,\mathrm{m/s}$ (Table 4.6) applied to the velocity servos, which creates a hip-thrusting torque. When the front legs touch down and the rear legs break contact, however, the front velocity servos dominate. Because they have a negative bias of $-0.23\,\mathrm{m/s}$ (Table 4.6), a shoulder-braking torque is generated, and the velocity decreases.

While the sagittal plane dynamics are fairly straightforward to analyze, the dynamics of the transverse and dorsal planes are more difficult because of the cross-coupling between roll, yaw, and the sagittal plane parameters. However, an analysis

of the yaw rate and roll rate plots (Fig. 4.8 (e) and (g), respectively) indicates that yaw rate fluctuates more slowly than the roll rate. This occurs because of two factors. First, the effective yaw moment of inertia of the body (I_{eff}^{zz}) is larger than its effective roll moment (I_{eff}^{xx}) [21]. Second, the net moments generated by foot contact forces are larger about the roll axis. As shown in Fig. 4.7, the vertical component of the contact force is, by far, the most dominant. Using simple mechanics, the maximum roll moment for a leg can be approximated using the following equation:

$$M_x = r_y F_z - r_z F_y \,, \tag{4.37}$$

where r_y and r_z are the vertical and lateral force moment arms, and F_z and F_y are the maximum vertical and lateral forces. Setting $r_y = 0.07\,\text{m}$ ($d/2$, Table 2.1), $r_z = -0.24\,\text{m}$ ($-l_0$, Table 2.1), $F_z = 125.0\,\text{N}$ and $F_y = 25.0\,\text{N}$ (Leg 3 [22], Fig. 4.7), the maximum roll moment is $14.75\,\text{N m}$. On the other hand, the maximum yaw moment can be approximated using the following:

$$M_z = r_x F_y - r_y F_x \,, \tag{4.38}$$

where r_x and r_y are the moment arms for the lateral and fore-aft forces, and F_x and F_y are the fore-aft and lateral forces. Setting $r_y = 0.07\,\text{m}$ ($d/2$, Table 2.1), $r_x = -0.20\,\text{m}$ ($-l_r$, Table 2.1), $F_x = 25.0\,\text{N}$ (Leg 3, Fig. 4.7), and F_y as given above, the magnitude of the maximum yaw moment is $6.75\,\text{N m}$. The combination of a larger yaw inertia and smaller yaw moment suggests that the body will undergo smaller acceleration about the yaw axis, which yields smaller fluctuations in yaw angular velocity. Consequently,

[21] Effective moments of inertia for the trunk are computed using a composite-rigid-body approach for the ab/ad joints: $I_{\text{eff}}^{zz} = 0.30\ \text{kg m}^2$ and $I_{\text{eff}}^{xx} = 0.15\ \text{kg m}^2$. Leg mass is neglected, although its inclusion does not change the overall result.

[22] Roll and yaw moments were computed for each leg at maximum leg compression. For both cases, Leg 3 generated the largest magnitude moments.

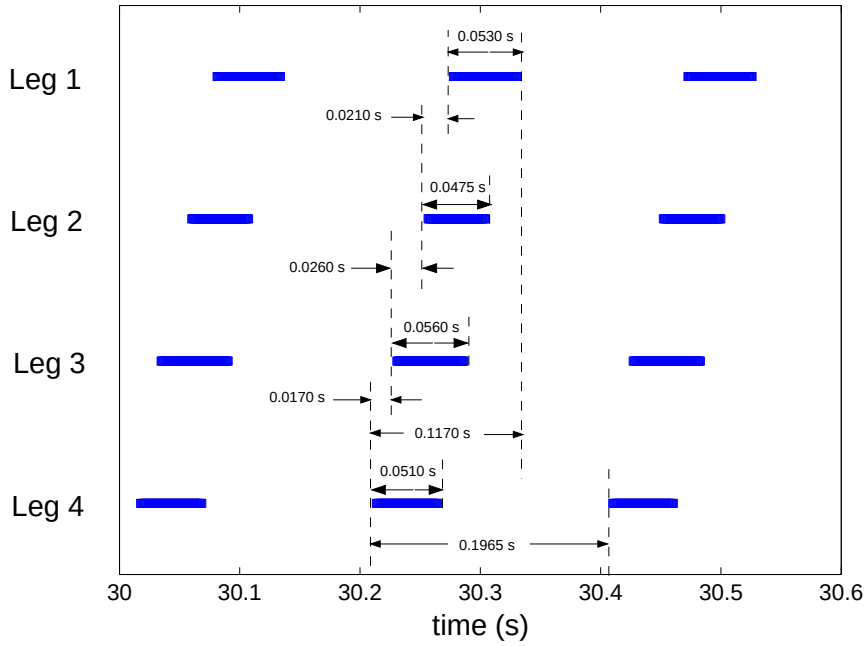


Figure 4.9: Transverse footfall sequence for the 3D gallop over three strides.

the yaw axis of the body may be easier to stabilize as it naturally has less sensitivity to variations in contact forces[23].

Another important aspect of the 3D gallop is its characteristic footfall sequence and duty factor. Figure 4.9 shows the footfall sequence for three strides. The footfall sequence yields the expected transverse gait, with a total duty cycle of approximately 60 %. The average stride period is just under 0.20 s[24], and the average duty cycle per leg is 26 %. The latter agrees with biological data, where the average duty cycle per leg for the transverse gallop was observed to be less than 40 % [29]. Furthermore,

[23]In fact, it was found that stable solutions exist without yaw control, although the quadruped ran in a large-radius arc.

[24]This value is somewhat less than the biological stride period for animals of similar mass, which ranges from 0.28 to 0.37 s [31]. The use of slightly larger leg stiffness values in the model and the lack of a flexible back, both of which would increase the total vertical stiffness, could explain the deviation.

Fig. 4.9 indicates that during a typical stride there are single-, double-, and triple-support phases, as there are in slow to moderate-speed biological gallops [28]. Single-leg support occurs when the leading hind leg touches down, lasting 17.0 ms, and after the leading fore leg lifts off, lasting 26.5 ms. Double support occurs when both hind and fore legs are in contact, lasting 26.0 and 17.5 ms, respectively, and when the diagonal leg pair (legs 2 and 3) is in contact, lasting 13.0 ms. Finally, brief triple support phases occur between legs 2, 3, and 4, and between legs 1, 2, and 3, lasting 8.0 and 9.0 ms, respectively. As with biological gallops, there are no periods of four-legged support.

Finally, as one of the goals for this study was to generate a visually realistic gait for the 3D gallop, a series of screen-captures for one stride of the gait is shown in Fig. 4.10. The quadruped starts at the top of the gathered flight phase (TOF) with the leading hind leg (leg 4) making contact first. The screen captures show several defining features of the biological gallop, including asymmetric footfalls, a significant gathered flight phase, and smoothness of the gait, as indicated by minimal vertical excursion of the body's center of mass and minimal pitch motion.

4.4.3 Torque and Power Analysis

In this section, the torque and power requirements for each joint are presented. Although the knee spring is simulated in parallel with the ideal actuator (Sect.2.2), the results given here assume that the parallel spring is *not* simulated during flight, and that the *sum* of the actuator and spring torques must be generated during stance. These assumptions effectively produce an ideal series elastic actuator, which appears to be the most efficient way to generate the large torque and power requirements for

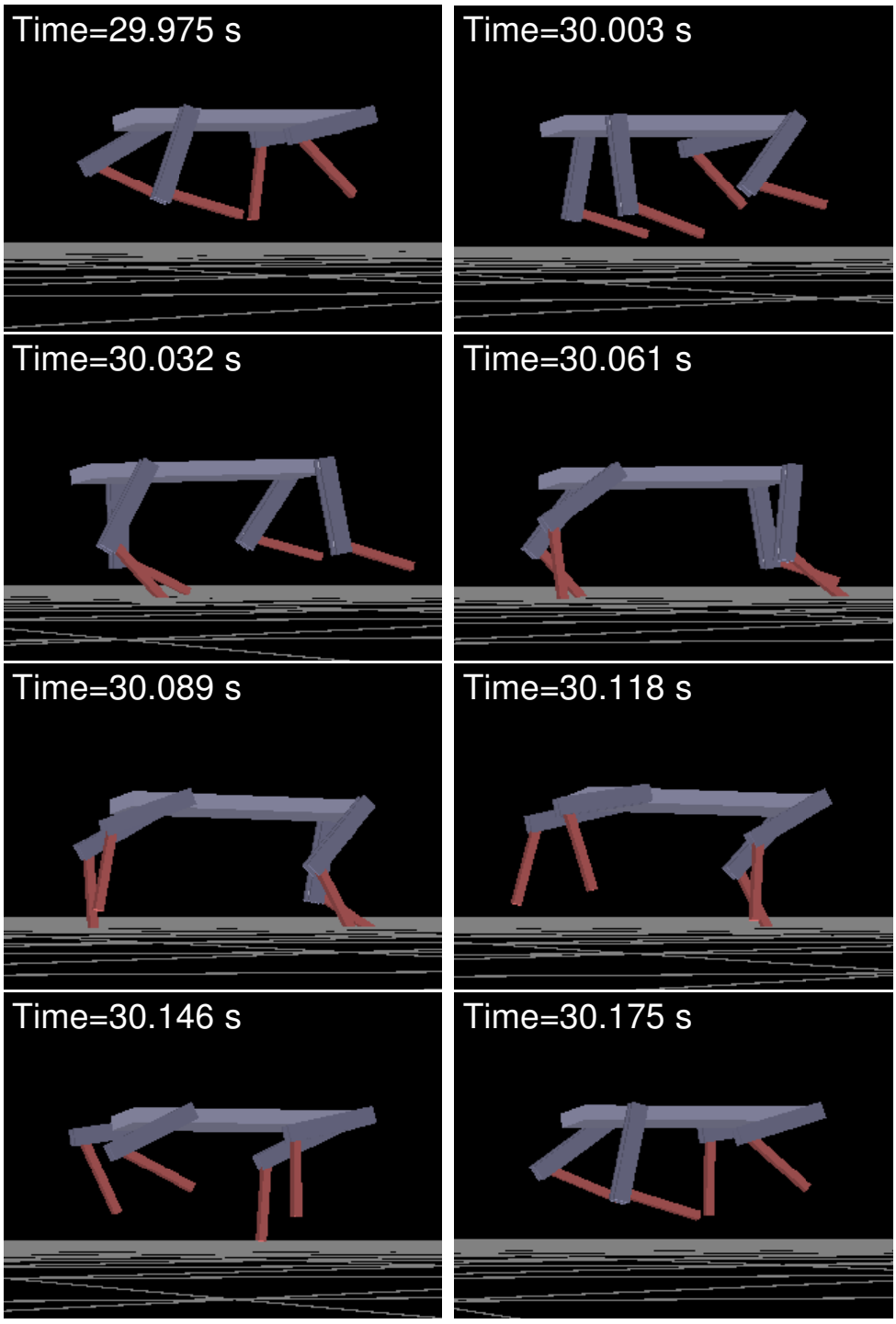


Figure 4.10: Screen captures of the 3D gallop over one stride, starting from the top of gathered flight (TOF).

galloping and maneuvering. Since the actuator is assumed to be ideal, and feedforward torque is used to compensate for the knee springs during position control of the knees (Sect. 2.3.3), converting to a series elastic configuration has no significant impact on the results of the simulation. For example, eliminating the knee springs during flight phases would not significantly change the total torque acting on the joint, which is given as follows:

$$\tau_{k_{\text{tot}}} = \tau_k + \tau_{k_s} \,, \tag{4.39}$$

where τ_k is the actuator torque and τ_{k_s} is the parallel spring torque. Substituting τ_k from Eq. (2.4), the total torque is

$$\begin{aligned} \tau_{k_{\text{tot}}} &= \tau_k + \tau_{k_s} \\ &= k^0_{p_k}\left(\theta_{k_d} - \theta_k\right) + k^0_{d_k}\left(\dot{\theta}_{k_d} - \dot{\theta}_k\right) + \tau_{\text{ffd}} + \tau_{k_s} \\ &= k^0_{p_k}\left(\theta_{k_d} - \theta_k\right) + k^0_{d_k}\left(\dot{\theta}_{k_d} - \dot{\theta}_k\right) \,, \end{aligned} \tag{4.40}$$

since $\tau_{\text{ffd}} = -\tau_{k_s}$ (Sect. 2.3.3). The feedforward term eliminates the spring torque when the spring is included at the joint. Eliminating the spring entirely during the flight phase would have no impact on the total torque acting on the knee when it is under PD control[25]. Furthermore, reporting the sum of the actuator and spring torques during stance does not change the total torque acting on the joint. Instead, it provides a more conservative estimate of the total torque required for actuation.

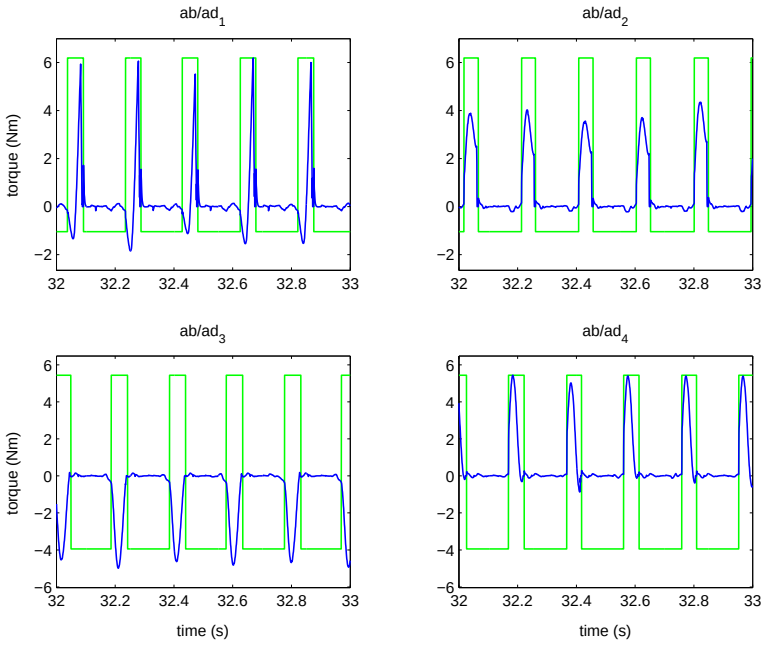
There are two main reasons for using a series elastic versus a parallel configuration. First, in a parallel configuration, the actuator must fight the spring during flight, which causes significantly larger peak power requirements. Although the spring stores

[25]The only exception would be during the FREE function, when the knee joint is in flight but not under PD control. However, the limited duration of the FREE function (0.025 s, Sect. 4.2.7) yields negligible differences between the series and parallel torque and power calculations.

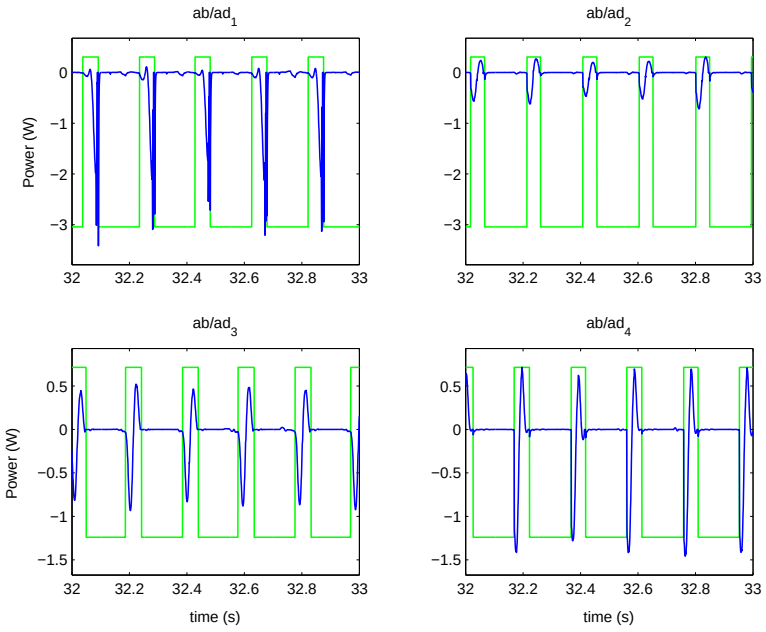
energy, the actuator must still generate sufficient power to compress it over a short period. The maximum amount of this power ultimately determines the actuator's minimum size. The presence of parallel springs would require larger, heavier actuators for the knee joint. Furthermore, biological evidence suggests that animals have series elastic configurations [90], as their limbs are transferred during flight with largely inertial resistance only.

Second, during stance, where a passive spring response is desired at the knees, a parallel actuator would appear as a large inertia with high damping due to its back-drive characteristics. Consequently, significant energy loss would occur unless the motor were physically detached from the knee joint, which would require an electric clutch. However, Schmiedeler [84] found performance limitations for the electric clutch in his one-legged hopper due to engagement times. The quadrupedal model in this work is smaller, has shorter stance periods, but still requires relatively high torque levels. Consequently, finding a small, light-weight clutch with sufficient holding torque but short engage and release times may not be possible. On the other hand, a series configuration would not require a clutch, but, instead, would require a sufficiently powerful motor to maintain one end of the spring while it flexes. Admittedly, finding a small, light-weight motor with sufficient power would be a challenge, although a preliminary survey of existing actuator technology suggests that there may be some feasible models whose peak intermittent power may be sufficient to generate the levels described below (e.g., Maxon brushless DC motors EC40 and EC32).

The torque and power for the ab/ad joints are shown in Fig. 4.11. The asymmetry among the ab/ad joint torque and power curves occur due to the asymmetric footfall sequence. Generally, the ab/ad joints must compensate for both roll and yaw motions



(a)

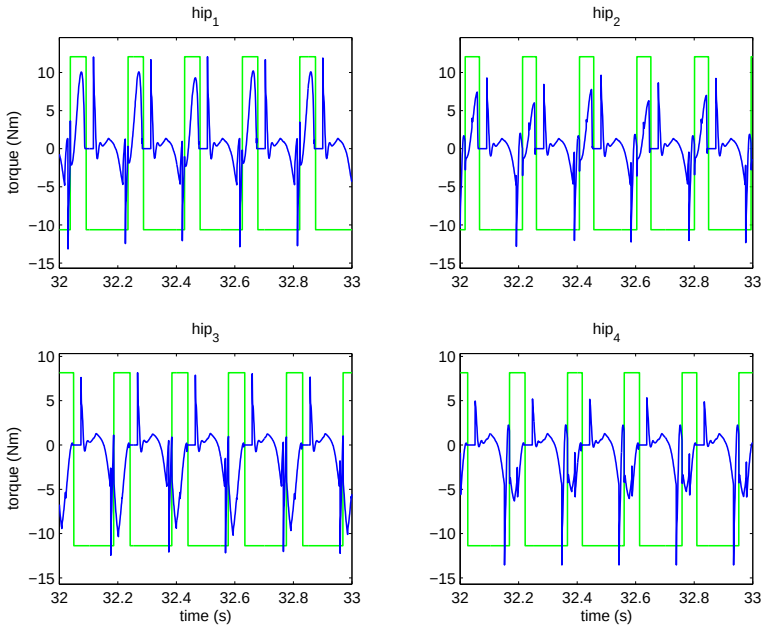


(b)

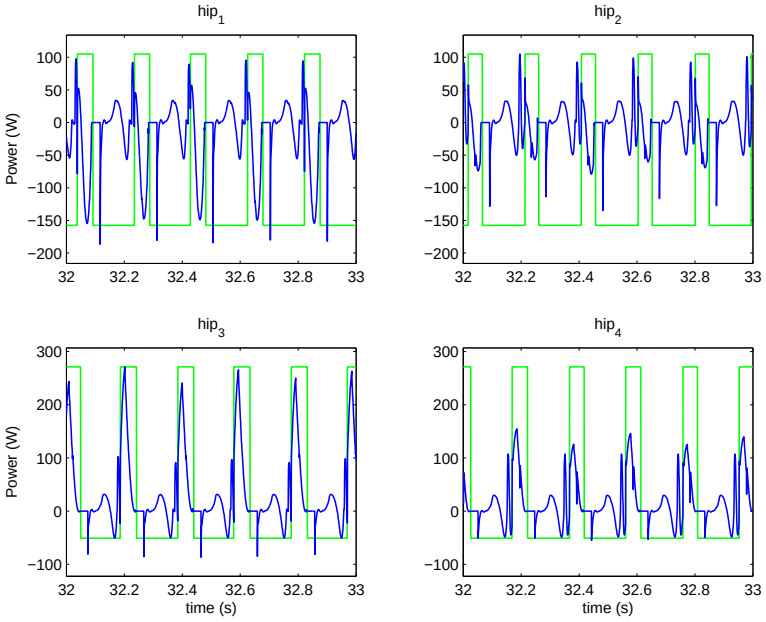
Figure 4.11: Torque and power requirements for the ab/ad axes during steady-state galloping. Square waveforms indicate leg contact when the level is high.

of the body, which are also affected by the hip and knee torques. Consequently, further study is necessary to analyze the relationship among ab/ad joint torques and body mechanics during stance. However, the ab/ad torque and power are small relative to the hip and knee joints, which is expected, since the ab/ad joints rotate by small amounts during flight and have small moment arms relative to the dominant vertical forces during contact. Lateral forces, which have the longest moment-arm about the ab/ad axes (Sect. 4.4.2), are generally small for steady-state galloping (Fig. 4.7). Consequently, the required torque for the ab/ad axes is not nearly as large as the torques for the hip or knees, both of which must generate significant horizontal and vertical forces during steady-state galloping.

The hip torque and power shown in Fig. 4.12 support this conclusion, as there is significantly larger peak torque and power required during stance. Positive torque and negative power during the front leg stance phases result from the shoulder-braking behavior described in Sect. 4.4.1. The shoulder torque is generally positive during stance to retard the retraction of the front legs. As the body's forward momentum forces the joints to rotate rearward with respect to the body, power is regenerated in the hip actuators due to the negative torque-speed product. The asymmetry in the front leg torque and power curves generally corresponds with the correction of the body's negative (CW) yaw rate during the front stance periods (Fig. 4.8 (e)). In the rear legs, hip torques are negative to forcibly retract the legs and propel the body forward. Consequently, the power is positive because the hips are moving in the same direction as the torque. Since the body's yaw rate is positive (CCW) during most of the rear leg stance period, hip joint 3, on the left side, exerts a larger driving torque to compensate.



(a)



(b)

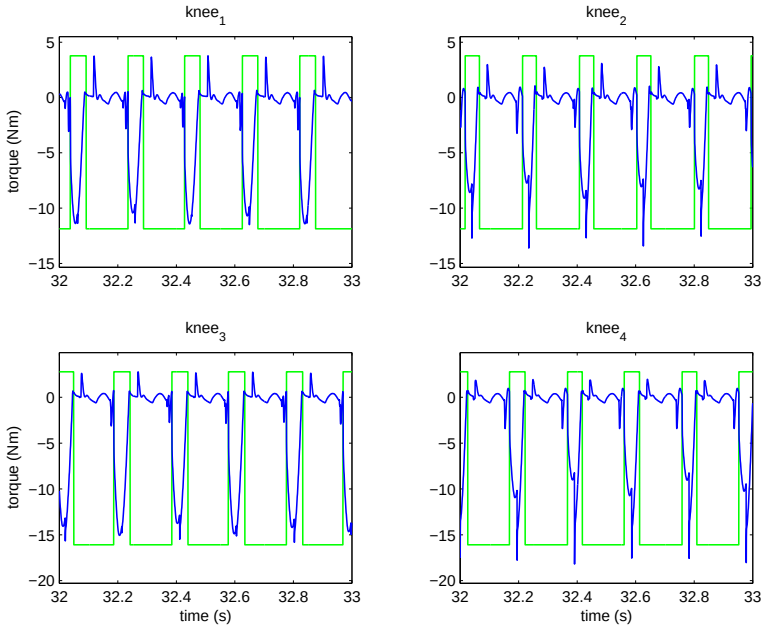
Figure 4.12: Torque and power requirements for the hip axes during steady-state galloping. Square waveforms indicate leg contact.

During flight phases, significant negative (thrusting) hip torques are required for all legs shortly before their contact phases. This coincides with the EARLY-RETRACTION function, during which the leg is actively rotated rearward. Consequently, the power spikes in the positive direction during this period. In general, however, significantly less torque and power are required throughout the flight phases before the legs begin their retraction.

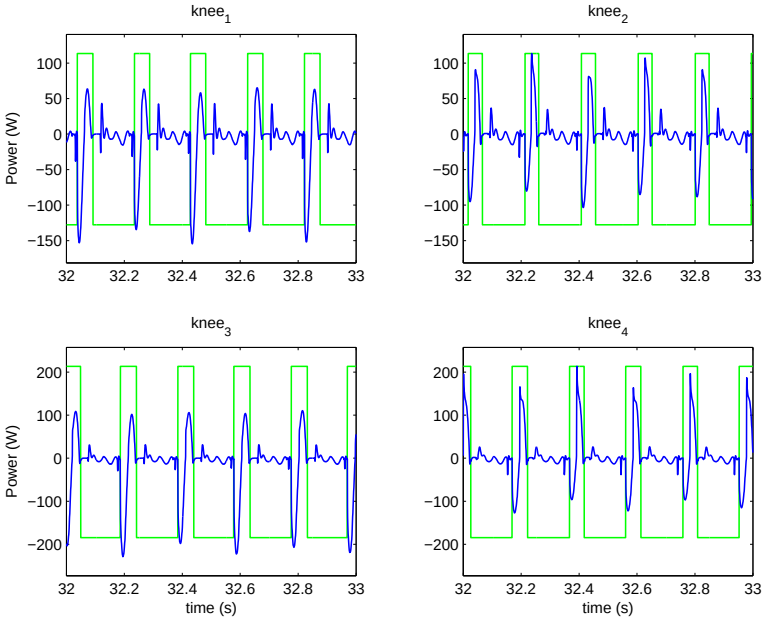
The knee torque and power are shown in Fig. 4.13. Minimal torque and power are required to transfer the shank during flight because only the relatively small inertial load of the shank (Table 2.1) is seen[26]. During stance, however, significant torque and power are required as the knee springs flex, then extend. During flexion, negative torque is required to resist the spring as it compresses, which results in negative power. During extension, the spring moves in the same direction as the torque, which results in positive power. The instantaneous adjustment of the knee spring anchor that occurs during stance (Sect. 4.2.5) can be seen as spikes in the contact phase torque in Fig. 4.13 (a). This injection of energy causes the associated power curve in Fig. 4.13 (b) to change sharply from negative to positive in each case. Larger positive power levels occur in knees 2 and 4, on the right side, while larger negative power occurs on the left side. It appears that power is injected on the right side, then withdrawn on the left, which would have a direct impact on the roll of the body. As stated above, however, further investigation is necessary given the significant coupling among ab/ad, hip, and knee torques during stance.

The total power summed across all joints is shown in Fig. 4.14. Here, it is assumed that negative power is regenerated, which reduces the requirements for positive, or

[26]Compare to the parallel configuration, where peak torques of 45.0 Nm and 1150 W are required to fight the knee springs during leg transfer.



(a)



(b)

Figure 4.13: Torque and power requirements for the knee axes during steady-state galloping. Note that an ideal series elastic configuration is assumed, where the knee springs are not simulated during flight, and the total actuator plus spring torque is reported during stance. Square waveforms indicate leg contact.

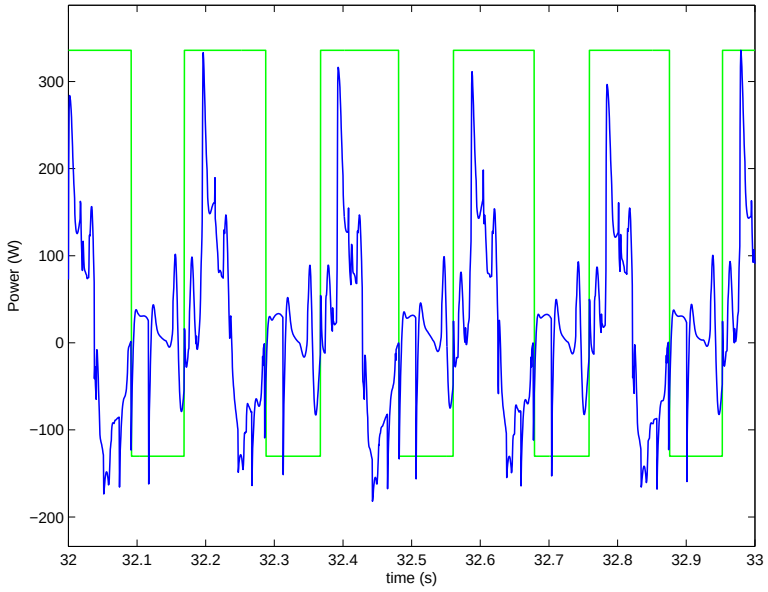


Figure 4.14: Total power required for all joints for steady-state galloping. The square-waveform indicates that one or more legs are in contact.

injected, power. This assumption appears reasonable based on the specifications for many modern digital amplifiers and power supplies. Four-quadrant control is typical for many amplifiers, which allows for precise control during regenerative periods. Furthermore, most power supplies can handle net regeneration of energy as long as there is sufficient capacitance to absorb the excess energy and prevent over-voltages. If there is insufficient capacitance, a shunt regulator may be added to dissipate the excess energy in the form of heat (I^2R) [91]. Thus the negative portion of the total power curve in Fig. 4.14 would be saturated at 0.0 W if it is assumed that there is insufficient capacitance to store the excess energy.

Although the assumption of regeneration is a reasonable one, its impact on the average power, an important efficiency metric, is significant. A thorough analysis on an experimental system would be necessary to determine the most realistic scenario.

Consequently, several different options will be used here in determining the average power and specific resistance in order to facilitate comparisons to other work. If regeneration is allowed, the average power for the period shown is 13.1 W, while the average power assuming no regeneration (i.e., the total power remains at or above 0.0 W) is 41.1 W. Finally, if the sum of the absolute values of the required power for each joint is used, the average power is 229.8 W.

The specific resistance of the model is a dimensionless number computed using the average power divided by the product of the weight times the average velocity [92]:

$$\kappa = \frac{P}{wv}\,, \tag{4.41}$$

where P is the average power, w is the weight of the system, and v is the average velocity. The specific resistance for this model would be 0.03 allowing regeneration, 0.11 without regeneration, and 0.59 using the absolute value approach. (All calculations assume a mass of 9.6 kg and a speed of 4.15 m/s.) Marhefka [89] found a specific resistance of 0.4 (using the absolute value approach) for a top speed of 7.0 m/s (7 body lengths/s), which is slightly slower in relative terms than the gallop produced here. Furthermore, he did not employ shoulder or hip torque during stance, which would explain much of the difference. The results of the power analysis are summarized in Table 4.7.

4.4.4 The Results of Using Early Retraction

In Sect. 4.2.4, it was explained that early retraction is used to maintain the relative leg phasing as well as to reduce impact losses when the foot touches down. To study the efficacy of early retraction in reducing the foot velocity, the touchdown velocity of each foot was measured across 60.0 sec of galloping. This velocity was then compared

Assumption	Average Power (W)	Specific Resistance (κ)
Regeneration allowed.	13.1	0.03
No net regeneration.	41.1	0.11
Absolute value of joint power requirements.	229.8	0.59

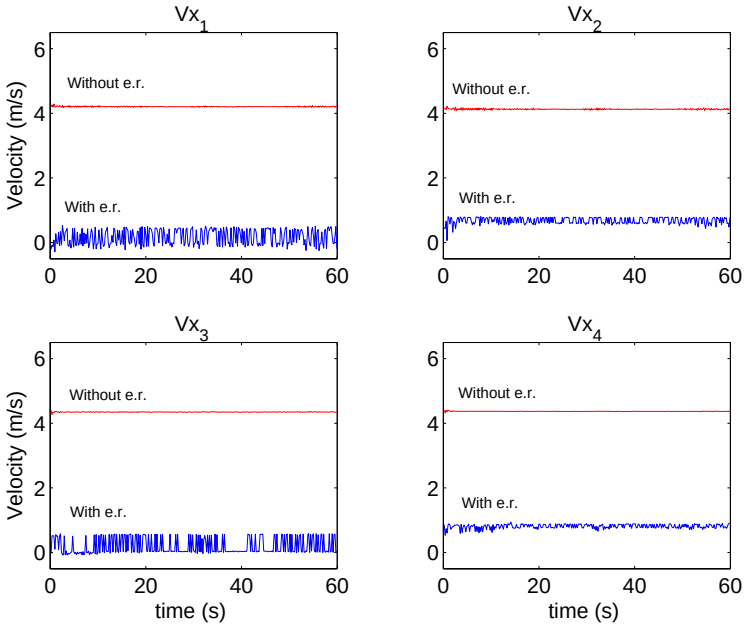
Table 4.7: Total power for the gallop under different assumptions.

to what the velocity of the foot *would have been* had it been stationary relative to the body at touchdown, but in the exact same touchdown position. A more telling experiment would be to evolve a new controller without early retraction. As this was not feasible, however, an estimate of the foot velocity using a stationary leg was computed as an alternative. The following equation, which describes rigid body kinematics, is used to compute the foot velocity at touchdown assuming no early retraction:

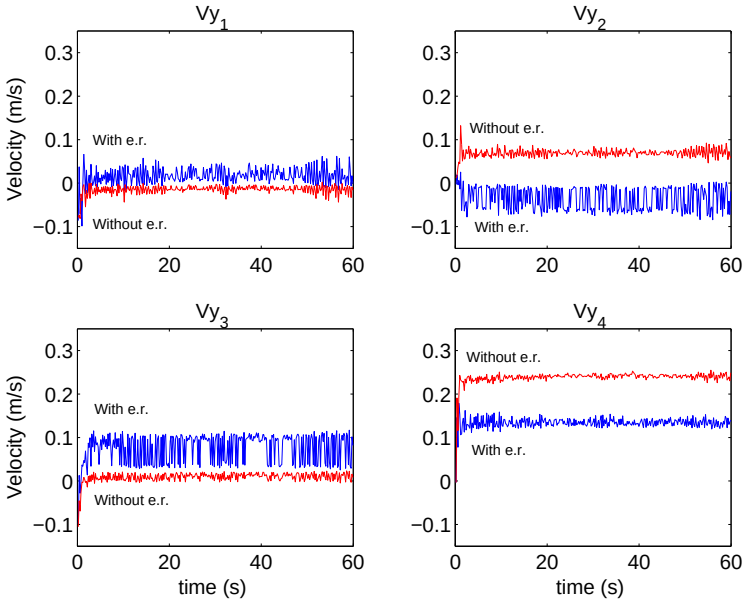
$$ {}^{e'}\boldsymbol{v}_{f_i} = {}^{e'}\mathbf{R}_e \left({}^{e}\boldsymbol{v}_b + {}^{e}\boldsymbol{\omega}_b \times {}^{e}\mathbf{R}_b \, {}^{b}\boldsymbol{p}_{f_i} \right) , \tag{4.42} $$

where ${}^{e'}\boldsymbol{v}_{f_i}$ is the velocity of foot i in the rotated frame $\{\boldsymbol{e}'\}$, where $\{\boldsymbol{e}'\}$ is obtained by rotating about $\hat{\boldsymbol{z}}_e$ (Fig. 2.1) by the body's yaw angle α, indicated by the transform ${}^{e'}\mathbf{R}_e$, ${}^{e}\boldsymbol{v}_b$ and ${}^{e}\boldsymbol{\omega}_b$ are the linear and angular velocities of the body in earth coordinates, ${}^{e}\mathbf{R}_b$ transforms from body to earth coordinates, and ${}^{b}\boldsymbol{p}_{f_i}$ is the position of foot i in body coordinates.

Figure 4.15 shows the actual velocity for each leg with early retraction versus the computed velocity without early retraction. The figure shows that the largest velocity component for all legs without early retraction is in the $\hat{\boldsymbol{x}}_{e'}$ direction, as expected, since the desired running direction is along this axis. The actual foot

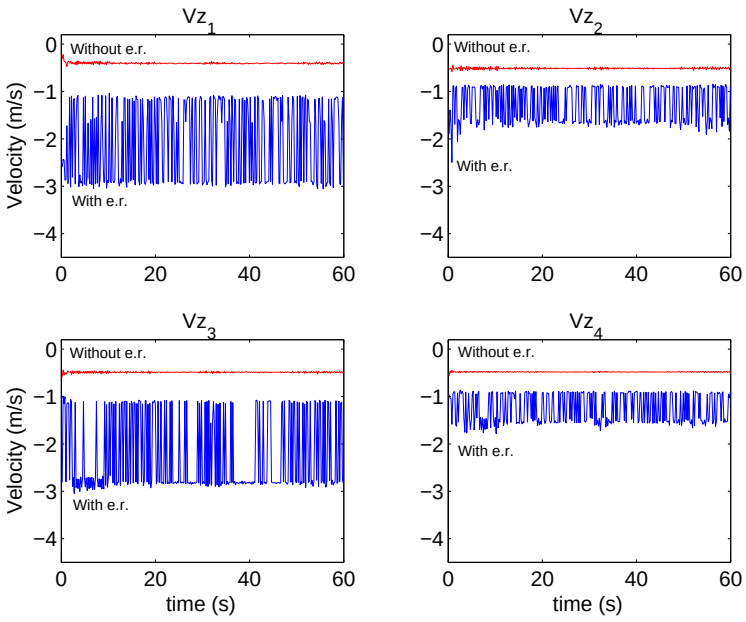


(a)

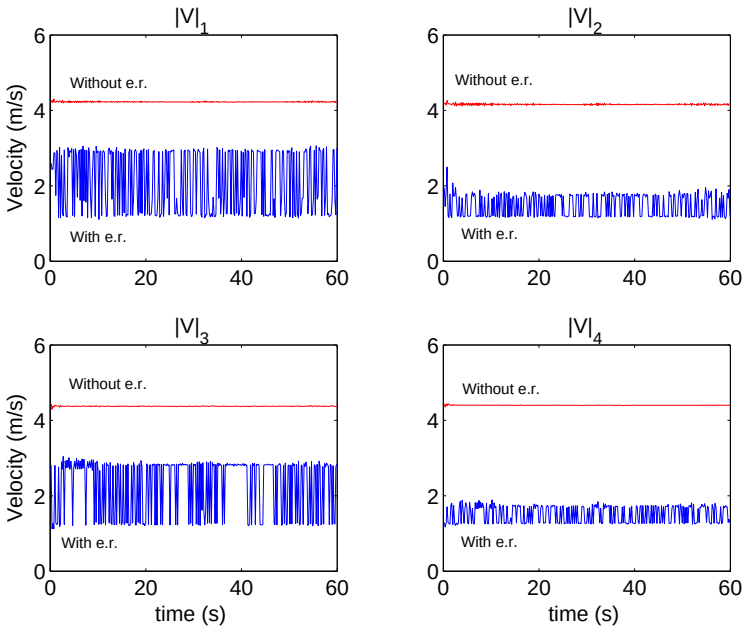


(b)

Figure 4.15: Foot velocity components and magnitudes with and without early retraction: (a) Velocity in $\hat{\boldsymbol{x}}_{e'}$ direction, and (b) velocity in $\hat{\boldsymbol{y}}_{e'}$ direction (cont'd on next page).



(c)



(d)

Figure 4.15: Foot velocity components and magnitudes with and without early retraction: (c) velocity in $\hat{\boldsymbol{z}}_e$ direction, and (d) magnitude of the foot velocity. The velocity without early retraction is computed assuming the leg is stationary with respect to the body at touchdown, but using the same touchdown position as that achieved during early retraction. (Cont'd from previous page.)

velocity along $\hat{\boldsymbol{x}}_{e'}$ using early retraction, however, is much smaller across all legs and is, on average, less than 1.0 m/s. In the $\hat{\boldsymbol{y}}_{e'}$ direction, the early retraction velocity components for the right-side legs (2 and 4) are smaller than the velocity without retraction, although the opposite is true for legs 1 and 3. This is most likely due to the fact that the body rolls over onto the left side faster during stance than it does in the opposite direction during flight (Fig. 4.8 (g)). On average, however, the velocity component along $\hat{\boldsymbol{y}}_{e'}$ is much less significant than the component along $\hat{\boldsymbol{x}}_{e'}$ since the quadruped is running largely in its own forward direction $\hat{\boldsymbol{x}}_{e'}$.

Unlike the $\hat{\boldsymbol{x}}_{e'}$ velocity component, however, the $\hat{\boldsymbol{z}}_e$ velocity with retraction is actually larger in magnitude, on average, than the computed velocity without retraction. This occurs because retracting the leg from a protracted position causes additional downward velocity of the foot. However, the total magnitude of the velocity in Fig. 4.15 (d) illustrates that the actual velocity with retraction is only about one-half as large as the computed velocity without it, averaged across all legs and over time. This occurs because the forward running velocity is the dominant linear velocity component of the body at touchdown. Thus, whenever this condition is true, early retraction tends to reduce the velocity of the foot relative to the ground, which would reduce the resulting impact force.

The average impact force with and without early retraction can be estimated using the simple contact model as follows[27]. If an average foot velocity of 4.0 m/s is assumed for the case without retraction, and 2.0 m/s is assumed with retraction, then the leg will incur a 200.0 N (2.1 body weights) force impact for the former, and a 100.0 N (1.1 body weights) force for the latter, using the ground damping constant of 50.0 N/m/s

[27]A more sophisticated contact model may produce more accurate results, although the speed of the contact point at touchdown should still be correlated with the impact force.

(Sect. 2.2). In Fig. 4.7, the average impact force from a typical stride is approximately 115.0 N, which is close to the estimate using early retraction. These results suggest that biological early retraction may significantly decrease impact forces at higher running speeds, when the forward velocity is the dominant velocity component at touchdown.

4.4.5 Stability Analysis

In this section, an analysis of the stability of the solution is presented by looking at the ranges of attraction for stable galloping and the impact of each evolved stage of the controller. Table 4.8 presents the range of attraction[28] for each body state variable around the nominal initial conditions described in Sect. 4.3 based on the requirement that the resulting gallop was stable for at least 20 strides. The table shows that the solution is particularly sensitive to initial height and lateral velocity, as both of these ranges are relatively small. However, longer periods of stable galloping were observed at some of the boundary values in Table 4.8, and evolved transitions from the turn, jump-start, and running jump were achieved from conditions outside those listed in the table, as discussed in the following chapters. The ranges given in the table provide a measure of sensitivity only for the nominal point described in Sect. 4.3 and only when varying one initial condition at a time. Further study would be required to better quantify the region of attraction for stable galloping, as it may not be a convex space.

[28]Mapping the entire region of attraction for all possible combinations of initial conditions was not practical given the dimensionality of the space. As such, the individual range of attraction for each body state variable was used to provide an indication of the gallop's sensitivity to each initial condition.

Variable	Range		Variable	Range	
Yaw (α)	$[-0.65, 0.75]$	rad	Pitch rate ($\dot{\beta}$)	$[-0.500, -0.325]$	rad/s
Pitch (β)	$[-0.050, 0.069]$	rad	Roll rate ($\dot{\gamma}$)	$[-0.75, 0.20]$	rad/s
Roll (γ)	$[-0.035, 0.015]$	rad	Forward velocity (${}^{e'}\dot{x}_b$)	$[4.05, 4.30]$	m/s
Height (${}^{e}z_b$)	$[0.2335, 0.2410]$	m	Lateral velocity (${}^{e'}\dot{y}_b$)	$[-0.01, 0.025]$	m/s
Yaw rate ($\dot{\alpha}$)	$[-0.06, 0.10]$	rad/s			

Table 4.8: Range of attraction for each body state variable for the 3D gallop using a 20 stride stability requirement.

As an illustration of how each component of the 3D gallop controller contributes to the overall stability of the solution, Fig. 4.16 shows the resulting solution after each stage of evolution. After the first stage in Fig. 4.16 (a), the solution is stable for just over four seconds with considerable oscillation in roll and yaw. After the second stage in (b), there is significant improvement in all variables, especially in roll and yaw, where both variables appear to be stabilizing. There remains instability in pitch, however, which ultimately destabilizes the system after about seven seconds. After stage three in Fig. 4.16 (c), the yaw error begins increasing after $t = 3.0\,\text{s}$, although both roll and yaw behave qualitatively the same as before. However, the solution is now stable for about nine seconds. Finally, after adding pitch control in plot (d), the solution is now stable for at least 60 s. An interesting result is that the steady-state values for all variables seem to remain the same from (c) to (d), indicating that the pitch controller somehow stabilizes the solution without significantly impacting the

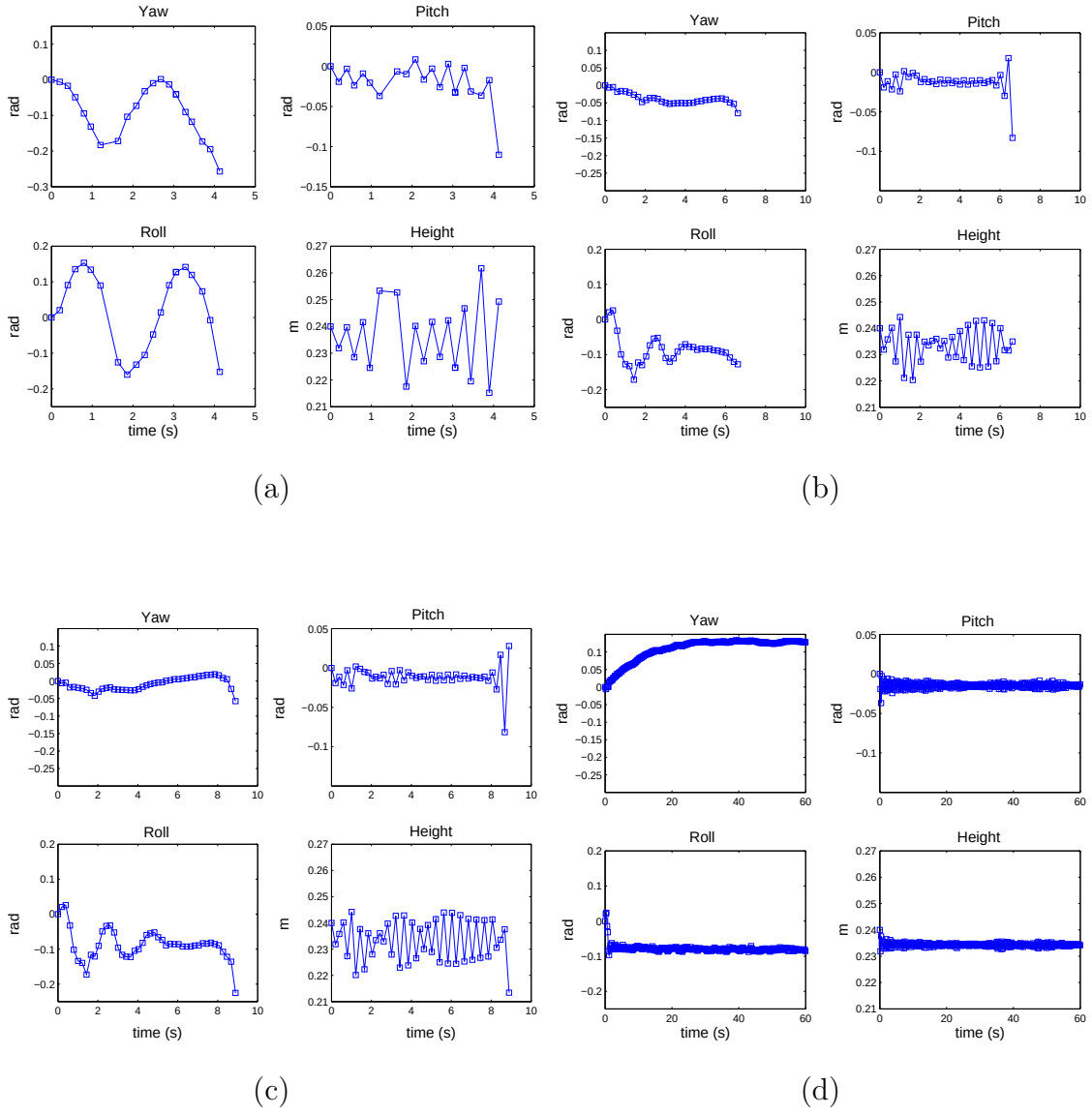


Figure 4.16: Stability results after each evolution stage for the 3D gallop: After evolving (a) sagittal plane parameters (d_{f_0}, v_{b_f}, v_{b_r}, and $\dot{\beta}_d$), (b) roll parameters (k_{p_γ}, k_{d_γ}, f_γ, k_{p_l}, and k_{d_l}), (c) yaw parameters (k_{p_α}, k_{d_α}, and f_α), and (d) pitch parameters (k_{p_β}, k_{d_β}, k'_{p_β}, and k'_{d_β}).

other portions of the controller. A thorough understanding of how the pitch controller stabilizes the 3D gallop remains an important topic for future investigation.

Although the focus of this study is on the 3D gallop, it is also worthwhile to note an early observation from the planar gallop, namely that the planar system appeared to be stable without any pitch control, a characteristic also observed in [30]. In fact, only the first stage of evolution was required for the planar case, as the system would easily settle into various stable patterns, including period-2 and period-1 gaits. Moreover, it was observed that the planar system appeared to have a strange attractor, as gradually increasing or decreasing the sagittal plane parameter d_{f_0}, the bias fore-aft energy distribution factor, caused the system to migrate from stable period-1 to period-2 solutions, then to more chaotic responses. Similar results have been observed for other systems employing compliant legs and intermittent contact phases [33]. In the spatial 3D domain, however, the solution remains unstable until pitch control is added. However, the stability of the planar gallop might suggest that if stable transverse and dorsal plane controllers could be found that minimally impacted the sagittal plane, the resulting gait could be stabilized in the sagittal plane without pitch feedback control.

4.5 Summary and Conclusions

In this chapter the controller for the gallop was presented. The controller is comprised of five of the leg primitive function discussed in Chap. 2, including TRANSFER, EARLY-RETRACTION, WAIT-FOR-TRIGGER, GALLOP-STANCE-CONTROL, and FREE. During TRANSFER, an "outward leg-rotation" scheme is used to compute the ab/ad angles based on roll and yaw errors. If the body is rolling over onto

one side, the ab/ad joints on that side are adjusted outward, while those on the other side are adjusted outward by a smaller, scaled amount. Likewise, if the body is yawing CCW, for example, the ab/ad joints for the diagonal leg pair 1 and 4 are adjusted outward, where leg 4 is adjusted by a small fraction of the front leg's adjustment. The outward leg-rotation method generates a sprawled posture to correct for roll and yaw errors, as opposed to the Raibert-style approach of using equal and opposite ab/ad deflections in the front and rear legs. The sprawled posture approach was found to be more stable due to the larger support area.

During WAIT-FOR-TRIGGER, a mixed temporal-spatial triggering scheme was used to enforce the desired leg phasing in a transverse pattern. Based on Herr and McMahon's work, the triggering method used here was modified to make it more robust to variations in body orientation during the first few strides of the gallop. In particular, it was found that using spatial cues to trigger leg 4, the leading hind leg, and leg 2, the leading front leg, was required to generate a stable gait pattern.

During GALLOP-STANCE-CONTROL, hip velocity servos similar to Herr and McMahon's strategy were employed. However, full position control was implemented by integrating the desired angular velocity of the leg. Furthermore, a simple energy distribution approach was used to allocate a constant amount of energy among the four legs to correct for roll and pitch errors. This approach was, perhaps, the most critical factor in stabilizing the gallop. Consisting of a difference equation control law for pitch and a simple PD control law for roll, the energy control equation guarantees that the total energy summed over all legs remains unchanged from stride to stride. Once a target energy value is computed for each leg, the knee spring rest position is adjusted at maximum compression to achieve the desired energy value.

Evolution of the control parameters and initial pitch rate was accomplished using a staged approach, where control functions were decoupled to a sufficient extent to permit partitioning of the parameter space. It was noted that evolving pitch control as a last stage was necessary for stable 3D solutions, although it was not necessary in preliminary planar studies. An interesting question is whether a 3D controller could be developed that minimally impacted the sagittal plane dynamics, which appear to require less feedback control.

In summary, the gallop generated here demonstrates largely biological characteristics and good stability in what appears to be a new set of results. The use of a staged approach with effective fitness criteria, reasonable heuristic control strategies, and effective parameterization were all important aspects of the overall approach used to generate solutions. The relative success of this method validates the use of evolutionary algorithms in combination with effective control architectures for investigating complex, dynamic behaviors in legged robot models that are based on non-trivial assumptions. Furthermore, the approach employed here promises to facilitate further investigation into the dynamics and stability of spatial galloping using legged robot models with comparable, or even greater, complexity as the one described here.

CHAPTER 5

THE HIGH-SPEED TURN

5.1 Introduction

The high-speed turn is a dynamic maneuver that results in a sharp change in trajectory in the horizontal plane. Typically used in quadrupedal mammals to avoid unexpected obstacles or evade a predator, the high-speed turn is an impressive maneuver that couples both yaw and roll motion, resulting in a bank. For sharp turns, the roll angle of the body can become extreme, as the legs rotate toward the outside of the turn. The result is that the vertical projection of the body's COM can lie completely outside the support throughout much of the turn, making the turn an extreme example of dynamic stability.

The high-speed quadrupedal turn has received virtually no attention in the literature for several reasons. First, the turn is an extreme application of dynamic stability, which is still challenging in its own right, even two decades after Raibert's pioneering work [32]. Dynamically stable systems such as this one have short, discrete contact phases, which represent the best opportunity to control the system. However, using standard feedback control to force the system back to equilibrium during the contact

phases can actually impede stabilization when the controller acts contrary to the system's natural dynamics [93]. Unfortunately, discovering these dynamics is often not trivial in systems with many degrees of freedom (DOF) and unconstrained spatial motion. Modeling the system analytically may not be possible unless a number of simplifying assumptions are made with respect to symmetry, mass distribution, and contact times [33, 44].

The second problem with studying the high-speed turn is that it is difficult to obtain high-quality video of the biological maneuver for analysis. The turn involves high speeds and covers significant area because of the trajectory change. This precludes studying it in a controlled laboratory setting on a motorized treadmill, which has been a common approach in studying various forms of legged locomotion [28, 31, 81, 82, 83, 94, 95].

As an alternative to the treadmill, force-plate analysis, motion-capture equipment, and dissection of similar test subjects can be used to infer the dynamics by measuring external forces, determining position and velocity of each body segment, and using computed inertial properties of the segments [96]. However, the turn presents two major problems with such an approach. First, training an animal to execute a sharp turn precisely over the force plate may be difficult, as animals tend to employ the turn out of necessity. Forcing the animal to run toward a known obstacle would most likely result in a gradual turn or significant deceleration before the turn, neither of which are characteristics of the desired behavior.

The third major problem would be obtaining repeatable results with respect to both the motion-capture equipment and the force-plate. During steady-state walking or running, animals employ a fairly regular gait pattern, although the data can still

be noisy and uneven, as in [81, 82]. It is not unreasonable to assume that a high-speed turn would suffer from even more irregularity, since an animal would use such a motion much less frequently than a straight-line gait, therefore limiting the training effect on muscles and neural pathways. Consequently, captured motion and force data may exhibit significant variability, making it difficult to extract a useful model.

Although the issues raised here with measuring a biological turn could certainly be overcome with enough time, effort, and funding, studying an artificial maneuver would seem to be a more practical alternative in the mean time. However, generating such a motion with an actual robot is not currently practical not only because of the difficulty in controlling it, but also due to the significant power required. Maneuvers involve significant periods of acceleration to alter the body's trajectory, which translates to large peak power requirements. Unfortunately, peak power requirements necessarily demand larger, heavier actuators, which further increases the required power, and so on. At the current time, both power sources and actuators are significant limitations in autonomous robots, a problem that must eventually be overcome in order to realize practical applications. Until then, off-board power supplies are the common alternative, which introduces the problem of managing a necessarily long, high-power umbilical during a high-speed 3D maneuver. Of course, any unexpected results during the maneuver could have disastrous consequences for both robot and handler. Consequently, it seems that studying high-speed quadrupedal maneuvers in an actual robot remains a distant goal.

While the other alternatives are clearly impractical at best, studying dynamic maneuvers with mathematical models in simulation seems to be the only viable approach at present. However, the lack of control strategies for complex dynamical

systems remains a major obstacle. Thus, the goal for this part of the study is to evolve a simulated high-speed turn during galloping and obtain a characterization of the resulting dynamics over a range of turn angles. Developing a complete turning controller that can accommodate a range of turn angles and conditions is beyond the scope of this work. However, studying the relationship between evolved open-loop control quantities, ground reaction forces, and the resulting body state at the end of the turn should provide a solid understanding of the basic mechanics of the maneuver and serve as a useful foundation for more in-depth investigation.

In the remaining sections, the control strategy and evolved parameters for the turn will be explained with respect to each of the leg primitive functions that it employs. The basic optimization problem will then be presented, with an emphasis on describing the fitness criteria used to evaluate solutions. Following that, an analysis of the data across a range of positive and negative turn angles will be presented in order to extract some simple, useful relations that may be exploited in the future development of a turning controller. Next, the results for two multiple-stride turns in each direction will be presented, including the torque and power requirements for each joint. Finally, a summary is given at the end.

5.2 Turn Controller

The controller for the high-speed turn is comprised of the same leg functions in the same sequence as that of the gallop (Fig. 4.1). However, there are two major differences between the two controllers. First, as stated above, there is no high-level, feedback controller used for the turn. Instead, all control variables are evolved, open-loop parameters, which are discussed in the following sections. Second, the turn

is evolved as a single stride, as opposed to the multiple-stride gallop discussed in Chap. 4. For the case of multiple turns in succession, as discussed in Sect. 5.7, each turn must be evolved independently.

5.2.1 TRANSFER

The TRANSFER function for the turn is similar to that of the gallop, although no control computations for the ab/ad joints are performed at TOF. Instead, the desired ab/ad angles $\theta^*_{a_i}$, $i = 1, \ldots, 4$, are open-loop control variables that must be evolved, while the virtual leg angles and lengths are identical to those of the gallop.

5.2.2 WAIT-FOR-TRIGGER

The WAIT-FOR-TRIGGER function for the turn is significantly different than its analog in the gallop controller. While the leg retraction is synchronized using temporal and spatial cues for the gallop, leg retraction for the turn is based purely on the spatial condition of the foot's proximity to the ground. Thus, each leg employs the FOOT-HEIGHT trigger (Table 2.6) with a threshold height of $h_t = 0.025\,\text{m}$. This triggering scheme is employed because visual analysis of high-speed turns suggests that the *inside leg* in each pair of legs in the front and rear appears to retract first for a given turning direction [85]. The inside legs are those legs closest to the center of the turn. For example, legs 1 and 3 on the left side of the body would be the inside legs for a CCW turn. Consequently, the FOOT-HEIGHT triggering scheme was selected because it produced natural-looking results, with the inside leg in each of the front and rear leg pairs touching down first, and required only one tunable parameter, the threshold height h_t, which was set to a reasonable value based on preliminary testing.

5.2.3 EARLY-RETRACTION

The EARLY-RETRACTION function is parameterized identically to that of the gallop. Because all legs are triggered when they are close to the ground, however, EARLY-RETRACTION in the turn results in a smaller average difference between the nominal leg angles used in TRANSFER and the actual touchdown angles of the legs.

5.2.4 GALLOP-STANCE-CONTROL

The GALLOP-STANCE-CONTROL function for the turn is also similar to that of the gallop, although no feedback control is used to compute the energy levels for each leg. Instead, each leg's target energy value E_i^*, $i = 1, \ldots, 4$, is an open-loop control quantity that must be evolved. Furthermore, the velocity bias values are no longer paired for the front and rear legs. Instead, four separate bias values $v_{b_i}^*$, $i = 1, \ldots, 4$, are evolved. However, the front biases are constrained to be negative, and the rear, positive, to create the shoulder-braking and hip-thrusting pitch stabilization that is important for galloping. Its use here is appropriate because the body's pitch and pitch rate are expected to behave similarly over the turn stride as they do in the gallop. Finally, the no-slip flag σ_{slip} is set to 1 for most of the testing, although Sect. 5.6 presents results obtained by setting $\sigma_{\text{slip}} = 0$.

5.2.5 FREE

The FREE function in the turn is identical to that of the gallop. However, as the turn often commands larger energy values for individual legs, the FREE function's role is more critical here because the knee spring rest position may have been adjusted more significantly than in the gallop. Consequently, more motion of the shank may

be required to return the rest position to its nominal value. A summary of the turn control parameters is given in Table 5.1.

5.2.6 Turn Initialization

Like the gallop, the turn is initialized using INIT-TOF. As the focus is on generating a turn from gallop conditions, however, the turn is initialized using the actual body and leg states after several seconds of galloping[29]. Consequently, $\sigma_{l_a} = 1$ to indicate that the actual leg states in $\boldsymbol{\theta}_l$ are used to initialize the legs. Table 5.2 summarizes the parameters used to initialize the turn.

5.3 Turn Optimization Problem

The turn optimization problem involves finding the open-loop control parameters for a range of turn angles from $[-35.0, 35.0]$ deg ($[-0.61, 0.61]$ rad) in 5.0 deg (0.087 rad) increments. The range was selected to achieve shallow-to-moderate turn angles with respect to a running speed of 4.15 m/s. The control parameters are given as follows:

$$\boldsymbol{\phi}_{\text{turn}} = \left[\theta^*_{a_1}, \ldots, \theta^*_{a_4}, v^*_{b_1}, \ldots, v^*_{b_4}, E^*_1, \ldots, E^*_4\right]^T , \tag{5.1}$$

where $\theta^*_{a_i}$ is the desired ab/ad position for leg i for TRANSFER (Sect. 5.2.1), and $v^*_{b_i}$ and E^*_i are the velocity bias and target energy for leg i for the GALLOP-STANCE-CONTROL function (Sect. 5.2.4). The range for each of these parameters is given in Table 5.3. The ab/ad angle ranges were constrained to be to one side of the nominal position to prevent unnatural, cross-legged solutions. The parameter ranges for the velocity biases were constrained to maintain the shoulder-braking and hip-thrusting

[29]The turn solutions are evolved after 26 strides of galloping following the jump-start maneuver discussed in the next chapter.

Function	Parameters	Description
TRANSFER	$\sigma_{a_e} = 0$, $\sigma_{h_e} = 0$, $\sigma_{k_d} = 1$, $\sigma_s = 0$, $T_r = 0.0$, $\boldsymbol{\theta}_{l_f} = [\theta^*_{a_i}, \theta_{v_i}, \frac{l_0}{2}]^T$, $T_t = 1.0$, $T = 0.1\,\mathrm{s}$	Transfer using evolved ab/ad angle $\theta^*_{a_i}$, and virtual leg angle θ_{v_i} given by $\theta_{v_i} = \begin{cases} 0.69 \text{ rad} & \text{if } i \leq 2 \\ 0.65 \text{ rad} & \text{otherwise,} \end{cases}$ where $i = 1, \ldots, 4$. Length l_0 is given in Table 2.1.
WAIT-FOR-TRIGGER	$L = i$, $R = 0$, $\delta = 0.0\,\mathrm{s}$, $h_t = 0.025\,\mathrm{m}$, $\sigma_{a_e} = 0$, $\sigma_{h_e} = 0$, $\sigma_{k_f} = 0$, $\sigma_s = 0$, $T_r = 0.0$, $\boldsymbol{\theta}_{l_d} = [\theta^*_{a_i}, \theta_{v_i}, l_0]^T$, $T = 1.0\,\mathrm{s}$	Hold using ab/ad and virtual leg angles $\theta^*_{a_i}$ and θ_{v_i} from TRANSFER (above) using trigger parameters $L = i$, $R = 0$ (FOOT-HEIGHT, Table 2.6), and $h_t = 0.025\,\mathrm{m}$.
EARLY-RE-TRACTION	$\sigma_{a_e} = 0$, $\theta_a = \theta^*_{a_i}$, $v_d = 4.15\,\mathrm{m/s}$, $T = 1.0\,\mathrm{s}$	Hold ab/ad joint at $\theta^*_{a_i}$ from TRANSFER (above) and rotate hip to achieve tangential foot velocity v_d.
GALLOP-STANCE-CONTROL	$v_d = 4.15\,\mathrm{m/s}$, $v_b = v^*_{b_i}$, $E_d = E_i$, $\sigma_{\text{slip}} = 1$, $T = 1.0\,\mathrm{s}$	Use velocity control with evolved velocity bias $v^*_{b_i}$. Use evolved target energy value E^*_i for leg i, $i = 1, \ldots, 4$, and use slip control.
FREE	$\sigma_{a_f} = 0$, $\sigma_s = 1$, $T_r = 0.75$, $T = 0.025\,\mathrm{s}$	Reset knee spring to the nominal position over 75% of period T while holding the ab/ad angle and allowing the hip to rotate freely.

Table 5.1: Summary of the leg function parameters for the turn.

Parameter	Value
σ_p	0
σ_{a_e}	0
σ_{h_e}	0
σ_{k_d}	1
σ_{l_a}	1
$\boldsymbol{\theta}_b = [\ \alpha,\ \beta,\ \gamma,$ $^{e}x_b, {}^{e}y_b, {}^{e}z_b,$ $\dot{\alpha},\ \dot{\beta},\ \dot{\gamma},$ $^{e'}\dot{x}_b, {}^{e'}\dot{y}_b, {}^{e}\dot{z}_b\ ]^T$	$[\ 0.03\,\text{rad},\ -0.01\,\text{rad},\ -0.08\,\text{rad},$ $0.0\,\text{m},\ 0.0\,\text{m},\ 0.23\,\text{m},$ $-0.07\,\text{rad/s}, -0.16\,\text{rad/s}, 0.39\,\text{rad/s},$ $4.17\,\text{m/s},\ 0.14\,\text{m/s},\ 0.0\,\text{m/s}\ ]^T$
$\boldsymbol{\delta}_b$	$\mathbf{0}_{12\times1}$
$\boldsymbol{\phi}_l = [\ \theta^1_{a_0}, \ldots, \theta^4_{a_0},$ $\theta^1_{v_0}, \ldots, \theta^4_{v_0},$ $d^1_0,\ \ldots,\ d^4_0,$ $\theta^1_{a_f}, \ldots, \theta^4_{a_f},$ $\theta^1_{v_f}, \ldots, \theta^4_{v_f},$ $d^1_f,\ \ldots,\ d^4_f\ ]^T$	$[\ 1.64\,\text{rad},\ 1.55\,\text{rad},\ 1.66\,\text{rad},\ 1.55\,\text{rad},$ $-0.82\,\text{rad}, -0.77\,\text{rad}, -0.78\,\text{rad}, -0.76\,\text{rad},$ $0.24\,\text{m},\ \ldots,\ 0.24\,\text{m},$ $1.66\,\text{rad},\ 1.55\,\text{rad},\ 1.66\,\text{rad},\ 1.55\,\text{rad},$ $0.69\,\text{rad},\ 0.69\,\text{rad},\ 0.65\,\text{rad},\ 0.65\,\text{rad},$ $0.12\,\text{m},\ \ldots,\ 0.12\,\text{m}\ ]^T$
$\boldsymbol{\theta}_l = [\ \theta_{a_1}, \ldots, \theta_{a_4},$ $\theta_{v_1}, \ldots, \theta_{v_4},$ $d_1,\ \ldots,\ d_4,$ $\dot{\theta}_{a_1}, \ldots, \dot{\theta}_{a_4},$ $\dot{\theta}_{v_1}, \ldots, \dot{\theta}_{v_4},$ $\dot{d}_1,\ \ldots,\ \dot{d}_4\ ]^T$	$[\ 1.64\,\text{rad},\ 1.55\,\text{rad},\ 1.66\,\text{rad},\ 1.55\,\text{rad},$ $-0.68\,\text{rad},\ -0.18\,\text{rad},\ 0.15\,\text{rad},\ 0.53\,\text{rad},$ $0.21\,\text{m},\ 0.13\,\text{m},\ 0.14\,\text{m},\ 0.21\,\text{m},$ $0.20\,\text{rad/s},\ 0.08\,\text{rad/s},\ -0.02\,\text{rad/s},\ 0.03\,\text{rad/s},$ $12.84\,\text{rad/s}, 21.49\,\text{rad/s}, 20.18\,\text{rad/s}, 13.05\,\text{rad/s},$ $-3.24\,\text{m/s},\ -1.64\,\text{m/s},\ 2.44\,\text{m/s},\ 3.24\,\text{m/s}\ ]^T$
$\boldsymbol{t}^* = [t^*_1, t^*_2, t^*_3, t^*_4]^T$	$[0.19,\ 0.44,\ 0.61,\ 0.82]^T$
T_t	$0.1\,\text{s}$

Table 5.2: Summary of the INIT-TOF parameters for initializing the turn after several seconds of galloping. In $\boldsymbol{\theta}_b$, $^{e}x_b = {}^{e}y_b = 0.0\,\text{m}$ for convenience.

Parameter	Turn Direction	Range	
$\theta^*_{a_i}$	CW	$[\frac{\pi}{2},\ (\frac{\pi}{2}+0.75)]$	rad
$\theta^*_{a_i}$	CCW	$[(\frac{\pi}{2}-0.75),\ \frac{\pi}{2}]$	rad
$v^*_{b_1}$,$v^*_{b_2}$	CW/CCW	$[-6.0,\ 0.0]$	m/s
$v^*_{b_3}$,$v^*_{b_4}$	CW/CCW	$[0.0,\ 6.0]$	m/s
E^*_i	CW/CCW	$[0.0,\ 20.0]$	J

Table 5.3: Ranges for each evolved parameter for the CW and CCW turns. The ranges for the ab/ad angles $\theta^*_{a_i}$ are constrained to one side of the nominal angle $\frac{\pi}{2}$ rad, based on turning direction, to prevent unnatural, cross-legged solutions.

stabilization employed during galloping, since a similar pitch motion is expected during the turn stride. In general, the parameter ranges were tuned through experimentation to provide a wide range of results while preventing the use of excessive joint torques or positions.

5.4 Fitness Function Criteria

In this section, the fitness function for the turn is presented. Like the gallop, the turn is evaluated using three different criteria. However, the criteria include two accuracy components, one for the overall orientation of the body at the TOF point at the end of the turn stride, and one for achieving the desired change in yaw. The third element is a correctness factor that includes achieving the desired footfall sequence for the given turning direction. The fitness vector is given as follows:

$$\boldsymbol{f}_{\mathrm{T}} = [f_{a'},\ f_{\Delta\alpha},\ f_{c'}]^T \quad . \tag{5.2}$$

Parameter	Value	Parameter	Value	Parameter	Value
λ_α	0.0	λ_y	0.0	$\lambda_{\dot{\gamma}}$	2.9
λ_β	11.5	λ_z	200.0	$\lambda_{\dot{x}}$	10.0
λ_γ	5.7	$\lambda_{\dot{\alpha}}$	0.0	$\lambda_{\dot{y}}$	10.0
λ_x	0.0	$\lambda_{\dot{\beta}}$	5.7	$\lambda_{\dot{z}}$	0.0

Table 5.4: Scaling factors used to compute the accuracy fitness component for the turn. Scaling factors of zero indicate "don't-care" conditions.

The first accuracy component is computed similar to the gallop's accuracy factor (Eq. (4.32)) and is given as follows:

$$f_{a'} = \frac{w_{a'}}{1 + \| \boldsymbol{\Lambda}' \left(\boldsymbol{\theta}_b - \boldsymbol{\theta}_{b_d} \right) \|^2} \tag{5.3}$$

where $w_{a'} = 500.0$, and $\boldsymbol{\theta}_{b_d}$ is given as follows:

$$\begin{aligned} \boldsymbol{\theta}_{b_d} &= \left[\; \alpha_d,\, \beta_d,\, \gamma_d,\, {}^e x_{b_d},\, {}^e y_{b_d},\, {}^e z_{b_d},\, \dot{\alpha}_d,\, \dot{\beta}_d,\, \dot{\gamma}_d,\, {}^{e'}\dot{x}_{b_d},\, {}^{e'}\dot{y}_{b_d},\, {}^e\dot{z}_{b_d} \; \right]^T \\ &= \left[\; 0,\;\; 0,\;\; 0,\;\; 0,\;\;\; 0,\;\;\; l_0,\;\; 0,\; \dot{\beta}_d^*,\; 0,\;\; v_d,\;\;\; 0,\;\;\; 0 \; \right]^T , \end{aligned} \tag{5.4}$$

where l_0 is the nominal leg length of 0.24 m (Table 2.1), $\dot{\beta}_d^*$ is the evolved pitch rate for the gallop of -0.47 rad/s (Table 4.6), and v_d is the desired forward running speed of 4.15 m/s. The diagonal elements in $\boldsymbol{\Lambda}'$ are given in Table 5.4. Unlike the gallop accuracy scaling factors (Table 4.5), the turn accuracy factors for both the yaw and yaw rate are don't-care conditions. This is because the yaw rate is generally not known in advance, and the yaw is included in the yaw accuracy factor. Furthermore, the roll and roll rate scaling factors have been relaxed to allow for the roll action which is inherent in the high-speed turn.

The yaw accuracy component is computed as follows:

$$f_{\Delta\alpha} = \frac{w_{\Delta\alpha}}{1 + \left(\lambda'_{\Delta\alpha} \left(\alpha - \alpha_d \right) \right)^2} , \tag{5.5}$$

where $w_{\Delta\alpha} = 100.0$, α_d is the target turn angle in the range $[-0.61, 0.61]$ rad at 0.087 rad increments (Sect. 5.3), α is the TOF yaw angle of the body, and $\lambda'_{\Delta\alpha}$ is the scaling factor for yaw, which is set to 114.6 to yield a 0.0087 rad (0.5 deg) tolerance.

Finally, the correctness factor is computed using a modified form of Eq. (4.35) as follows:

$$f_{c'} = f_{\mathrm{f}} \left(w_{c'} \left(\sum_{k=1}^{4} t_{c_k} - t_e^* \right) \right) , \tag{5.6}$$

where the scaling factor $w_{c'}$ is set to 60.0, t_{c_k} is the amount of time before an error is encountered in leg k, t_e^* is the length of time after the first error occurs in any leg, and f_{f} is the footfall factor, computed as

$$f_{\mathrm{f}} = \left(\frac{s}{4} \right) \left(\frac{n_c}{4} \right) \left(\frac{1}{1 + w_{\mathrm{td}} \left(\Delta t_{\mathrm{td}_f} + \Delta t_{\mathrm{td}_r} \right) + w_{\mathrm{IF}} \left(t_{\mathrm{IF}_f} + t_{\mathrm{IF}_r} \right)} \right) , \tag{5.7}$$

where s is the number of correct footfalls, n_c is the total number of contact periods for the stride, Δt_{td_f} and Δt_{td_r} are the front and rear touchdown separation times between each leg in the pair, t_{IF_f} and t_{IF_r} are the front and rear inter-leg flight (IF) times, w_{td} is a weight factor for the touchdown separation time, and w_{IF} is a weight factor for the IF time. The values of w_{td} and w_{IF} are given below. The total number of contact periods n_c is given by

$$n_c = \sum_{i=1}^{4} c_i \, , \tag{5.8}$$

where c_i is the number of contact periods for leg i.

The first element in Eq. (5.7) rewards solutions with the maximum number of correct footfalls, where the desired sequence is 3-4-1-2 for a CCW turn and 4-3-2-1 for a CW turn (due to the use of the FOOT-HEIGHT triggering scheme discussed in Sect. 5.2.2). For example, if a CCW solution consists of the actual footfall sequence

3-4-2-1, then $\frac{s}{4} = 0.5$, since only the first two footfalls are correct. The second term in Eq. (5.7) rewards solutions that achieve at least one contact per leg, since many solutions are possible in which one or more legs never make contact. The maximum fitness for the second term is 1, which occurs when every leg makes contact.

The third term in Eq. (5.7) rewards solutions that have no excess touchdown separation time (Δt_{td_f} and Δt_{td_r}) and no flight phases (t_{IF_f} and t_{IF_r}) between each leg in the front and rear pairs. Excess leg spread is measured by the difference in touchdown times between the front or rear legs. For the front legs, the excess leg spread factor is computed as follows:

$$\Delta t_{\mathrm{td}_f} = \max\left(0,\ |t_{\mathrm{td}_1} - t_{\mathrm{td}_2}| - t_{\mathrm{td}_t}\right) , \tag{5.9}$$

where t_{td_1} is the touchdown time for leg 1, and t_{td_t} is a threshold time of 50.0 ms. The excess leg spread for the rear legs is computed in a similar manner. Any touchdown delays between the front or rear legs in excess of the threshold is penalized in Eq. (5.7) by w_{td}, which is set to 100.0 so that the third term evaluates to 0.5 for a total excess touchdown delay of 10.0 ms, assuming no IF time.

The inter-leg flight times t_{IF_f} and t_{IF_r} are measured between the liftoff of one leg and the touchdown of the other. For example, t_{IF_r} is computed as follows:

$$t_{\mathrm{IF}_r} = \max\left(0, t_{\mathrm{td}_{t_r}} - t_{\mathrm{lo}_{l_r}}\right) , \tag{5.10}$$

where the subscript t_r is the index of the trailing rear leg (e.g., $t_r = 3$ for a CW turn, assuming the correct footfall sequence), subscript l_r is the index of the leading hind leg, and t_{lo_i} is the lift-off time for leg i. If the trailing leg touches down *after* the leading leg lifts off, the IF time will be positive. Otherwise, the IF time is zero. The factor w_{IF} is set to 900.0 so that the third term in Eq. (5.7) evaluates to 0.1 for a

total of 10.0 ms of IF time, assuming no excess leg spread. Thus, $f_{\mathrm{f}} = 1$ only if all of the following are true:

- All legs have made contact,
- All legs have touched down in the proper sequence,
- There has been no excess leg spread between front or between rear legs, and
- There has been no IF time between front or between rear legs.

Multiplying f_{f} times the remainder of the correctness factor in Eq. (5.6) generates the final measure of correctness, where the scaling factor $w_{c'} = 60.0$ yields a maximum fitness of approximately 100.0. In this way, the fitness sharing factor of 6.5 generates approximately 16 clusters. The sharing factors for the other two criteria are also set to 6.5, as both of these components are scaled to produce maximum values near 100.0.

5.5 Results

A total of 10 trials of 250 generations per trial were run for each turn angle from -35.0 to 35.0 deg at 5.0 deg increments. Results were obtained by combining all archives into a single set, calculating a new Pareto front, then selecting all individuals with fitness values greater than $\boldsymbol{\eta}_{\mathrm{T}} = [0.0, 50.0, 50.0]^T$. The first component, the overall accuracy criterion $f_{a'}$, was thresholded at 0.0 since increasing turn angles require larger roll angles and rates, which causes the overall accuracy to degrade to very small values toward the boundaries of the turn angle range. The yaw accuracy and correctness components $f_{\Delta\alpha}$ and $f_{c'}$ where thresholded at 50.0 to select the better solutions along these dimensions. For $f_{\Delta\alpha}$, a threshold value of 50.0 eliminated solutions with yaw angles not within 0.0087 rad (0.5 deg) of the desired angle.

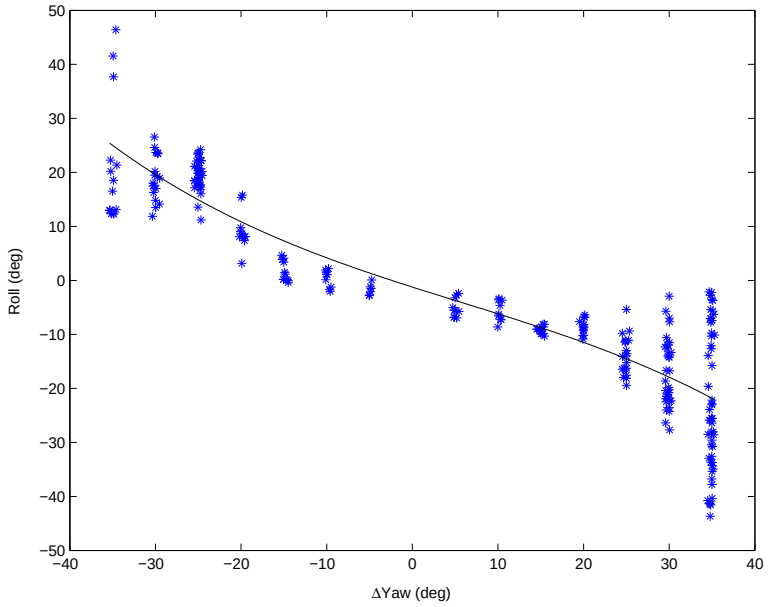


Figure 5.1: Roll vs. change in yaw for the turn. The solid line is the least-squares regression line for the actual data. The banking nature of the turn results in the coupling between roll and yaw.

The results for the TOF roll vs. the change in yaw are plotted in Fig. 5.1. The figure illustrates the strong correlation between yaw angle and the final roll angle of the body. This relationship demonstrates that the high-speed turn is more of a bank, where both roll and yaw are coupled together, as stated previously.

The simple conical pendulum model demonstrates the basic principle, as shown in Fig. 5.2. The angle of the conus ζ formed by the pendulum gives the tilt angle of the force $\boldsymbol{T}$ and is found using the following relation:

$$\tan(\zeta) = \frac{{v_t}^2}{rg}, \tag{5.11}$$

where v_t is the tangential velocity of the block, r is the radius of its horizontal circular trajectory, and g is gravity. To compare the conical pendulum model with the actual

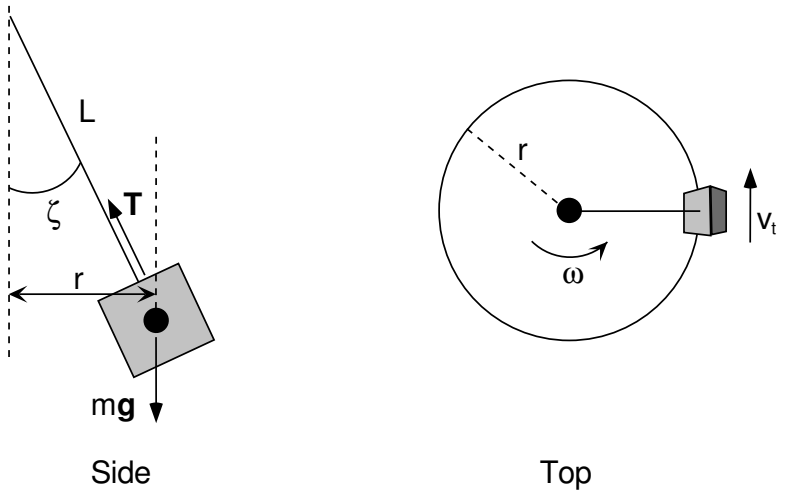


Figure 5.2: The conical pendulum model for the high-speed turn. The angle of the conus ζ gives the tilt angle of the required force $\boldsymbol{T}$ and is given by $\tan(\zeta) = \omega^2 r/g = (v_t)^2/rg$, where ω is the angular velocity, v_t is the tangential velocity, and r is the radius of the horizontal circular trajectory.

data, each data point's computed angle ζ_c is calculated using the relation in Eq. (5.11) as follows:

$$\zeta_c = \arctan \frac{\left({}^{e'}\dot{x}_b\right)^2}{r_c\, g}\,, \tag{5.12}$$

where ${}^{e'}\dot{x}_b$ is the forward velocity in the rotated frame $\{\boldsymbol{e'}\}$ (Sect. 2.4.4), which is approximately equal to the tangential velocity v_t, and r_c is the computed turn radius. The computation of the turn radius is illustrated in Fig. 5.3. The change in position $|\Delta \boldsymbol{p}|$ and the change in yaw angle $\Delta\alpha$ are used to estimate the turn radius using the Law of Cosines, as follows:

$$r_c = \sqrt{\frac{(\Delta p)^2}{2\,(1-\cos\Delta\alpha)}}\quad . \tag{5.13}$$

The estimated turn radius r_c for each data point using this calculation is shown in Fig. 5.4.

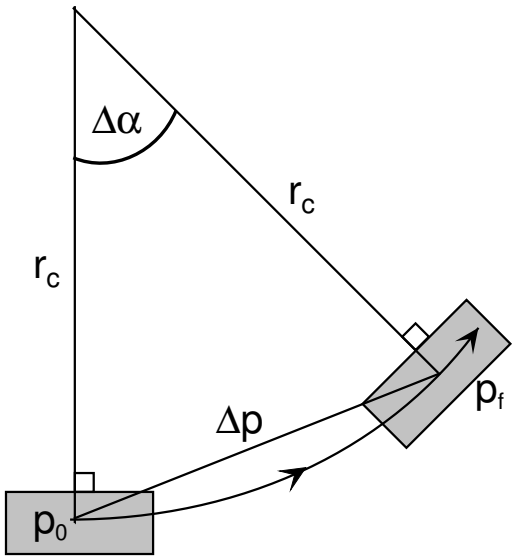


Figure 5.3: Computation of the turn radius. The magnitude of the change in position $|\Delta \boldsymbol{p}|$ and the change in yaw angle $\Delta\alpha$ are used to compute the radius r_c using the Law of Cosines: $r_c = \sqrt{(\Delta p)^2/2(1 - \cos \Delta\alpha)}$.

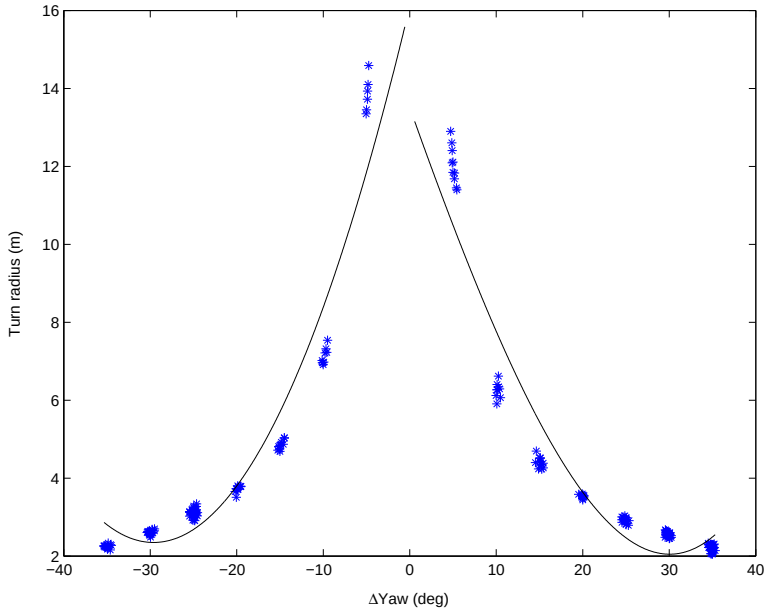


Figure 5.4: Estimated turn radius r_c vs. change in yaw for the high-speed turn. The solid curves are computed using least-squares regression with a second-order polynomial function. The turn radius computation is shown in Fig. 5.3.

Although there are several methods for computing the turn radius, the geometric approach described above tends to fit the trajectory data fairly well, producing reasonable results based on multiple-stride turning, discussed in Sect. 5.7. However, another approach for determining the turn radius is to use the average change in velocity, where velocity is measured in earth coordinates. This approach is illustrated in Fig. 5.5. The average angular velocity ω_{ave}, the magnitude of the change in velocity $|\Delta \boldsymbol{v}|$, and the turn stride period Δt are used to estimate the turn radius. The average angular velocity is computed as follows:

$$\omega_{\text{ave}} = \frac{\dot{\alpha}_0 + \dot{\alpha}_f}{2}\,, \tag{5.14}$$

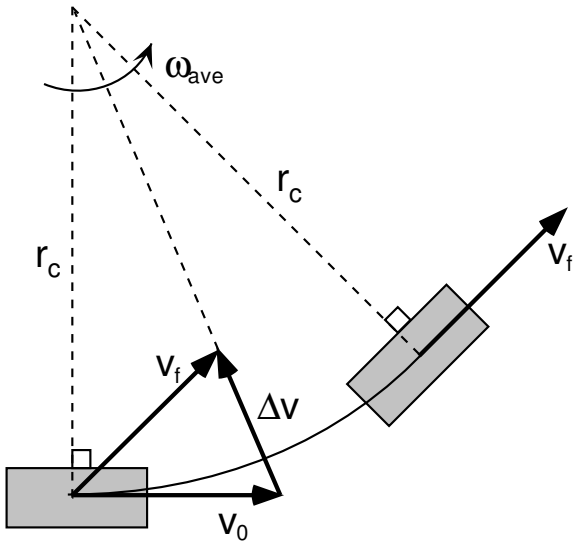


Figure 5.5: Alternative computation of the turn radius. The magnitude of the change in velocity $|\Delta v|$ and the average angular velocity ω_{ave} are used to compute the radius of curvature r_c using $|\omega_{\text{ave}}|^2 r_c = |\Delta \boldsymbol{v}/\Delta t|$.

where $\dot{\alpha}_0$ is the initial TOF yaw rate (Table 5.2) and $\dot{\alpha}_f$ is the final TOF yaw rate. The average angular velocity is shown in Fig. 5.6. The change in velocity is given by

$$\Delta \boldsymbol{v} = [\Delta v_x,\, \Delta v_y,\, \Delta v_z]_e^T\,, \tag{5.15}$$

where Δv_x and Δv_y are shown in Figs. 5.8 and 5.9. Δv_z is zero since the initial and final velocities along $\hat{\boldsymbol{z}}_e$ are zero at TOF. The turn stride period Δt is measured for each individual and shown in Fig. 5.7. The magnitude of the change in velocity $|\Delta \boldsymbol{v}|$ is then used to find the effective turn radius r_c as follows:

$$r_c = \frac{|\Delta \boldsymbol{v}|}{\Delta t \left(\omega_{\text{ave}}\right)^2} \quad . \tag{5.16}$$

This approach assumes the effective turn radius is related to the change in heading velocity, which is caused by the average centripetal acceleration throughout the turn

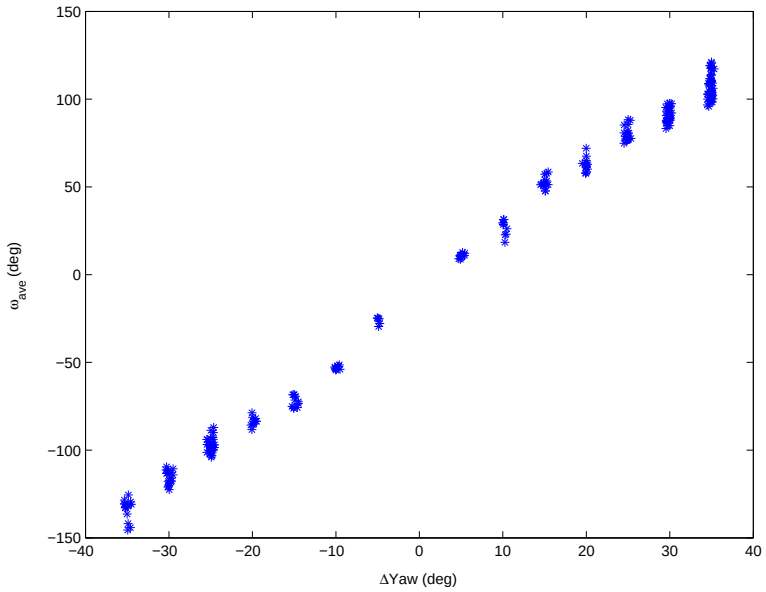


Figure 5.6: Average angular velocity vs. change in yaw for the high-speed turn. As expected, the average angular velocity is proportional to the requested change in yaw angle.

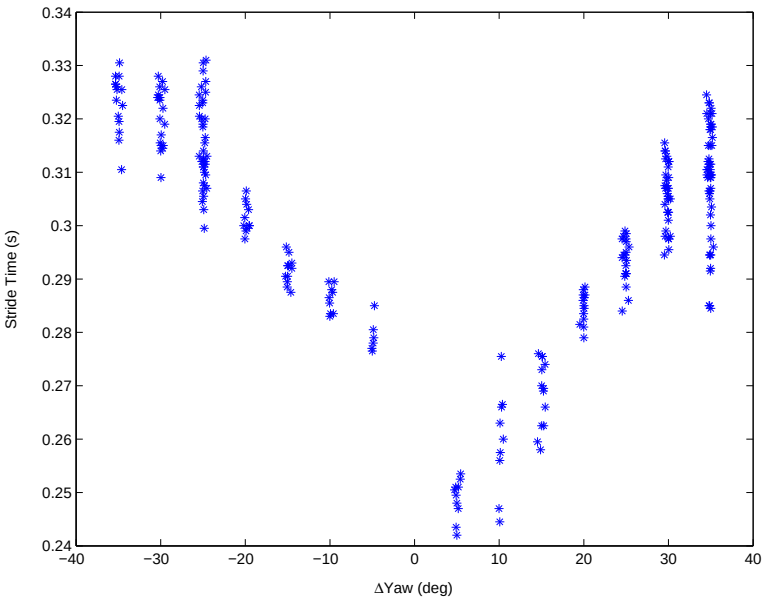


Figure 5.7: Stride period for the turn. At larger turn angles, more aerial time occurs due to larger ab/ad angles (Fig. 5.12).

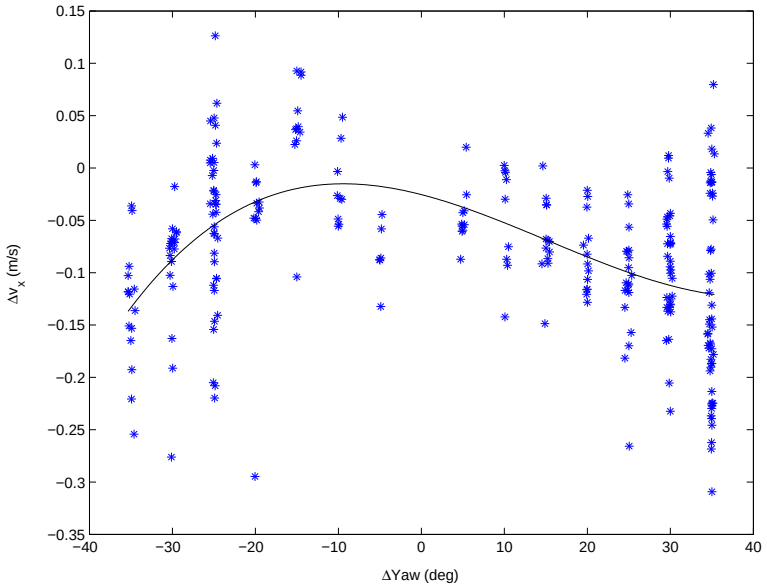


Figure 5.8: Change in Δv_x for the turn. The change is fairly small because the centripetal acceleration is perpendicular to $\hat{\boldsymbol{x}}_e$.

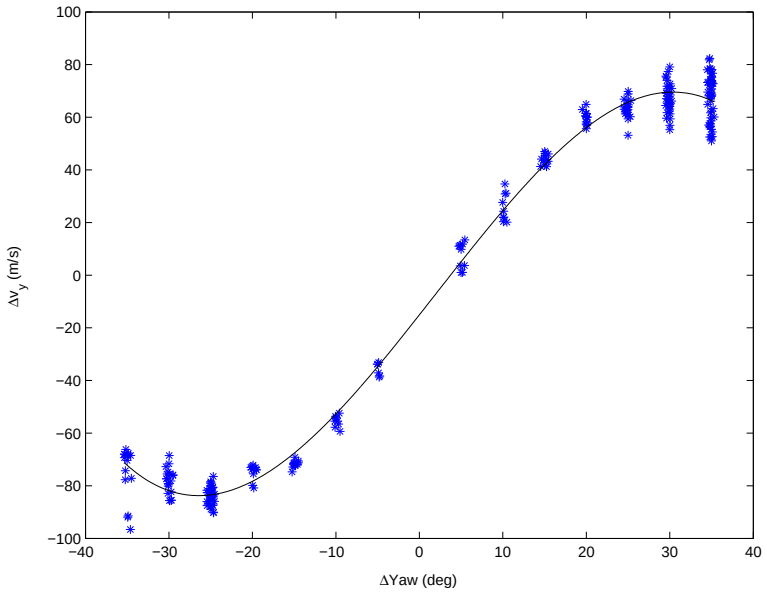


Figure 5.9: Change in Δv_y for the turn. The magnitude of the change is strongly correlated with the yaw angle, since the centripetal acceleration acts along $\hat{\boldsymbol{y}}_e$.

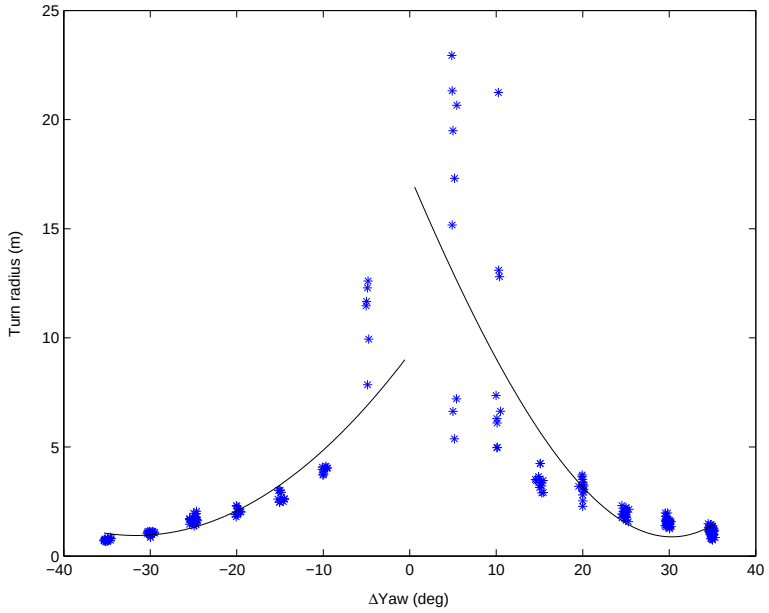


Figure 5.10: Estimated turn radius vs. change in yaw using the velocity approach for computing the turn radius (Fig. 5.5). The solid curves are computed using least-squares regression with a second-order polynomial function.

stride. A plot of the estimated turn radius using this approach is shown in Fig. 5.10. Although the values are not significantly different than those in Fig. 5.4, there appears to be more noise in the data, which is not unexpected since velocity is typically noisier than position. Furthermore, the turn radius appears to be understated for larger magnitude turn angles based on the results in Sect. 5.7. Consequently the geometric approach is used for the remaining analysis.

Once the turn radius is computed using the geometric approach (Fig. 5.4), the tilt angle of the required reaction force ζ_c is computed using Eq. (5.12). The results are plotted in Fig. 5.11. If the model applies to the data generated for the turn, then the ab/ad angles should fit the predicted angles in Fig. 5.11, assuming that the forces act mostly along the leg. The ab/ad angles with respect to the body's

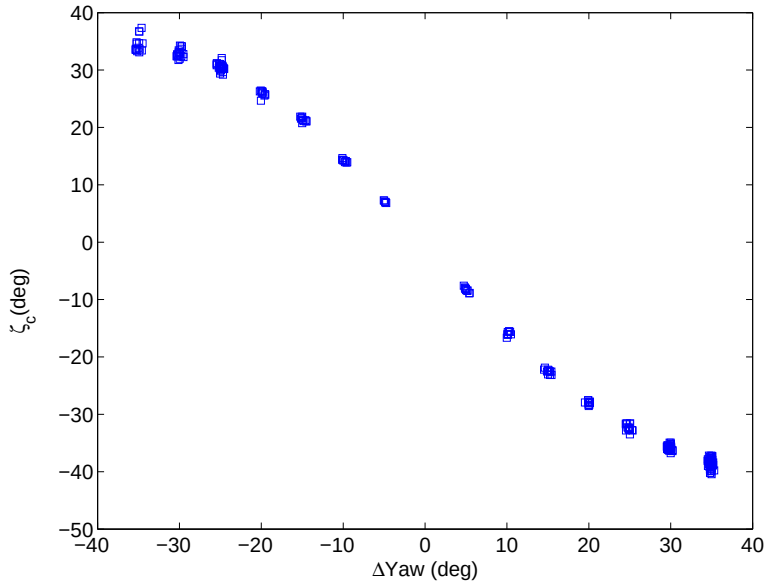


Figure 5.11: Computed tilt angle ζ_c vs. change in yaw for the turn. ζ_c is computed for each data point using the conical pendulum model in Fig. 5.2 and the geometric approach for computing the turn radius r_c (Fig. 5.3).

normal (i.e., $\Delta\theta_{a_i} = \theta_{a_i} - \frac{\pi}{2}$) versus the computed tilt angle ζ_c for each data point are shown in Fig. 5.12. Note that the roll angle of the body is neglected in this analysis since it remains small during the contact phases (i.e., less than 5.0 deg on average for the largest turn angles). The figure shows that the computed tilt angles (shown as squares) match the ab/ad angles more closely for the front legs and for larger requested turn angles in both directions. Interestingly, the computed tilt angle has a much looser fit for the rear legs, although it does appear to match leg 4's ab/ad angles for positive requested turn angles. These results would suggest that the front legs are chiefly responsible for generating the required centripetal acceleration for the turn, while the rear legs are responsible for motion in a different plane, presumably the sagittal plane, which is tangential to the turn.

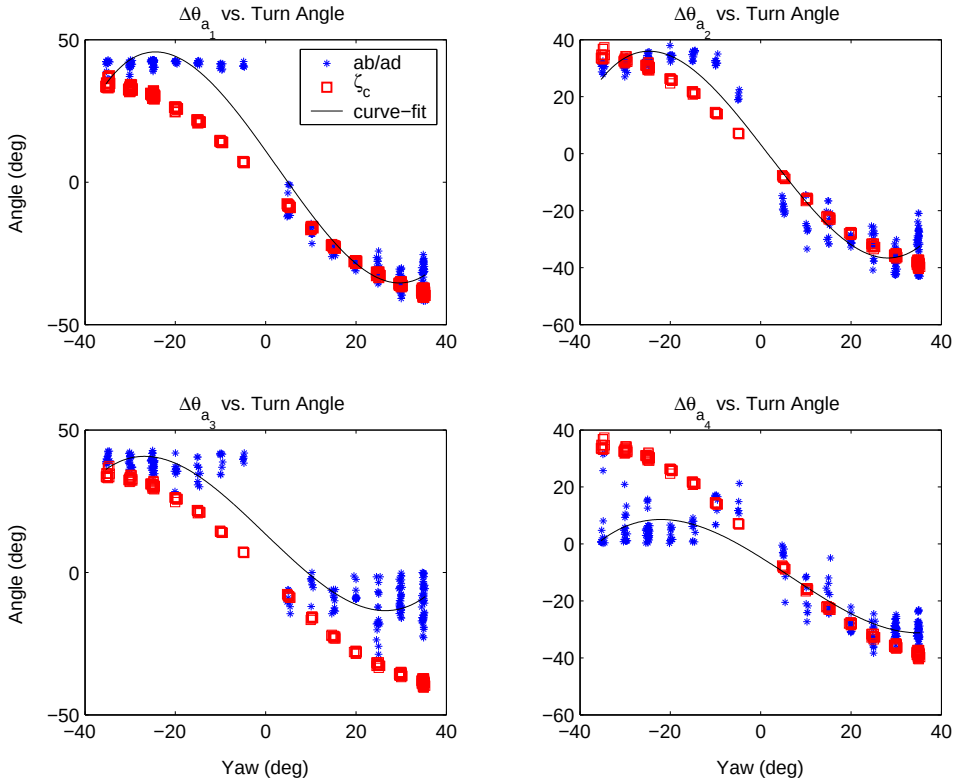


Figure 5.12: Evolved ab/ad angles with respect to the body's normal (i.e., $\Delta\theta_{a_i} = \theta_{a_i} - \frac{\pi}{2}$) versus change in yaw angle for the turn. The roll angle of the body is neglected since it remains small on average during contact. Squares indicate the computed tilt angle ζ_c for the conical pendulum model, and the solid line is a third-order polynomial regression line for the evolved ab/ad data points, which are shown as asterisks.

Although the front leg ab/ad angles seem to match the model, a more telling result would be if the projection of the average ground reaction force vector onto the transverse plane demonstrated a tilt angle that agreed with the model. This transverse plane tilt angle for the average ground reaction force is computed using the average lateral and vertical force components as follows:

$$\zeta_{F_{\text{ave}}} = \arctan\left(\frac{{}^{e'}F^{y}_{\text{ave}}}{{}^{e}F^{z}_{\text{ave}}}\right), \tag{5.17}$$

where ${}^{e'}F^{y}_{\text{ave}}$ is the average lateral contact force in the rotated frame $\{\boldsymbol{e}'\}$ (Sect. 2.4.4) and ${}^{e}F^{z}_{\text{ave}}$ is the average vertical force in earth coordinates. The computed transverse plane tilt angle of the ground reaction forces $\zeta_{F_{\text{ave}}}$ is plotted against the computed tilt angle ζ_c in Fig. 5.13. The figure shows that there is better agreement between the predicted tilt angle ζ_c and the transverse plane tilt angle $\zeta_{F_{\text{ave}}}$ in the front legs. Furthermore, the front, inside legs agree with the model more closely than the front, outside legs. The poor agreement between the model and the rear legs suggests that they are not as critical in producing the horizontal force required for turning. On the other hand, the front legs tend to act like the cord in the conical pendulum model, producing the lateral force to generate the necessary centripetal acceleration for the turn.

This means that lateral forces in the front legs should increase with turn angle, which is shown in Fig. 5.14. Generally, the lateral forces are proportional to the turn angle such that larger CCW turns ($\Delta\alpha > 0$) require larger lateral forces in the positive $\hat{\boldsymbol{y}}_{e'}$ direction, toward the inside of the turn. This result agrees qualitatively with Jindrich and Full's Linear Maneuverability Number (LMN) [42], which is the ratio of the lateral impulse to the linear momentum of the system. The LMN suggests that to achieve larger yaw angles, a larger lateral impulse must be delivered, assuming

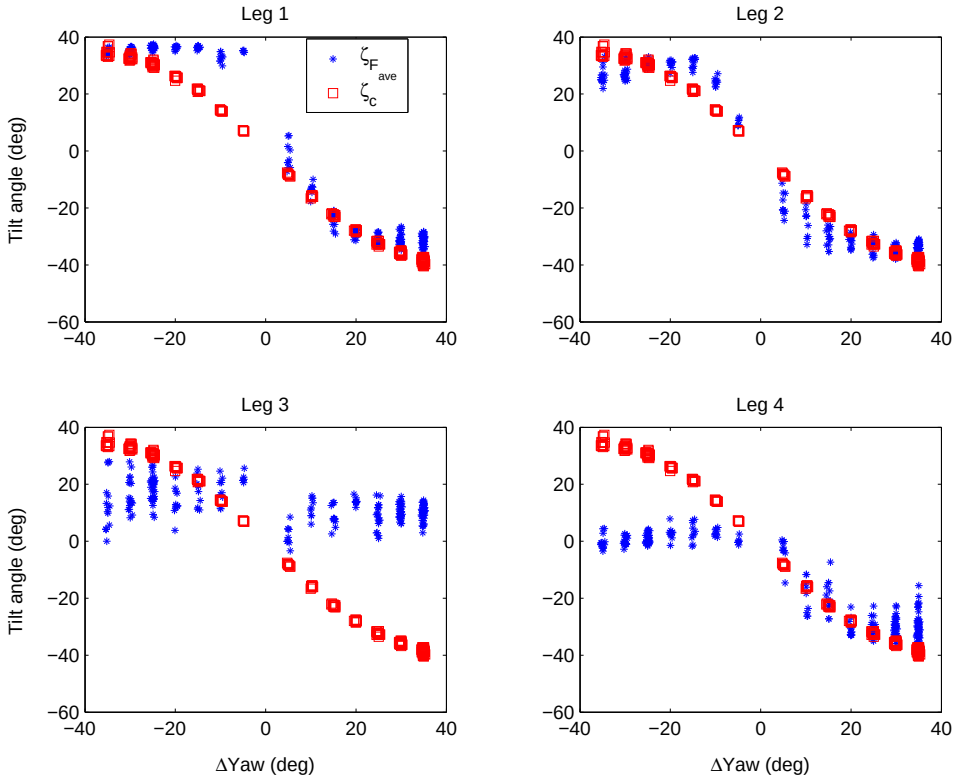


Figure 5.13: Tilt angle $\zeta_{F_{\text{ave}}}$ of the ground reaction force vector in the transverse plane ('*') vs. computed tilt angle ζ_c (squares) using the conical pendulum model. The front legs agree with the computed model more closely than the rear legs, suggesting that the front legs are more critical for turning.

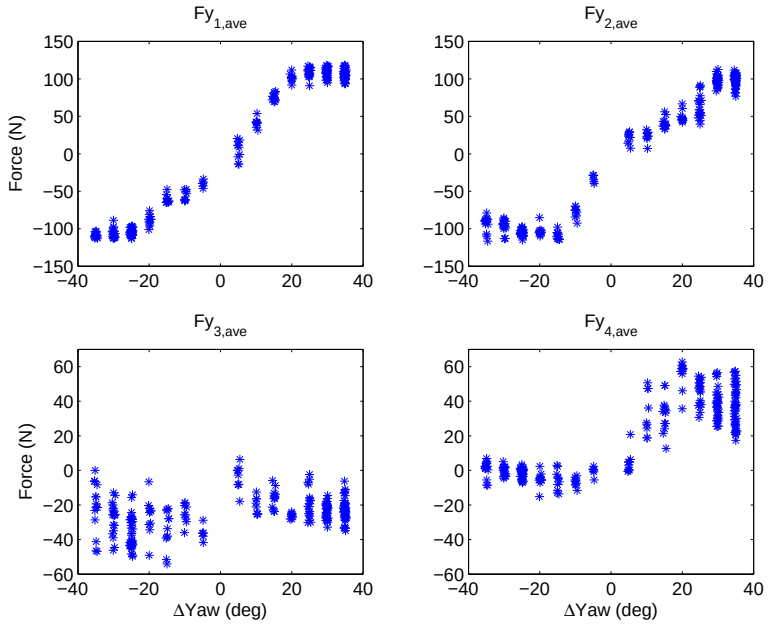


Figure 5.14: Lateral contact forces vs. change in yaw angle for each of the legs. In general, the lateral forces in the front legs are proportional to the turn angle.

a constant linear momentum. In a later study, Full et al. [93] found that the front legs were more effective in generating turns in hexapedal insects, which seems to agree with the quadrupedal data presented here.

However, Full et al. also found that the *outside* legs generated the largest overall forces for turning in hexapods. Due to the sprawled posture and alternating tripod gait, hexapod turns involve inward-directed lateral forces in the outside legs and opposing lateral forces in the inside legs [42]. The outside legs are chiefly responsible for creating the lateral impulse to generate the centripetal acceleration, while the inside legs actually counter this effort. In the quadrupedal turn described here, both the inside and outside legs are rotated in the same direction to produce the turn, as they are in biological quadruped turns [85]. Consequently, the inside legs generate

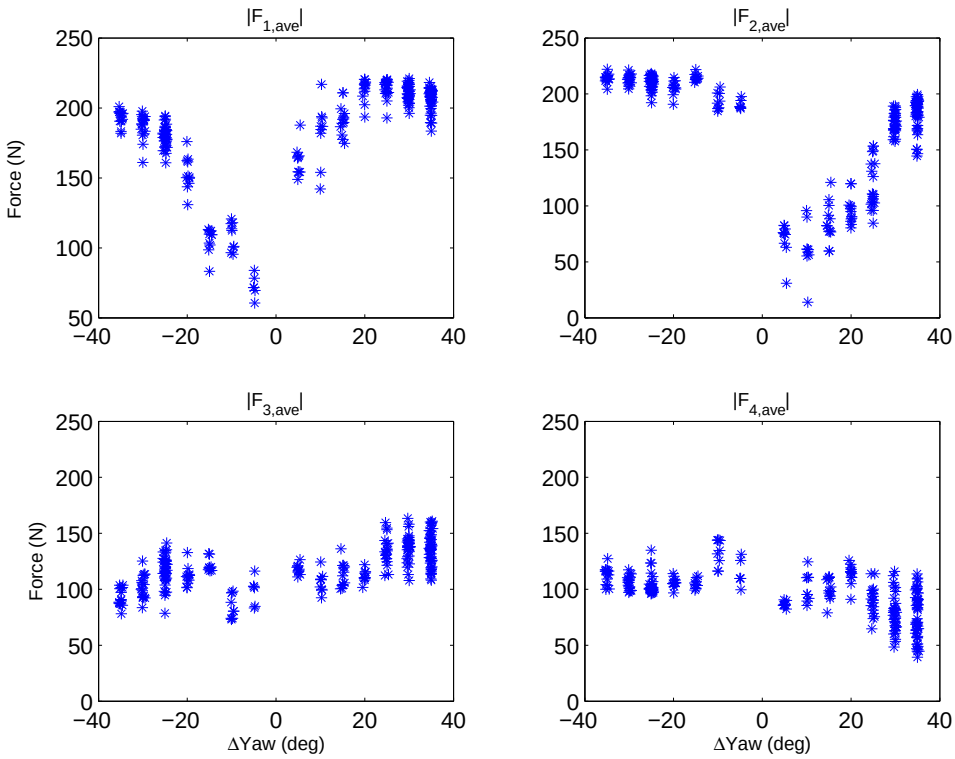


Figure 5.15: Magnitude of total average contact forces vs. change in yaw angle for each of the legs for the turn. Front "inside" legs play a significant role for the turn. For example, for a CCW turn of 20 deg (ΔYaw = 20 deg), leg 1 is the front inside leg, while leg 2 is the front outside leg. Total force for leg 1 at ΔYaw = 20 deg is about 200 N, while the total force for leg 2 is about 100 N. A similar, but opposite, result is obtained for a CW turn of 20 deg.

commensurate lateral forces with the outside legs, as shown in Fig. 5.14. Furthermore, the front, inside legs tend to generate the largest magnitude in overall force, as shown in Fig. 5.15, presumably because of the front, inside foot's close proximity to the vertical projection of the body's COM.

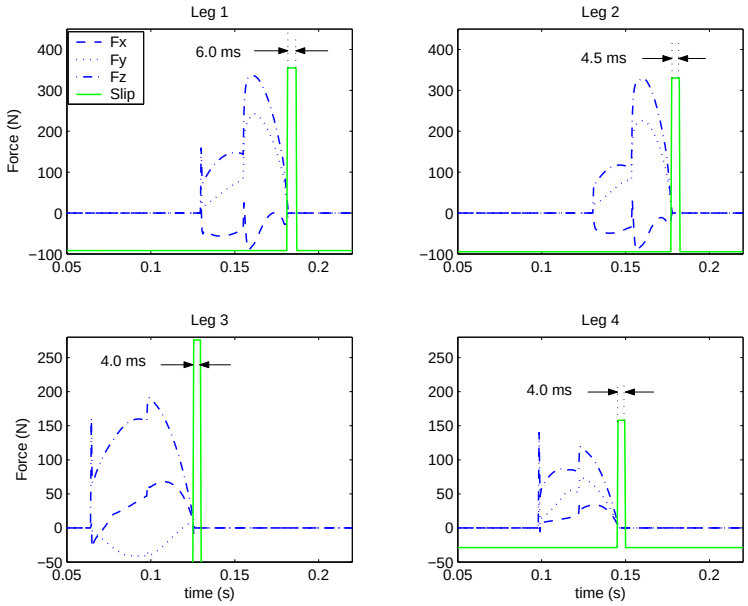
While these findings are not surprising given the different styles of turning in hexapods and quadrupeds, the implications for control are significant. In hexapods, slightly altering the foot placement of the front, outside leg is sufficient to create

a significant lateral impulse for turning [42, 93]. However, tripod support is still maintained for a statically stable configuration. For high-speed quadrupedal turns, on the other hand, the front and outside rear legs tend to rotate in the same direction with respect to the vertical to produce the required lateral forces. The front legs are chiefly responsible for turning, and due to their reaction forces, the body is rotated about both its yaw and roll axes to produce a banked turn. Even at moderate turn angles, the significant rotation of the ab/ad angles can cause the vertical projection of the COM to fall near the edge or entirely outside the support of the feet.

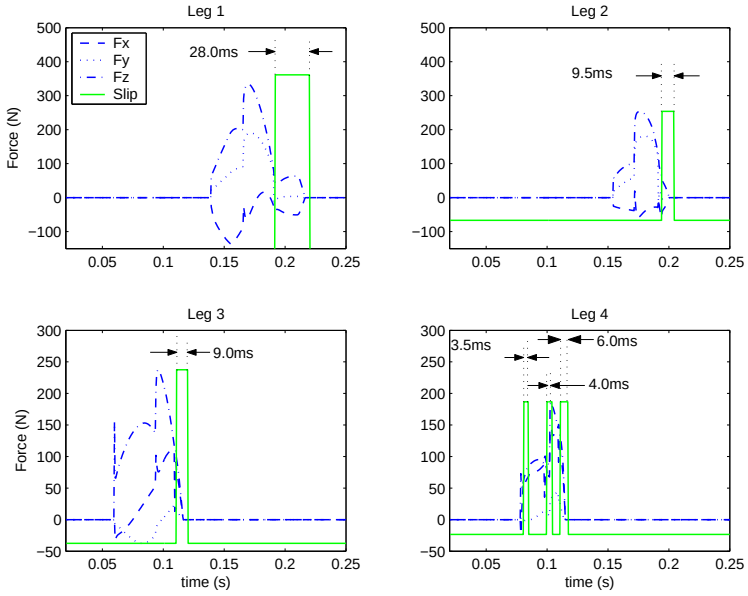
5.6 The Impact of Enforcing the No-Slip Condition

For the turn data presented above, the no-slip algorithm (Sect. 4.2.6) was employed during the stance phase for each individual. To study the impact of this algorithm on the results, another set of data was generated for the same range of turn angles, but with the no-slip algorithm disabled ($\sigma_{\text{slip}} = 0$ in GALLOP-STANCE-CONTROL, Table 5.1). As shown in Fig. 4.7, the no-slip algorithm can generally limit slipping to all but the first or last few milliseconds of each contact phase during galloping. The algorithm performed similarly for turning. When the algorithm was not used, significant slipping occurred, which is illustrated in Fig. 5.16.

The presence of significant slipping when the no-slip algorithm was disabled caused an overall degradation of the quality of solutions. This was noticeable because fewer solutions were generated after thresholding the Pareto sets. For example, there were over 250 solutions generated after thresholding for the no-slip case, while there were only about 175 solutions with slipping. Furthermore, a visual analysis of the slipping solutions revealed many instances of unexpected leg motion following contact. This



(a)



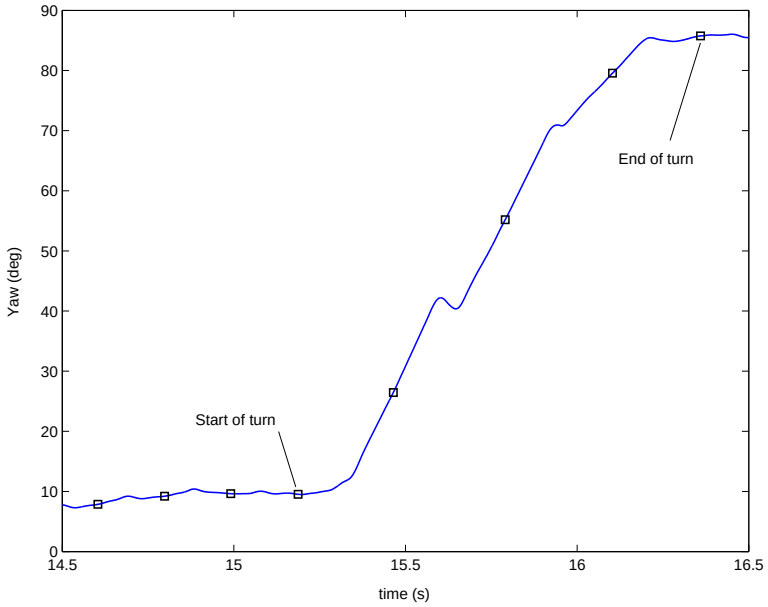
(b)

Figure 5.16: Typical turn stride for a CCW turn of 35.0 deg when the no-slip algorithm is (a) employed and (b) not employed. Durations of slipping are indicated in each plot. Significant slipping is noticeable in legs 1 and 4, plot (b), when the algorithm is not employed.

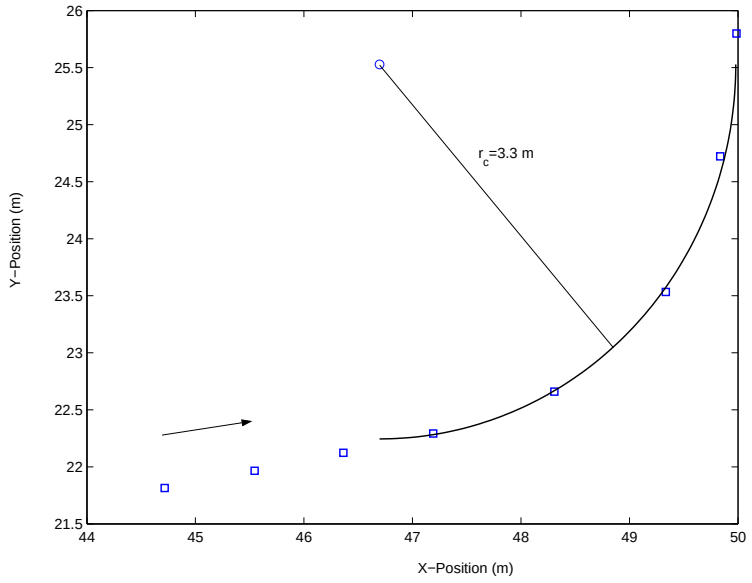
occurred because the hip torque values during contact were no longer bounded by the no-slip algorithm, causing the foot to slip and break contact at larger velocities. This resulted in additional leg retraction during the FREE phase, causing the thigh link to bounce off its internal joint limit (Table 2.2), which lead to a brief period of chaotic motion. While this type of behavior did not affect every individual, its occurrence was noted in a significant percentage of the solutions. However, no known instances of this undesired behavior occurred for the case when the no-slip algorithm was enabled.

5.7 Multiple-Stride Turns

Due to limited friction and limited torque, the quadruped is able to turn by a limited angle during each stride. In order to produce significant changes in trajectory, multiple-stride turns must be employed. For these behaviors, several turn strides are evolved in succession. Each turn stride is evolved independently and initialized with the final TOF conditions from the previous turn stride. Figures 5.17 and 5.18 illustrate two multiple-stride turns in each direction that were used during the final sequence discussed in Chap. 6. In both figures, the average change in yaw angle per stride is approximately 19.0 deg. The estimated radius of curvature of the trajectory was computed by fitting a circle to the data points using nonlinear least-squares regression in MATLAB. The average turn radius for both directions is about 3.4 m, which agrees with the estimated radius computed in Fig. 5.4. For both turns, a maximum change in yaw angle was sought instead of using the target values in Eq. (5.5). Furthermore, roll and roll rate were penalized only if their magnitudes

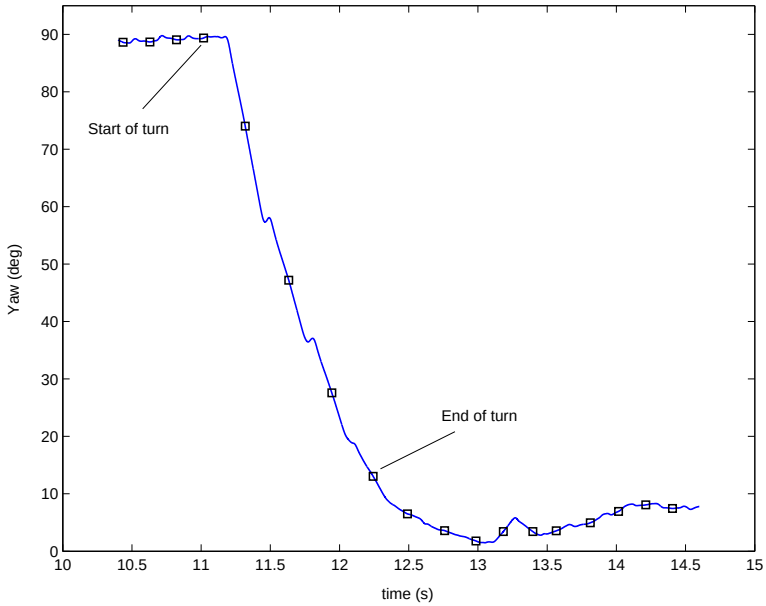


(a)

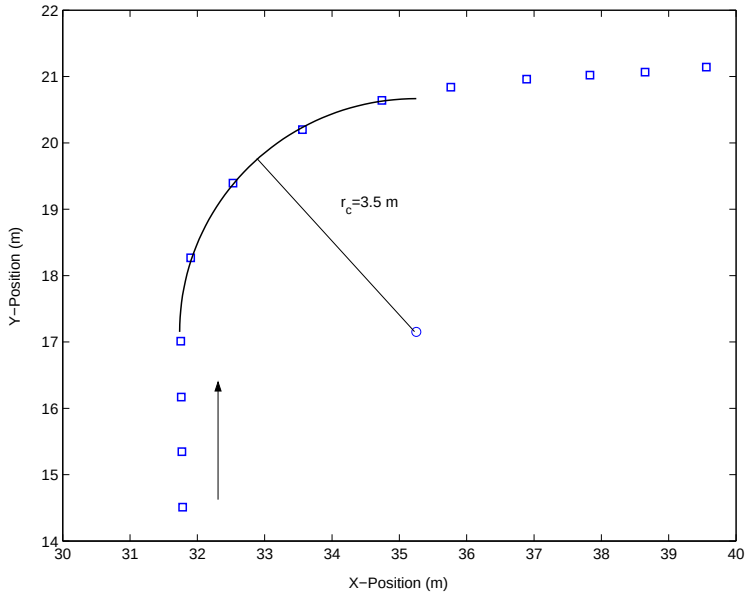


(b)

Figure 5.17: Multiple-stride turn in the CCW direction: (a) Yaw angle at TOF, indicated by squares, and (b) trajectory with circle-fitting. The arrow indicates direction of travel.



(a)



(b)

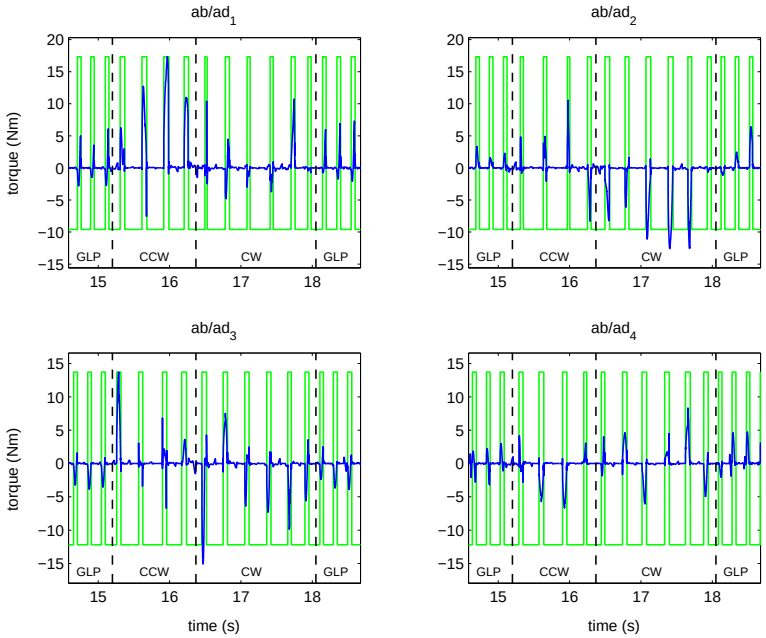
Figure 5.18: Multiple-stride turn in the CW direction: (a) Yaw angle at TOF, indicated by squares, and (b) trajectory with circle-fitting. The arrow indicates direction of travel.

exceeded 0.3 rad (17.2 deg) and 0.4 rad/s (25.5 deg/s), respectively. These values represent the maximum values from which the quadruped can still successfully transition back into a stable gallop after the turn. Using these criteria, it appears that a 90-degree turn can be accomplished in about 5 turn strides. The performance limit for a multiple-stride turn in this system appears to originate from limitations on the region of attraction for the gallop, as opposed to the properties of the turn itself.

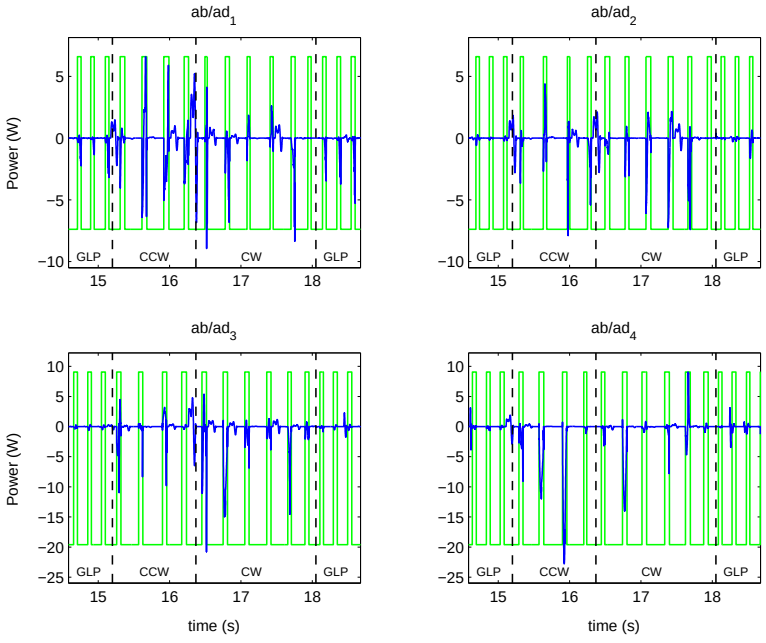
5.8 Torque and Power Analysis for Multiple-Stride Turns

In this section, the torque and power requirements for each joint, as well as the total power across all joints, are presented for two multiple-stride turns in succession. The torque and power requirements for the ab/ad joints are shown in Fig. 5.19. Torque levels in (a) for the front ab/ad joints are generally higher during the turn than during steady-state galloping (Fig. 4.11), especially for the front, inside legs (i.e., leg 1 for the CCW turn, leg 2 for the CW turn). Torque levels for the rear legs are occasionally higher, especially in the outside legs. This is not surprising since less correlation was noted between the rear ab/ad angles and the turn angle (Sect. 5.5). For the power curves in Fig. 5.19 (b), significant periods of negative ab/ad power occur in all legs as the controller attempts to regulate the ab/ad angles during stance. Overall, ab/ad power levels are much larger in magnitude during the turn than during steady-state galloping (Fig. 4.11).

The torque and power for the hip joints are shown in Fig. 5.20. Generally, both the torque and power for all hip joints is significantly higher during both stages of the turn as opposed to steady-state galloping (Fig. 4.12). In the front legs, larger positive (shoulder-braking) torques occur during the turn, which would, in combination with

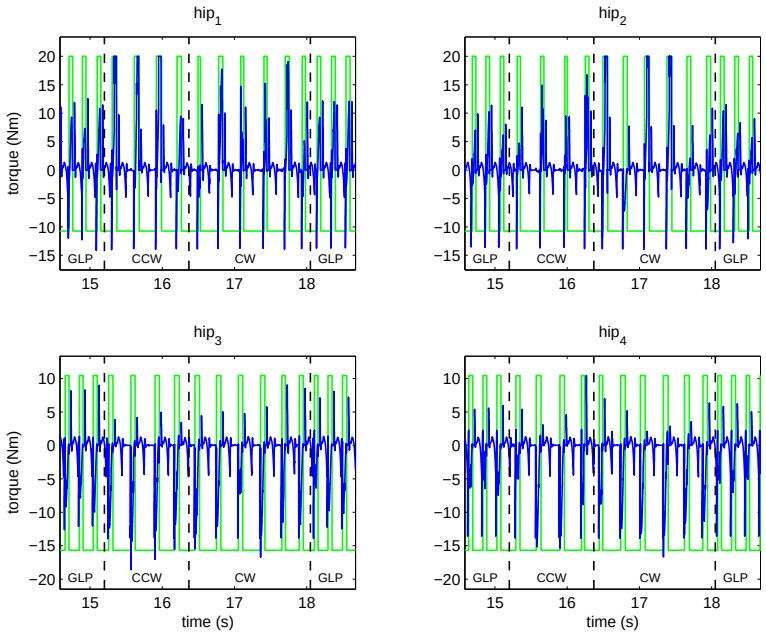


(a)

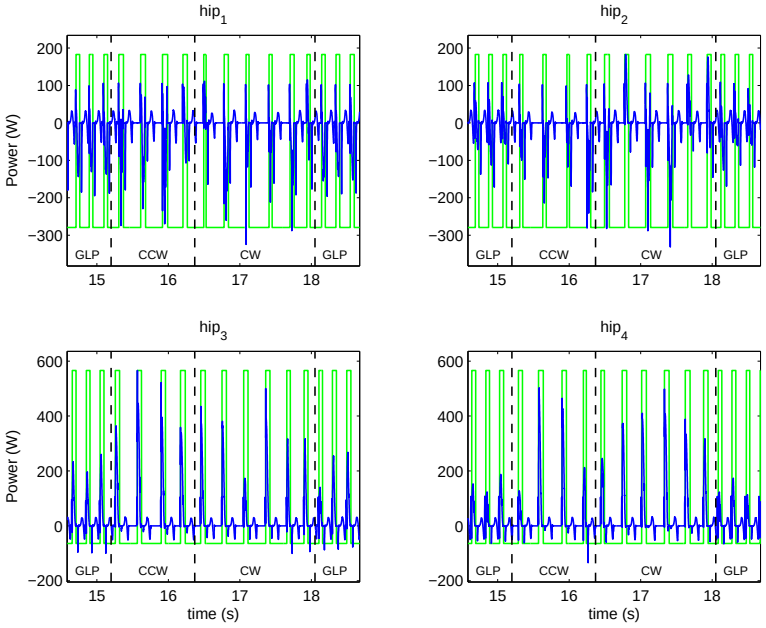


(b)

Figure 5.19: Torque and power requirements for the ab/ad axes during two successive multiple-stride turns. The square waveforms indicate leg contact, and the separate stages of the turn are labeled in each plot.



(a)



(b)

Figure 5.20: Torque and power requirements for the hip axes during two successive multiple-stride turns. Square waveforms indicate leg contact, and the separate stages of the turn are labeled in each plot.

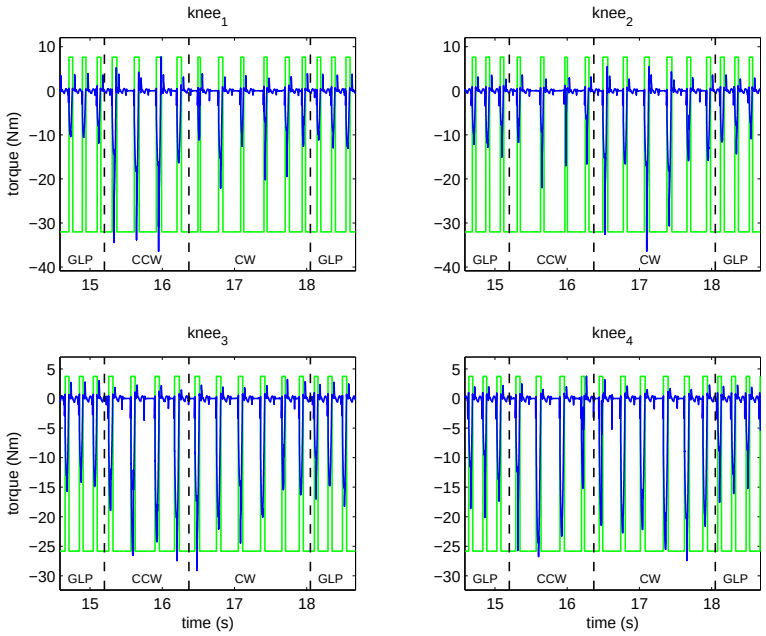
the ab/ad torques, produce a sharply inward-directed reaction force. To compensate, larger torques are required in the rear legs to maintain the tangential velocity of the body during the turn. Consequently, the front hip power during the turn appears as larger negative values than during steady-state galloping. In contrast, the rear hip power appears as larger positive values. The shoulder-braking and hip-thrusting behavior appears to be exaggerated during the turning maneuver.

The torque and power for the knee joints (series configuration, Sect. 4.4.3) are shown in Fig. 5.21. Knee torques (and power) are largest for the front, inside legs for each turning direction, although all legs exhibit higher torque values at some point during the turn. This suggests that much of the lateral force is actually being generated by the knee joints, which have been rotated to one side of the body's normal or the other by the ab/ad joints. Significant negative power occurs in the rear legs, indicating that the rear knee springs are forcibly being compressed, which would result from the driving rear hip torques discussed above.

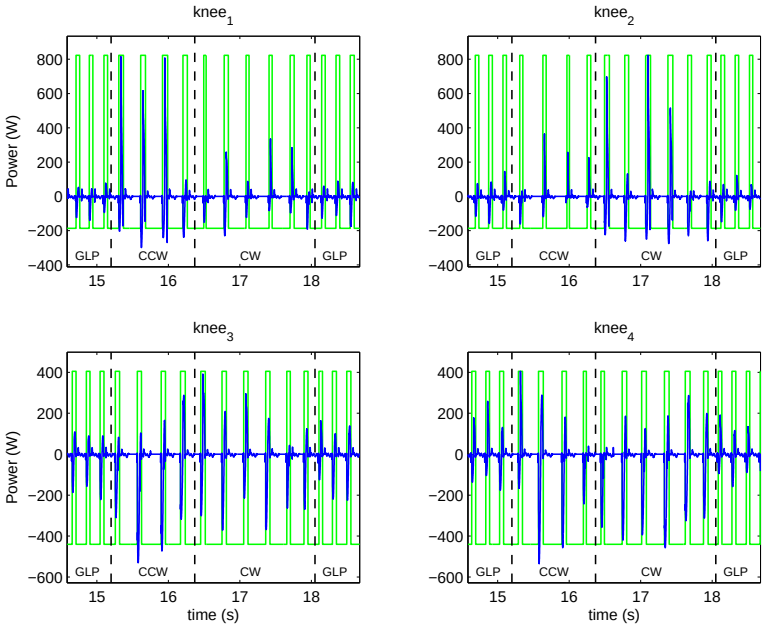
The total power summed across all joints for the turn is given in Fig. 5.22. Like the gallop, the assumption is that negative power offsets positive power (Sect. 4.4.3). As expected, the largest spikes in total power occur during the turn itself, as the legs must alter the forward momentum of the body. The largest positive power requirements occur in the front, inside legs for each turn and in both of the rear legs in both turns.

5.9 Summary and Conclusions

In this chapter, the control strategy and optimization problem for the high-speed turn were presented. The turn utilizes the same leg primitive functions in the same sequence as the gallop, although a high-level feedback controller was not developed



(a)



(b)

Figure 5.21: Torque and power requirements for the knee axes during two successive multiple-stride turns. Square waveforms indicate leg contact, and the separate stages of the turn are labeled in each plot. A series elastic configuration is assumed, as described in Sect. 4.4.3.

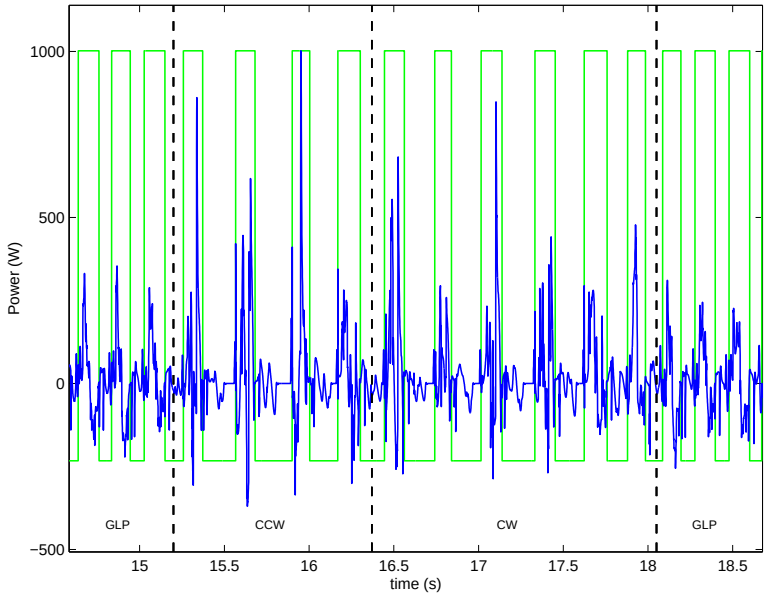


Figure 5.22: Total power required for all joints for two successive multiple-stride turns. The square waveform indicates that one or more legs are in contact, and the separate stages are labeled.

for the turn. Instead, open-loop control quantities were evolved on a single-stride basis. Multiple-stride turns were generated by evolving each turn stride independently, where each successive turn stride is initialized with the final TOF conditions from the preceding turn stride.

The results presented here appear to be new ones for an articulated-leg, quadrupedal model running at a relatively fast speed. Turn data collected for hexapedal insects shares some similarities, especially in two regards: (1) the front legs tend to be responsible for creating the required lateral impulse to turn the body, and (2) the magnitude of the turn dictates the amount of lateral impulse required, assuming a fixed linear momentum. Beyond these two characteristics, the high-speed quadrupedal turn appears to be different, at least for this model. The rotation of the legs

under the body causes the front, inside leg of the turn to experience approximately equal inward-directed lateral forces as the front, outside leg. In contrast, the outside legs tend to produce the largest inward-directed lateral forces for hexapedal turns. Furthermore, the front, inside leg experiences the largest magnitude in force levels during the quadrupedal turn.

For the banked turn, the conical pendulum model provides a reasonable fit for the front ab/ad angles and tilt angle of the front leg reaction force vector in the transverse plane. This suggests that the reaction forces in the front legs tend to act like the cord in the conical pendulum model, generating the necessary centripetal acceleration to produce the turn. The function of the rear legs appears to be more closely related to maintaining the forward velocity of the body tangential to the turn.

The banked turn appears to be the most stable mode of turning at high speeds for this system. One other turning mode was tested during preliminary trials, which employed the approach utilized by Raibert [21, 32]. In this method, the front leg ab/ad angles are deflected in one direction, while the rear ab/ad angles are deflected in the opposite direction. This approach was able to produce the largest turn angle in one stride, although the system immediately became unstable due to large *outward* roll rates.

The results of this analysis should facilitate the development of a high-speed turning controller, which is the next logical step in the study of the high-speed turn. Recognizing that (1) roll and yaw tend to be coupled together for high-speed turns, and (2) the front legs generally obey the properties of the conical pendulum model with respect to ab/ad angles and reaction forces, should influence the design of the controller. Thus, a controller might be developed that uses the conical pendulum

model to determine required ab/ad angles for the front legs given the running speed and desired turn radius. However, improvements to the stability of the gallop controller may be required before such a turning controller is feasible. The turning controller must not only maintain stability during the turn stride, but also successfully handle transitions to and from galloping. As such, enlarging the gallop's region of attraction by improving its robustness would be a necessary first step.

CHAPTER 6

STARTING, STOPPING, AND JUMPING

6.1 Introduction

In this chapter, the jump-start, running jump, and sudden stop are described. The last two behaviors briefly interrupt and terminate dynamic running, respectively. Consequently, each one is initialized during dynamically stable motion. The running jump represents a change in trajectory, although much of the motion is confined to the sagittal plane. On the other hand, the sudden stop represents a dramatic change in velocity, from the steady-state running velocity to a stationary pose, in a minimal amount of time. Finally, the jump-start begins in a statically stable configuration, although it terminates in a dynamically stable running gait.

In nature, all of these maneuvers are prevalent. The running jump is used to clear small obstacles, while the sudden stop may be employed when there is no way to circumnavigate a large obstacle while maintaining forward motion. Additionally, the sudden stop may actually be employed in extremely sharp turns. This type of "hairpin turn" actually involves stopping quickly, then leaping off into a new direction [85]. As such, the jump-start is also used whenever rapid acceleration is required, especially in predator-prey scenarios.

As stated in Sect. 5.1, quadrupedal dynamic maneuvers have received little attention in robotics research. While Wong and Orin studied static jumps [41], and running jumps in a bounding robot [40], both studies were restricted to the sagittal plane. As of the writing of this dissertation, there are no known examples of running jumps or sudden stops in a 3D galloping quadruped. Furthermore, Marhefka [89] studied jump-starts to initialize quadruped running gaits, although his study was planar and consisted of a single rigid body.

The first goal for this last part of the study on quadrupedal running and maneuvering is focused on finding biological-mode solutions to the running jump and sudden stop in order to understand their underlying mechanics. A high-level steady-state controller for the running jump is beyond the scope of this work, although a more robust method for stopping is illustrated. The second goal is to find a method of quickly accelerating into a steady gallop from a rest position, as might be required in an actual robot. Rather than use a sequence of different gaits to gradually accelerate into a gallop (e.g., walking, trotting, cantering, then galloping), the objective here is to achieve maximum acceleration using a jump-start followed by an accelerating bound.

In the following sections, the jump-start, the sudden stop, then the running jump will be presented. In each case, the strategy for achieving the biological-mode solution will be explained, followed by an analysis of the underlying mechanics. In addition, the torque and power requirements for each joint will be presented. The final evolved sequence, from the jump-start to the sudden stop, will be described next, which provides an impressive demonstration of the power of the evolutionary approach

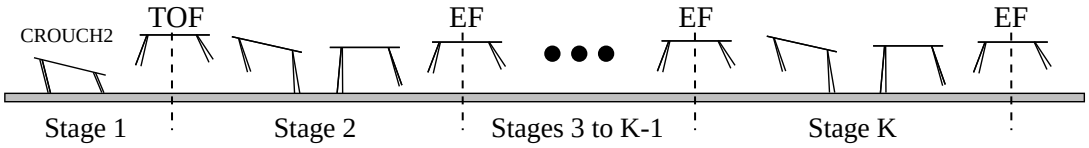


Figure 6.1: Diagram of the jump-start maneuver. Stage 1 consists of the crouch and the leap; Stage 2 consists of the landing; Stages 3 to $K-1$ consist of accelerating bounding strides, and Stage K is the transition stride into galloping. "EF" refers to the top-of-flight point in the extended flight phase, which occurs after the rear legs, but not the fore, have touched down. The total number of stages K depends on how much acceleration can be achieved per stage.

combined with the control architecture described earlier. The resulting sequence represents what appears to be the first example of 3D galloping and maneuvering in a quadrupedal model. Finally, a summary is provided at the end.

6.2 The Jump-Start

The jump-start is one of the more complex maneuvers which consists of several stages, as shown in Fig. 6.1. Stage 1 consists of the crouch and the leap, Stage 2 consists of the landing, Stages 3 to $K-1$ consist of accelerating bounding strides, and Stage K is the transition stride into stable galloping. The number of stages K is not known in advance but depends on how much acceleration is achieved during each bounding stage (which is equivalent to one stride). As discussed in Sect. 6.2.4, $K = 6$ was required to achieve the desired running speed of $4.15\,\mathrm{m/s}$. Note that the top-of-flight points are labeled "EF" to signify that the flight phase is considered extended flight, because the rear legs have touched more recently than the front legs during the current stride. A state diagram for the entire maneuver is given in Fig. 6.2.

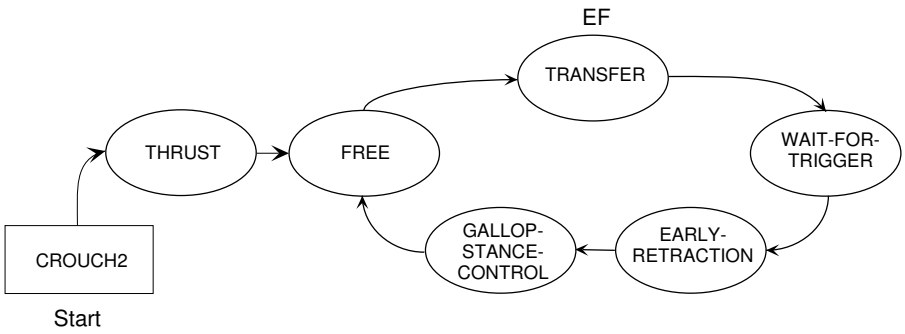


Figure 6.2: State diagram for the jump-start maneuver. Each leg begins in the coordinated function CROUCH2, then transitions to THRUST. When the leg breaks contact, it enters the loop in the FREE function, cycling around the loop until the maneuver transitions into galloping. Each cycle is terminated at the EF (top-of-extended-flight) point and represents a single stage of bounding (Stages 3 to $K-1$).

After Stage 1, each stage consists of one cycle of the repeated portion of the state diagram (from EF to EF), which is identical to that of the gallop in Fig. 4.1. Each of the stages is discussed in the following sections.

6.2.1 Stage 1: The Leap

Stage 1 consists of the sequence CROUCH2-THRUST-FREE-TRANSFER, where all legs remain in CROUCH2 for the same amount of time, since it is a coordinated function (Sect. 2.4.3). All four parameters of CROUCH2, front and rear virtual leg lengths l_f and l_r, and front and rear virtual leg angles ${}^e\theta_{v_f}$ and ${}^e\theta_{v_r}$, are evolved. When the leg enters the THRUST function (Sect. 2.3.7), the torque values and timing parameters for the hip and knee are evolved to generate the required torque profile to produce the leap. When the leg breaks contact, it enters the loop in Fig. 6.2 in the FREE function, then transitions to TRANSFER. At some point during TRANSFER, the quadruped reaches a top-of-flight point, which, as stated above, is labeled EF to

signify the top-of-extended-flight since the rear legs have made contact more recently. At EF, Stage 1 is terminated; subsequent stages are also initialized and terminated at the EF points. A summary of the parameters for Stage 1, as well as the remaining stages, is given in Table 6.1. Note that leg parameters are paired so that both front legs have the same parameters, as do both rear legs. Thus, the entire first stage of the JUMP-START is basically a planar maneuver since the legs on both sides of the mid-line behave the same[30]. Furthermore, different parameters are required depending on the stage, which is denoted in the table as a subscript S. The parameter ranges for each of the evolved parameters for Stage 1, as well as the remaining stages, are given in Table 6.2.

The choice of THRUST parameters to evolve was based on preliminary experiments conducted on stationary leaps. During these experiments, it was observed that the hip torque profile generally required two torque values, the first of which should start at normalized time $t = 0.0$. The start time of the second torque value, t_{h_2}, required evolution. For the knees, only the first torque value τ_{k_1} and start time t_{k_1} required evolution. As stated in Sect. 2.3.7, the initial torque commanded at the knee joint just prior to the THRUST function's initialization is maintained from $t = 0.0$ to $t = t_{k_1}$. This permits the knee springs to be held in place, then released. Finally, the overall period of $T = 0.5\,\mathrm{s}$ was chosen as this provided adequate time for the THRUST function to complete a significant power stroke.

After the leg breaks contact, it transitions from the THRUST function to the FREE function. Nominal parameters from galloping were chosen for the FREE function, although the period was set to a minimal time of 0.5 ms for the first stage to

[30]Negligible ab/ad torques are required to stabilize the system due to numerical precision.

Function	Parameters	Description
CROUCH2	$l_f = l_f^*$, $l_r = l_r^*$, ${}^e\theta_{v_f} = {}^e\theta_{v_f}^*$, ${}^e\theta_{v_r} = {}^e\theta_{v_r}^*$, $T = 0.025\,\text{s}$	Execute crouch using evolved front and rear leg lengths l_f^* and l_r^*, and front and rear virtual leg angles ${}^e\theta_{v_f}^*$ and ${}^e\theta_{v_r}^*$.
THRUST	$\tau_{h_1} = \begin{cases}\tau_{h_{1,f}}^* \text{ if } i \leq 2\\ \tau_{h_{1,r}}^* \text{ otherwise,}\end{cases}$ $t_{h_1} = 0.0$, $\tau_{k_1} = \begin{cases}\tau_{k_{1,f}}^* \text{ if } i \leq 2\\ \tau_{k_{1,r}}^* \text{ otherwise,}\end{cases}$ $t_{k_1} = \begin{cases}t_{k_{1,f}}^* \text{ if } i \leq 2\\ t_{k_{1,r}}^* \text{ otherwise,}\end{cases}$ $T = 0.5\,\text{s}$ | $\tau_{h_2} = \begin{cases}\tau_{h_{2,f}}^* \text{ if } i \leq 2\\ \tau_{h_{2,r}}^* \text{ otherwise,}\end{cases}$ $t_{h_2} = \begin{cases}t_{h_{2,f}}^* \text{ if } i \leq 2\\ t_{h_{2,r}}^* \text{ otherwise,}\end{cases}$ $\tau_{k_2} = 0.0$, $t_{k_2} = 1.0$,	Execute leap forward by implementing torque profiles for the hip and knee joints with evolved values $\tau_{h_{1,f}}^*$, $\tau_{h_{1,r}}^*$, $\tau_{h_{2,f}}^*$, $\tau_{h_{2,r}}^*$, $t_{h_{2,f}}^*$, $t_{h_{2,r}}^*$, $\tau_{k_{1,f}}^*$, $\tau_{k_{1,r}}^*$, $t_{k_{1,f}}^*$, and $t_{k_{1,r}}^*$.
TRANSFER	$\sigma_{a_e} = 0$, $\sigma_{h_e} = 0$, $\sigma_s = 0$, $\sigma_{k_d} = 1$, $T_r = 0.0$, $\boldsymbol{\theta}_{l_f} = [\frac{\pi}{2}, \theta_{v_i}, \frac{l_0}{2}]^T$, $T_t = 1.0$, $T = \begin{cases}0.25\,\text{s} & \text{if } S = 1,\\ 0.125\,\text{s} & \text{if } S = 2,\\ 0.1\,\text{s} & \text{otherwise.}\end{cases}$	Transfer using θ_{v_i} as follows: $\theta_{v_i} = \begin{cases}0.69\,\text{rad} & \text{if } i \leq 2,\\ 0.65\,\text{rad} & \text{if } i > 2,\end{cases}$ where $i = 1, \ldots, 4$, l_0 is the nominal leg length (Table 2.1), and S is the stage.
WAIT-FOR-TRIGGER	$L = i$, $R = 4$, $\delta = 0.0$, $h_t = l_0$, $\sigma_{a_e} = 0$, $\sigma_{h_e} = 0$, $\sigma_{k_f} = 0$, $\sigma_s = 0$, $T_r = 0.0$, $\boldsymbol{\theta}_{l_d} = [\frac{\pi}{2}, \theta_{v_i}, l_0]^T$, $T = 1.0\,\text{s}$	Hold at ab/ad angle θ_{v_i} given in TRANSFER (above). Use $R = 4$ (HIP-HEIGHT, Table 2.6) with $h_t = l_0$ (Table2.1) for leg i, $i = 1, \ldots, 4$.
EARLY-RETRACTION	$\sigma_{a_e} = 0$, $\theta_a = \frac{\pi}{2}\,\text{rad}$, $v_d = 4.15\ \frac{m}{s}$, $T = 1.0\,\text{s}$	Hold ab/ad joint at nominal value of $\frac{\pi}{2}$ rad and rotate hip to achieve tangential foot velocity v_d.
GALLOP-STANCE-CONTROL	$v_d = 4.15\ \frac{m}{s}$, $v_b = \begin{cases}v_{b_{f,S}}^* & \text{if } i \leq 2,\\ v_{b_{r,S}}^* & \text{otherwise.}\end{cases}$ $\sigma_{\text{slip}} = 1$, $E_d = \begin{cases}E_{f,S}^* & \text{if } i \leq 2,\\ E_{r,S}^* & \text{otherwise.}\end{cases}$ $T = 1.0\,\text{s}$	Use velocity control with evolved velocity biases $v_{b_{f,S}}^*$ and $v_{b_{r,S}}^*$ and evolved target energy value $E_{f,S}^*$ and $E_{r,S}^*$, where S is the stage. Use slip control.
FREE	$\sigma_{a_f} = 0$, $\sigma_s = 1$, $T_r = 0.75$, $T = \begin{cases}0.5\text{ ms} & \text{if } S = 1,\\ 25.0\text{ ms} & \text{otherwise.}\end{cases}$	Reset knee spring over $T_r = 75\,\%$ of the period T, where S is the stage. Hold the ab/ad angle while allowing the hip to rotate freely.

Table 6.1: Summary of the parameters for the jump-start (Fig. 6.2).

Parameter	Stage(S)	Range
l_f^*, l_r^*	1	[0.06, 0.24] m
${}^e\theta_{v_f}^*, {}^e\theta_{v_r}^*$	1	[1.05, 2.09] rad
$\tau_{h_{1,f}}^*, \tau_{h_{1,r}}^*, \tau_{h_{2,f}}^*, \tau_{h_{2,r}}^*$	1	[−10.0, 10.0] Nm
$t_{h_{2,f}}^*, t_{h_{2,r}}^*, t_{k_{1,f}}^*, t_{k_{1,r}}^*$	1	[0.0, 1.0]
$\tau_{k_{1,f}}^*, \tau_{k_{1,r}}^*$	1	[−20.0, 0.0] Nm
$v_{b_{f,S}}^*, v_{b_{r,S}}^*$	$2, \ldots, K$	[−10.0, 10.0] m/s
$E_{f,S}^*, E_{r,S}^*$	$2, \ldots, K$	[0.0, 10.0] J

Table 6.2: Ranges for each evolved parameter for the jump-start.

prevent the leg from swinging freely for any significant amount of time. This was done because of the potentially large torque values acting on the joints during stance, none of which are bounded during the THRUST function to prevent slipping. Consequently, the leg could break contact at a large velocity, which could destabilize the system. An almost immediate transition to the TRANSFER function is necessary to maintain control over the leg. However, eliminating the FREE function from Stage 1 was not practical due to encoding issues[31].

After the FREE function, the leg transitions to TRANSFER, where again, the nominal set of parameters from galloping were used, with two exceptions. First, the ab/ad angles are fixed at $\theta_{a_i} = \frac{\pi}{2}$ for $i = 1, \ldots, 4$. Second, the period is set to a larger value of 0.25 s for the initial leap, since this leap is expected to be significantly larger in height than the typical flight phases that occur during galloping. During

[31]Encoding a repeating cycle of functions is generally a much more efficient method of representation than encoding each function in sequential order. Consequently, there is considerable motivation for representing each behavior with a minimal set of functions, especially if that set can be repeated in a cyclic fashion.

subsequent stages, the period for the TRANSFER function is reduced to the typical value of 0.1 s used for galloping (Sect. 4.2.1).

The fitness vector for Stage 1 of the jump-start $\boldsymbol{f}_{\mathrm{J}_1}$ consists of three elements:

$$\boldsymbol{f}_{\mathrm{J}_1} = \left[f_h^{\mathrm{J}_1},\, f_c^{\mathrm{J}_1},\, f_p^{\mathrm{J}_1} \right]^T , \tag{6.1}$$

where $f_h^{\mathrm{J}_1}$ rewards accuracy with respect to TOF height, $f_c^{\mathrm{J}_1}$ is a correctness and accuracy factor, and $f_p^{\mathrm{J}_1}$ rewards reasonable pitch rates. The height factor $f_h^{\mathrm{J}_1}$ is computed as follows:

$$f_h^{\mathrm{J}_1} = \frac{w_h}{1 + \left(\lambda_h \left({}^e z_b - h_d\right)\right)^2} , \tag{6.2}$$

where $w_h = 100.0$, $\lambda_h = 16.67$, which produces a threshold value of 0.06 m, ${}^e z_b$ is the TOF height in earth coordinates, and h_d is the desired height. h_d is set to $1.2\, l_0$ (Table 2.1) to produce a jump of sufficient height for leg clearance, but not excessively high such that the jumping time is unnecessarily inflated.

The correctness and accuracy factor $f_c^{\mathrm{J}_1}$ is computed as follows:

$$f_c^{\mathrm{J}_1} = \frac{w_c f_x f_{\mathrm{lc}}}{1 + \left(\lambda_\beta \left(\beta - \beta_d\right)\right)^2} , \tag{6.3}$$

where $w_c = 100.0$, f_x is a jumping distance factor, f_{lc} is a leg contact factor, $\lambda_\beta = 11.5$, yielding a threshold of approximately 5.0 deg, β is the TOF pitch, and β_d is the desired pitch angle, which is set to 0.0 rad for level pitch orientation at TOF. Roll, yaw, roll rate, and yaw rate are ignored because the jump-start is essentially a planar maneuver[32].

[32]Rarely, an individual will incur error conditions that cause instability, yielding a large roll rate and/or yaw rate. To discourage such solutions, individuals with $\dot{\alpha} > \epsilon$ and/or $\dot{\gamma} > \epsilon$, where $\epsilon = 0.001$ rad/s, are discarded from the working population $W(k)$.

The jumping distance factor f_x rewards the horizontal distance traveled and is computed as follows:

$$f_x = w_x \left(\text{sign}\left(\Delta x\right)\right) \left(\Delta x\right)^2 , \tag{6.4}$$

where w_x is set to 5.0 so that a horizontal leap of 0.45 m (a good benchmark) produces a contribution of about 1.0, Δx is the change in x-position in earth coordinates, and sign$(\cdot)$ is the signum function. This term maximizes forward distance traveled as opposed to forward velocity. Including a velocity term yields solutions with large *instantaneous* velocities at TOF, although these solutions involve large joint torques during the THRUST function, which cause the feet to slip significantly. (Recall that the THRUST function does not employ the no-slip algorithm used in GALLOP-STANCE-CONTROL.) Consequently, undesired leg motion can occur following the stance phase. On the other hand, rewarding distance traveled yields larger *average* velocities, but discourages slipping-mode solutions.

The leg contact factor f_{lc} rewards solutions where only one contact period occurs per leg and no feet remain in contact when the jump is terminated. Occasionally, individuals may error out (via excessive body orientation or excessive joint angles) before reaching TOF, at which time their fitness is computed in the normal way. However, one or more legs will, more than likely, still be in contact. To penalize these solutions, the leg contact factor is computed as follows:

$$f_{\text{lc}} = \frac{1}{1 + \sum_{i=1}^{4} \left(w_{\sigma_c} \sigma_{c_i} + \left(w_{n_c} \left(c_i - 1 \right) \right)^2 \right)} , \tag{6.5}$$

where w_{σ_c} is a penalty factor for the final contact condition, σ_{c_i} is the final contact condition for leg i, where $\sigma_{c_i} = 1$ if the leg is still in contact, w_{n_c} is a penalty factor for the incorrect number of contact periods, and c_i is the number of contact periods

for leg i. The penalty factors were set to $w_{\sigma_c} = 10.0$ and $w_{n_c} = 5.0$ to balance the contributions of each term. The fitness factor f_{lc} will achieve its maximum value of 1.0 when all legs have made exactly one contact during the leap, and no legs are still in contact when the leap is terminated.

The final fitness criterion is a factor for pitch rate, the desired value of which is not known precisely before the evolution. However, positive (nose-down) pitch rates are desired so that the quadruped lands on its front legs after the jump, like its biological analog. The pitch rate criterion is computed as follows:

$$f_p^{\text{J1}} = \frac{w_p}{1 + \left(\lambda_{\dot{\beta}} \max\left(|\dot{\beta} - \dot{\beta}_c| - t_{\dot{\beta}},\, 0.0\right)\right)^2}\,, \tag{6.6}$$

where w_p is the fitness scaling factor of 100.0, $\lambda_{\dot{\beta}}$ is a scaling factor for pitch rate errors, $\dot{\beta}$ is the actual TOF pitch rate, $\dot{\beta}_c$ rad/s is the center of the acceptable pitch rate range, and $t_{\dot{\beta}}$ is a pitch rate tolerance that determines the half-width of the pitch rate range. The center $\dot{\beta}_c$ and half-width $t_{\dot{\beta}}$ were tuned experimentally to 2.0 rad/s and 1.0 rad/s, respectively, while the threshold value was set to 0.5 rad/s, which yields $\lambda_{\dot{\beta}} = 2.0$. The max$(\cdot)$ function penalizes the pitch rate only if it is outside the acceptable range of $[\dot{\beta}_c - t_{\dot{\beta}},\, \dot{\beta}_c + t_{\dot{\beta}}]$. This approach was taken to allow for a variety of reasonable pitch rates instead of forcing the solution toward a particular target value, as this value was not known *a priori*.

6.2.2 Stages 2 to K-1

The goal for Stages 2 through $K - 1$ is to accelerate the quadruped so that its forward running speed is close to the desired running speed of 4.15 m/s. This is required because the gallop has a limited range of attraction for velocity (Table 4.8). Leg parameters are paired to produce a planar bound, which simplifies the problem

by eliminating the need for significant roll and yaw control, which are both required for a spatial gait like the gallop. Simple PD control is sufficient to maintain the nominal 90 deg angle for the ab/ad joints during each stage.

Stage 2 and the subsequent stages through $K-1$ are initialized at the top-of-flight point where the previous stage terminated. During Stage 2, the quadruped executes a landing from the initial leap, where the front feet come down first (due to the positive, nose-down pitch rate), followed by the rear. Stage 2 terminates at the next top-of-flight point, although this point is considered to be the top-of-extended-flight (EF) because the rear legs have made contact most recently[33]. In general, however, each stage utilizes the same leg primitive functions in the same sequence as the gallop, as illustrated in Fig. 6.2. The remaining stages 3 through $K-1$ also end in extended flight, with a nose-down pitch rate. This means that the transition stage K will require a reversal of pitch rate since galloping involves a nose-up pitch rate during flight.

As indicated in Table 6.1, there are four parameters that must be found during each stage S: $v^*_{b_{f,S}}$ and $E^*_{f,S}$, the velocity bias and target energy value for the front legs, and $v^*_{b_{r,S}}$ and $E^*_{r,S}$, the corresponding parameters for the rear legs. Each stage is evolved independently to minimize the parameter space.

The fitness vector for Stages 2 through $K-1$ consists of two elements, as follows:

$$\boldsymbol{f}_{\mathrm{J}_2} = [f_a^{\mathrm{J}_2}, f_v^{\mathrm{J}_2}]^T \ , \tag{6.7}$$

[33]During a typical gallop stride, the opposite is true: At the top-of-flight point, the front legs will have made contact most recently, yielding a gathered flight phase.

where $f_a^{\text{J}_2}$ is an accuracy factor, and $f_v^{\text{J}_2}$ is a velocity factor. The accuracy factor is computed as follows:

$$f_a^{\text{J}_2} = f_{\text{lc}} f_t \left(\frac{w_{a_2}}{1 + \| \boldsymbol{\Lambda} \left(\boldsymbol{\theta}_b - \boldsymbol{\theta}_{b_d} \right) \|^2} \right) \tag{6.8}$$

where w_{a_2} is a fitness scaling factor of 100.0, the diagonal elements of $\boldsymbol{\Lambda}$ are given in Table 4.5, and $\boldsymbol{\theta}_{b_d}$ is given as

$$\begin{aligned} \boldsymbol{\theta}_{b_d} &= \left[\; \alpha_d, \, \beta_d, \, \gamma_d, \, {}^e x_{b_d}, \, {}^e y_{b_d}, \, {}^e z_{b_d}, \, \dot{\alpha}_d, \;\; \dot{\beta}_d, \;\; \dot{\gamma}_d, \, {}^{e'}\dot{x}_{b_d}, \, {}^{e'}\dot{y}_{b_d}, \, {}^e\dot{z}_{b_d} \; \right]^T \\ &= \left[\; 0, \;\; 0, \;\; 0, \;\; 0, \;\;\; 0, \;\;\; l_0, \;\; 0, \; -\dot{\beta}_d^*, \; 0, \;\; v_d, \;\;\; 0, \;\;\; 0 \; \right]^T , \end{aligned} \tag{6.9}$$

where $\dot{\beta}_d^*$ is the target pitch rate for the gallop (-0.47 rad/s, Table 4.6), l_0 is the nominal leg length (Table 2.1), and $v_d = 4.15\,\text{m/s}$. The target pitch rate is reversed for this maneuver because the flight phases are extended flight, with nose-down pitch motion.

The term f_{lc} in Eq. (6.8) is the leg contact factor computed in Eq. (6.5), and f_t is the fitness with respect to elapsed time, computed as follows:

$$f_t = \frac{w_t}{1 + \left(\lambda_t \left(\Delta t \right) \right)^2} , \tag{6.10}$$

where w_t is a weighting factor of 2.25, λ_t is 3.16, and Δt is the time elapsed during the stage, from the initial EF point to the current EF. The weight factors yield $f_t \approx 1.0$ for $\Delta t = 0.35$, where the latter was found to be a good benchmark duration for an accelerating bound stride. Including a time-based term encourages minimal stride times, which results in maximal average acceleration.

The second fitness criterion $f_v^{\text{J}_2}$ isolates the EF forward velocity, which provides an indication of how close the quadruped is to achieving the target running velocity:

$$f_v^{\text{J}_2} = \frac{w_v}{1 + \left(\lambda_v \left({}^{e'}\dot{x}_b - v_d \right) \right)^2} , \tag{6.11}$$

where $w_v = 100.0$, $\lambda_v = 1.0$, ${}^{e'}\dot{x}_b$ is the forward velocity of the body, and v_d is the desired running speed of 4.15 m/s. Inclusion of the velocity term in both fitness criteria f_a^{J2} and f_v^{J2} accomplishes two things. First, it isolates the velocity in f_v^{J2}, creating specific pressure on this variable. Second, it rewards solutions along both dimensions, making velocity a type of "global requirement." Caution must be used when taking this approach, however, as using fitness criteria that have too much coupling among them can degrade the Pareto front, often resulting in a single scalar solution. In this case, however, both criteria exhibited sufficient independence to avoid this problem.

6.2.3 Stage K

The last stage of the jump-start maneuver, Stage K, consists of a transition to galloping. The value of K, which is the total number of stages for the jump-start maneuver, can only be determined once the quadruped reaches sufficient forward velocity to successfully transition to a gallop. As stated above, the transition stage must also correct the nose-down, extended flight pitch rate to match the nose-up pitch rate in the gallop. Consequently, the forward velocity should be close to the target value to avoid simultaneous changes to both velocity and pitch rate, as such a compound adjustment appears to be difficult to achieve based on the results of preliminary testing.

The leg primitive functions, sequence, and evolved parameters for Stage K are identical to those of the previous stages. However, the fitness function employed for this stage is given as follows:

$$\boldsymbol{f}_{\text{J}K} = [f_a,\, f_s,\, f_c]^T \ , \tag{6.12}$$

where f_a, f_s, and f_c are computed in Eq.'s (4.32 - 4.35). Thus, the fitness criteria for this function are exactly equal to those for galloping. This approach evaluates fitness indirectly by examining the fitness of the subsequent gallop since a stable galloping gait is the ultimate goal for the jump-start maneuver.

6.2.4 Jump-Start Results

A total of 10 trials were run with $S = 32$ individuals and $N = 100$ generations for each stage of the jump-start maneuver. Multiple acceptable solutions were found, although only one particular solution is discussed here. The evolved parameters for this solution are given in Table 6.3. The sagittal plane body state variables are shown in Fig. 6.3. The height, shown in Fig. 6.3 (b), shows that the initial leap is slightly higher than the desired value of 0.29 m (Sect. 6.2.1), although subsequent EF heights are close to the nominal running height of 0.24 m. The final top-of-flight pitch angle in plot (a) is fairly small, at an average of -0.10 rad (-5.7 deg) (nose-up) for stages 2 through 6. However, the pitch rate (plot (c)) is significantly positive at the end of stages 1 through 5, which indicates a nose-down pitch motion, characteristic of extended flight. Finally, the forward velocity with respect to the body is shown in plot (d). During the leap in stage 1, the quadruped can accelerate from 0.0 to 2.0 m/s in about 0.36 s (5.56 m/s^2), although some deceleration occurs prior to reaching the first TOF at $t = 0.52$ s. It was observed through experimentation that the limit on desired TOF jumping height is chiefly responsible for this ceiling. When the desired jumping height is increased, larger velocities are generated. At lower heights, the TOF velocity is limited even when the torque ranges are increased. It appears that

Parameter	**Stage** (S)	**Value**	**Parameter**	**Stage** (S)	**Value**
l_f^*	1	0.16 m	$E_{r,S}^*$	2	2.29 J
l_r^*	1	0.15 m	$v_{b_{f,S}}^*$	3	9.01 m/s
${}^e\theta_{v_f}^*$	1	1.98 rad	$v_{b_{r,S}}^*$	3	0.40 m/s
${}^e\theta_{v_r}^*$	1	1.77 rad	$E_{f,S}^*$	3	1.31 J
$\tau_{h_{1,f}}^*$	1	-1.47 Nm	$E_{r,S}^*$	3	1.28 J
$\tau_{h_{1,r}}^*$	1	-1.84 Nm	$v_{b_{f,S}}^*$	4	6.44 m/s
$\tau_{h_{2,f}}^*$	1	6.64 Nm	$v_{b_{r,S}}^*$	4	6.44 m/s
$\tau_{h_{2,r}}^*$	1	3.57 Nm	$E_{f,S}^*$	4	1.29 J
$t_{h_{2,f}}^*$	1	0.91	$E_{r,S}^*$	4	1.85 J
$t_{h_{2,r}}^*$	1	0.64	$v_{b_{f,S}}^*$	5	0.50 m/s
$t_{k_{1,f}}^*$	1	0.50	$v_{b_{r,S}}^*$	5	8.10 m/s
$t_{k_{1,r}}^*$	1	0.52	$E_{f,S}^*$	5	2.90 J
$\tau_{k_{1,f}}^*$	1	-17.84 Nm	$E_{r,S}^*$	5	3.42 J
$\tau_{k_{1,r}}^*$	1	-16.57 Nm	$v_{b_{f,S}}^*$	6	-6.55 m/s
$v_{b_{f,S}}^*$	2	9.99 m/s	$v_{b_{r,S}}^*$	6	0.59 m/s
$v_{b_{r,S}}^*$	2	6.81 m/s	$E_{f,S}^*$	6	3.82 J
$E_{f,S}^*$	2	0.47 J	$E_{r,S}^*$	6	0.64 J

Table 6.3: Evolved parameters for the jump-start.

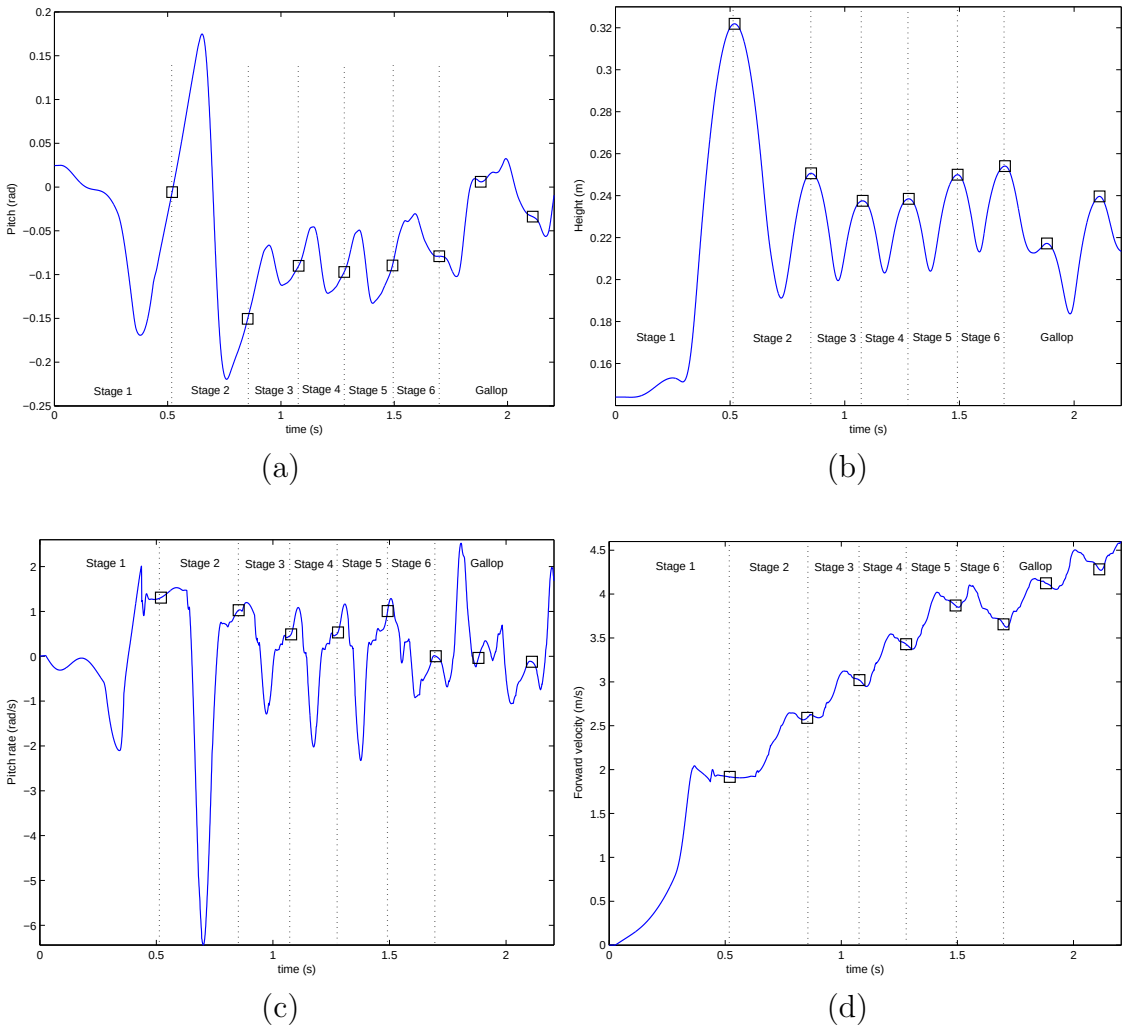


Figure 6.3: Sagittal plane body state variables for the jump-start: (a) pitch, (b) height, (c) pitch rate, and (d) forward velocity ${}^{e'}\dot{x}_b$. Squares indicate EF (top-of-extended-flight) during Stages 1 - 6, and TOF (gathered) thereafter.

finite friction and the limit on desired jumping height are jointly responsible for the ceiling on achievable TOF velocity during a leap.

The average acceleration during each subsequent stage in Fig. 6.3 (d) also appears limited at $2.0\,\mathrm{m/s^2}$ for stages 3 to 5. Because of the ceiling on forward acceleration, multiple strides are required to produce sufficient forward velocity to transition to a gallop. Preliminary experiments were conducted to determine an approximate range of speeds for a successful transition, where the minimum value appeared to be about $3.5\,\mathrm{m/s}$. This value is significantly outside the range of attraction for forward velocity for the gallop, which is listed in Table 4.8 as $[4.05, 4.30]\,\mathrm{m/s}$. However, each range in the table was determined by varying each body state variable individually. The experiments run for the jump-start, as well as the turn and running jump, indicate that a much larger region of attraction for the gallop exists when multiple body state variables are varied simultaneously. Consequently, a lower transition speed can be achieved.

The foot forces for the jump-start maneuver are shown in Fig. 6.4. The vertical forces show a large spike in the rear legs (legs 3 and 4) when they touch down at $t = 0.69\,\mathrm{s}$ in Stage 2 after the initial leap. This is not an intuitive result, since the quadruped lands on its front legs, where one would expect the larger vertical forces to occur. However, when the front legs make contact at $t = 0.64\,\mathrm{s}$, a large moment is induced about the body's pitch axis of rotation, causing a large negative pitch rate, indicated by the local minimum of $-6.4\,\mathrm{rad/s}$ at $t = 0.70\,\mathrm{s}$ in Fig. 6.3 (c). In fact, this negative pitch rate is more than 6.0 times larger in magnitude than the pitch rate of $1.04\,\mathrm{rad/s}$ when the front legs touch down at $t = 0.64\,\mathrm{s}$. Consequently, this causes the hind quarters to rotate downward at a much larger velocity, generating a

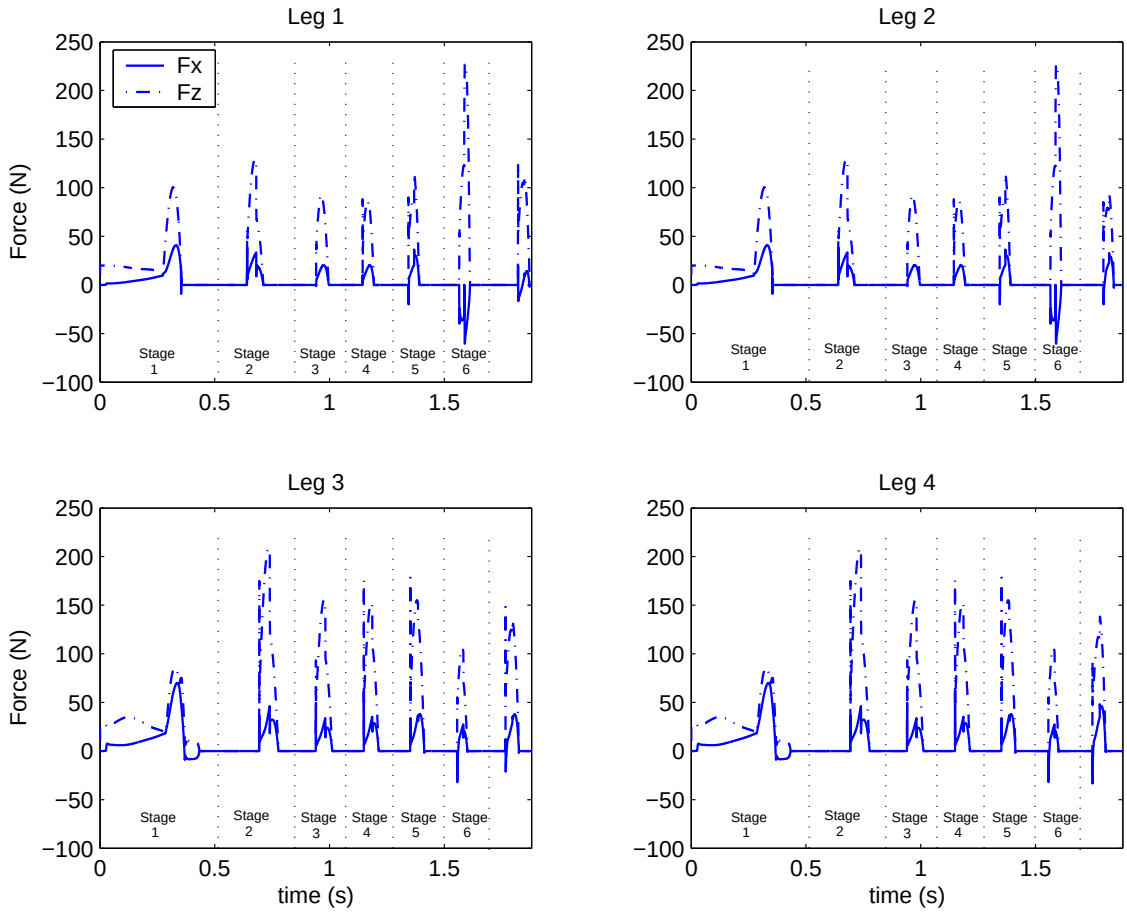


Figure 6.4: Foot forces for the jump-start maneuver in the forward and vertical directions. The no-slip algorithm is employed during GALLOP-STANCE-CONTROL to prevent slipping during Stages 2 through 6. Both the vertical and forward forces change during Stage 6 in the front legs to transition to galloping.

larger reaction force. The opposite appears to be true during the landing stage in horses after jumping over a hurdle [97], however. The difference may be related to system compliance and active control during landing. Quadrupedal mammals have significant compliance and many more DOF than the simple model used here. Thus, impacts may be absorbed more effectively and actively directed through the center of mass instead of being transformed into large moments. For example, when jumping down off a platform, cats generally absorb much of the vertical impact in their front legs and throughout their skeletal structure, as the line of force appears to intersect the center of mass [98]. This reduces the kinetic energy of the system and, since the force acts mostly through the center of mass, does not cause significant rotation on landing, reducing the impact in the hind legs.

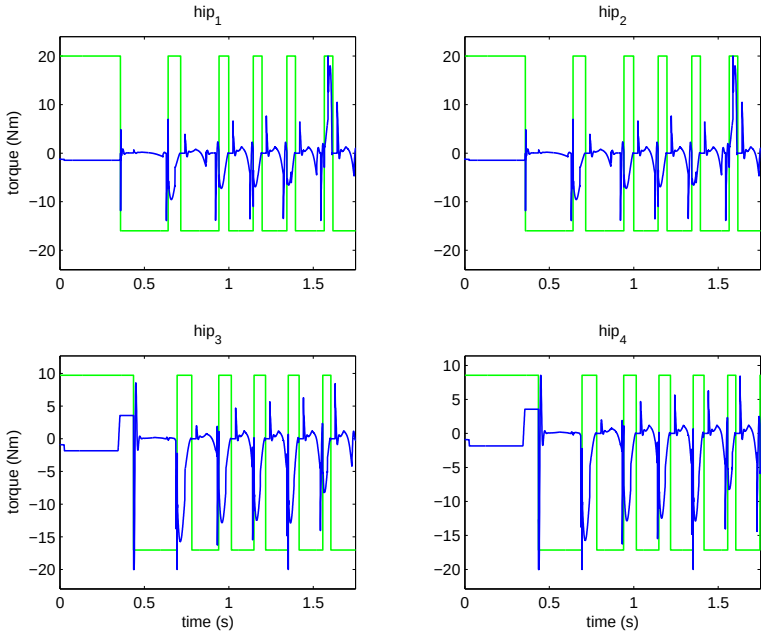
During subsequent stages, the vertical forces in the simulated model's rear legs remain smaller. In the front legs, however, there is a spike in vertical force during Stage 6 as the quadruped transitions to a gallop. This spike coincides with a reversal of forward reaction forces in the front legs. Up until this point, the forward reaction forces have been positive in all legs to accelerate the body in the forward direction. However, in Stage 6, the shoulders begin to brake, and the knees deliver a much larger injection of energy than in previous strides ($E^*_{f,6} = 3.82\,\mathrm{J}$ in Table 6.3). The combined effect of shoulder-braking and knee-thrusting in the front legs causes the pitch rate at top-of-flight to reverse to the negative, nose-up direction required for galloping. The result is that the rear legs come down first during the subsequent stride, the first stride of galloping, to yield the proper footfall phasing.

6.2.5 Torque and Power Analysis for the Jump-Start

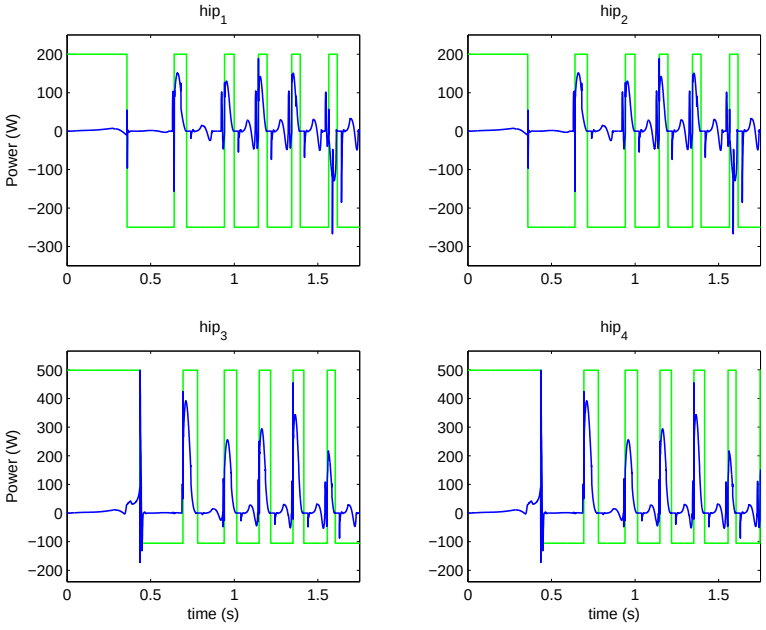
In this section, the torque and power for the hip and knee joints, as well as the total power, are presented. Ab/ad torque and power are excluded since the jump-start is largely a planar maneuver and negligible ab/ad actuation is required. The torque and power for the hip joints are shown in Fig. 6.5. During the first stage, relatively little power (Fig. 6.5 (b)) is required of the hips to produce the leap[34]. The reason for this is that most of the forward impulse is generated by the knees. Thus, the hips must rotate the legs to the proper angle to correctly orient the reaction force produced by the knees. During the last part of the rear leg contact phases, the rear hip torque in Fig. 6.5 (a) goes positive, which helps extend the articulated leg as the knees thrust. During the remaining stance phases, however, both the front and rear legs exhibit larger negative torques and positive power, as the quadruped gradually accelerates to the desired running speed. This is in contrast with the gallop, where positive, shoulder-braking torques in the front retard the forward motion but help stabilize pitch (Sect. 4.4.1). Consequently, the shoulder power is generally negative during galloping (Sect. 4.4.3). During the last stage of the jump-start ($t = 1.60\,\mathrm{s}$), the shoulder torque reverses to act as a brake, resulting in the regeneration of power. This helps produce the negative (nose-up) pitch rate of the body to facilitate the transition to galloping, as stated in Sect. 6.2.4.

The torque and power for the knees (series configuration, Sect. 4.4.3) are shown in Fig. 6.6. Both the front and rear knee torques coincide during the first contact

[34]The large spikes in torque and power that occur after each leg initially breaks contact are a result of bringing the leg back under active control. Reducing the amplitude of the spikes may be possible by temporarily limiting hip torque or using different PD gains for the hip joints during the brief period after lift-off. During this time, however, the hip trajectory tracking error may increase.

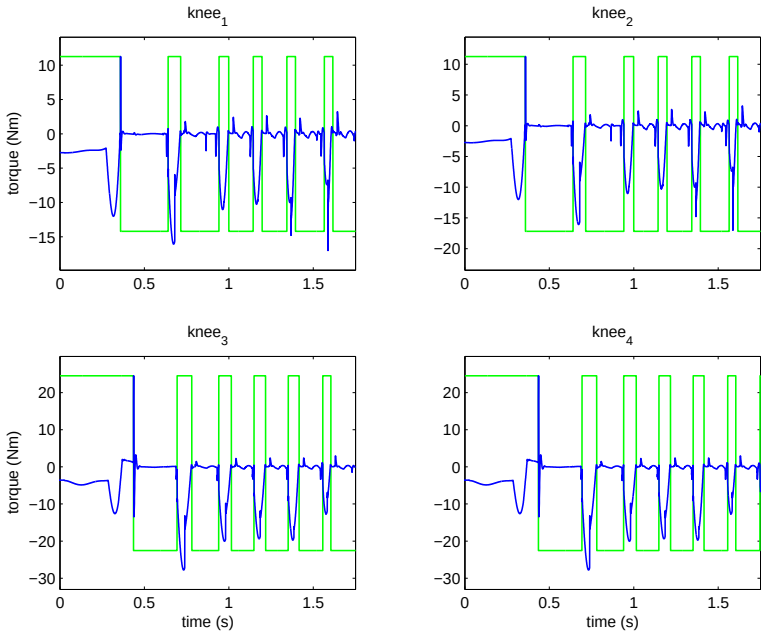


(a)

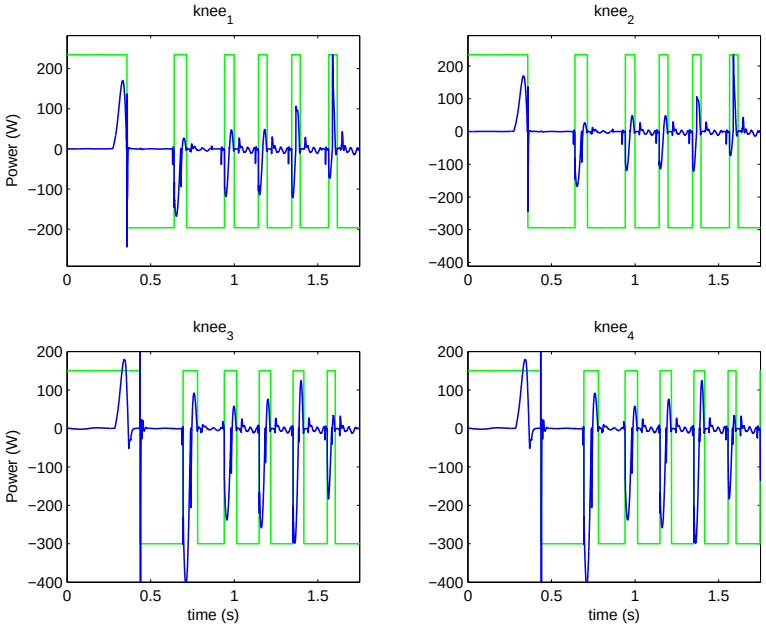


(b)

Figure 6.5: Torque and power requirements for the hip axes during the jump-start. Square waveforms indicate leg contact when the level is high. The long contact at the beginning occurs during Stage 1, while the shorter contact periods occur in Stages 2 through 6.



(a)



(b)

Figure 6.6: Torque and power requirements for the knee axes during the jump-start. Square waveforms indicate contact.

phase to produce a nearly simultaneous knee thrust in all legs. It is during this period that the knee power reaches its maximum positive value. During the next contact phase, when the quadruped lands after the initial leap, all knee joints actively remove energy from the knee springs to prevent a large, vertical rebound. This is indicated by the "reverse spike" (i.e., reduction of magnitude) in torque and energy levels during the second contact phase in each leg. During subsequent contact phases, energy is removed in the rear, which helps stabilize pitch. The knee power during these contact phases is mostly negative, since both the front and rear hips actively retract the legs, which forcibly compresses the knee springs. During the last contact phase, however, the front knees inject a significant amount of power to correct the pitch rate for the transition to galloping.

The total power is shown in Fig. 6.7. As in Sect. 4.4.3, negative power offsets positive power. The total power reaches its first peak during the knee thrust in the first contact period. Once the legs break contact, there is a brief spike in power as hips and knees are quickly brought back under PD control following the THRUST function, where large torques are applied to the joints. For this reason, the first FREE function after the initial leap is effectively eliminated (Table 6.1) so that the legs will not continue rotating at excessive rates once they break contact. Generally, the power remains positive during the subsequent contact phases. On average, the peak values during contact are slightly greater than those for the gallop (Fig. 4.14), which makes sense since more power is needed during periods of acceleration.

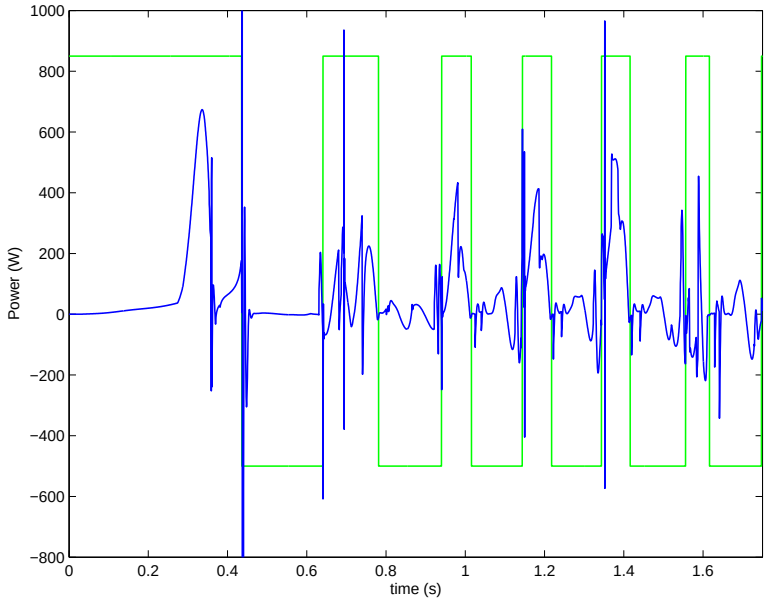


Figure 6.7: Total power required for all joints for the jump-start. The square waveform indicates that one or more legs are in contact when high.

6.3 The Sudden Stop

The sudden stop is used to terminate galloping and consists of the state diagram shown in Fig. 6.8. Two possible termination paths exist for each leg for the sudden stop, as indicated by the dotted arrow in Fig. 6.8. In the first case, when the leg does *not* break contact during GALLOP-STOP-CONTROL, the leg terminates here. In the second case, when the leg does break contact, it transitions to TRANSFER, then to HOLD, where it terminates. Two termination conditions were included in the maneuver to recognize the two possibilities that could occur, as it was not known *a priori* which route any particular solution would take. In either case, each leg is held at specified ab/ad and hip angles and at the nominal leg length of l_0. This allows the quadruped to achieve a stable standing position at the end of the maneuver.

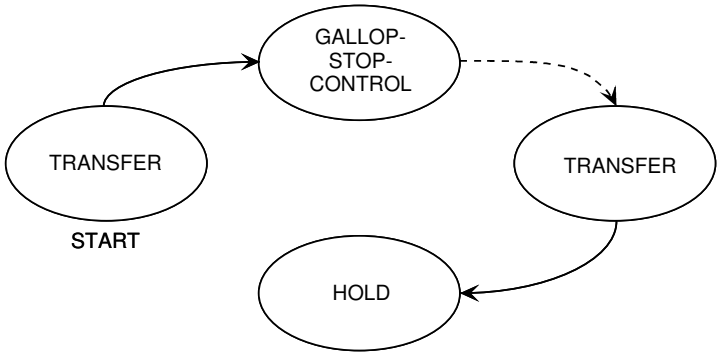


Figure 6.8: State diagram for each leg for the sudden stop. The dotted arrow indicates a potential transition. For example, if a leg breaks contact during GALLOP-STOP-CONTROL, it transitions to TRANSFER, then terminates in HOLD. If it remains in contact, however, the leg terminates in GALLOP-STOP-CONTROL.

A summary of the leg parameters for the sudden stop maneuver is given in Table 6.4. During the first TRANSFER function (TRANSFER 1 in Table 6.4), the ab/ad angles are rotated to evolved positions $\theta^*_{a_{\mathrm{L}}}$ and $\theta^*_{a_{\mathrm{R}}}$ for the left side legs (legs 1 and 3) and right side legs (legs 2 and 4), respectively. The virtual leg angles are set to evolved values $\theta^*_{v_{1,f}}$ and $\theta^*_{v_{1,r}}$ for the front and rear legs, respectively. The typical double-spline is employed for the knees ($\sigma_{k_d} = 1$), and the spline transfer time is $T_t \cdot T = 0.1\,\mathrm{s}$, which is the same value used for galloping (Sect. 4.2.1). However, the total period T is increased to $0.25\,\mathrm{s}$ so that each leg remains in TRANSFER until it touches down. For $t > T_t \cdot T$, each leg joint is held at its final position (Sect. 2.3.4).

When the leg touches down, it enters the GALLOP-STOP-CONTROL function, which is described in the next section. If the leg remains in contact during GALLOP-STOP-CONTROL, then it terminates inside this function. However, if it breaks contact, the leg transitions to a second TRANSFER function (TRANSFER 2, Table 6.4). During this leg transfer, the ab/ad angles are rotated to the same angles

Function	Parameters	Description
TRANSFER 1	$\sigma_{a_e} = 0, \quad \sigma_{h_e} = 0,$ $\sigma_s = 1, \quad \sigma_{k_d} = 1,$ $T_r = 0.0, \quad \boldsymbol{\theta}_{l_f} = [\theta^*_{a_i}, \theta^*_{v_{1,i}}, \frac{l_0}{2}]^T,$ $T_t = 0.4, \quad T = 0.25$	Transfer using evolved ab/ad angle $\theta^*_{a_i}$ given by $\theta^*_{a_i} = \begin{cases} \theta^*_{a_\mathrm{L}} & \text{if } i = 1 \vee i = 3 \\ \theta^*_{a_\mathrm{R}} & \text{otherwise,} \end{cases}$ and evolved virtual leg angle $\theta^*_{v_{1,i}}$ $\theta^*_{v_{1,i}} = \begin{cases} \theta^*_{v_{1,\mathrm{f}}} & \text{if } i \leq 2 \\ \theta^*_{v_{1,\mathrm{r}}} & \text{otherwise,} \end{cases}$ where $i = 1, \ldots, 4$. Length l_0 is given in Table 2.1.
GALLOP-STOP-CONTROL	$k_{p_a} = 1077.7, \; k_{d_a} = 4.3176,$ $k_{p_h} = k^*_{p_h}, \quad k_{d_h} = k^*_{d_h},$ $k_{p_k} = k^*_{p_k}, \quad k_{d_k} = k^*_{d_k},$ $E_d = E^*_d, \quad \boldsymbol{\theta}_{l_d} = [\theta^*_{a_i}, \theta^*_{v_{2,i}}, l_0]^T,$ $\sigma_{a_e} = 0, \quad \sigma_{h_e} = 1,$ $T_s = 0.15, \quad T_t = 0.5,$ $T = 1.0\,\mathrm{s}$	During the first stage ($t < T_s{\cdot}T$), use evolved gains $k^*_{p_h}$, $k^*_{d_h}$, $k^*_{p_k}$, and $k^*_{d_k}$ and evolved energy value E^*_d for all legs to remove kinetic energy from the system. During the second stage ($t_s \cdot T < t < (T_s + T_t)\,T$), transfer to evolved ab/ad angle $\theta^*_{a_i}$ (above), length l_0, and evolved virtual leg angle $\theta^*_{v_{2,i}}$ given as follows: $\theta^*_{v_{2,i}} = \begin{cases} \theta^*_{v_{2,\mathrm{f}}} & \text{if } i \leq 2 \\ \theta^*_{v_{2,\mathrm{r}}} & \text{otherwise,} \end{cases}$ for $i = 1, \ldots, 4$. During the last stage ($t > (T_s + T_t)\,T$), hold the leg at $\boldsymbol{\theta}_{l_d}$.
TRANSFER 2	$\sigma_{a_e} = 1, \quad \sigma_{h_e} = 1,$ $\sigma_s = 1, \quad \sigma_{k_d} = 0,$ $T_r = 0.5, \quad \boldsymbol{\theta}_{l_f} = [\theta^*_{a_i}, \theta^*_{v_{2,i}}, l_0]^T,$ $T_t = 1.0, \quad T = 0.25$	Transfer using $\theta^*_{a_i}$ from TRANSFER 1 (above), and evolved leg angle $\theta^*_{v_{2,i}}$ (above).
HOLD	$\sigma_{a_e} = 1, \quad \sigma_{h_e} = 1,$ $\sigma_{k_f} = 0, \quad \sigma_s = 1,$ $T_r = 0.01, \quad \boldsymbol{\theta}_{l_d} = [\theta^*_{a_i}, \theta^*_{v_{2,i}}, l_0]^T,$ $T = 1.0\,\mathrm{s}$	Hold using ab/ad angle $\theta^*_{a_i}$ (TRANSFER 1), and virtual leg angle $\theta^*_{v_{2,i}}$ (GALLOP-STOP-CONTROL). Use inertial control for both joints, and reset the knee spring over $T_r{\cdot}T$.

Table 6.4: Summary of the parameters for the sudden stop. TRANSFER 1 is the first TRANSFER function in the sequence (Fig. 6.8), while TRANSFER 2 is the second.

$\theta^*_{a_\mathrm{R}}$ and $\theta^*_{a_\mathrm{L}}$ used during the first TRANSFER function. However, the virtual leg angles are rotated to new angles $\theta^*_{v_{2,f}}$ and $\theta^*_{v_{2,r}}$, which are the target angles used during GALLOP-STOP-CONTROL. Furthermore, inertial control is used for both the ab/ad and hip joints to maintain these angles with respect to the inertial frame instead of with respect to the body. This approach was found to be more robust when the quadruped bounces to a stop, which causes significant pitch, roll, and yaw motion during flight. Maintaining ab/ad and hip joint angles with respect to the inertial frame allows for automatic correction to compensate for the motion of the body.

The knee spring rest position is also reset during TRANSFER 2, using $T_r = 0.5T$ and $T = 0.25\,\mathrm{s}$. This is required because the GALLOP-STOP-CONTROL adjusts the knee spring rest position during stance to achieve the target energy value. The knee transfer spline is a single spline for the second TRANSFER function ($\sigma_{k_d} = 0$) with the target length of l_0, since the final stance position should be at the nominal standing height. The transfer time for all joints is set to $0.25\,\mathrm{s}$, which is longer than the transfer period used during galloping, since there is typically more flight time incurred during the sudden stop maneuver.

After TRANSFER 2 expires, the leg transitions to HOLD, where it terminates. The HOLD parameters are similar to those of TRANSFER 2, including the target ad/ad, hip, and length values. This allows a consistent leg position to be maintained starting from the last stage of the GALLOP-STOP-CONTROL function, which is discussed next.

6.3.1 GALLOP-STOP-CONTROL

The GALLOP-STOP-CONTROL function, which was introduced in Table 2.4, is the leg primitive function used during the stance phase of the sudden stop maneuver and is chiefly responsible for stopping the quadruped's motion. The GALLOP-STOP-CONTROL function is divided into three separate stages, determined by the parameters T_s, T_t, and T. The first parameter T_s determines the length of the first stage, which is $T_s \cdot T$. During the first stage, when $t < T_s \cdot T$, PD parameters given by k_{p_a}, k_{d_a}, k_{p_h}, k_{d_h}, k_{p_k}, and k_{d_k} (not to be confused with the default PD parameters given in Table 2.5) are used with the target values specified in $\boldsymbol{\theta}_{l_d}$ to damp out the motion of the joints. Typically, the ab/ad PD parameters are set to the default values used during GALLOP-STANCE-CONTROL (Sect. 4.2.5), $k_{p_a} = 1077.7$ and $k_{d_a} = 4.3176$, which were found to be sufficient to hold the ab/ad joints even during sliding stops. The remaining gains are evolved with relatively large damping values to dissipate the kinetic energy of the system.

If the leg reaches maximum compression during the first stage, the knee spring energy is set to the desired value E_d by adjusting the rest position of the spring in the same manner as is done in GALLOP-STANCE-CONTROL. Typically the energy is set to a very small value so that most of the knee spring energy is removed. The T_s value is set to $0.15T$, where $T = 1.0\,\text{s}$, to provide adequate time for energy removal during the first stage.

Although the damping for the ab/ad and hip joints are applied throughout the first stage, the knee damping is applied only during leg compression and only when $t > 0.01\,\text{s}$. Limiting the knee damping to this time period is necessary to prevent chattering of the foot contact. Because the shank link is relatively light, the knee

spring is stiff, and there exists a hard force boundary at the surface, a chattering contact (i.e., limit cycle) can develop as the leg becomes unloaded, especially when the damping value is changed from the hand-tuned value given in Table 2.1. On the other hand, applying knee damping during compression does not cause this problem because the loaded leg appears to be much less sensitive to changes in the knee damping value.

During the second stage, when $T_s \cdot T < t < (T_s + T_t)\, T$, the leg is transferred from its initial position to the target position by executing the TRANSFER function with σ_{a_e}, σ_{h_e}, $\boldsymbol{\theta}_{l_d}$, T_t, and T (assuming TRANSFER parameters $\sigma_{k_d} = \sigma_s = 0$, and $T_r = 0.0$). The PD gains used during this stage are identical to those used during GALLOP-STANCE-CONTROL (Sect. 4.2.5). Inertial control is used at the hip joint ($\sigma_{h_e} = 1$ in Table 6.4), as this method compensates for the remaining pitch motion of the body. In addition, the knee spring rest position is gradually returned to its nominal value over the entire length of this stage, $T_t \cdot T$. However, if the leg breaks contact at or before the knee spring rest position is fully restored, the subsequent function must reset the spring, as described in the previous section.

Finally, when $t > (T_s + T_t)\, T$, the leg is held at the target values specified in $\boldsymbol{\theta}_{l_d}$ using the PD gains from GALLOP-STANCE-CONTROL (Sect. 4.2.5). In the optimal case, the leg remains in GALLOP-STOP-CONTROL for all three stages and terminates in the final stage. If it breaks contact, which happens during bouncing stops, it proceeds to the second TRANSFER function and the final HOLD function, which generally increases the required stopping time.

6.3.2 Sudden Stop Initialization

The sudden stop maneuver is designed to bring the quadruped to a halt from typical TOF galloping conditions. Although the maneuver could be evolved for one particular set of conditions as long as the termination point is known in advance, an alternative approach is employed here. Using the perturbation capabilities of the INIT-TOF function (Sect. 2.4.4), average TOF galloping conditions are perturbed using a noise function to generate initial conditions for the sudden stop. This was done to determine whether a valid stopping solution could be evolved without using the actual galloping conditions during the evolution. The use of noise during evolution has been shown to produce more robust solutions in environments that were not encountered during evolution [99, 100], although these examples involved far simpler, statically stable robots. While a complete analysis of using noise during evolution is beyond the scope of this work, the methodology presented here should provide enough data to facilitate further study.

The parameters for the INIT-TOF function are listed in Table 6.5. The TOF galloping conditions in $\bar{\boldsymbol{\theta}}_b$ were computed based on the average body state for the first 60 s of galloping (Fig. 4.6). The noise scaling values in $\boldsymbol{\delta}_b$ were determined based roughly on the range of values for each body state variable over the same period. As stated in Sect. 2.4.4, the body state used for initialization is obtained by perturbing the nominal state. The body state is given as follows:

$$\boldsymbol{\theta}'_b = \bar{\boldsymbol{\theta}}_b + \boldsymbol{\Psi}\boldsymbol{\delta}_b \,, \tag{6.13}$$

where $\boldsymbol{\theta}'_b$ is the body state used to initialize the quadruped, $\boldsymbol{\Psi}$ is a diagonal matrix of random values on the interval $[-1.0, 1.0]$, and $\bar{\boldsymbol{\theta}}_b$ is given in Table 6.5.

Parameter	Value
σ_p	1
σ_{a_e}	0
σ_{h_e}	0
σ_{k_d}	1
σ_{l_a}	0
$\bar{\boldsymbol{\theta}}_b = [\ \bar{\alpha},\ \bar{\beta},\ \bar{\gamma},\ {}^{e}x_b,\ {}^{e}y_b,\ {}^{e}\bar{z}_b,\ \dot{\bar{\alpha}},\ \dot{\bar{\beta}},\ \dot{\bar{\gamma}},\ {}^{e'}\dot{\bar{x}}_b,\ {}^{e'}\dot{\bar{y}}_b,\ {}^{e}\dot{\bar{z}}_b\ ]^T$	$[\ 0.0\,\text{rad},\ \text{-}0.025\,\text{rad},\ \text{-}0.075\,\text{rad},\ 1.0\,\text{m},\ 1.0\,\text{m},\ 0.235\,\text{m},\ \text{-}0.05\,\text{rad/s},\ \text{-}0.25\,\text{rad/s},\ 0.4\,\text{rad/s},\ 4.17\,\text{m/s},\ 0.14\,\text{m/s},\ 0.0\,\text{m/s}\ ]^T$
$\boldsymbol{\delta}_b = [\ \delta_\alpha,\ \delta_\beta,\ \delta_\gamma,\ \delta_x,\ \delta_y,\ \delta_z,\ \delta_{\dot{\alpha}},\ \delta_{\dot{\beta}},\ \delta_{\dot{\gamma}},\ \delta_{\dot{x}},\ \delta_{\dot{y}},\ \delta_{\dot{z}}\ ]^T$	$[\ 0.0\,\text{rad},\ 0.025\,\text{rad},\ 0.025\,\text{rad},\ 0.0\,\text{m},\ 0.0\,\text{m},\ 0.003\,\text{m},\ 0.075\,\text{rad/s},\ 0.25\,\text{rad/s},\ 0.125\,\text{rad/s},\ 0.02\,\text{m/s},\ 0.02\,\text{m/s},\ 0.0\,\text{m/s}\ ]^T$
$\boldsymbol{\phi}_l = [\ \theta^1_{a_0},\ \ldots,\ \theta^4_{a_0},\ \theta^1_{v_0},\ \ldots,\ \theta^4_{v_0},\ d^1_0,\ \ldots,\ d^4_0,\ \theta^1_{a_f},\ \ldots,\ \theta^4_{a_f},\ \theta^1_{v_f},\ \ldots,\ \theta^4_{v_f},\ d^1_f,\ \ldots,\ d^4_f\ ]^T$	$[\ \pi/2\,\text{rad},\ \ldots,\ \pi/2\,\text{rad},\ -0.35\,\text{rad},\ \ldots,\ -0.35\,\text{rad},\ 0.24\,\text{m},\ \ldots,\ 0.24\,\text{m},\ \pi/2\,\text{rad},\ \ldots,\ \pi/2\,\text{rad},\ 0.69\,\text{rad},\ 0.69\,\text{rad},\ 0.65\,\text{rad},\ 0.65\,\text{rad},\ 0.12\,\text{m},\ \ldots,\ 0.12\,\text{m}\ ]^T$
$\boldsymbol{\theta}_l$	$\mathbf{0}_{24\times1}$
$\boldsymbol{t}^* = [t^*_1,\ t^*_2,\ t^*_3,\ t^*_4]^T$	$[0.3,\ 0.4,\ 0.6,\ 0.7]^T$
T_t	0.1 s

Table 6.5: Summary of the INIT-TOF parameters for initializing the sudden stop. $\bar{\boldsymbol{\theta}}_b$ is the average TOF body state from the first 60 s of galloping (Fig. 4.6), while $\boldsymbol{\delta}_b$ contains the perturbation ranges applied to $\bar{\boldsymbol{\theta}}_b$. The body state used for initialization is computed using Eq. (6.13).

The transfer spline values in $\boldsymbol{\phi}_l$ are nominal starting and ending conditions for the legs used for the gallop (Table 4.3). While these values are not true averages from the actual galloping data, they were sufficient for the purposes of this study, as the legs do not have a significant impact on the body's dynamics during flight.

6.3.3 Sudden Stop Optimization Problem

The sudden stop optimization problem is defined by the following vector of evolved parameters:

$$\boldsymbol{\phi}_{\mathrm{S}} = \left[\theta^*_{a_{\mathrm{L}}},\, \theta^*_{a_{\mathrm{R}}},\, \theta^*_{v_{1,f}},\, \theta^*_{v_{1,r}},\, \theta^*_{v_{2_f}},\, \theta^*_{v_{2_r}},\, k^*_{p_h},\, k^*_{d_h},\, k^*_{p_k},\, k^*_{d_k},\, E^*_d\right]^T, \tag{6.14}$$

where $\theta^*_{a_{\mathrm{L}}}$ and $\theta^*_{a_{\mathrm{R}}}$ are the left and right ab/ad angles used in all of the leg primitive functions (Fig. 6.8), $\theta^*_{v_{1,f}}$ and $\theta^*_{v_{1,r}}$ are the front and rear virtual leg angles used during the first TRANSFER function, $\theta^*_{v_{2,f}}$ and $\theta^*_{v_{2,r}}$ are the front and rear virtual leg angles used during the remaining leg functions, gains $k^*_{p_h}$, $k^*_{d_h}$, $k^*_{p_k}$, and $k^*_{d_k}$ are used during the first stage of the GALLOP-STOP-CONTROL (Sect. 6.3.1), and E^*_d is the target energy value for the knee springs at maximum compression during the first stage of GALLOP-STOP-CONTROL. The ranges for each of these evolved parameters is given in Table 6.6. The ranges for $\theta^*_{v_{2,f}}$ and $\theta^*_{v_{2,r}}$ were set to produce a slightly sprawled posture of the legs in the fore-aft direction for more stability during the final stance phase of the sudden stop maneuver. The range for $k^*_{p_k}$ includes negative values to offset the stiffness of the knee spring, which is 38.0 Nm/rad (Table 2.1). This allows the algorithm to increase or decrease the compliance of the knee joint with respect to its nominal value.

Parameter	Range
$\theta^*_{a_{\mathrm{L}}}$	[1.57, 1.97] rad
$\theta^*_{a_{\mathrm{R}}}$	[1.17, 1.57] rad
$\theta^*_{v_{1,f}}$, $\theta^*_{v_{1,r}}$	[0.52, 0.87] rad
$\theta^*_{v_{2,f}}$	[0.0, 0.25] rad
$\theta^*_{v_{2,r}}$	[−0.25, 0.0] rad
$k^*_{p_h}$	[0.0, 500.0] Nm/rad
$k^*_{d_h}$	[0.0, 5.0] Nm/(rad/s)
$k^*_{p_k}$	[−38.0, 38.0] Nm/rad
$k^*_{d_k}$	[0.0, 1.0] Nm/(rad/s)

Table 6.6: Ranges for each evolved parameter for the sudden stop. The range for k_{p_k} includes negative values to offset the stiffness of the knee spring (Table 2.1).

The fitness function for the sudden stop maneuver consists of two elements, a combined accuracy and correctness component and a time component. The fitness vector is given as follows:

$$\boldsymbol{f}_{\mathrm{S}} = [f_a^{\mathrm{S}},\, f_t^{\mathrm{S}}]^T \ , \tag{6.15}$$

where f_a^{S} is the combined accuracy and correctness factor, and f_t^{S} is the time factor. The first component includes elements for the correct number of footfalls and the final orientation of the body. The calculation is given as follows:

$$f_a^{\mathrm{S}} = \left(\frac{\sum_{i=1}^{4} \sigma_{c_i}}{4} \right) \left(\frac{w_a^{\mathrm{S}}}{1 + w_l \sum_{i=1}^{4} (c_i - 1)^2 + w_\theta ||\boldsymbol{\Lambda}_{\mathrm{S}} \left(\boldsymbol{\theta}_b^s - \boldsymbol{\theta}_{b_d}^s \right) ||^2} \right) \ , \tag{6.16}$$

where σ_{c_i} is the contact state for leg i at the termination of the maneuver, w_a^{S} is the accuracy fitness scaling factor of 100.0, $w_l = 0.1$, c_i is the number of contact periods

Parameter	Value	Parameter	Value	Parameter	Value
λ_α	28.6	λ_y	0.0	$\lambda_{\dot{\gamma}}$	5.0
λ_β	28.6	λ_z	200.0	$\lambda_{\dot{x}}$	10.0
λ_γ	28.6	$\lambda_{\dot{\alpha}}$	5.0	$\lambda_{\dot{y}}$	10.0
λ_x	0.0	$\lambda_{\dot{\beta}}$	5.0	$\lambda_{\dot{z}}$	10.0

Table 6.7: Scaling factors used for the sudden stop. Scaling factors of zero indicate "don't-care" conditions.

for leg i, $w_\theta = 0.1$, $\mathbf{\Lambda}_\mathrm{S}$ is a diagonal matrix containing the scaling values given in Table 6.7, $\boldsymbol{\theta}_b^s$ is the final stance-phase body state, and $\boldsymbol{\theta}_{b_d}^s$ is the desired stance-phase body state, given as follows:

$$\begin{aligned}\boldsymbol{\theta}_{b_d}^s &= \left[\ \alpha_d^s,\, \beta_d^s,\, \gamma_d^s,\, {}^e x_{b_d}^s,\, {}^e y_{b_d}^s,\, {}^e z_{b_d}^s,\, \dot{\alpha}_d^s,\, \dot{\beta}_d^s,\, \dot{\gamma}_d^s,\, {}^{e'}\dot{x}_{b_d}^s,\, {}^{e'}\dot{y}_{b_d}^s,\, {}^e\dot{z}_{b_d}^s\ \right]^T \\ &= \left[\ 0,\ \ 0,\ \ 0,\ \ 0,\ \ 0,\ \ l_0,\ \ 0,\ \ 0,\ \ 0,\ \ 0,\ \ 0,\ \ 0\ \right]^T , \end{aligned} \tag{6.17}$$

where l_0 is given in Table 2.1. The final desired body state consists of zero roll, pitch, and yaw[35], arbitrary x- and y-positions (since $\lambda_x = \lambda_y = 0.0$ in Table 6.7), a desired z-position equal to the nominal standing height of l_0, and zero angular and linear velocity. Furthermore, the desired number of contacts for each leg is one.

The criterion f_t^S encourages minimal stopping time and is computed as follows:

$$f_t^\mathrm{S} = \frac{w_t^\mathrm{S}}{1 + t_\mathrm{stop}} , \tag{6.18}$$

where $w_t^\mathrm{S} = 100.0$, and t_stop is the stopping time, computed as

$$t_\mathrm{stop} = t_f - \min_i \left(t_{\mathrm{td}_i}\right) , \tag{6.19}$$

[35]Here it is assumed that the quadruped is heading in the x-direction with desired yaw angle $\alpha = 0.0\,\mathrm{rad}$.

where t_f is the time at which the maneuver terminates, and t_{td_i} is the earliest touchdown time incurred by leg i. The maneuver is terminated 0.25 s after all velocity components of the body's velocity vector have settled to within their specified thresholds, which are given in $\boldsymbol{\epsilon}$, as follows:

$$\boldsymbol{\epsilon} = \left[\varepsilon_{\mathrm{ang}},\, \varepsilon_{\mathrm{ang}},\, \varepsilon_{\mathrm{ang}},\, \varepsilon_{\mathrm{lin}},\, \varepsilon_{\mathrm{lin}},\, \varepsilon_{\mathrm{lin}}\right]^T , \tag{6.20}$$

where $\varepsilon_{\mathrm{ang}}$ is the angular velocity threshold, and $\varepsilon_{\mathrm{lin}}$ is the linear velocity threshold. These variables were tuned to produce a sufficiently stationary configuration at termination, yielding $\varepsilon_{\mathrm{ang}} = 0.1\,\mathrm{rad/s}$ and $\varepsilon_{\mathrm{lin}} = 0.05\,\mathrm{m/s}$. Each velocity component must settle to within $[-\varepsilon, \varepsilon]$, similar to the 2% settling time computed for linear systems. When all elements have settled to within their thresholds and have remained there for at least 0.25 s, the maneuver terminates. The stopping time is measured from the first leg contact to the time when the system has sufficiently come to rest. Solutions with minimal stopping time receive higher fitness, as these solutions represent more sudden stops.

6.3.4 Sudden Stop Results

The sudden stop was evolved using 10 trials, $S = 32$ individuals, and $N = 100$ generations. Furthermore, because each individual is initialized randomly (Sect. 6.3.2), 10 iterations were run per individual and the fitness averaged across all iterations. Multiple acceptable solutions were found, including both sliding and bouncing stops, although only one sliding stop is shown here. The sliding stop represents what appears to be the most effective method for stopping based on minimal stopping time.

The evolved parameters are given in Table 6.8. Both the front and rear virtual leg angles $\theta^*_{v_{1,f}}$ and $\theta^*_{v_{1,r}}$ in Table 6.8 are set at protracted angles for touchdown

Parameter	Value	Parameter	Value
$\theta^*_{a_{\mathrm{L}}}$	1.82 rad	$k^*_{p_h}$	1.52 Nm/rad
$\theta^*_{a_{\mathrm{R}}}$	1.31 rad	$k^*_{d_h}$	3.43 Nm/(rad/s)
$\theta^*_{v_{1,f}}$	0.55 rad	$k^*_{p_k}$	-33.60 J
$\theta^*_{v_{1,r}}$	0.59 rad	$k^*_{d_k}$	0.71 J
$\theta^*_{v_{2_f}}$	0.21 rad	E^*_d	0.03 J
$\theta^*_{v_{2_r}}$	0.0 rad		

Table 6.8: Evolved parameters for the sudden stop.

to produce a rearward fore-aft force component to retard the quadruped's forward motion. Both the hip and knee proportional gains, $k_{p_h} = 1.52\,\mathrm{Nm/rad}$ and $k_{p_k} = -33.60\,\mathrm{Nm/rad}$, result in small effective compliance values at each joint. The negative knee proportional gain k_{p_k} must be added to the stiffness of the knee spring, $k_s = 38.0\,\mathrm{Nm/rad}$ (Table 2.1), to compute the effective compliance, which is $4.4\,\mathrm{Nm/rad}$. The evolved damping gains, $k^*_{d_h} = 3.43\,\mathrm{Nm/(rad/s)}$ and $k^*_{d_k} = 0.71\,\mathrm{Nm/(rad/s)}$, are large in relation to the effective proportional gains and produce significant dissipation of kinetic energy during impact.

Results for the sliding-style stop are shown in Fig. 6.9. The figure shows that all velocity components have settled to within their specified thresholds at $t = 1.80\,\mathrm{s}$ (indicated by the arrow in Fig. 6.9 (b)), which results in termination at $t = 2.05\,\mathrm{s}$. The forward velocity ${}^c\dot{x}_b$ in plot (d) decelerates to almost zero in about $0.5\,\mathrm{s}$, mainly because the feet maintain good contact throughout this period. The contact forces for each foot are shown in Fig. 6.10. Slipping is prevalent in the first $0.5\,\mathrm{s}$ in all legs, which indicates that the quadruped is sliding to a halt. Although leg 4 briefly breaks

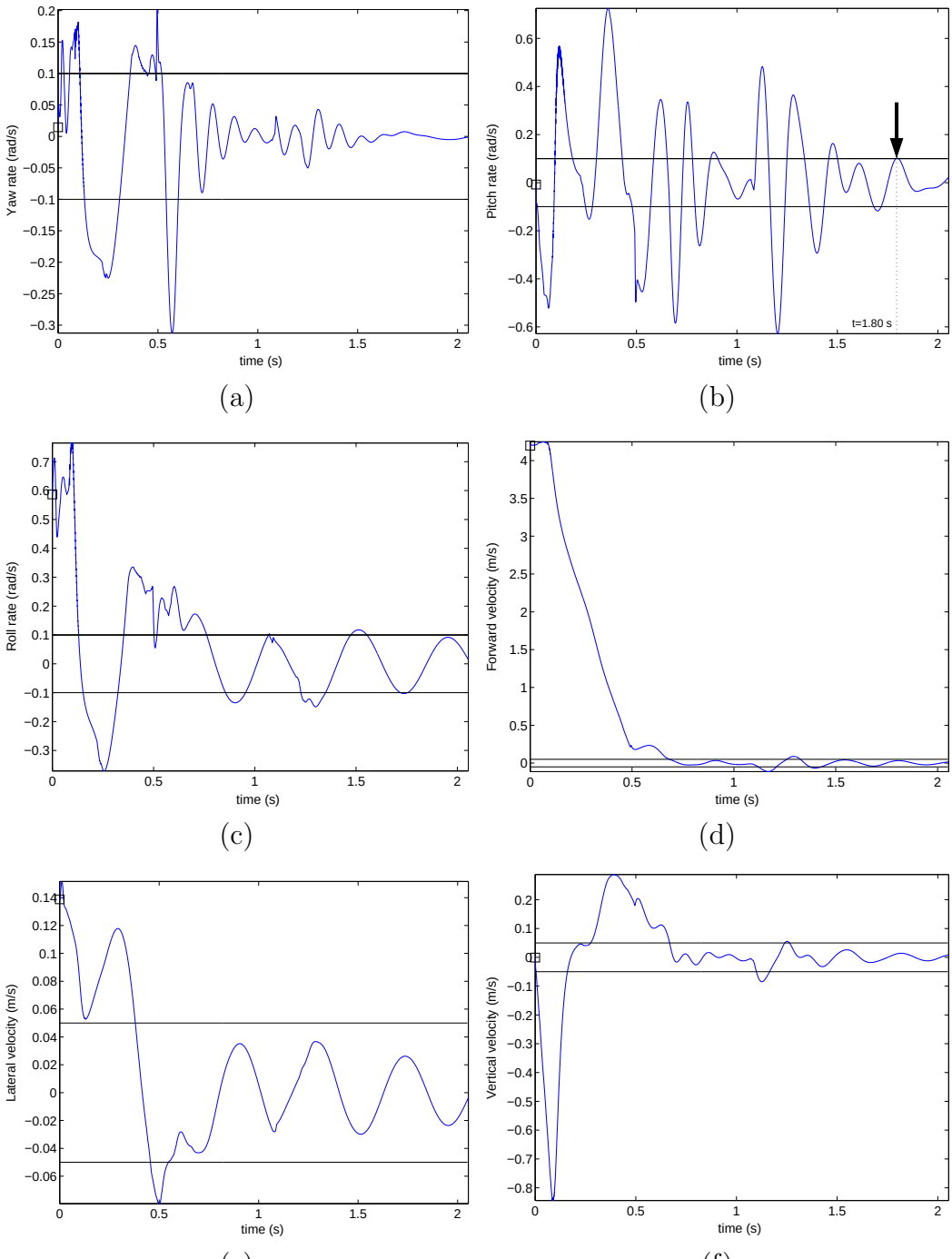


Figure 6.9: Results for the sudden stop: (a) Yaw rate, (b) pitch rate (c) roll rate, (d) forward velocity, (e) lateral velocity, and (f) vertical velocity. Stopping thresholds are indicated in each plot, and the arrow in (b) is the time when all velocities have entered the envelope ($t = 1.80\,\mathrm{s}$), causing the maneuver to terminate 0.25 s later.

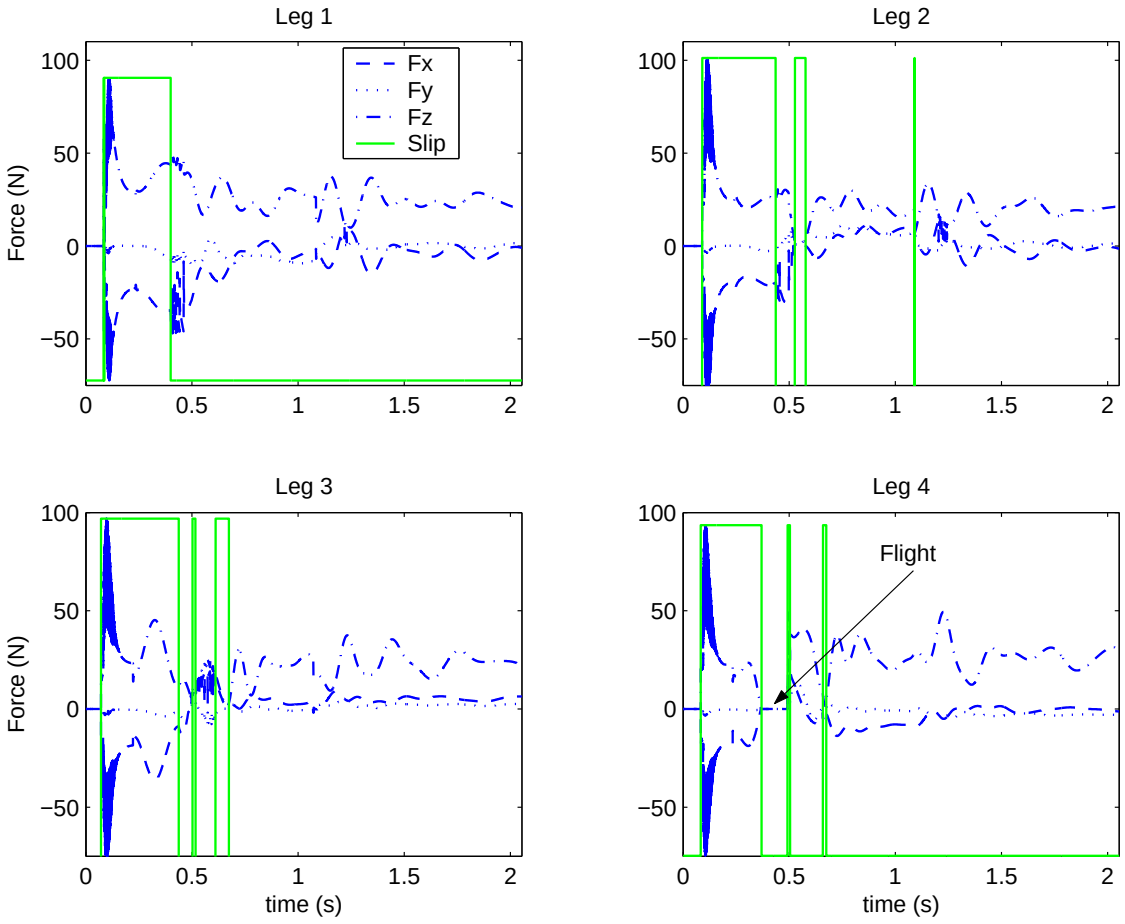


Figure 6.10: Foot forces for the sudden stop. The square waveform in each plot indicates slipping when it is high, which means the quadruped is sliding to a halt. Note that leg 4 briefly breaks contact from 0.37 to 0.49 s, indicated by zero foot forces during this period.

contact, at about $t = 0.4\,\mathrm{s}$, the remaining legs maintain continuous contact once they touch down. Although sliding was not the preferred mode of stopping, it does appear that this approach produces the most rapid deceleration. Other solutions that did not involve sliding resulted in a bouncing stop, which requires more time.

Perhaps the most significant aspect of the sliding-style stop is that this mode was not anticipated during the design of the stopping sequence and the fitness criteria. In fact, the desired mode was a single-contact, non-sliding stop, where the quadruped "sticks" a landing, much like a gymnast. However, it appears that this solution is not possible due to limited friction. In hindsight, this result is not surprising, since the jump-start could not accelerate the body to full running speed in one leap (Sect. 6.2.4)[36]. For the stop, it would follow that decelerating from the steady-state running speed of $4.15\,\mathrm{m/s}$ to zero is also not possible.

The sliding-style stop represents one of the more exciting features of evolution, which is the generation of *emergent* solutions — solutions that might not be predicted in advance but arise naturally from the properties of the system. These solutions, while often not the desired form, can provide additional insight into the underlying problem. In this case, the sliding-style stop is actually more realistic with respect to biological stopping than the original goal. Quadrupedal mammals often slide to a halt when stopping suddenly or executing a very sharp turn [85]. However, sliding with multiple contact points is a complex, nonlinear phenomenon that is highly dependent on stance configuration, joint-level control gains, and the contact model itself. Given these factors, it was not known in advance that a sliding-style stop was even possible

[36]During testing, the sudden stop was evolved prior to the jump-start. Consequently, the jump-start results were not available during design and testing of the sudden stop maneuver.

for this system. The evolutionary algorithm found a realistic and efficient method to stop despite the fact that this was not the original goal.

Robustness of the Sudden Stop

As stated above, noise is used during initialization in order to produce stopping solutions for the range of conditions expected during steady-state galloping. Ten iterations were run for each individual, with random initialization for each iteration using the average body state and perturbation ranges given in Table 6.5. In this section, the results of testing the above solution using 150 random stopping strides (about 50% of the total strides) during the first 60 s of galloping are presented. While more testing would be necessary to better quantify the robustness of the solution, the results do provide sufficient evidence that acceptable solutions can be evolved to conditions not explicitly introduced during evolution if (1) the specific variables of uncertainty can be identified, (2) the range of the uncertainty for each variable can be quantified, and (3) noise is introduced to model each uncertain variable. The results are shown in Fig. 6.11. The figure shows that the stopping time in (a) varies between 1.1 s and 2.0 s, with successful stops produced in all but the first two cases. The relevant conditions of the body prior to the stop are given in plots (b) through (i). The initialization range is indicated for each body state variable, which shows that there are some instances when the actual galloping conditions exceed the initialization range. In most of these cases, except for the first two (the squares in Fig. 6.11), the quadruped is still able to stop, which means that the solution's robustness extends beyond the conditions used for testing. The first two cases cause errors (where the body falls over) because the quadruped has not reached steady-state conditions, causing multiple body state variables to fall outside the initialization ranges used during evolution.

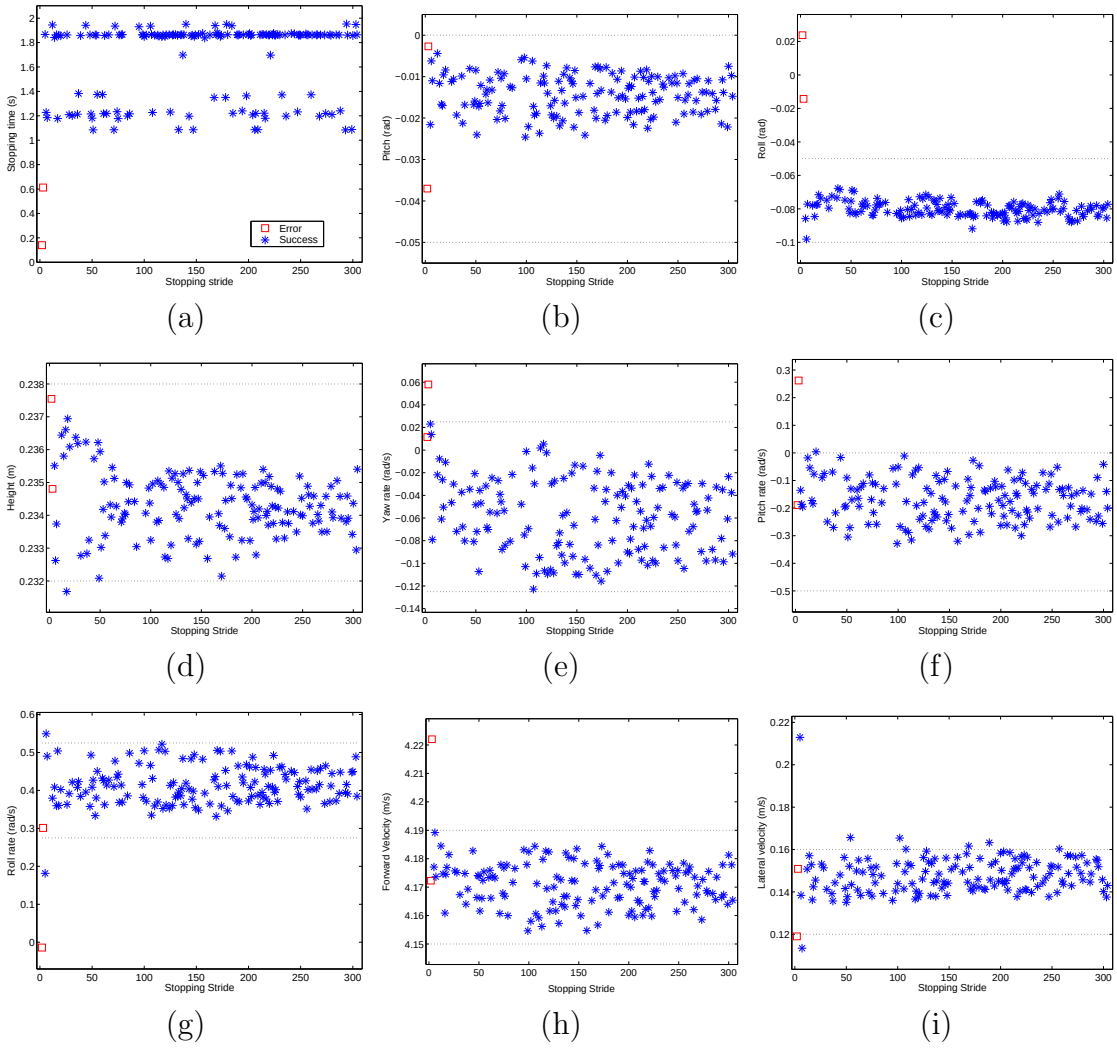
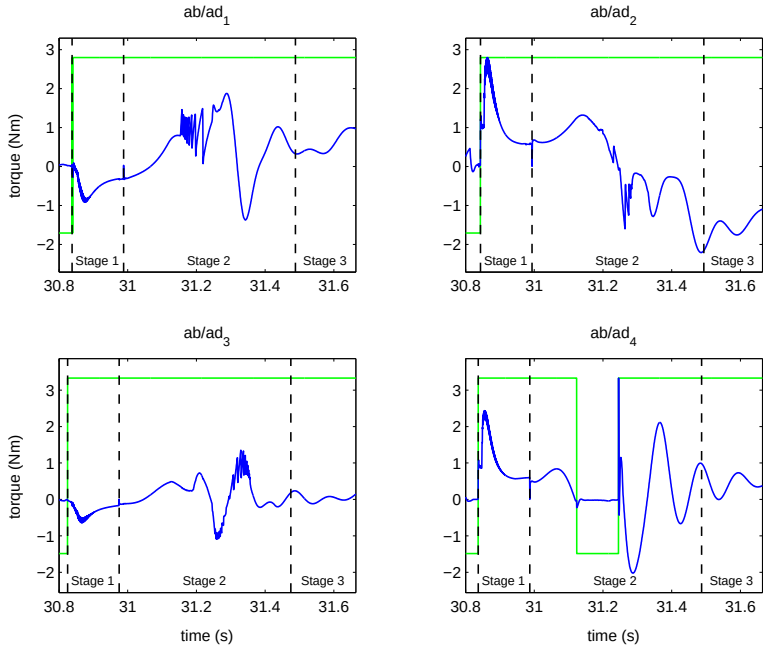


Figure 6.11: Results of testing the sudden stop solution from random galloping strides during the first 60 s of galloping: (a) Stopping time, (b) pitch, (c) roll, (d) height, (e) yaw rate, (f) pitch rate, (g) roll rate, (h) forward velocity, and (i) lateral velocity. Body state variables in (b) through (i) represent the TOF conditions prior to the stopping maneuver. Dotted lines indicate the initialization range for each variable, given in Table 6.5, and squares indicate error conditions (unsuccessful stops).

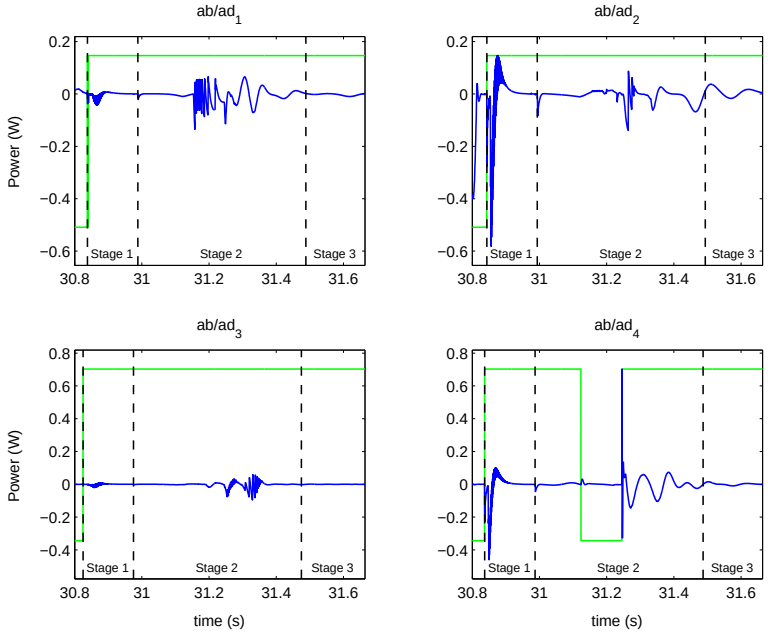
6.3.5 Torque and Power Analysis for the Sudden Stop

In this section, the torque and power requirements for each joint, and in total, are presented. The torque and power for the ab/ad axes are shown in Fig. 6.12. Generally both the torque and power levels for the ab/ad joints are less in the sudden stop than for steady-state galloping (Fig. 4.11). Since the same ab/ad PD gains are used for both the GALLOP-STANCE-CONTROL and GALLOP-STOP-CONTROL functions, the reason must be the stability of the stance posture. During the sudden stop, a symmetric, sprawled posture is employed. The ab/ad angles in Table 6.8 are 1.82 and 1.31 rad for the left and right side, respectively, which produces an approximately even outward rotation of the ab/ad joints on each side of about 0.26 rad (14.9 deg) from the perpendicular position. This symmetric, sprawled stance is much easier to maintain given sufficient surface friction and largely forward motion of the body as it slides to a halt. In the gallop, however, an asymmetric stance is adopted (Table 4.6) in an effort to combat the yaw and roll moments produced during the transverse footfall sequence. Consequently, more ab/ad torque is required on average.

The torque and power for the hip joints are shown in Fig. 6.13. Significant chatter is present during the first stage of GALLOP-STOP-CONTROL in the hip joints, which is indicated by the solid portion of the torque curves in Fig. 6.13 (a). This chatter occurs due to sensitivity of the system with respect to knee damping during periods of light loading of the knee joints. Because of the stiff spring constants of the knees and in the contact model, significant chatter can develop as an artifact of the simulation. As discussed in Sect. 6.3.1, damping is restricted to periods of knee spring compression to prevent this reaction from becoming excessive. However, it was not possible to completely eliminate the chatter during periods shortly after touchdown.

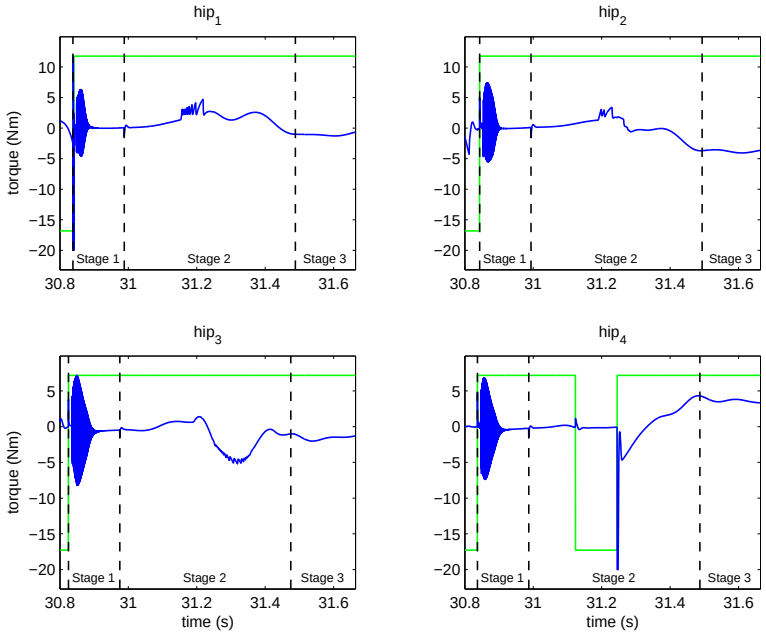


(a)

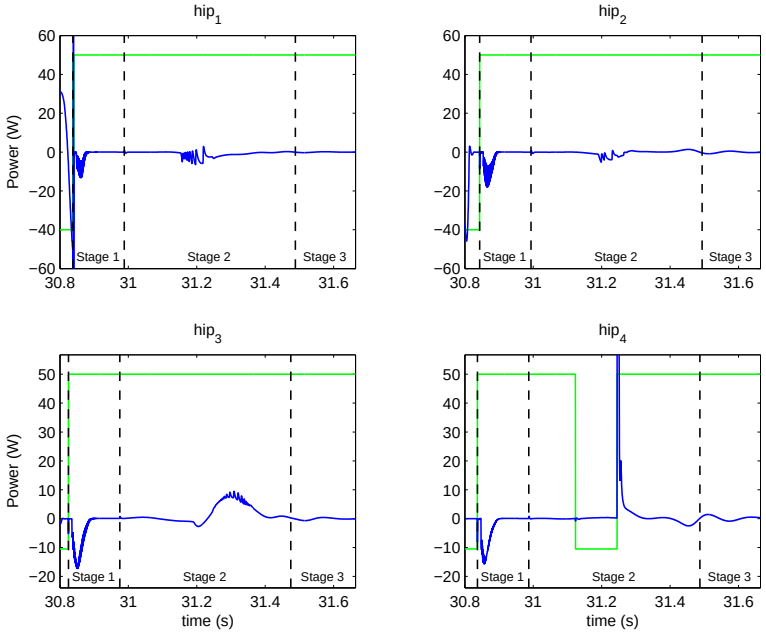


(b)

Figure 6.12: Torque and power requirements for the ab/ad axes during the sudden stop. Square waveforms indicate leg contact, and stages for the GALLOP-STOP-CONTROL function are labeled.



(a)



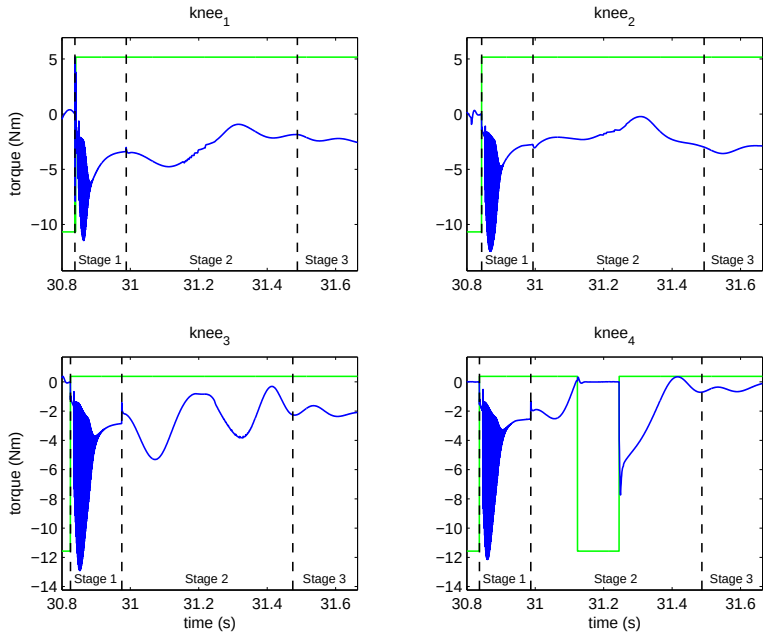
(b)

Figure 6.13: Torque and power requirements for the hip axes during the sudden stop. Square waveforms indicate leg contact, and stages for the GALLOP-STOP-CONTROL function are labeled.

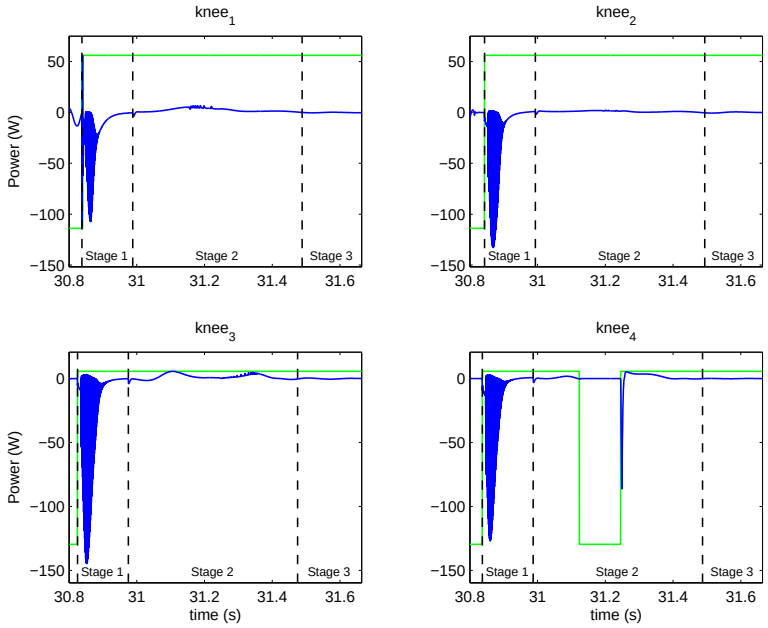
A different contact model with nonlinear damping, such as the one used in [101], may improve the response of the system during periods of light loading.

During the period of chatter, the hip power remains negative, which indicates that the actuators absorb the energy of the system on average. During stage 2, when the legs are transferred to their final positions, positive power is required, especially in the rear legs, since the final virtual leg angles are not the same as the touchdown angles. Note that leg 4 actually breaks contact briefly during the second stage, which causes a spike in torque and power when it makes contact again. During the last stage, the torque and power oscillate as the system settles to rest. Much of this oscillation is due to the passive response of the knee springs.

The torque and power for the knee joints (series configuration, Sect. 4.4.3) are shown in Fig. 6.14. As stated above, significant chatter is present during the early stance phases due to chattering of the contact points at the ends of the shank. However, it is currently unclear whether a sliding-style stop would be possible without the presence of this chatter. Nevertheless, in the current solution, the knees are able to effectively absorb the bulk of the energy during the first portion of the stance phase. Negative power levels in the knees are, on average, an order of magnitude larger than those in the hips. The ability to dissipate the body's kinetic energy is closely related to the evolved damping gains, which are large relative to the effective compliance of the knees, as discussed in Sect. 6.3.4. The use of the knees to absorb much of the kinetic energy of the system is reasonable based on the significant knee flexion seen during biological stops and landings. However, an interesting question is whether improved stopping ability could be achieved in a model with additional passive joints in the legs, assuming the same surface friction coefficients.



(a)



(b)

Figure 6.14: Torque and power requirements for the knee axes during the sudden stop. A series configuration is assumed, as described in Sect. 4.4.3. Square waveforms indicate leg contact, and stages for the GALLOP-STOP-CONTROL function are labeled.

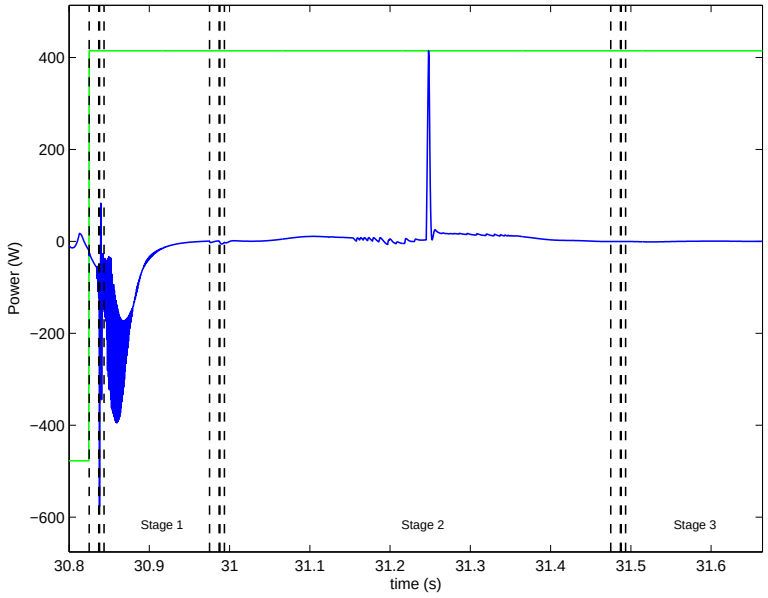


Figure 6.15: Total power required for all joints for the sudden stop. The square waveform indicates that one or more legs are in contact, and the stage transitions for each leg are shown.

The total power for all joints is shown in Fig. 6.15, again, assuming that negative power offsets positive power. As expected, the power is largely negative during the early portion of the stance phase, where the negative power of the knee joints dominate the response. One interesting observation is that the absorption of power from the system occurs during a relatively brief period of time of less than 0.1 s. However, the forward velocity of the body declines more gradually over about 0.5 s (Fig. 6.9), which implies that the remaining kinetic energy must be absorbed externally by the ground contact. Once the body starts sliding, there is little that can be done to actively decelerate. This is the basic motivation behind anti-lock brakes: If sliding can be reduced, maximum braking can be achieved. The same may be true for this model, which suggests that increased surface friction may allow for shorter stops. However,

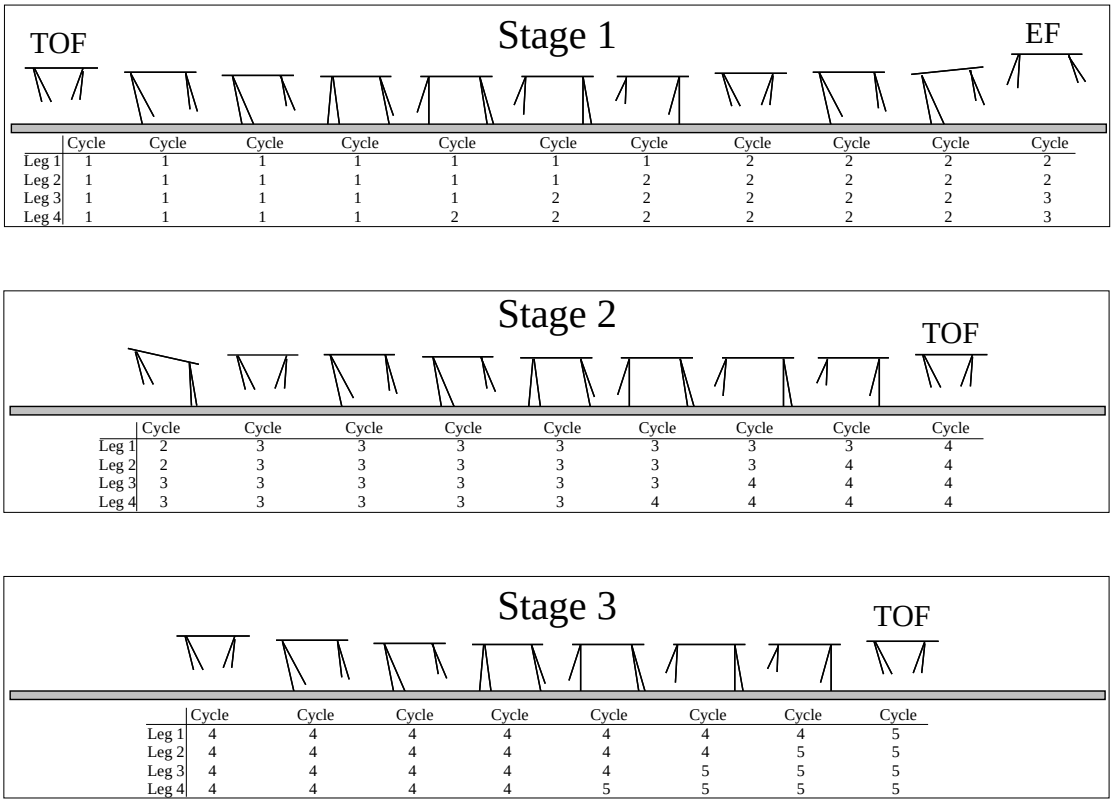
the stopping ability would eventually saturate based on the system's compliance, damping gains, and geometry. Further study of the stopping ability of the system would be required to determine this upper limit.

6.4 The Running Jump

The final maneuver is the running jump, which is executed during galloping. Like the jump-start, the running jump consists of multiple stages, as shown in Fig. 6.16. Stage 1 consists of the preparatory stride and the first half of the jump, which is terminated at the top-of-extended flight (EF). Stage 2 consists of the landing and the recovery stride, and Stage 3 is used to transition back to galloping. The state diagram for each leg is identical to that of the gallop (Fig. 4.1). However, leg parameters are referenced by *leg cycle* (LC), which indicates how many times the leg has executed the cycle of leg functions given in Fig. 4.1. Leg cycles are incremented after each leg completes a contact phase and are necessary because some stages (stages 1 and 2) contain more than one cycle of leg functions (e.g., either the front or the rear legs touch down twice during the stage). The parameter summary is given in Table 6.9, and the parameter ranges for each evolved parameter are given in Table 6.10. Each stage is evolved separately and discussed in the following sections.

6.4.1 Stage 1: Preparation and Jump

During the first stage of the running jump (Fig. 6.16), the quadruped executes a preparatory stride during which the front legs deliver a significant vertical thrust to lift the front part of the body off the ground. Following this, the rear legs touch down for the second time to provide the forward and upward acceleration needed to produce the jump. The stage is terminated when the body reaches the apex of



	Cycle	Cycle	Cycle	Cycle	Cycle	Cycle	Cycle	Cycle	Cycle	Cycle	Cycle
Leg 1	1	1	1	1	1	1	1	2	2	2	2
Leg 2	1	1	1	1	1	1	2	2	2	2	2
Leg 3	1	1	1	1	1	2	2	2	2	2	3
Leg 4	1	1	1	1	2	2	2	2	2	2	3

	Cycle	Cycle	Cycle	Cycle	Cycle	Cycle	Cycle	Cycle	Cycle
Leg 1	2	3	3	3	3	3	3	3	4
Leg 2	2	3	3	3	3	3	3	4	4
Leg 3	3	3	3	3	3	3	4	4	4
Leg 4	3	3	3	3	3	4	4	4	4

	Cycle	Cycle	Cycle	Cycle	Cycle	Cycle	Cycle	Cycle
Leg 1	4	4	4	4	4	4	4	5
Leg 2	4	4	4	4	4	4	5	5
Leg 3	4	4	4	4	4	5	5	5
Leg 4	4	4	4	4	5	5	5	5

Figure 6.16: Stages for the running jump. Stage 1 consists of the preparatory stride and the beginning of the jump itself, which terminates in EF (top-of-extended flight). Stage 2 consists of the landing and the recovery stride, and stage 3 is the transition stride back to galloping. Leg cycles, indicated beneath each figure, indicate how many times each leg has performed the cycle of leg functions given in Fig. 4.1 and are incremented after each leg completes a contact phase. The jump is modeled after photographic plates of horses jumping [102].

Function	Parameters	Description
TRANSFER	$\sigma_{a_e}=0,\quad \sigma_{h_e}=\begin{cases}1 & \text{if } \mathrm{LC}_i=2\\ 0 & \text{otherwise,}\end{cases}$ $\sigma_{k_d}=1,\quad \sigma_s=1,$ $T_r=0.0,\quad \boldsymbol{\theta}_{l_f}=[\theta_{a_{i,\mathrm{LC}_i}},\theta_{v_{i,\mathrm{LC}_i}},\frac{l_0}{2}]^T,$ $T_t=1.0,\quad T=\begin{cases}T^* & \text{if } i\leq 2,\ \mathrm{LC}_i=2\\ T^* & \text{if } i>2,\ \mathrm{LC}_i=3\\ 0.1\,\mathrm{s} & \text{otherwise.}\end{cases}$	Transfer using ab/ad angle $\theta_{a_{i,\mathrm{LC}_i}}$: $\theta_{a_{i,\mathrm{LC}_i}}=\begin{cases}1.65\text{ rad} & \text{if } i\in[1,3]\,,\ \mathrm{LC}_i>3,\\ 1.55\text{ rad} & \text{if } i\in[2,4]\,,\ \mathrm{LC}_i>3,\\ \frac{\pi}{2}\text{ rad} & \text{otherwise,}\end{cases}$ where LC_i is the leg cycle for leg i. Use virtual leg angle $\theta_{v_{i,\mathrm{LC}_i}}$ as follows: $\theta_{v_{i,\mathrm{LC}_i}}=\begin{cases}\theta^*_{v_{f,\mathrm{LC}_i}} & \text{if } i\leq 2,\mathrm{LC}_i\in[2,\,3]\,,\\ \theta^*_{v_{r,\mathrm{LC}_i}} & \text{if } i>2,\mathrm{LC}_i\in[2,\,3]\,,\\ 0.69\text{ rad} & \text{if } i\leq 2,\ \mathrm{LC}_i\notin[2,\,3]\,,\\ 0.65\text{ rad} & \text{if } i>2,\ \mathrm{LC}_i\notin[2,\,3]\,,\end{cases}$ where $\theta^*_{v_{f,\mathrm{LC}_i}}$ and $\theta^*_{v_{r,\mathrm{LC}_i}}$ are evolved values. Use evolved period T^* for the front legs, cycle 2, and for the rear legs, cycle 3.
WAIT-FOR-TRIGGER	$L=L_i,\qquad R=\begin{cases}3 & \text{if } i=1,\ \mathrm{LC}_i=2\\ R_i & \text{otherwise.}\end{cases}$ $h_t=0.24\,\mathrm{m},\quad \delta=\begin{cases}1.5\,\mathrm{ms} & \text{if } i=1,\ \mathrm{LC}_i=2,\\ \delta_i & \text{otherwise.}\end{cases}$ $\sigma_{a_e}=0,\quad \sigma_{h_e}=\begin{cases}1 & \text{if } \mathrm{LC}_i=2\\ 0 & \text{otherwise.}\end{cases}$ $\sigma_{k_f}=0,\quad \sigma_s=0,$ $T_r=0.0,\quad \boldsymbol{\theta}_{l_d}=[\theta_{a_{i,\mathrm{LC}_i}},\theta_{v_{i,\mathrm{LC}_i}},l_0]^T,$ $T=1.0\,\mathrm{s}$	Hold at ab/ad and virtual leg angles $\theta_{a_{i,\mathrm{LC}_i}}$ and $\theta_{v_{i,\mathrm{LC}_i}}$ (above) for leg i, $i=1,\ldots,4$ and leg cycle LC_i. Trigger parameters L_i, R_i, and δ_i are given in Table 2.6, except for $i=1$, $LC_i=2$, where $R=3$ (ER-DELAY, Table 2.6) and $\delta=1.5\,\mathrm{ms}$.
EARLY-RETRACTION	$\sigma_{a_e}=0,\qquad \theta_a=\theta_{a_{i,\mathrm{LC}_i}},$ $v_d=4.15\,\mathrm{m/s},\quad T=1.0\,\mathrm{s}$	Hold ab/ad joint at $\theta_{a_{i,\mathrm{LC}_i}}$, given in TRANSFER (above). Rotate the hip rearward to achieve tangential foot velocity v_d.
GALLOP-STANCE-CONTROL	$v_d=4.15\,\frac{m}{s},$ $v_b=\begin{cases}v^*_{b_{f,\mathrm{LC}_i}} & \text{if } i\leq 2,\ \mathrm{LC}_i<4,\\ v^*_{b_{r,\mathrm{LC}_i}} & \text{if } i>2,\ \mathrm{LC}_i\in[2,\,3]\,,\\ -0.23\,\frac{m}{s} & \text{if } i\leq 2,\ \mathrm{LC}_i\geq 4,\\ 0.52\,\frac{m}{s} & \text{if } i>2,\ \mathrm{LC}_i\notin[2,\,3]\,.\end{cases}$ $E_d=E^*_{i,\mathrm{LC}_i},$ $\sigma_{\text{slip}}=1,$ $T=1.0\,\mathrm{s}$	Use velocity control with evolved front and rear velocity biases $v^*_{b_{f,\mathrm{LC}_i}}$ and $v^*_{b_{r,\mathrm{LC}_i}}$; use evolved target energy value E^*_{i,LC_i} for leg i, $i=1,\ldots,4$, and leg cycle LC_i. Use slip control.
FREE	$\sigma_{a_f}=0,\qquad \sigma_s=1,$ $T_r=0.75,\quad T=0.025\,\mathrm{s}$	Reset knee spring to the nominal position over 75% of period T while holding the ab/ad angle and allowing the hip to rotate freely.

Table 6.9: Summary of the leg function parameters for the running jump. LC_i denotes the leg cycle for leg i.

Parameter	**Cycle** (LC_i)	**Range**	
T^*	2-3	[0.2, 0.4]	s
$\theta^*_{v_{f,\mathrm{LC}_i}}, \theta^*_{v_{r,\mathrm{LC}_i}}$	2,3	[0.52, 0.87]	rad
$v^*_{b_{f,\mathrm{LC}_i}}$, $v^*_{b_{r,\mathrm{LC}_i}}$	1-3	[−10.0, 10.0]	m/s
E^*_{1,LC_i},E^*_{2,LC_i}	1	[0.0, 30.0]	J
E^*_{3,LC_i},E^*_{4,LC_i}	1	[2.0, 5.0]	J
$E^*_{i,\mathrm{LC}_i}, i = 1, \ldots, 4$	≥ 2	[0.0, 30.0]	J

Table 6.10: Evolved parameter ranges for the running jump.

the flight phase, which is considered extended flight (EF), since the rear legs have contacted the ground most recently.

The evolved parameters for the first stage include the velocity bias for the front legs prior to their first contact phase ($v^*_{b_{f,1}}$), the target energy amounts for all legs during the first contact phase ($E^*_{i,1}$, $i = 1, \ldots, 4$), the rear leg virtual leg angle used before their second contact phase ($\theta^*_{v_{r,2}}$), the rear leg velocity bias during their second contact phase ($v^*_{b_{r,2}}$), and the two target energy amounts for the rear legs during their second contact phase ($E^*_{i,2}$, $i > 2$). The front leg velocity bias $v^*_{b_{f_1}}$ and front target energy amounts $E^*_{i_1}$, $i \leq 2$, are critical in generating a negative pitch moment and significant vertical thrust, respectively, to lift the front part of the body off the ground. Once the front legs break contact, the evolved transfer time T^* is used to determine the length of time to transfer the front legs during the leap.

On the other hand, the rear legs are not critical to the jump during their first cycle. Consequently, their energy amounts $E^*_{3,1}$ and $E^*_{4,1}$ during their first contact phase are limited by restricting the parameter range to relatively small values (Table 6.10).

This keeps the rear leg energy values close to their average values during steady-state galloping. During the second cycle, however, the rear legs are much more important. For this cycle, the target virtual leg angle $\theta^*_{v_{r,2}}$ is evolved, and the inertial control flag for the TRANSFER function σ_{h_e} is set to 1 (Table 6.9). Inertial control is necessary because the body will generally be pitching nose-up at a significant rate due to the action of the front legs during their first contact cycle. Consequently, controlling the rear leg virtual leg angle with respect to the body would introduce significant variation in this angle before touchdown. When the rear legs touch down after this flight phase, the evolved velocity bias $v^*_{b_{r,2}}$ and target energy amounts $E^*_{3,2}$ and $E^*_{4,2}$ generate the vertical and forward thrust necessary to launch the body into the leap. During the subsequent flight phase, the rear legs are transferred to new positions (evolved during Stage 2) using the transfer time T^*.

The fitness vector for the first stage of the running jump includes three elements, as follows:

$$\boldsymbol{f}_{\mathrm{R}_1} = \left[f_a^{\mathrm{R}_1},\, f_v^{\mathrm{R}_1},\, f_p^{\mathrm{R}_1} \right]^T , \tag{6.21}$$

where $f_a^{\mathrm{R}_1}$ is an accuracy factor, $f_v^{\mathrm{R}_1}$ is the velocity factor, and $f_p^{\mathrm{R}_1}$ is a factor for pitch rate. The accuracy factor $f_a^{\mathrm{R}_1}$ is computed as follows:

$$f_a^{\mathrm{R}_1} = f_h^{\mathrm{J}_1} \left(\frac{w_a^{\mathrm{R}_1}}{1 + \| \boldsymbol{\Lambda}_{\mathrm{R}_1} (\boldsymbol{\theta}_b - \boldsymbol{\theta}_{b_d}) \|^2} \right) , \tag{6.22}$$

where $w_a^{\mathrm{R}_1}$ is a scaling factor of 100.0, the diagonal elements of $\boldsymbol{\Lambda}_{\mathrm{R}_1}$ are given in Table 6.11, and $\boldsymbol{\theta}_{b_d} = [\alpha_0,\, 0.0,\, \ldots,\, 0.0]^T$. The desired yaw angle in $\boldsymbol{\theta}_{b_d}$ is set to the initial yaw angle α_0 to maintain the initial heading during the jump. The remaining elements in $\boldsymbol{\theta}_{b_d}$ are set to 0.0, including ${}^e z_{b_d}$, $\dot{\beta}_d$, and ${}^{e'}\dot{x}_{b_d}$, since these three components are addressed by other elements of the fitness function. Consequently, $\lambda_z = \lambda_{\dot{\beta}} =$

Parameter	Value	Parameter	Value	Parameter	Value
λ_α	28.6	λ_y	0.0	$\lambda_{\dot{\gamma}}$	5.7
λ_β	11.5	λ_z	0.0	$\lambda_{\dot{x}}$	0.0
λ_γ	28.6	$\lambda_{\dot{\alpha}}$	5.7	$\lambda_{\dot{y}}$	10.0
λ_x	0.0	$\lambda_{\dot{\beta}}$	0.0	$\lambda_{\dot{z}}$	0.0

Table 6.11: Scaling factors used to compute the accuracy fitness component for the first stage of the running jump. Scaling factors of zero indicate "don't-care" conditions.

$\lambda_{\dot{x}} = 0.0$ in Table 6.11. The height fitness factor f_h^{J1} in Eq. (6.22) is computed using Eq. (6.2) with $h_d = 0.36\,\mathrm{m}$. The value of h_d was selected based on photographic plates of horses jumping [102].

The second element of the fitness vector f_v^{R1} is the velocity factor and is computed using Eq. (6.11). Finally, the third element of the fitness vector f_p^{R1} is the pitch rate factor, computed using Eq. (6.6) with $t_{\dot{\beta}} = 2.0\,\mathrm{rad/s}$, so that the acceptable pitch range is widened to $[1.0, 5.0]\,\mathrm{rad/s}$. A wider pitch range than the one used for the first stage of the jump start (Sect. 6.2.1) is necessary because the larger desired jumping height results in larger TOF pitch rates.

6.4.2 Stage 2: Landing and Recovery

In Fig. 6.16, Stage 2 consists of the landing and recovery stride. During landing, the front feet touch down, and during recovery, the rear legs, then the front, touch down. Evolved parameters for the second stage include the front virtual leg angle $\theta_{f,2}^*$, front velocity bias $v_{b_{f,2}}^*$, and the front leg target energy amounts $E_{1,2}^*$ and $E_{2,2}^*$, where all four parameters are for the second leg cycle, when the front legs touch down during the landing. These parameters are responsible for absorbing the impact of landing

and correcting the body's pitch motion so that it comes down properly on the rear legs. During the preparation for touchdown, the front legs are controlled inertially ($\sigma_{h_e} = 1$ in TRANSFER, Table 6.9) to compensate for the significant pitch motion of the body. Furthermore, leg 1, the trailing front leg, is triggered at approximately the same time as leg 2, the leading front leg ($R = 3$ (ER-DELAY), $\delta = 1.5\,\text{ms}$, for $i = 1$ and $\text{LC}_i = 2$ in WAIT-FOR-TRIGGER, Table 6.9). This causes both front legs to touch down almost simultaneously, which stabilizes the roll motion of the body during landing.

After the front legs make contact, the rear legs touch down. The evolved rear virtual leg angle $\theta^*_{r,3}$, rear velocity bias $v^*_{b_{r,3}}$, and target rear energy amounts $E^*_{3,3}$, and $E^*_{4,3}$ comprise the beginning of the recovery stride, wherein the body's vertical and rotational kinetic energy must be converted back to forward and rotational kinetic energy. Finally, the evolved front virtual leg angle $\theta^*_{f,3}$, front velocity bias $v^*_{b_{f,3}}$, and target energies $E^*_{1,3}$ and $E^*_{2,3}$ are used during cycle 3 to complete the recovery stride[37].

The fitness function for the second stage of the running jump is given as follows:

$$\boldsymbol{f}_{\text{R}_2} = [f_a^{\text{R}_2},\, f_{\text{lc}}^{\text{R}_2}]^T\,, \tag{6.23}$$

where $f_a^{\text{R}_2}$ is an accuracy factor and $f_{\text{lc}}^{\text{R}_2}$ is a leg contact factor. The accuracy factor is computed as follows:

$$f_a^{\text{R}_2} = \frac{w_a^{\text{R}_2}}{1 + \|\boldsymbol{\Lambda}\,(\boldsymbol{\theta}_b - \boldsymbol{\theta}_{b_d})\,\|^2}\,, \tag{6.24}$$

[37]Note that $\theta^*_{v_{f,3}}$ and $\theta^*_{v_{r,3}}$ are used as temporary target angles in the beginning of cycle 4 (Fig. 6.16) since $\theta^*_{v_{f,4}}$ and $\theta^*_{v_{r,4}}$ are not evolved until Stage 3.

where $w_a^{\mathrm{R2}} = 100.0$, and $\boldsymbol{\theta}_{b_d}$ is given as

$$\begin{aligned}\boldsymbol{\theta}_{b_d} &= \left[\ \alpha_d,\, \beta_d,\, \gamma_d,\, {}^{e}x_{b_d},\, {}^{e}y_{b_d},\, {}^{e}z_{b_d},\, \dot{\alpha}_d,\ \ \dot{\beta}_d,\ \ \dot{\gamma}_d,\, {}^{e'}\dot{x}_{b_d},\, {}^{e'}\dot{y}_{b_d},\, {}^{e}\dot{z}_{b_d}\ \right]^T \\ &= \left[\ \alpha_0,\ 0,\ \ 0,\ \ \ 0,\ \ \ \ 0,\ \ \ l_0,\ \ 0,\ \dot{\beta}_{\mathrm{ave}},\ 0,\ \ v_d,\ \ \ \ 0,\ \ \ \ 0\ \right]^T . \end{aligned} \tag{6.25}$$

The desired yaw angle α_d in Eq. (6.25) is set to the initial yaw angle α_0, and the desired pitch rate is set to $\dot{\beta}_{\mathrm{ave}} = -0.15\,\mathrm{rad/s}$, the average pitch rate for galloping. The scaling factors in $\boldsymbol{\Lambda}$ are identical to those used for the gallop and are given in Table 4.5.

The leg contact fitness criterion in Eq. (6.23) is computed as follows:

$$f_{\mathrm{lc}}^{\mathrm{R2}} = \frac{w_{\mathrm{lc}}^{\mathrm{R2}}}{1 + \sum_{i=1}^{4} (c_i - c_{d_i})}\,, \tag{6.26}$$

where $w_{\mathrm{lc}}^{\mathrm{R2}} = 100.0$, c_i is the number of contact periods for leg i during stage 2, and c_{d_i} is the desired number of contact periods for leg i, given as

$$c_{d_i} = \begin{cases} 2 & \text{if } i \leq 2\,, \\ 1 & \text{otherwise.} \end{cases} \tag{6.27}$$

The leg contact fitness factor $f_{\mathrm{lc}}^{\mathrm{R2}}$ generates maximum fitness for individuals with two contacts for each front leg and one contact for each rear during stage 2.

6.4.3 Stage 3: Transition to Galloping

The third stage of the running jump consists of a normal stride (with a single contact per leg) and serves as a transition back to galloping. Assuming a good solution has been found for the previous stage, Stage 3 requires only four evolved energy values $E_{i,4}^*$, $i = 1, \ldots, 4$. The velocity biases are set to the evolved values for the gallop (Table 4.6), while the ab/ad angles are set to average values achieved during steady-state galloping (See $\theta_{a_{i,\mathrm{LC}_i}}$, $\mathrm{LC}_i > 3$, in TRANSFER, Table 6.9.).

Like the jump-start transition stage, the fitness function for the running jump transition is identical to that of the gallop, since achieving a stable galloping solution is the ultimate goal. The fitness vector contains the accuracy, stability, and correctness criteria given in Eqs. (4.32) - (4.35).

6.4.4 Results for the Running Jump

Like the jump-start, ten trials were run for each stage of the running jump using a population size of $S = 32$ individuals. However, for stages 1 and 2 with more parameters, $N = 250$ generations were run, while stage 3 required only $N = 50$ generations. Multiple acceptable solutions were generated, although only one is presented here. The evolved parameter values for this solution are given in Table 6.12. The other solutions were similar, with minor variations in EF body state variables such as height, pitch rate, and forward velocity.

Results for the relevant body state variables are given in Fig. 6.17. During Stage 1, there are significant spikes in pitch and pitch rate (Fig. 6.17 (b) and (e), $t = 22.2\,\mathrm{s}$) as the front portion of the body is lifting off the ground. This is caused by the front leg thrust, indicated by the large impulse in the front leg contact forces (Fig. 6.18) during stage 1 at $t = 22.1\,\mathrm{s}$. The pitch and pitch rate (Fig. 6.17 (b) and (e)) reverse as the height (Fig. 6.17 (d)) increases during the remainder of stage 1. This occurs as the hindquarters lift off the ground, caused by thrusting of the rear legs, which is indicated by the spike in the rear vertical contact forces at $t = 22.2\,\mathrm{s}$ (Fig. 6.18).

During the jump, the velocity (Fig. 6.17 (f)) tends to increase due to the forward thrust of the front legs. Up until the running jump, the fore-aft component of the ground reaction force for the front legs is negative ("Fx" in Fig. 6.18, legs 1 and

Parameter	**Cycle** (LC_i)	**Value**	**Parameter**	**Cycle** (LC_i)	**Value**
T^*	2-3	0.25 s	E^*_{4,LC_i}	1	4.24 J
$\theta^*_{v_{f,\mathrm{LC}_i}}$	2	0.53 rad	E^*_{1,LC_i}	2	13.01 J
$\theta^*_{v_{r,\mathrm{LC}_i}}$	2	0.85 rad	E^*_{2,LC_i}	2	14.69 J
$\theta^*_{v_{f,\mathrm{LC}_i}}$	3	0.69 rad	E^*_{3,LC_i}	2	24.05 J
$\theta^*_{v_{r,\mathrm{LC}_i}}$	3	0.69 rad	E^*_{4,LC_i}	2	7.31 J
$v^*_{b_{f,\mathrm{LC}_i}}$	1	6.40 m/s	E^*_{1,LC_i}	3	1.51 J
$v^*_{b_{f,\mathrm{LC}_i}}$	2	0.63 m/s	E^*_{2,LC_i}	3	3.19 J
$v^*_{b_{r,\mathrm{LC}_i}}$	2	0.56 m/s	E^*_{3,LC_i}	3	1.07 J
$v^*_{b_{f,\mathrm{LC}_i}}$	3	-8.01 m/s	E^*_{4,LC_i}	3	0.00 J
$v^*_{b_{r,\mathrm{LC}_i}}$	3	-1.12 m/s	E^*_{1,LC_i}	4	1.89 J
E^*_{1,LC_i}	1	19.08 J	E^*_{2,LC_i}	4	7.09 J
E^*_{2,LC_i}	1	21.51 J	E^*_{3,LC_i}	4	13.78 J
E^*_{3,LC_i}	1	2.59 J	E^*_{4,LC_i}	4	1.20 J

Table 6.12: Evolved parameters for the running jump.

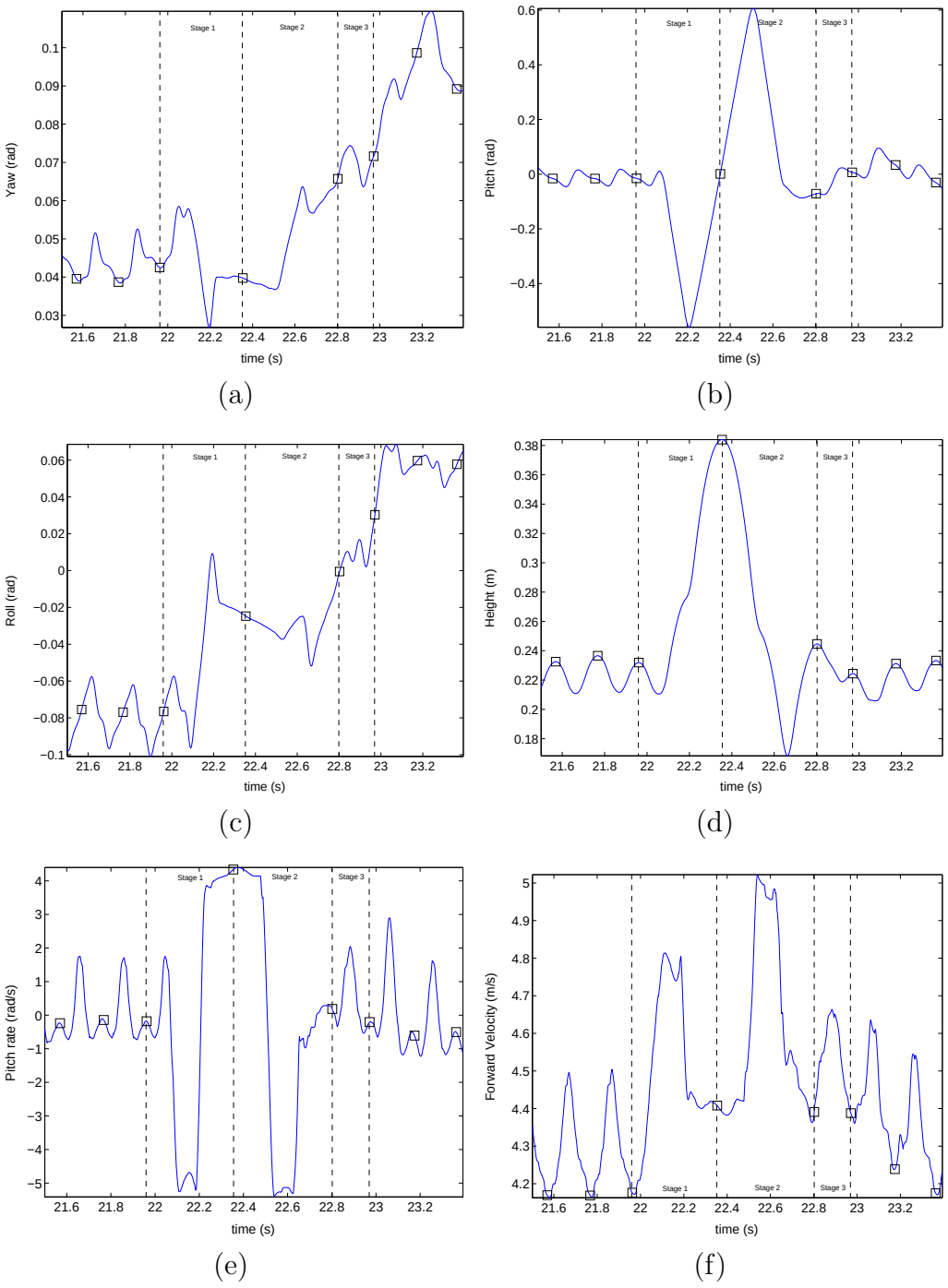


Figure 6.17: Results for the running jump: (a) Yaw, (b) pitch, (c) roll, (d) height, (e) pitch rate, and (f) forward velocity. Squares indicate top-of-gathered-flight (TOF) or top-of-extended-flight (EF). Stages are labeled in each plot.

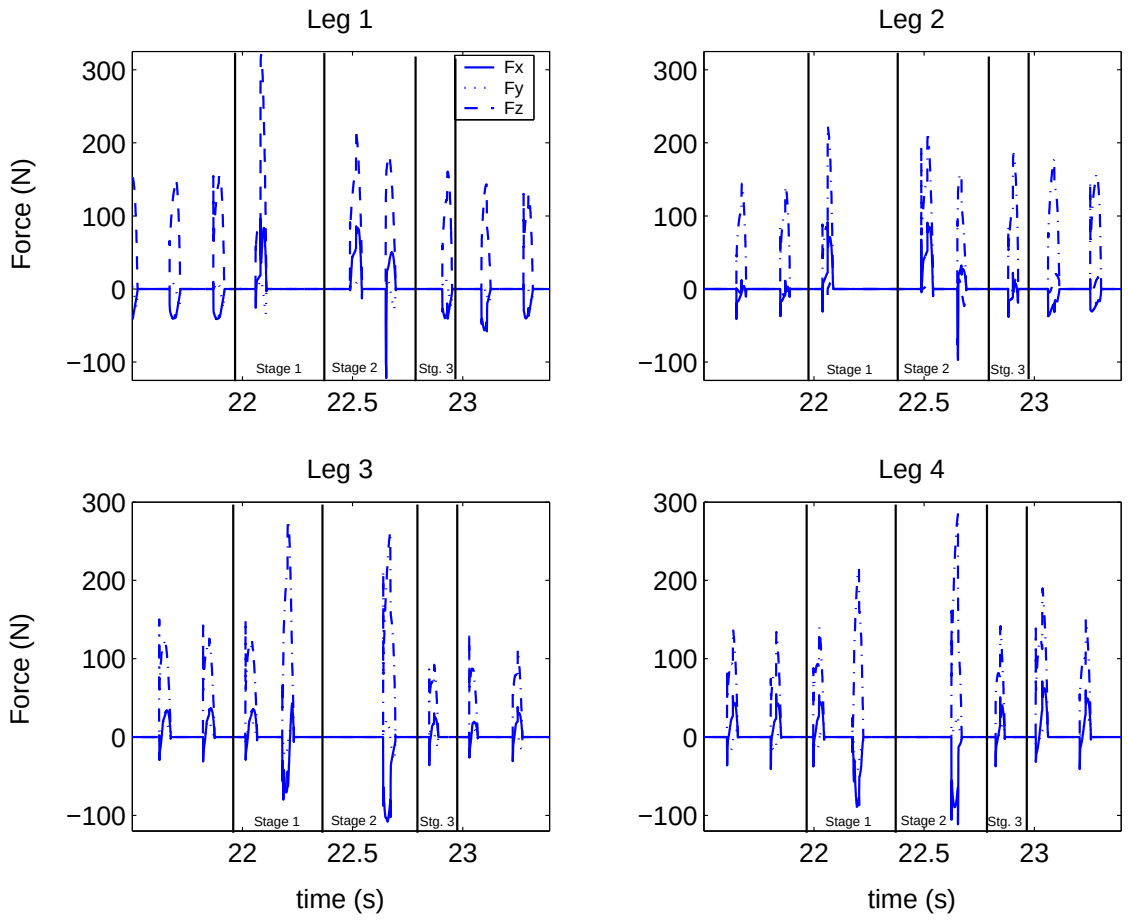


Figure 6.18: Foot forces for each leg for the running jump. Stages are labeled in each plot.

2, $t < 22.0\,\mathrm{s}$), due to the shoulder-braking action of the front legs during galloping (Sect. 4.2.5). During the first stage of the running jump, however, the front legs shift to shoulder *thrusting* ($v^*_{b_{f,1}} = 6.40\,\mathrm{m/s}$, Table 6.12), which causes the forward velocity to increase during Stage 1. More importantly, however, the positive fore-aft component of the front leg ground reaction forces ("Fx" in Fig. 6.18, legs 1 and 2, Stage 1) helps rotate the body upward for the leap by acting *in concert* with the front vertical ground reaction forces ("Fz", Fig. 6.18, legs 1 and 2, Stage 1). Both forces generate a pitch moment that rotates the body in the nose-up direction. This is exactly the opposite of what happens during galloping, when shoulder-braking generates negative fore-aft reaction forces in the front legs that *counter* the nose-up moment created by the vertical ground reaction forces.

In the hind legs, a similar behavior is observed, although the protracted leg angle ($\theta^*_{v_{r,2}} = 0.85\,\mathrm{rad}$, Table 6.12) contributes more to this effect during Stage 1 than the hip velocity bias ($v^*_{b_{r,2}} = 0.56\,\mathrm{rad/s}$, Table 6.12). The ground reaction forces in the fore-aft direction generally reverse during the second cycle of Stage 1 ("Fx" in Fig. 6.18, legs 3 and 4, $t = 22.2\,\mathrm{s}$), which effectively yields a hip-braking behavior. During Stage 2, the same thing occurs, but chiefly because of the negative velocity bias in the rear legs ($v^*_{b_{r,3}} = -1.12\,\mathrm{m/s}$, Table 6.12). These reversals of the fore-aft component of the ground reaction force in the rear legs *add* to the nose-down pitch moments created by the vertical ground reaction forces. Again, this is exactly the opposite of what happens during galloping. For the running jump, however, the rear legs must reverse the large nose-up pitch moment created by the front legs in Stage 1 in order to launch the body into flight. During Stage 2, the rear legs must counter the rapid downward rotation of the hindquarters during landing. The combination of

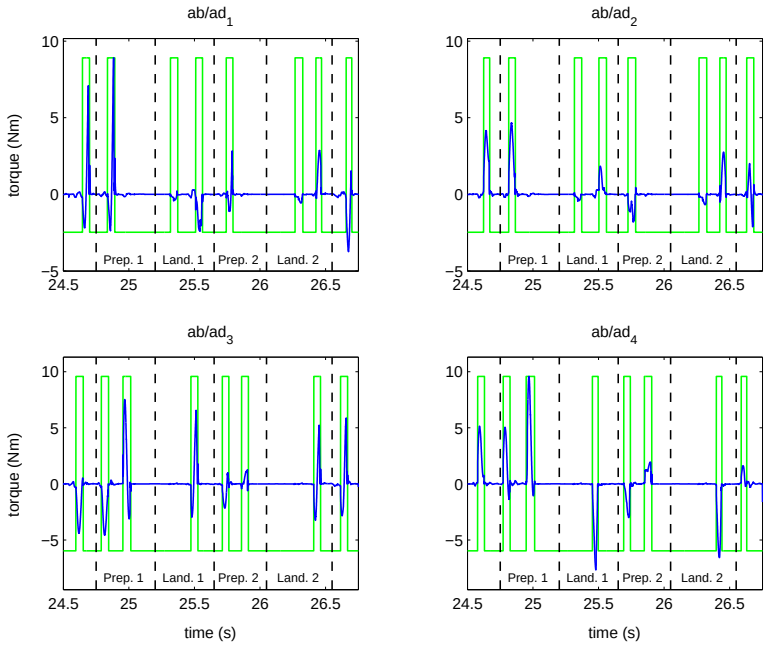
shoulder-thrusting and hip-braking create the necessary pitch moments to rotate the body upward for the leap and downward for the landing.

During landing, there is a spike first in the front legs at $t = 22.5\,\mathrm{s}$, then a larger impulse in the rear legs at $t = 22.65\,\mathrm{s}$, followed by another front leg impulse at $t = 22.66\,\mathrm{s}$. This pattern indicates that the rear legs actually absorb more force due to the downward rotation of the hindquarters during landing, as explained in Sect. 6.2.4. Finally, in all three stages for the running jump, there is noticeable asymmetry in vertical force levels between the left-side and right-side legs. This is necessary to compensate for the roll of the body during and after the jump (Fig. 6.17 (c)). Relatively small lateral forces are required to prevent large changes in yaw (Fig. 6.17 (a)).

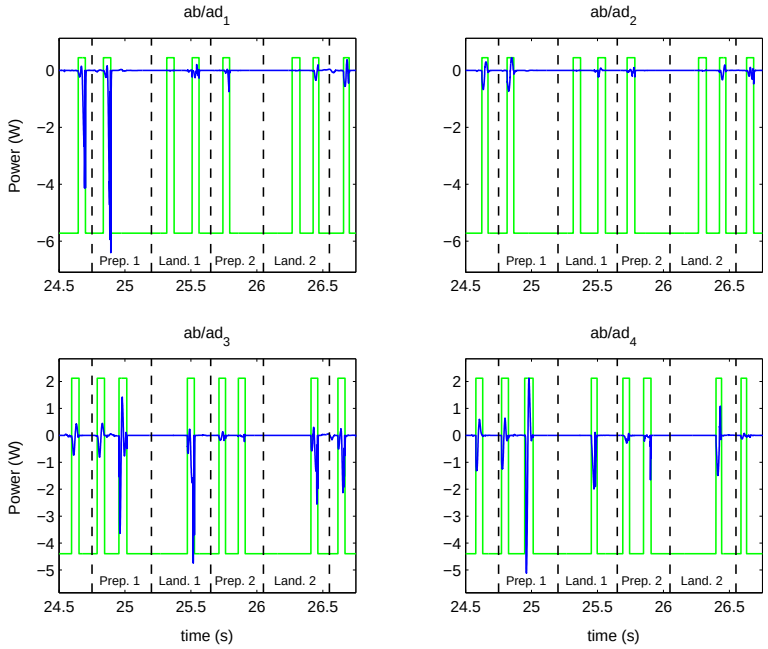
6.4.5 Torque and Power Analysis for the Running Jump

In this section, the torque and power analysis for the running jump is presented. The torque and power for the ab/ad joints are shown in Fig. 6.19. Torque and power levels spike in most legs during the last stance period before the first leap (first preparation stage, Fig. 6.19). This occurs because of the large knee thrusts that are exerted during take-off, discussed below, which generate larger moments about the body's roll axis. Consequently, larger ab/ad torque and power levels are necessary to stabilize the ab/ad joints. In general, however, it appears that less ab/ad torque is required during the remainder of the jumping maneuver than in galloping. This could be a result of enforcing the nominal ab/ad angles of $\frac{\pi}{2}\,\mathrm{rad}$ for all legs during the jump. Consequently, the knees become largely responsible for stabilizing roll and roll rate, both of which generally stay much closer to zero throughout the jump.

The torque and power for the hip joints are shown in Fig. 6.20. During the contact

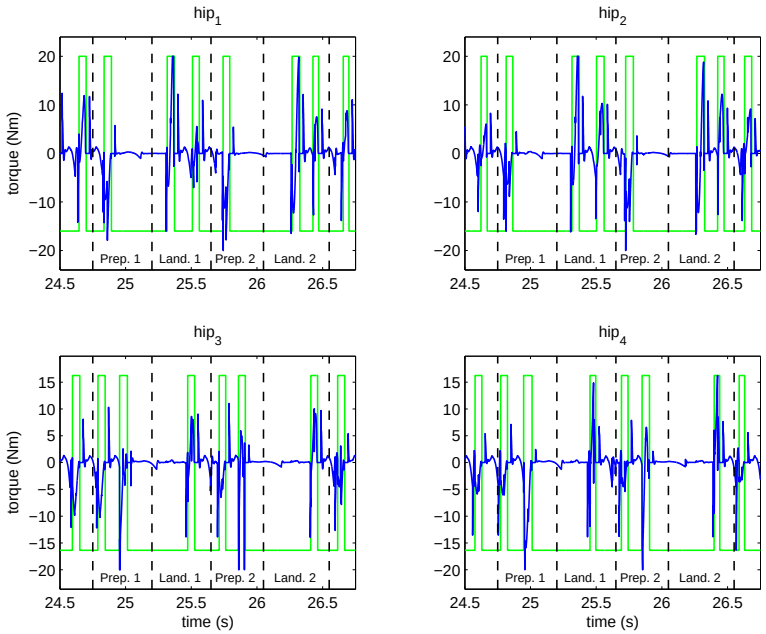


(a)

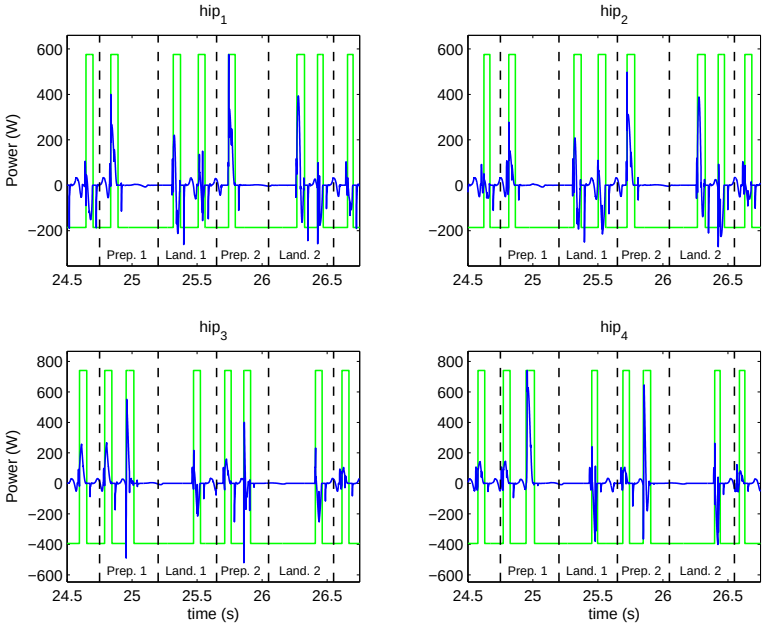


(b)

Figure 6.19: Torque and power requirements for the ab/ad axes during two successive running jumps. Square waveforms indicate leg contact, and the preparatory and landing stages of each jump are labeled.



(a)



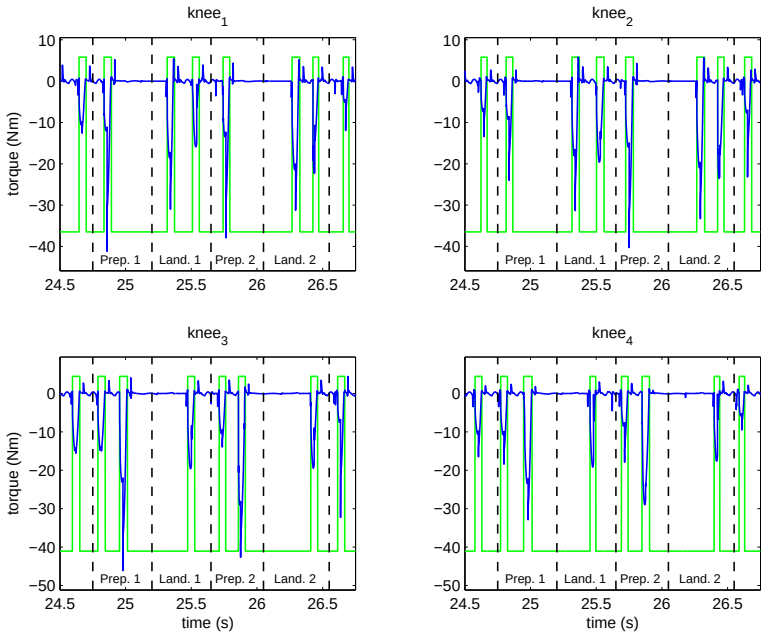
(b)

Figure 6.20: Torque and power requirements for the hip axes during two successive running jumps. Square waveforms indicate leg contact, and the preparatory and landing stages of each jump are labeled.

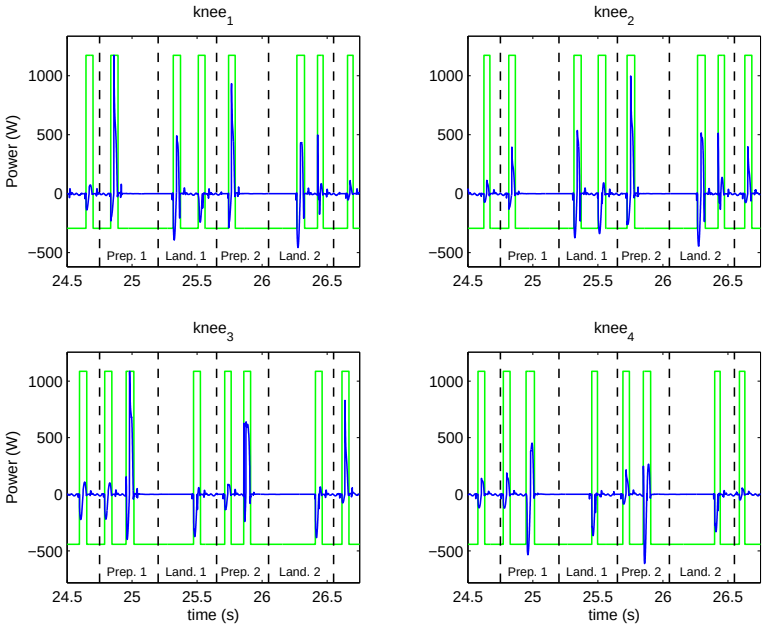
phase in the first preparatory stage, the front hip torques are negative, as both front legs thrust the body forward and upward into the leap. Consequently, significant positive power is required at the joints during this period. In the rear legs, there is even more driving torque and power required during the take-off stages. During landing, the front legs exhibit a shift between negative and positive power, which is precisely out-of-phase with that of the knees, discussed below. In the rear legs, the hip power is mostly negative during stance, which suggests that the rear hips are absorbing energy from the system.

The torque and power for the knee joints (series configuration, Sect. 4.4.3), are shown in Fig. 6.21. Significant negative (thrusting) torques are required in all knee joints during take-off, as expected. This generates peak positive power requirements in excess of 1000 W for some legs. Naturally, significant power absorption is required during landing, in order to absorb the excess rotational and linear kinetic energy of the system. As stated above, the knee power during landing is out-of-phase with the hip power, which suggests that the two joints fight each other during this period. However, the knee power is larger in magnitude throughout the stance phase, which means that the knee joints are chiefly responsible for absorbing and transforming the kinetic energy of the system during landing.

The total power for all joints (allowing regeneration) is shown in Fig. 6.22. During the take-off stance phases before each leap, the total positive power requirements exceed 1500 W. During the landing phases, significant power regeneration occurs to absorb the excess kinetic energy of the system. However, positive power during landing is also required in the hip and knee joints to redirect the energy of the system. The results shown here suggest that building a small quadruped capable of



(a)



(b)

Figure 6.21: Torque and power requirements for the knee axes during two successive running jumps. Square waveforms indicate leg contact, and the preparatory and landing stages of each jump are labeled. A series elastic configuration is assumed, as described in Sect. 4.4.3.

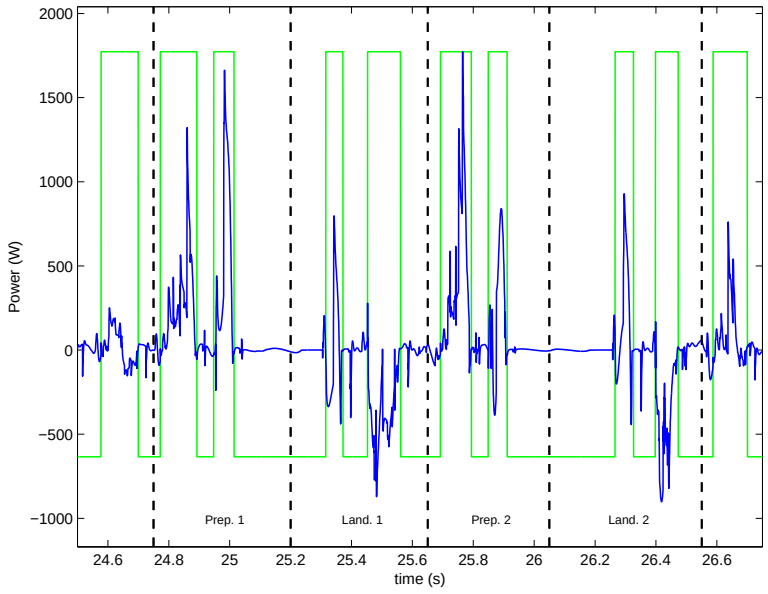


Figure 6.22: Total power required for all joints for two successive running jumps. The square waveform indicates that one or more legs are in contact, and the separate stages are labeled.

this type of performance would be difficult. In general, achieving over 1 kW of power in a single actuator would require a large, heavy DC motor. Although hydraulic actuators generally have better power density, designing a small, light-weight, stand-alone hydraulic system of sufficient power would still be a challenge. Furthermore, a power supply capable of handling the total power levels shown here would probably have to remain off-board due to its size and weight. For these reasons, the realization of an autonomous legged robot capable of the performance demonstrated here seems doubtful, at least until significant advances are made in actuation and power source technologies.

6.5 The Galloping and Maneuvering Sequence

After initial results were obtained for all of the desired behaviors, including galloping, turning, the jump-start, the sudden stop, and the running jump, a complete sequence of these behaviors was evolved as a final demonstration. Each behavior was evolved independently, as well as the transitions between behaviors, since developing a real-time controller capable of 3D maneuvering is beyond the scope of this work. However, the gallop solution described in Sect. 4.4 was not altered in any way, other than its initial conditions, which were evolved as part of the transition stages for the various maneuvers.

The sequence consists of the evolved behaviors given in Table 6.13. The final trajectory for the quadruped is shown in Fig. 6.23. Four multiple-stride turning sequences were included, two for each turning direction. The second pair of turning sequences were placed back-to-back with no intervening gallop to determine if a transition could be found from a CCW to a CW turn, where TOF roll, roll rate, and yaw rate would require a reversal of sign. Furthermore, three running jumps were included, with the last two placed back-to-back to determine if a single transition stride between the two jumps (i.e., the second running jump in Table 6.23 has only 2 stages instead of 3) would be sufficient to generate a higher, longer jump than the first. In both cases, successful transitions were found. The entire sequence covers approximately 135 m and takes 31.75 s of simulated time, yielding an average velocity of 4.25 m/s (about 9.66 body lengths/s). This provides an indication of the speed and the maneuvering capability of the quadrupedal model.

Behavior	**Description**	**Strides/Stages**
Jump-start	Leap into a gallop.	6
Gallop	Maintain heading along $\hat{\boldsymbol{x}}_e$.	27
Turn	Execute multiple-stride turn in the CCW (positive yaw) direction.	6
Gallop	Maintain approximate heading along $\hat{\boldsymbol{y}}_e$.	11
Turn	Execute multiple-stride turn in the CW direction.	7
Gallop	Maintain approximate heading along $\hat{\boldsymbol{x}}_e$.	11
Turn	Execute multiple-stride turn in the CCW direction.	4
Turn	Execute multiple-stride turn in the CW direction.	6
Gallop	Maintain approximate heading along $\hat{\boldsymbol{x}}_e$.	20
Running jump	Execute running jump with desired height $h_d = 0.36\,\text{m}$.	3
Gallop	Maintain approximate heading along $\hat{\boldsymbol{x}}_e$.	10
Running jump	Execute running jump with desired height $h_d = 0.42\,\text{m}$.	2
Running jump	Execute running jump with desired height $h_d = 0.48\,\text{m}$.	3
Gallop	Maintain approximate heading along $\hat{\boldsymbol{x}}_e$.	20
Sudden stop	Execute sudden stop and maintain approximate orientation pointing along $\hat{\boldsymbol{x}}_e$.	1

Table 6.13: Evolved behaviors for the final galloping and maneuvering sequence.

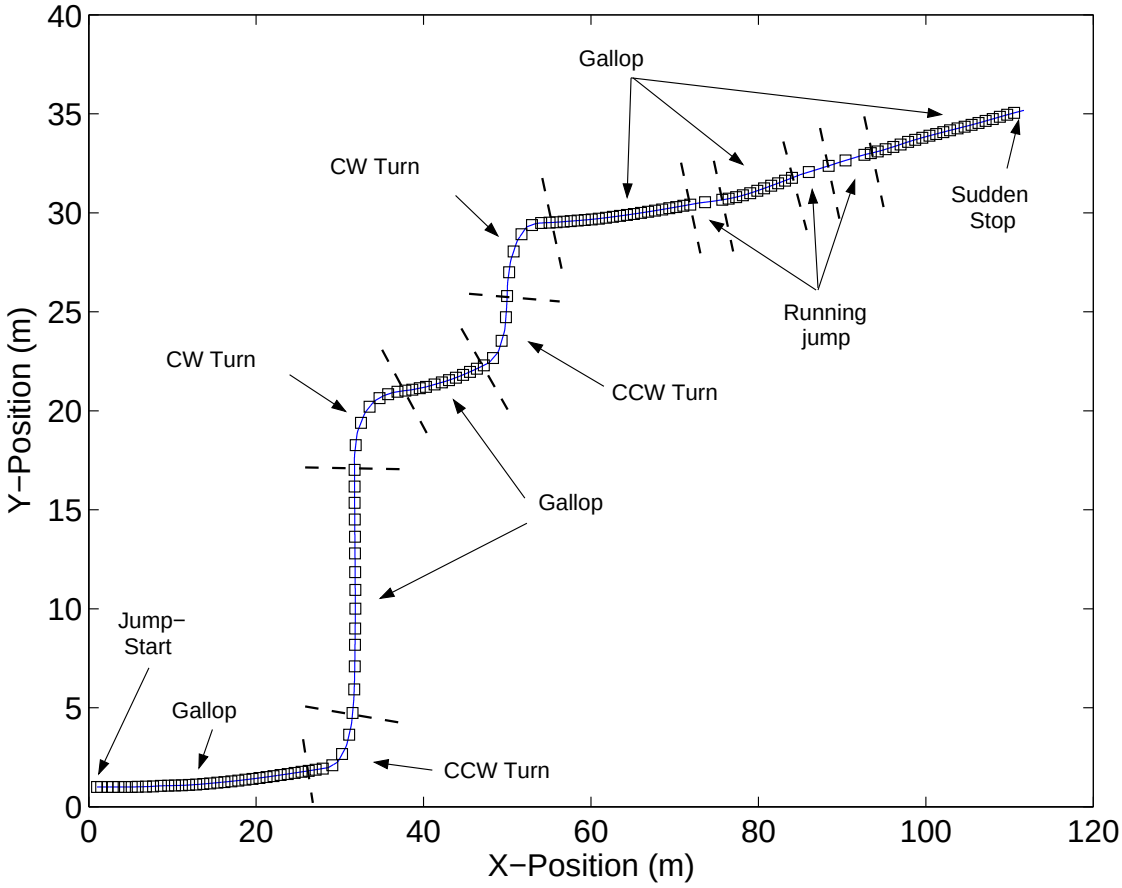


Figure 6.23: Trajectory for the final galloping and maneuvering sequence.

6.6 Summary and Conclusions

In this chapter, the jump-start, sudden stop, and running jump were presented. The jump-start is a multi-stage behavior that begins with the CROUCH2 coordinated function, followed by a THRUST function for each leg. The desired jumping height is kept low to minimize the total time, although doing so bounds the TOF velocity achievable during the first stride, as surface friction is limited. The front legs and rear legs are paired starting in the THRUST function so that the maneuver remains planar until the first stride of galloping. After the initial leap, multiple bounding strides are required to increase the body's forward velocity to the desired running speed. Parameters for each stage of the jump-start are evolved independently using separate fitness criteria.

Unlike the jump-start, the sudden stop is evolved as a single-stage maneuver and does not use the state diagram for the gallop. The main leg primitive function employed in the sudden stop is GALLOP-STOP-CONTROL, which is divided into several stages. However, the parameters for all stages are evolved simultaneously since the legs are generally in contact and there is no convenient flight-phase cutoff point. During the first stage of GALLOP-STOP-CONTROL, evolved PD gains for the hip and knee are used to dissipate the kinetic energy of the system during the contact phase. Generally, the effective proportional gains for both the hip and knee are evolved to be small values relative to the larger damping gains. If the leg reaches maximum compression during the first stage of GALLOP-STOP-CONTROL, the knee spring rest position is adjusted to achieve the evolved energy level, which further removes energy from the system.

During the second stage of GALLOP-STOP-CONTROL, the ab/ad, hip, and knee joints are transitioned to final positions for a standing posture, which is used to terminate the sequence. If, however, the leg breaks contact at some point during GALLOP-STOP-CONTROL, the leg is transferred during flight using the TRANSFER leg function and subsequently held in place using the HOLD function. The maneuver is terminated once each component of the velocity vector settles to within an ϵ envelope, similar to the 2% settling time for second-order linear systems.

Since the leg may or may not break contact during GALLOP-STOP-CONTROL, there are two basic modes for stopping: a continuous contact mode, and a multiple-contact, bouncing mode. The former represents a sliding stop, since limited friction prevents the system from "sticking" a landing. The sliding stop was an unanticipated solution found by the MOGA, although it is a more realistic mode, since animals typically skid to a halt on limited-friction running surfaces. Furthermore, the generation of emergent behaviors represents one of the more interesting aspects of evolution. Unanticipated solutions may not only offer additional insight into the underlying problem, but may also provide better, more robust solutions than the anticipated ones.

The final maneuver is the running jump. Like the jump-start, the running jump is a multiple-stage maneuver. Furthermore, it is comprised of the same leg primitive functions and the same state diagram as the gallop. However, the first two stages require multiple contact periods for the front and rear legs. Consequently, parameters are referenced with respect to leg cycles, which are delimited by contact phases. In contrast to the gallop, the running jump utilizes shoulder-thrusting and hip-braking to generate the requisite pitch moments for the jump and recovery stages.

Finally, a complete sequence of galloping and maneuvering was presented as a final demonstration of the results obtained in this study. Although a real-time controller for maneuvering was not developed, the sequence illustrates that successful transitions between galloping and maneuvering are still possible given the limited region of attraction for the gallop.

CHAPTER 7

Summary and Future Work

7.1 Summary

Legged biological systems demonstrate remarkable speed and agility, as they are able to run at high speeds over uneven and varying terrain, turn sharply, stop quickly, and leap over obstacles. This requires not only a finely tuned mechanical system but also a sophisticated controller capable of stabilizing a dynamic system in which there are recurring periods of static instability. In the field of legged robotics, these feats are far from conquered, as Raibert's pioneering work of the 1980s still represents some of the best results for dynamic running in a quadrupedal machine. Furthermore, because of the relative lack of progress in this area, it is not surprising that very little attention has been paid to the subject of quadrupedal dynamic maneuvers, which represent more extreme examples of dynamic stability.

Considerable effort has been made to develop a rigorous mathematical approach for modeling dynamic systems with intermittent contacts to facilitate the development of stable control strategies. However, such analytical methods frequently fail when assumptions are relaxed. For systems with numerous degrees of freedom, non-trivial inertial properties, and for asymmetric, fully spatial running gaits, approaches like

Poincaré return map analysis, which has been the workhorse for studying Raibert-inspired dynamical legged machines, can yield an intractable set of equations where numerical integration is required. In this dissertation, a practical and effective alternative method has been introduced that has generated what appear to be new solutions for a simulated quadrupedal machine that gallops, turns, jumps, and stops in unconstrained spatial coordinates. While this method cannot replace an analytical approach, it does offer a practical means of generating and analyzing solutions to complex problems using a system that is based on nontrivial assumptions. Consequently, this approach could be used in parallel with a more traditional technique to determine the impact of the various simplifying assumptions required under the latter.

To generate the variety of behaviors investigated here, a hierarchical control architecture was developed that utilizes parametric motor primitives at the lowest levels. These primitives provide the basic functionality to perform a wide variety of behaviors, including transferring the leg during flight (or contact), holding the leg at a fixed position, thrusting the leg using a multilevel torque profile, synchronizing the motion of the legs, and executing servo-based control over the leg joints during the contact phase. Sequential composition of these primitives for each leg, and the parallel combination across multiple legs, is used to create specific, complex behaviors. Once trained, these behaviors may be added to a library of behaviors which is drawn upon later to assemble high-level sequences.

Currently, the study of motor primitives has remained very general, with applicability to fairly simple systems with many fewer degrees of freedom than the quadrupedal model studied here. Furthermore, the motor primitives, or basis behaviors, are

often quite simple, involving a few motor neurons that respond to a set of stimuli. Thus, the application of this concept to a legged robot with 18 DOF and intricate, biological-mode target behaviors is one contribution of this work.

In order to train each behavior, a multiobjective genetic algorithm (MOGA) is employed to find parameters for each maneuver. A non-dominated sorting routine and fitness sharing approach are used to rank individuals by domination count and encourage lateral distribution across the Pareto front. Vector-valued fitness functions must be developed for goal behaviors, the use of which recognizes the inherent trade-offs associated with most engineering optimization problems, including the target behaviors studied here. While the components of the evolutionary algorithm have been adopted from the relatively broad field of research in this area, the application of this approach to the specific problem of dynamically stable running gaits and dynamic maneuvers appears to be new.

In general, the application of evolutionary algorithms to robotics control problems has become quite popular, resulting in an emerging field called evolutionary robotics. The main thrust of this body of work is focused on generating complex, emergent behavior from single- or multiple-agent mobile robots. However, the robots themselves are, almost always, simple systems that are statically stable. Consequently, very general controllers and evolutionary strategies can be employed to obtain interesting and illuminating results. For the current problem, however, there exists a significant bootstrapping problem in generating usable behaviors, which are generally complex and specific. Consequently, the approach adopted in this dissertation presents a practical and effective methodology, where a flexible control architecture is combined with the MOGA to achieve specific, complex dynamic behaviors that can

be evolved using relatively few generations and trials. This is possible mainly because the control architecture provides an efficient and effective way to parameterize each target behavior, which reduces the time required to find an initial acceptable solution (i.e., to bootstrap the process).

The use of the MOGA and control architecture developed in this dissertation has resulted in what appears to be the first biological-mode, fully spatial gallop in an articulated-leg model. While one example of a gallop-like gait has been demonstrated in the robot Scout II [25], the resulting gait did not resemble the biological one and failed to demonstrate any form of control over heading, as it moved in a tight, circular trajectory. In contrast, the gallop developed here demonstrates many of the features found in the biological gait, including early leg retraction, phase-locked leg motion, heading regulation, and relative smoothness with respect to minimal height excursion and pitch motion. A simple energy-control approach is employed to ensure that the same total energy level is achieved in the knee springs at maximum compression. Distribution of the energy among the legs is used to control pitch and roll, and hip velocity servos based on Herr and McMahon [30] are used to regulate velocity during stance. A linear top-of-flight feedback controller based on an outward-leg rotation strategy is used to compute ab/ad angles for additional control over roll and yaw. The basic approach is to use an asymmetric sprawled posture to correct for both roll and yaw errors. The yaw control strategy differs from that used by Raibert [21, 32], where differential ab/ad deflections in the front and rear are used to correct yaw, as Raibert's approach generates a narrower stance and was found to be unstable for this system.

In addition to the gallop, this dissertation has also presented what appears to be the first set of results for high-speed turning in a 3D quadrupedal model. It was found that the most stable high-speed turn is generated using a bank, where the body's roll and yaw are coupled together. For increasing turn angles, larger ab/ad deflections are necessary. Using a simple conical pendulum model, it is possible to predict the required ab/ad angles and tilt angle for the reaction force vector for the front legs, especially at larger desired turn angles. This suggests that the front legs tend to act like the cord of the pendulum, such that the reaction force vector in the transverse plane points in the same direction as the cord would, which provides the necessary centripetal acceleration to produce the turn. These results suggest that a real-time turning controller might be developed that adjusts front leg ab/ad angles using this simple model.

Solutions for the jump-start, sudden stop, and running jump were also presented in this dissertation. The jump-start requires an initial leap from a crouched configuration and subsequent bounding strides to gradually increase the quadruped's speed for a successful transition to galloping. From galloping, the quadruped executes a running jump using shoulder-thrusting and hip-braking (in contrast to the gallop, where shoulder-braking and hip-thrusting are employed) to generate the required pitch moments during the leap and recovery. Finally, to stop suddenly, the legs are placed at protracted angles for touchdown, and compliant, high-damping gains are used at the hips and knees to damp out kinetic energy. At maximum compression, energy is removed from the knee springs to further reduce the total energy of the system. Although a single-contact, non-sliding stop was originally desired, the best solution

found by the MOGA was a sliding stop. This solution turned out to be the most realistic mode of stopping given the limited friction of the surface.

Finally, a complete sequence of galloping, turning, jumping, and stopping was presented as a final demonstration of the results produced in this work. Although a real-time maneuvering controller is beyond the scope of this work, the final sequence demonstrates that successful transitions to and from 3D galloping are possible, despite the limited region of attraction for the gallop.

7.2 Future Work

The results presented in this dissertation have covered a relatively broad area in order to more fully explore the potential of the methodology promulgated within. However, this approach has also generated a broad range of issues that require more in-depth investigation, which should now be possible based on the contributions described above. These issues are listed below.

1. The current solution found for the gallop exhibits stable characteristics, although the robustness of the stability could be improved. The ability to handle uneven terrain and noisy (and minimal) sensor data would make the solution more amenable to hardware implementation. Improving the robustness of the solution should also lead to a better region of attraction when initializing the gait, which would facilitate the development of maneuvering controllers that could execute successful transitions to and from galloping.

2. Developing a gallop controller capable of running at different speeds, and changing from one speed to another, would allow the quadruped to accelerate or decelerate, to a limited extent, without requiring a specific maneuver. This would

also facilitate transitioning into a gallop, especially from a lower speed gait like the trot. Again, this capability would make the galloping results much more useful for a real-time hardware implementation.

3. Although the current gallop can maintain a heading with only a slight error, adding tight trajectory control would make it a more useful gait. Furthermore, gradual, but commanded changes in heading could be accomplished using a turning controller at shallow angles.

4. Multiple-stride turns demonstrated in this work were evolved on a stride-per-stride basis. However, using the results from Chap. 5 could lead to a real-time controller capable of producing multiple-stride turns that transition to and from galloping, especially if the gallop were more robustly stable.

5. Likewise, a jumping controller could be developed, as opposed to individually evolving each stage of a running jump. Desired jumping height or distance could be used as an input to generate different sized jumps.

6. Development of a high-level controller that could select from various galloping or maneuvering controllers could provide the ability to generate many different sequences. With sensing ability, such a controller could react to stimuli from the environment in choosing the current behavior, which would afford the opportunity to explore high-level behavior with a dynamical system.

7. Improving the simulation model by increasing the leg mass and decreasing the body's roll inertia would make the model more realistic and minimize the "reality gap" between the simulation and an actual robot. Furthermore, additional

compliance and additional DOF in the model may make the mechanical system easier to stabilize. An articulated back with passive compliance might facilitate the natural rejection of roll and yaw disturbances during galloping, which would largely reduce the required stabilization problem to the sagittal plane. Furthermore, articulated, compliant ankle and toe joints with a knee-forward design in the rear legs could be more energetically optimal with respect to the jump-start and running jump power strokes.

8. The control architecture is powerful in its flexibility and minimal parameterization. However, reduction of evolved behaviors into compact, coordinated functions remains a non-trivial step that requires hard-coding the various parameters, leg primitives, and sequence into a more concise form. Maintaining the data in its original form tends to require more memory and makes the architecture less modular. A more efficient representation of coordinated functions may be possible that facilitates this process.

9. Currently, cyclic behaviors like the gallop require a complex sequence of leg primitive functions. The ability to generate such motions using a simple neural oscillator would further simplify the architecture. However, this approach would necessarily require integration with a high-level, TOF feedback controller in order to be effective. Furthermore, the ability to expand the controller to include leg-level reflexes could be important in generating more robust locomotion algorithms.

10. The evolutionary algorithm employed in this work made use of vector-valued fitness functions. As explained in Sect. 4.4, the Pareto front for galloping was

fractured and discontinuous, suggesting that the fitness function may have had too much cross-coupling between variables, or that the fitness criteria were simply not entirely appropriate for the underlying problem. In either case, significant optimization could be done with respect to fitness criteria, sharing parameters, population size, number of generations, etc. Further optimization may lead to even better solutions for the gallop than the one described here.

11. While the results of this work support the study of high-speed quadrupedal locomotion and maneuvering in simulation, the ultimate goal for pursuing such research is the development of an experimental system capable of these behaviors. To that end, the construction of a small quadrupedal galloping and maneuvering machine using the control development methodology outlined here would further demonstrate the power of the evolutionary approach and the effectiveness of the various leg primitive functions. Some limiting assumptions may be necessary at first, including constraining the system to be planar, which would dramatically simplify control and reduce the number of required actuators. Furthermore, an off-board power supply may be necessary to minimize the total system mass. However, studying high-speed gaits like the gallop and dynamic maneuvers in such a system would still provide invaluable data on the actuation and control requirements for other experimental legged systems.

12. Apart from quadrupedal locomotion and maneuvering, much of the approach described in this work may also be applicable to more general legged systems, including bipeds, hexapods, etc. Furthermore, the evolutionary algorithm and many of the leg primitive functions have applicability for more general legged

locomotion algorithms. While minor modifications may be necessary, evolving different walking or running gaits, jumps, turns, etc., should be possible.

13. The purpose of evolution in this work was to find various parameters for the leg- and coordinated-level functions for each target behavior. However, the control architecture also allows for the evolution of the *sequence* of leg- and coordination-level functions, as well. Expanding the evolution capabilities in this way would allow for an entirely new set of solutions and far more interesting results. As stated in Sect. 6.3.4, generation of emergent solutions is one of the more exciting features of evolution and can often generate very effective strategies.

14. In addition to evolving the sequence of leg functions, the ability to evolve system design parameters and structure would be a useful tool for optimizing different link geometries, joint stiffness values, and morphologies for a given task. For example, solutions to certain running gaits may be significantly improved through the proper choice of compliance for the legs, as this value directly impacts the stride frequency of the system. An optimal stride frequency may exist for a given inertia matrix that yields a resonant running frequency with more favorable stability characteristics.

15. Most of the low-level implementation details of the control architecture and evolutionary algorithm have been omitted from this dissertation for the sake of brevity. For example, the construction of the various sequences using leg primitive and coordinated functions requires the manual creation of an ASCII-based configuration file. However, due to the complexity of the architecture

and number of parameters, developing such files is time-consuming and often results in user-induced errors or unwanted behaviors. Consequently, one key issue that should be addressed in order to make the software more user-friendly is the development of a graphical user interface (GUI). The GUI would then be used to automate the generation of configuration files, reducing errors and unwanted behaviors, and dramatically decreasing the lead-time to develop new behaviors and sequences.

In summary, this dissertation has presented a flexible control architecture coupled with a powerful multiobjective evolutionary approach to find solutions to several problems in high-speed dynamic locomotion and maneuvering that have, heretofore, not been demonstrated in simulation or hardware due to limitations in more traditional, analytical approaches. While successful analytical methods may eventually be found, the approach championed here provides both an immediate avenue for research and an excellent validation tool for model-based and heuristic-style control strategies. Most importantly, the methodology presented in this work requires simplification only to the extent required by the user in order to develop a testable strategy. Systems of arbitrary complexity may be evolved so long as the user can develop a plausible control method and provide reasonable search bounds on the parameters of interest. Hopefully, the results presented in this work will facilitate the development of high-speed locomotion algorithms for an actual robot in the near future.

BIBLIOGRAPHY

[1] F. A. Jenkins, "Limb posture and locomotion in the virginia oppossum (*Didelphis marsupialis*) and in other non-cursorial mammals," *Journal of Zoology, London*, vol. 165, pp. 303–315, 1971.

[2] R. B. McGhee, "Finite state control of quadruped locomotion," in *Proceedings of the Second International Symposium on External Control of Human Extremities*, Dubrovnik, Yugoslavia, 1966.

[3] R. S. Mosher, "Test and evaluation of a versatile walking truck," in *Proceedings of the Off-Road Mobility Research Symposium, International Society for Terrain Vehicle Systems*, Washington, D.C., 1968, pp. 359–379.

[4] R. B. McGhee, D. E. Orin, D. R. Pugh, and M. R. Patterson, "A hierarchically-structured system for computer control of a hexapod walking machine," in *Theory and Practice of Robots and Manipulators, Proceedings of RoManSy-84 Symposium*, A. Morecki, Ed. London: Hermes Publishing Co., 1985, pp. 375–381.

[5] S. M. Song and K. J. Waldron, *Machines That Walk: The Adaptive Suspension Vehicle*. Cambridge, Mass.: The MIT Press, 1989.

[6] D. Pugh, E. Ribble, V. Vohnout, T. Bihari, T. Walliser, M. Patterson, and K. Waldron, "Technical description of the Adaptive Suspension Vehicle," *The International Journal of Robotics Research*, vol. 9, pp. 24–42, April 1990.

[7] E. Krotkov and R. Simmons, "Performance of a six-legged planetary rover: Power, positioning, and autonomous walking," in *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, Nice, France, 1992, pp. 169–174.

[8] K. S. Espenschied, R. D. Quinn, H. J. Chiel, and R. D. Beer, "Leg coordination mechanisms in the stick insect applied to hexapod robot locomotion," *Adaptive Behavior*, no. 4, pp. 455–468, 1993.

[9] J. Schofield, "Dante survives the inferno," *Design News*, pp. 68–74, September 1994.

[10] K. Arikawa and S. Hirose, "Development of quadruped walking robot TITAN-VIII," in *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, 1996, pp. 208–214.

[11] K. S. Espenschied, R. D. Quinn, R. D. Beer, and H. J. Chiel, "Biologically based distributed control and local reflexes improve rough terrain locomotion in a hexapod robot," *Robotics and Autonomous Systems*, vol. 18, pp. 59–64, 1996.

[12] G. M. Nelson, R. D. Quinn, R. J. Bachman, and W. C. Flannigan, "Design and simulation of a cockroach-like hexapod robot," in *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, Albuquerque, NM, April 1997, pp. 1106–1111.

[13] S. Peng, C. P. Lam, and G. R. Cole, "A biologically inspired four legged walking robot," in *Proceedings of the 2003 IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan, 2003, pp. 2024–2030.

[14] K. Nonami, "Development of mine detection robot COMET-III," *JSME International Journal, Series C*, vol. 46, no. 3, pp. 881–890, 2003.

[15] C. Georgiades, A. German, A. Hogue, H. Liu, C. Prahacs, A. Ripsman, R. Sim, L.-A. Torres, P. Zhang, M. Buehler, G. Dudek, M. Jenkin, and E. Milios, "AQUA: An aquatic walking robot," in *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, 2004, pp. 3525–3531.

[16] R. Hodoshima, T. Doi, Y. Fukuda, S. Hirose, T. Okamoto, and J. Mori, "Development of TITAN XI: A quadruped walking robot to work on slopes. Design of system and mechanism," in *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, 2004, pp. 792–797.

[17] D. Spenneberg, K. McCullough, and F. Kirchner, "Stability of walking in a multilegged robot suffering leg loss," in *Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA)*, 2004, pp. 2159–2164.

[18] M. Albrecht, T. Backhaus, S. Planthaber, H. Stöpler, D. Spenneberg, and F. Kirchner, "AIMEE: A four-legged robot for RoboCup rescue," in *Proceedings of the 8th International Conference on Climbing and Walking Robots (CLAWAR)*, London, U.K., 2005, pp. 1003–1010.

[19] J. Hilljegerdes, D. Spenneberg, and F. Kirchner, "The construction of the four legged prototype robot ARAMIES," in *Proceedings of the 8th International Conference on Climbing and Walking Robots (CLAWAR)*, London, U.K., 2005, pp. 335–342.

[20] W. A. Lewinger, M. S. Branicky, and R. D. Quinn, "Insect-inspired, actively compliant hexapod capable of object manipulation," in *Proceedings of the 8th International Conference on Climbing and Walking Robots (CLAWAR)*, London, U.K., 2005, pp. 65–72.

[21] M. H. Raibert, "Trotting, pacing, and bounding by a quadruped robot," *Journal of Biomechanics*, vol. 23, pp. 79–98, 1990.

[22] J. Furushu, S. Akihito, S. Masamichi, and K. Eichi, "Realization of a bounce gait in a quadruped robot with articular-joint-type legs," in *Proceedings of the 1995 IEEE International Conference on Robotics and Automation (ICRA)*, Nagoya, Japan, 1995, pp. 697–702.

[23] H. Kimura, S. Akiyama, and K. Sakurama, "Realization of dynamic walking and running of the quadruped using neural oscillator," *Autonomous Robots*, vol. 7, pp. 247–258, 1999.

[24] D. Papadopoulos and M. Buehler, "Stable running in a quadruped robot with compliant legs," in *Proceedings of the 2000 IEEE International Conference on Robotics and Automation (ICRA)*, San Francisco, CA, 2000, pp. 444–449.

[25] J. A. Smith and I. Poulakakis, "Rotary gallop in the untethered quadrupedal robot Scout II," in *Proceedings of the International Conference on Intelligent Robots and Systems (IROS) 2004*, Sendai, Japan, 2004, pp. 2556–2561.

[26] D. F. Hoyt and C. R. Taylor, "Gait and the energetics of locomotion in horses," *Nature*, vol. 292, pp. 239–240, 1981.

[27] P. P. Gambaryan, *How Mammals Run: Anatomical Adaptations*. New York: John Wiley & Sons, 1974.

[28] A. E. Minetti, L. P. Ardigò, E. Reinach, and F. Saibene, "The relationship between mechanical work and energy expenditure of locomotion in horses," *The Journal of Experimental Biology*, vol. 202, pp. 2329–2338, 1999.

[29] T. A. McMahon, "The role of compliance in mammalian running gaits," *Journal of Experimental Biology*, vol. 115, pp. 263–282, 1985.

[30] H. M. Herr and T. A. McMahon, "A galloping horse model," *The International Journal of Robotics Research*, vol. 20, no. 1, pp. 26–37, 2001.

[31] N. C. Heglund and C. R. Taylor, "Speed, stride frequency and energy cost per stride: How do they change with body size and gait?" *Journal of Experimental Biology*, vol. 138, pp. 301–318, 1988.

[32] M. H. Raibert, *Legged Robots that Balance*. Cambridge, Mass.: MIT Press, 1986.

[33] A. F. Vakakis, J. W. Burdick, and T. K. Caughey, "An 'interesting' strange attractor in the dynamics of a hopping robot," *International Journal of Robotics Research*, vol. 10, no. 6, 1991.

[34] P. Nanua and K. J. Waldron, "Instability and chaos in quadruped gallop," *Journal of Mechanical Design*, vol. 116, pp. 1096–1101, 1994.

[35] R. Ringrose, "Self-stabilizing running," in *Proceedings of the 1997 IEEE International Conference on Robotics and Automation (ICRA)*, 1997, pp. 487–493.

[36] D. W. Marhefka, D. E. Orin, J. P. Schmiedeler, and K. J. Waldron, "Intelligent control of quadruped gallops," *IEEE/ASME Transactions on Mechatronics*, vol. 8, no. 4, pp. 446–456, December 2003.

[37] D. P. Krasny, "An analysis of high-speed running gaits in a quadruped robot using an evolutionary optimization strategy," Master's thesis, The Ohio State University, Columbus, Ohio, 2002.

[38] D. P. Krasny and D. E. Orin, "Achieving periodic leg trajectories to evolve a quadruped gallop," in *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, vol. 3, Taipei, Taiwan, September 2003, pp. 3842–3848.

[39] J. Hodgins and M. H. Raibert, "Biped gymnastics," *International Journal of Robotics Research*, vol. 9, no. 2, pp. 115–132, 1990.

[40] H. C. Wong and D. E. Orin, "Quadruped standing and running jumps," in *Video Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Atlanta, GA, May 1993.

[41] ——, "Control of a quadruped standing jump over irregular terrain obstacles," *Autonomous Robots*, vol. 1, pp. 111–129, 1995.

[42] D. L. Jindrich and R. J. Full, "Many-legged maneuverability: Dynamics of turning in hexapods," *The Journal of Experimental Biology*, vol. 202, pp. 1603–1623, 1999.

[43] M. Berkemeier, "Modeling the dynamics of quadrupedal running," *International Journal of Robotics Research*, vol. 17, no. 9, pp. 971–985, September 1998.

[44] R. T. M'Closkey and J. W. Burdick, "An analytical study of simple hopping robots with vertical and forward motion," in *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramenta, CA, April 1991, pp. 1392–1397.

[45] V. Hamburger, K. Berns, F. Iida, and R. Pfeifer, "Standing up with motor primitives," in *Proceedings of the 8th International Conference on Climbing and Walking Robots (CLAWAR)*, London, U.K., 2005, pp. 383–390.

[46] A. G. Feldman, "Superposition of motor programs – I. Rhythmic forearm movements in man," *Neuroscience*, vol. 5, pp. 81–90, 1980.

[47] F. A. Mussa-Ivaldi, S. F. Giszter, and E. Bizzi, "Linear combinations of primitives in vertebrate motor control," in *Proceedings of the National Academy of Science*, vol. 91, 1994, pp. 7534–7538.

[48] R. Brooks, "A robust layered control system for a mobile robot," *IEEE Journal of Robotics and Automation*, vol. RA-2, no. 1, pp. 14–23, 1986.

[49] W. Neibauer, "Locomotion with articulated legs in pipes or ducts," *Robotics and Autonomous Systems*, vol. 11, pp. 163–169, 1993.

[50] D. M. Bevly, S. Farritor, and S. Dubowsky, "Action module planning and its application to an experimental climbing robot," in *Proceedings of the 2000 IEEE International Conference on Robotics and Automation (ICRA)*, 2000, pp. 4009–4014.

[51] M. A. Arbib, "Perceptual structures and distributed motor control," in *Handbook of Physiology, Section 2: The Nervous System, Vol. II, Motor Control, Part 1*, V. B. Brooks, Ed. American Physiological Society, 1981, pp. 1449–1480.

[52] R. W. Paine and J. Tani, "Motor primitive and sequence self-organization in a hierarchical recurrent neural network," *Neural Networks*, vol. 17, pp. 1291–1309, 2004.

[53] R. Amit and M. J. Mataric, "Parametric primitives for motor representation and control," in *Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA)*, 2002, pp. 863–868.

[54] E. Drumwright, O. C. Jenkins, and M. J. Mataric, "Exemplar-based primitives for humanoid movement classification and control," in *Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA)*, 2004, pp. 140–145.

[55] H. H. F. Buchner, H. H. C. M. Savelberg, H. C. Schamhardt, and A. Barneveld, "Inertial properties of Dutch Warmblood horses," *Journal of Biomechanics*, vol. 30, no. 6, pp. 653–658, 1997, Technical Note.

[56] S. McMillan, D. E. Orin, and R. B. McGhee, "DynaMechs: An object oriented software package for efficient dynamic simulation of underwater robotic vehicles," in *Underwater Robotic Vehicles: Design and Control*, J. Yuh, Ed. Albuquerque, NM: TSI Press, 1995, pp. 73–98.

[57] J. J. Craig, *Introduction to Robotics: Mechanics and Control*. Reading, Mass.: Addison-Wesley Publishing Co., 1989, pp. 48–49.

[58] T. Bäck, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford: Oxford University Press, 1996.

[59] D. Floreano and F. Mondada, "Evolutionary neurocontrollers for autonomous mobile robots," *Neural Networks*, vol. 11, pp. 1461–1478, 1998.

[60] S. Nolfi and D. Floreano, *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. MIT Press, 2000.

[61] J. Urzelai and D. Floreano, "Evolutionary robots with fast adaptive behavior in new environments," in *Proceedings of the Third International Conference on Evolvable Systems (ICES)*, Edinburgh, Scotland, UK, 2000, pp. 241–251.

[62] L. Steels, "Emergent functionality in robotics agents through on-line evolution," in *Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*, R. A. Brooks and P. Maes, Eds., 1994, pp. 8–14.

[63] J. C. Gallagher, R. D. Beer, K. S. Espenchied, and R. D. Quinn, "Application of evolved locomotion controllers to a hexapod robot," *Robotics and Autonomous Systems*, vol. 19, no. 1, pp. 95–103, 1996.

[64] T. Gomi and K. Ide, "Evolution of gaits of a legged robot," in *Proceedings of the 1998 IEEE World Congress on Computational Intelligence*, vol. 1, 1998, pp. 159–164.

[65] G. S. Hornby, M. Fujita, S. Takamura, T. Yamamoto, and O. Hanagata, "Autonomous evolution of gaits with the Sony quadruped robot," in *Proceedings of the Genetic and Evolutionary Computation Conference*, W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith, Eds., vol. 2. Orlando, Florida, USA: Morgan Kaufmann, 1999, pp. 1297–1304.

[66] C. A. Coello, "An updated survey of GA-based multiobjective optimization techniques," *ACM Computing Surveys*, vol. 32, no. 2, pp. 109–143, 2000.

[67] D. W. Corne, K. Deb, P. J. Fleming, and J. D. Knowles, "The good of the many outweighs the good of the one: Evolutionary multi-objective optimization," *IEEE Connections Newsletter*, vol. 1, no. 1, pp. 9–13, Feb. 2003.

[68] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: Formulation, discussion, and generalization," in *Genetic Algorithms: Proceedings of the Fifth International Conference*. San Mateo, CA: Morgan-Kaufmann, July 1993, pp. 416–423.

[69] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.

[70] K. Deb and A. Pratap, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[71] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1994.

[72] M. Jamshidi, L. dos Santos Coelho, R. A. Krohling, and P. J. Fleming, *Robust Control Systems with Genetic Algorithms*. Boca Raton, FL: CRC Press, 2003.

[73] T. Bäck, U. Hammel, and H. P. Schwefel, "Evolutionary computation: Comments on the history and current state," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, April 1997.

[74] X. Yao, "Global optimisation by evolutionary algorithms," in *Proceedings of the Second AIZU International Symposium on Parallel Algorithms/Architecture Synthesis*. Aizu-Wakamatsu, Japan: IEEE Computer Society Press, Los Alamitos, CA, March 1997, pp. 282–291.

[75] K. M. Passino, *Biomimicry for Optimization, Control, and Automation*. London: Springer-Verlag, 2005.

[76] D. P. Krasny and D. E. Orin, "Generating high-speed dynamic running gaits in a quadruped robot using an evolutionary search," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 34, no. 4, pp. 1685–1696, Aug. 2004.

[77] K. Deb and T. Goel, "Controlled elitist non-dominated sorting genetic algorithm for better convergence," in *Proceedings of the First International Conference on Evolutionary Multi-criterion Optimization*, 2001, pp. 67–81.

[78] J. Knowles and D. Corne, "The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimisation," in *Proceedings of the Congress of Evolutionary Computation*, Washington, D. C., July 1999, pp. 98–105.

[79] R. M. Alexander, A. S. Jayes, and R. F. Ker, "Estimates of energy cost for quadrupedal running gaits," *Journal of Zoology*, vol. 190, pp. 155–192, 1980.

[80] P. Nanua and K. J. Waldron, "Energy comparison between trot, bound, and gallop using a simple model," *ASME Journal of Biomechanical Engineering*, vol. 117, pp. 466–473, 1995.

[81] C. T. Rubin and L. E. Lanyon, "Limb mechanics as a function of speed and gait: A study of functional strains in the radius and tibia of horse and dog," *Journal of Experimental Biology*, vol. 101, pp. 187–211, 1982.

[82] A. A. Biewener and C. R. Taylor, "Bone strain: A determinant of gait and speed?" *Journal of Experimental Biology*, vol. 123, pp. 383–400, 1986.

[83] C. T. Farley and C. R. Taylor, "A mechanical trigger for the trot-gallop transition in horses," *Science*, vol. 253, pp. 306–308, 1991.

[84] J. P. Schmiedeler, "The mechanics of and robotic design for quadrupedal galloping," Ph.D. dissertation, The Ohio State University, Columbus, Ohio, 2001.

[85] J. Boswall, "Nova: Animal Olympians," 1980, educational television program.

[86] M. Hildebrand, "Walking and running," in *Functional Vertebrate Morphology*, M. Hildebrand, D. M. Bramble, K. F. Liem, and D. B. Wake, Eds. Cambridge: Belknap Press, 1985, pp. 25–65.

[87] ——, "The adaptive significance of tetrapod gait selection," *American Zoologist*, vol. 20, pp. 97–103, 1977.

[88] H. M. Herr, G. T. Huang, and T. A. McMahon, "A model of scale effects in mammalian quadrupedal running," *The Journal of Experimental Biology*, vol. 205, pp. 959–967, 2002.

[89] D. W. Marhefka, "Fuzzy control and dynamic simulation of a quadruped galloping machine," Ph.D. dissertation, The Ohio State University, Columbus, Ohio, 2000.

[90] J. E. Pratt and B. T. Krupp, "Series elastic actuators for legged robots," in *Proceedings of SPIE–The International Society for Optical Engineering*, vol. 5422, 2004, pp. 135–144.

[91] Advanced Motion Controls, Inc., "Power supply selection document," Camarillo, CA 93012, 2003, www.a-m-c.com.

[92] G. Gabrielli and T. H. von Karman, "What price speed?" *Mechanical Engineering*, vol. 72, no. 10, pp. 775–781, 1950.

[93] R. J. Full, T. Kubow, J. Schmitt, P. Holmes, and D. Koditschek, "Quantifying dynamic stability and maneuverability in legged locomotion," *Integrative and Comparative Biology*, vol. 42, pp. 149–157, 2002.

[94] M. A. Fedak, N. C. Heglund, and C. R. Taylor, "Energetics and mechanics of terrestrial locomotion II: Kinetic energy changes of the limbs and body as a function of speed and body size in birds and mammals," *Journal of Experimental Biology*, vol. 79, pp. 23–40, 1982.

[95] C. T. Farley, J. Glasheen, and T. A. McMahon, "Running springs: Speed and animal size," *Journal of Experimental Biology*, vol. 185, pp. 71–86, 1993.

[96] M. G. Pandy, V. Kumar, N. Berme, and K. J. Waldron, "The dynamics of quadrupedal locomotion," *ASME Journal of Biomechanical Engineering*, vol. 110, pp. 230–237, 1988.

[97] H. C. Schamhardt, H. W. Merkens, V. Vogel, and C. Willekens, "External loads on the limbs of jumping horses at take-off and landing," *American Journal of Veterinary Research*, vol. 54, no. 5, pp. 675–680, 1993.

[98] P. A. McKinley, J. L. Smith, and R. J. Gregor, "Responses of elbow extensors to landing forces during jump downs in cats," *Experimental Brain Research*, vol. 49, pp. 218–228, 1983.

[99] O. Miglino, H. H. Lund, and S. Nolfi, "Evolving mobile robots in simulated and real environments," *Artificial Life*, vol. 2, pp. 417–434, 1995.

[100] N. Jakobi, "Half-baked, ad-hoc and noisy: Minimal simulations for evolutionary robotics," in *Advances in Artificial Life: Proceedings of the 4th European Conference on Artificial Life*, P. Husbands and I. Harvey, Eds., 1997, pp. 348–357.

[101] D. W. Marhefka and D. E. Orin, "A compliant contact model with nonlinear damping for simulation of robotic systems," *IEEE Transactions on Systems, Man, and Cybernetics–Part A: Systems and Humans*, vol. 29, no. 6, pp. 566–572, November 1999.

[102] E. Muybridge, *Animals in Motion*. New York: Dover Publications, 1979.