Sensor Filtering and State Estimation of a Fast Simulated Planar Bipedal Robot

Stefano Rossi and S. Andrew Gadsden

Abstract The development of bipedal humanoid robots is a very prevalent area of research today. Legged robots have many advantages over wheeled robots on rough or uneven terrains. Due to the rapid growth in robotics, it is unavoidable that legged robots will be adapted for everyday household settings. However, the agile bipedal robots possesses many design and control challenges. Model based control of humanoid robots relies on the accuracy of the state estimation of the model's constituents. The spring loaded inverted pendulum (SLIP) is frequently used as a fundamental model to analyze bipedal locomotion. In general, it consists of a stance phase and a flight phase, employing different strategies during these phases to control speed and orientation. Due to the underactuation and hybrid dynamics of bipedal robots during running, estimating the state of the robot's appendages can be challenging. In this paper, various Kalman estimation techniques are combined with sensor data fusion to predict the spatial state of a fast simulated planar SLIP model.

Keywords State estimation · Bipedal robot · Kalman filter

1 Introduction

Unlike fixed based robots, bipedal robots have a floating base and are high degree of freedom dynamical systems. They can move around complex environments and state estimation is a crucial part of controlling such a system. For a controller using the model dynamics to compute feed-forward torques, the state estimator needs to provide the orientation, linear and angular velocities of each component.

S. Rossi (✉) · S. Andrew Gadsden
College of Engineering and Physical Sciences, University of Guelph, Guelph, ON, Canada
e-mail: srossi@uoguelph.ca

S. Andrew Gadsden
e-mail: gadsden@uoguelph.ca

F. Janabi-Sharifi and W. Melek (eds.), *Advances in Motion Sensing and Control for Robotic Applications*, Lecture Notes in Mechanical Engineering,
https://doi.org/10.1007/978-3-030-17369-2_1

The introduction of accurate full body state estimates as well as force interactions with the external environment throughout the phases of rapid locomotion considerably increases the agility of legged robots [1]. Empirical studies of proprioception controllers relying on inertial measurement unit (IMU) feedback has resulted in significantly improved performance of legged robots [2]. Performance improved on varying terrain such as slopes and broken terrain [3], as well as introducing entirely new behaviors such as flips [4] and unique gaits [5]. In mobile robotics, the concept of sensor data fusion has already been widely used on wheeled vehicles. However, in the field of legged robotics, there has only been limited study in applying sensor fusion to control robot behavior. The implementation of a cost effective sensor suite to deliver full body state estimation relevant to motor control remains a challenge in legged locomotion, due to the severe drift and sensitivity of low cost IMU packages. While sensor fusion has been applied to the estimation and control of walking bipedal robots [6], it has yet to be tested on a biped undergoing rapid locomotion.

In this paper the effectiveness of state estimation of a rapid biped robot using common IMU sensors will be tested in a simulated environment. Various methods of integrating and filtering these sensors will be explored with the goal of providing a basis for cost effective alternatives in researching rapid bipedal locomotion control. Research in the development of agile bipedal robotics is rapidly growing, yet state estimation remains difficult due to the hybrid dynamics and non-linearity inherit to rapid locomotion [7]. The aim of this experiment is to demonstrate that accurate state estimation of rapid locomotion robots can be done utilizing simple models and cost effective sensor data fusion as opposed to closed form motion planning.

An overview of the background and previous works related to rapid locomotion will be outlined. A framework of the design goals for a sensor based state estimator will be extrapolated from previous works, as well as a description of the methods and model to be used.

2 Background and Previous Work

2.1 Sensory Pose Estimation of Legged Robots

Due to the rapid development of modern legged robots, the underlying principles of dynamically stable locomotion have been revealed. As a result, more research is being done on how to increase a robot's performance and agility through sensory feedback and data fusion algorithms.

One of the earliest attempts [8] of estimating a legged robot's global pose was performed on a hexapod robot, the Ambler. The position of the feet was calculated from the motor input commands and encoder measurements on the joints. More recently, pose estimation techniques for a hexapod were developed by fusing IMU data, vision and leg odometry [8]. Regarding dynamical gaits, an extended Kalman filter (EKF) based body estimation approach only using proprioceptive sensors and leg kinematics was introduced in a hexapod [9]. With regards to biped robots, sensor fusion utilizing observer based data and preview control has be utilized for stabilizing walking motion on a 3 dimensional linear inverted pendulum model [10]. The use of only proprioceptive sensor fusion has been applied to bipedal robots and feedback control applications, but only in the context of stable walking and turning [6], and not yet to rapid locomotion.

Based on an assessment of the previous works, it is conjectured that a suitable filter for general biped estimation should (1) only use proprioceptive sensors (2) make no assumptions about the outside environment or gait and (3) be easily adaptable to any general rapid bipedal locomotion platform.

2.2 Estimation Framework

The Kalman Filter (KF) is an optimal state estimation strategy, and is widely used in the field of control and estimation theory [11–14]. The Kalman filter provides an estimate of the state along with a corresponding covariance matrix which outlines the uncertainty of the estimate. The Kalman filter involves two steps, the prediction step where the previous state estimates are propagated through the system to produce a priori estimates, and the update step where the a priori estimates are combined with the current observed measurements to refine the state estimate, referred to as the a posteriori state estimate.

2.3 SLIP Model

To study running in its simplest form, a single legged planar running machine was built [15] in 1984, to be later recognized as the Spring Loaded Inverted Pendulum model (SLIP model). Although the robot only had one leg, the main principle is identical to a biped, and the SLIP model is widely used today to study dynamically stable running. The original machine used a pneumatic leg simulate a telescopic passive spring and was capable of exerting a thrust force. A single running cycle consists of a stance and flight phase. During the stance phase, the leg supports the body and remains in a fixed position on the ground. In this phase, the robot tips like an inverted pendulum while the spring undergoes compression and then thrust. During stance, there is no chance to move the foot placement to control position. In order to change the foot position, the robot jumps to flight phase where the leg is unloaded and free to swing. Marc Raibert developed a simple control strategy for simple legged robots which allowed them to perform dynamically stable running, as well as regulate speed and body attitude. This control method relied on measurements of the forward speed and body attitude [16].

3 Simulated Model

The model used in this paper was created in VREP and modelled after the Raibert planar biped [17]. The model consists a rectangular main body, with two actuated hip joints, which connect to telescopic legs (Fig. 1). The leg acts as a passive spring/damper during the compression portion of the stance phase, and are capable of applying a thrust force. The robot is attached to a spherical joint at the hip by a 5 m massless boom. This eliminates 3DOF from the model, its yaw, roll and lateral movement. The feet consist of spheres and have perfect friction with the ground ($\mu = 1$) (Table 1).

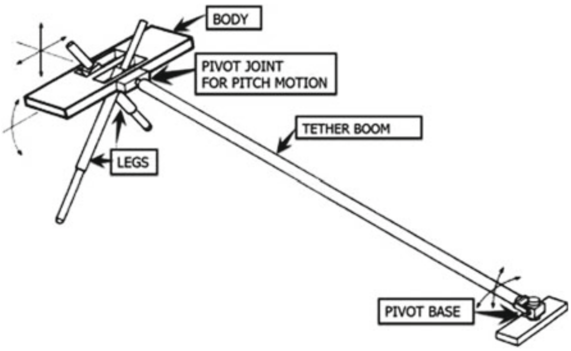

Fig. 1 Simulated biped model assembly

Table 1 Symbols and descriptions

Symbol	Description
θ	Leg angle relative to body vertical
φ	Body angle with respect to horizontal
φ_d	Desired body angle
r	Leg length at equilibrium
z	Body height from contact surface
x	Body position on track
$\dot{x}$	Body velocity
$\dot{x}_d$	Desired body velocity
x_r	Foot distance from body CoG
M_B	Mass of body
M_L	Mass of leg
I_B	Body mass moment of inertia
g	Gravity
T_{st}	Time of Stance
K_L	Spring constant of leg
τ	Control torque of active hip

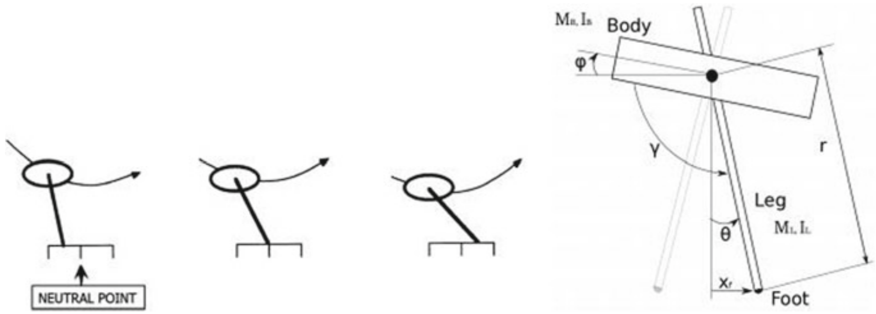

Fig. 2 Free body diagram (right) and illustration of how touchdown foot placement effects the takeoff trajectory (left)

The simulation used Open Dynamics Engine as its physics engine due to its accuracy modelling spring damper systems. The main control loop is implemented directly in VREP via a child script. The running motion of the robot can be described in 5 phases: flight, touchdown, compression, thrust and takeoff. The main working principle behind the speed control is foot placement. Because the leg acts as a spring-damper, the time of the stance phase (compression and thrust) can be approximated by:

$$T_{st} = \frac{\pi}{\omega} = \pi\sqrt{\frac{M_B + M_L}{K_L}} \tag{1}$$

Foot placement has a direct effect on the resultant velocity at takeoff (Fig. 2). If the foot is placed directly at the halfway point throughout the stance (neutral point), the stance phase is symmetric and the takeoff velocity is the same as the touchdown.

$$x_{f0} = \frac{\dot{x}T_{st}}{2} \tag{2}$$

Any deviation from the neutral point results in a non-zero horizontal acceleration. Placing the foot before the neutral point results in positive acceleration in the forward direction, as more of the vertical velocity is converted to horizontal, and vice versa (Fig. 2). Therefore, foot position on touchdown is used accelerate to a desired speed. This is regulated by proportional control. The algorithms for foot placement and corresponding hip angles are:

$$x_f = \frac{\dot{x}T_{st}}{2} + k_{\dot{x}}(\dot{x} - \dot{x}_d) \tag{3}$$

$$\gamma_d = \emptyset - \sin^{-1}\left(\frac{\dot{x}T_{st}}{2} + \frac{k_{\dot{x}}(\dot{x} - \dot{x}_d)}{r}\right) \tag{4}$$

Body attitude is maintained by applying a torque about the hip during the stance phase. Since angular momentum is conserved during flight, the friction between the foot and the ground provides an opportunity to correct the angular momentum of the entire system. To servo the body to a desired attitude, the control torque is applied is:

$$\tau = -k_p(\varphi - \varphi_d) - k_v(\dot{\varphi}) \tag{5}$$

where k_p and k_v are constants, set to 80 and 20 respectively. During the stance and flight phases, the idle leg mirrors the active leg to cancel out angular momentum.

4 State Estimation

4.1 Body Pose Estimation

The body angle of the robot is determined using a KF. The measurements consisted of a gyroscope located at the center of the body and the applied control torque during stance phase. Force sensors located on the feet of the robot were also used to provide a binary reading, indicating if the robot is making contact with the ground. The gyroscope was simulated using a reference point where the Euler angles were read, and simulation noise and drift was added. The control torque is an input based on the current body angle and estimated angular velocity. These values were streamed from VREP in real time to MATLAB using a remote API, where the Kalman filtering occurred. The estimated state was then streamed back into VREP to compute the control outputs for the next time step. The true body attitude and angular velocity were also streamed to MATLAB for comparison at each time step (Table 2).

Table 2 Simulation parameters

Parameter	Value	Description
t	0.001	Simulation time step, in s
I	4.708e−02	Body mass moment of inertia, in kg m^2
ϕ_d	0	Desired body angle, in rad
r	0.5575	Leg length uncompressed, in m
$\dot{x}_d$	2	Desired body velocity, in m/s
M_B	9.246	Mass of body, in kg
M_L	0.478	Mass of leg, in kg
T_{st}	0.178	Time of Stance, in s
K_L	2700	Spring constant of leg, in N/m

Estimation Methods. Three methods were used for comparison to estimate the pose and angular velocity of the body. The first method was only using the gyroscope to compute the current angle and angular velocity.

$$\varphi_i = \varphi_{i-1} + t\dot{\varphi}_i \tag{6}$$

The second method utilized the input torque applied during stance along with the gyroscope. If the force sensor indicates the robot is currently in flight, the control torque measurement is disregarded. The hip control torque only has a large effect on body attitude when a foot is in contact with the ground.

$$\varphi_i = \varphi_{i-1} + t\dot{\varphi}_i + \frac{\tau}{2I}t^2 \tag{7}$$

The third method was the implementation of the Kalman filter utilizing (7) as the system model. Again the system changes during flight, the control torque is disregarded.

$$\begin{bmatrix} \varphi \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \varphi \\ \dot{\varphi} \end{bmatrix} + \begin{bmatrix} \frac{t^2}{2I} \\ t/I \end{bmatrix} \begin{bmatrix} \tau \\ \tau \end{bmatrix} \tag{8}$$

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{9}$$

Measurement and system noise were then added to the system to simulate real world conditions. These values were assumed to be Gaussian and white. These values were approximated, and it was assumed most of the noise would stem from the sensors, so the system noise was several orders of magnitude less than the measurement noise.

Results. The simulation ran for 3000 time steps, where the robot took six steps. The KF performed the best with an RMSE approximately six times lower than that of the gyroscope estimation alone, and approximately half that of the gyroscope and the torque estimation (Table 3). As per the position estimation results, the gyroscope estimation drifts over time, while the combined estimation overshoots the extreme tilt angles (Fig. 3). The KF accurately accounts for these deviations and best represents the true angle of the body. Although the Kalman filter is able to filter out most of the noise in the angular velocity estimation, it does seem to have a bias. This may be due to the Kalman filter counteracting the bias of the gyroscope, as the state error

Table 3 Root mean square error of the three body angle estimation methods

	Gyroscope angle estimation (rad^2)	Gyro & Torque angle estimation (rad^2)	Kalman filter angle estimation (rad^2)
RMSE	0.031735	0.009148	0.005355

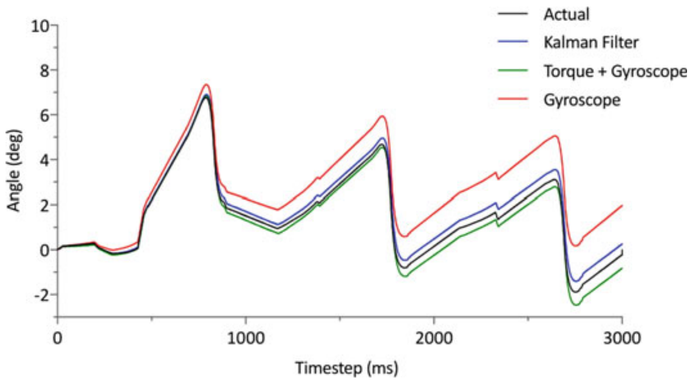

Fig. 3 The estimated body angle according to the Kalman filter, the gyroscope, and the gyroscope combined with the control torque model

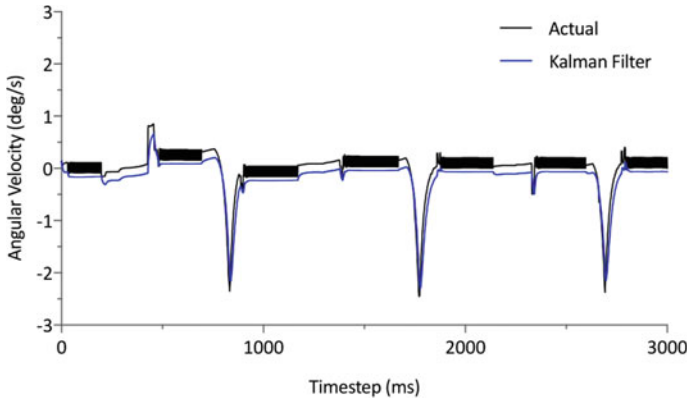

Fig. 4 The estimated angular velocity of the model according to the Kalman filter

covariance of both the position and the velocity converge quickly over time (Figs. 5 and 6). The estimated state of the body was streamed back into the VREP simulation to predict the next control outputs, and the robot performed just as well as if reading the true states, running at its desired speed of 2 m/s with a standard deviation of 0.15 m/s (Fig. 4).

4.2 Leg State Estimation

The state of the leg is estimated using a sensor fusion between an accelerometer and gyroscope located at the hip joint of the model. The virtual accelerometer consists of a reference mass attached to a force sensor. These values were streamed from VREP

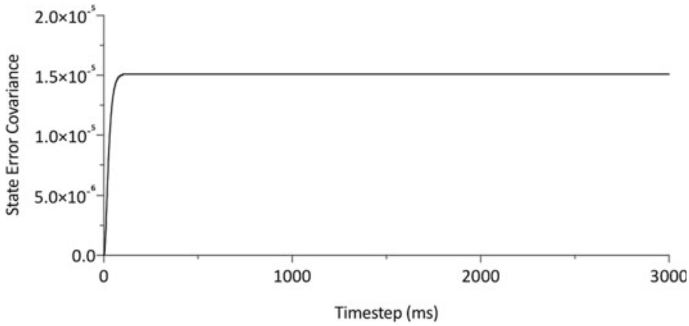

Fig. 5 The Kalman filter state error covariance of the angle

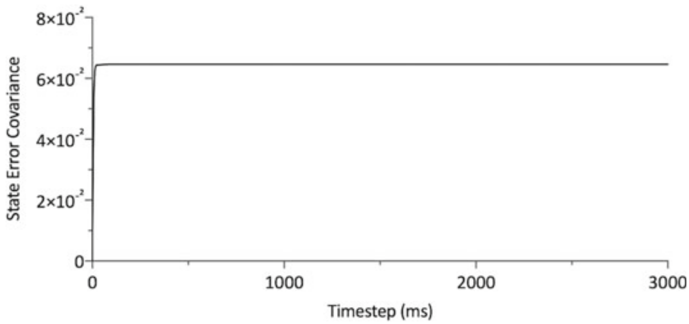

Fig. 6 The Kalman filter state error covariance of the angular velocity

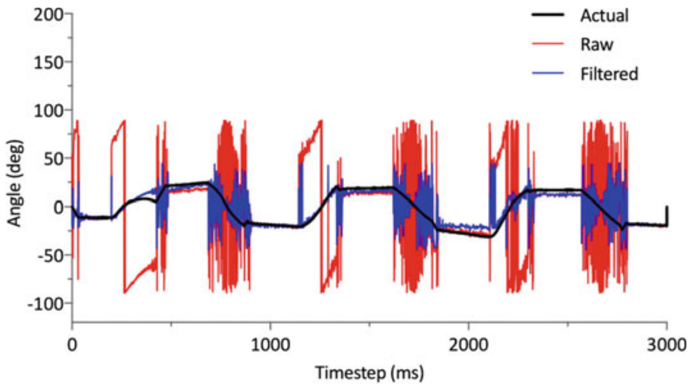

Fig. 7 The raw accelerometer data from the leg compared to the corrected data

in real time to MATLAB using a remote API. They were then filtered and streamed back into VREP to calculate the control inputs for the next time step. The true angle and angular velocity of the leg were also streamed to MATLAB for comparison.

Estimation Methods. Due to the impact forces of touchdown and the harmonic motion of the robot, the calculated accelerometer angle of the leg is very noisy and biased during the stance phase. The signal was processed through two corrective algorithms before it was filtered to provide better estimations. The first uses accelerometer data from the body rather than the leg. Because the body of the biped remains relatively level (±5°) compared to the sweep range of the leg (±40°), the calculated accelerometer angle of the body is subtracted from that of the leg to correct for the rising and falling motion of the model. The data is then put through a low pass filter to correct for the large spikes occurring during impact at touchdown. The angle of the leg was estimated using two methods, a complimentary filter and a Kalman. The complementary filter combined the calculated angle from the gyroscope sensor and the accelerometer sensor.

$$\theta_{comp} = \alpha\left(\theta_{gyro}\right) + (1 - \alpha)(\theta_{accel}), \alpha = 0.9 \tag{10}$$

The state of the leg was also estimated using a Kalman filter. Measurement and system noise were then added to the system to simulate real world conditions. These values were assumed to be Gaussian and white. These values were approximated, and it was assumed most of the noise would stem from the sensors, so the system noise was several orders of magnitude less than the measurement noise.

$$\begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \varphi \\ \dot{\varphi} \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{11}$$

$$Q = \begin{bmatrix} \frac{t^3}{3} & \frac{t^2}{2} \\ \frac{t^2}{2} & t \end{bmatrix}, R = \begin{bmatrix} 10^{-4} & 0 \\ 0 & 10^{-1} \end{bmatrix} \tag{12}$$

Results. As seen in Figs. 8 and 9, the KF outperformed the complementary filter with an RMSE almost half that of the complementary (Table 4). The complementary performed fairly well, however after a certain time the gyroscopic drift had too large of an effect on the estimation. The KF was able to filter out most of the noise from the accelerometer reading (Fig. 8). While the position tracking estimation was smooth, very little of the velocity estimation noise was filtered out. This is primarily due to the low measurement noise covariance applied to the gyroscope data, and a large noise covariance applied to the accelerometer. The estimated state of the active leg was streamed back into the VREP simulation to predict the next control outputs in real time. The robot performed just as well as if reading the true states, running at its desired speed of 2 m/s with a deviation of 0.13 m/s (Fig. 7).

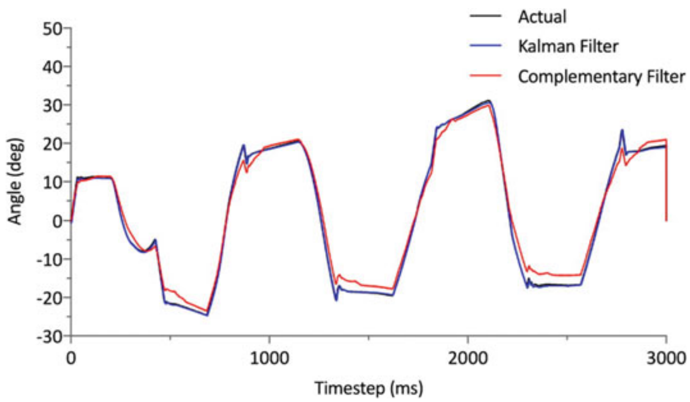

Fig. 8 The estimated leg angle according to the Kalman filter and the complimentary filter

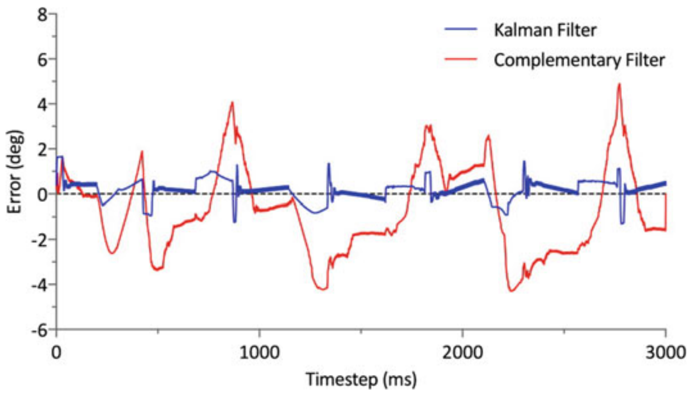

Fig. 9 The leg angle tracking error of the Kalman filter and the complimentary filter over simulation time

Table 4 Root mean square error of the Kalman and complimentary angle estimation methods

	Complementary filter angle estimation (rad^2)	Kalman filter angle estimation (rad^2)
RMSE	0.0658437	0.0388306

5 Conclusions and Future Work

5.1 Conclusions

Preliminary virtual experimentation showed that the implementation of a Kalman filter in combination with proprioceptive sensors is very effective at estimating of the states of the model constituents, outperforming other non-adaptive simple filters.

Much of the sensor and system noise was filtered to provide accurate readings and estimates. From these preliminary results, more extensive implementations of multiple Kalman filters could improve the overall performance of simple robotic bipeds. The system matrix has the largest effect on the performance of the Kalman filter. Overall, it has been successfully demonstrated to use Kalman filtering techniques to estimate the state of a simple bipedal model undergoing fast locomotion just using sensor data. The techniques outlined above would be suitable for many other biped configurations and physical platforms in the real world.

5.2 Future Work

While Kalman filter estimations for the state of the body and the legs were successful, the state estimation of the entire model dynamics of the with respect to the world frame has yet to be implemented. More accurate linearized state equations for the stance phase would be very useful in improving the filters, and allowing the implementation of an extended Kalman filter. Multiple linearized models for the different states would allow the implementation IMM-EKF strategy [18]. Although the simulated model is easier to work with, a physical model would be the next step in experimentally tuning the filtering techniques. Ideally, to minimize computation, one filter would implement multiple system models for each phase of locomotion, and accurately switch between them, accurately predicting the changing center of gravity and accounting for the hybrid dynamics. With the multiple sensory readings, this system also lends itself to a neural net configuration, either for the state estimation or as a main control strategy. Applying a Kalman filter to adaptively train the neural net may yield a more successful and adaptable control strategy compared the Raibert controller implemented.

References

1. Saranli U (2002) Dynamic locomotion in a hexapod robot. PhD dissertation, Ann Arbor, University of Michigan
2. Komsuoglu H, McMordie D, Saranli U, Moore N, Buelher M, Koditschek DE (2001) Proprioception based behavioral advances in a hexapod robot. In: IEEE international conference on robotics and automation, Seoul, Korea
3. Weingarten J, Grof RE, Koditschek DE (2004) A framework for the coordination of legged robot gaits. In: IEEE conference on robotics, automation and mechatronics, Singapore
4. Saranli U, Rizzi A, Koditschek D (2009) Model-based dynamic self righting maneuvers for a hexapedal robot. Int J Robot Res 23(9):903–918
5. McMordie D, Buehler M (2001) Towards pronking with a hexapod robot. In: International conference on climbing and walking robots
6. Pei-Chen C, Yi Chen J, Kai Huang C, Ching Lu J, Chun Lin P (2015) Sensor data fusion for body state estimation in a bipedal robot and its feedback control application for stable walking. Sensors 15(3):4925–4946

7. Sreenath K, Park H-W, Poulakakis I, Grizzle J (2013) Embedding active force control within the compliant hybrid zero dynamics to achieve stable, fast running on MABEL. Int J Robot Res 32(3):324–345
8. Chilian A, Hirschmuller H, Gorner M (2008) Multisensor data fusion for robust pose estimation of a six-legged walking robot. In: IEEE/RSJ international conference on intelligent robots and systems, San Francisco
9. Lin P-C, Komsuoglu H, Koditschek D (2006) Sensor data fusion for body state estimation in a hexapod robot with dynamical gaits. IEEE Trans Rob 22(5):932–943
10. Urbann O, Tasse S (2013) Observer based biped walking control, a sensor fusion approach. Auton Robots 35:37–49
11. Afshari HH, Gadsden SA, Habibi SR (2017) Gaussian filters for parameter and state estimation: a general review and recent trends. Sig Process 135:218–238
12. Gadsden SA, Song Y, Habibi SR (2013) Novel model-based estimators for the purposes of fault detection and diagnosis. IEEE/ASME Trans Mechatron 18(4)
13. Raibert MH (1984) Hopping in legged systems-modeling and simulation for the 2D one-legged case. IEEE Trans Syst Man Cybern 14:3
14. Raibert MH (1986) Legged robots that balance. MIT Press, Cambridge, Mass
15. Hodgins J (1985) Planar Biped. MIT Leg Laboratory, Boston, Mass
16. Gadsden AS, Habibi SR, Kirubarajan T (2010) A novel interacting multiple model method for nonlinear target tracking. In: International conference on information fusion, Edinburgh
17. Reinstein M, Hoffmann M (2011) Dead reckoning in a dynamic quadruped robot: Inertial navigation system aided by a legged odometer. In: IEEE international conference on robotics and automation, Shanghai
18. Roston GP, Krotkov EP (1992) Dead reckoning navigation for walking robots. In: IEEE/RSJ international conference on intelligent robots and systems, Raleigh