2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Congress Center Hamburg
Sept 28 - Oct 2, 2015. Hamburg, Germany

From One-legged Hopping to Bipedal Running and Walking: A Unified Foot Placement Control Based On Regression Analysis

Yangwei You, Zhibin Li, Darwin G. Caldwell and Nikos G. Tsagarakis

Abstract— **This paper aims at developing a unified and adaptive foot placement control for legged robots. The locomotion control of legged robots can be classified into three parts as body height control, body attitude control, and forward velocity control. In our study, the body attitude is controlled at stance phase by the hip actuator, and the height is controlled by the motion of the stance leg. In this case, the foot placement has a nearly linear correlation with forward velocity. Hereby, a generic foot placement controller is developed to control the forward velocity based on the online linear regression analysis of their coupled correlation. Our proposed algorithm is capable of adjusting the control parameters automatically, and is featured by good adaptability and higher control accuracy that outperforms the empirical tuning. The very same controller is able to produce stable hopping with accurate forward velocity tracking even with unknown mass offset, as well as stable bipedal running and walking with accurate velocity tracking.**

I. Introduction

Inspired by the high versatility and adaptability of animals on rough terrains, a lot of research has been done to develop legged robots which are expected to walk and run in a natural environment [1], [2], [3], [4], [5], [6]. One of the most important control actions for the legged robots is to know where to place the foot [7], which affects the stability of the locomotion remarkably. Some works have been done to study the relationship between the foot placement and the system stability [8], [7], in which foot placement indicator and estimator are proposed as a measure of balance.

For bipedal walking, many control methods have been developed based on the zero moment points theory, and the linear inverted pendulum model (LIPM) is often used for deciding the foot placement [9], [10], [11]. The analytic locomotion trajectory can be derived in this way. But it requires the robot to move like a linear inverted pendulum which is not natural.

Biological research revealed that the movement of legged animals can be represented better by a spring loaded inverted pendulum (SLIP) model [12]. Based on this idea, Raibert developed some successful legged robots from a one-legged hopper to a quadruped [13]. He used foot placement to control the forward velocity of these robots by a very simple equation. This method has been applied to many other robots successfully [14], [15]. To improve this method, Raibert proposed tabular control [16] and Russell proposed the approximate optimal control [17]. Both need lots of experimental data and are performed offline. Consequently, they cannot respond quickly to the change of the system or the environment.

To address this problem, we propose a novel foot placement control based on the online linear regression analysis to identify the relation between the foot placement, forward velocity and its change. The proposed method demands very small quantity of empirical data and works online to ensure quick responses to the disturbances. It can track the desired forward velocity accurately, and is not sensitive to the modeling error such as incorrect position of center of mass or the motion profile used by the leg, such as LIPM, SLIP or the like. High adaptation is demonstrated by the successful implementation from one-legged hopping to bipedal running and walking as shown in Fig. 1.

This paper is organized as follows. In Section II, the foot placement control algorithm based on the online linear regression analysis is presented. In Section III, the new method is applied to a one-legged hopper in simulation, and the comparison with Raibert's method is also studied. In Section IV, the method is extended to a bipedal robot for running and walking. The paper ends with conclusions and an outlook to future research.

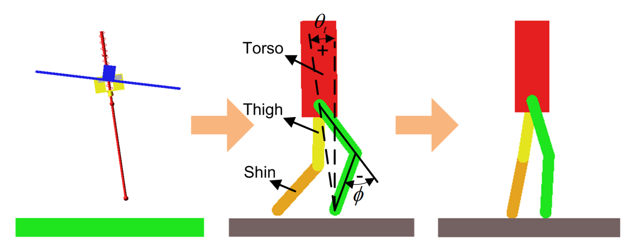

Fig. 1. Generic foot placement control from one-legged hopping to bipedal running and walking.

II. Foot Placement Control Based On Online Linear Regression Analysis

One of the most common control methods for legged robots is the three-part controller proposed by Raibert, in which the body attitude is controlled by the torque applied on the hip joint during the stance, the body height is regulated by the thrust energy injected into the system, and the forward velocity is modulated by adjusting the foot placement [13]. To warrant the overall performance of these controllers, it is very critical to design the foot placement control algorithm carefully. To investigate the relation between the foot placement and the forward velocity, a spring loaded inverted pendulum model (SLIP) can be used for primary analysis which closely resembles the locomotion of legged animals.

The SLIP model contains a point mass located on a spring as Fig. 2. In the figure, the spring is replaced by a force, which is more generic. Its dynamic equation can be written as

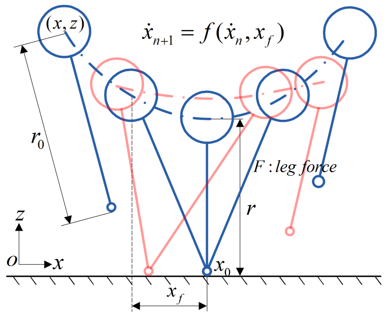

Fig. 2. Simplified model of legged robots.

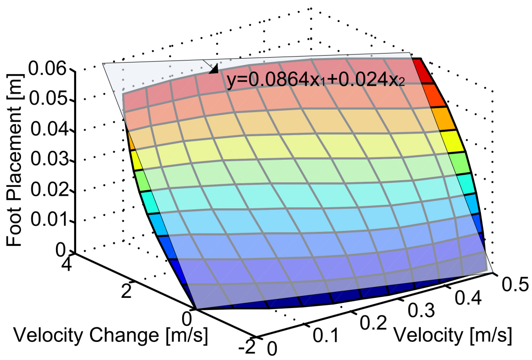

Fig. 3. Correlation between forward velocity and foot placement.

(1), in which m, k, r_0, r and g are separately the mass, spring stiffness, free length and compression length of spring and gravity constant. x_f is the distance between the end of leg and hip joint along the horizontal direction at the first instant of the touch-down, which represents the foot placement. x, and z are the position of the point mass along the horizontal and vertical directions while x_0 and z_0 are the ones of the end of leg.

$$\begin{aligned} \text{Flight}: & \ \ddot{x} = 0; \ddot{z} = -g; \\ \text{Stance}: & \ \ddot{x} = k(x - x_0)(r_0 - r)/(mr); \\ & \ \ddot{z} = k(z - z_0)(r_0 - r)/(mr) - g; \\ & \ r = \sqrt{(x - x_0)^2 + (z - z_0)^2}. \end{aligned} \tag{1}$$

Suppose the total mechanical energy is conserved, let the SLIP model hop one step with different initial forward velocities $\dot{x}$ and foot placements x_f. The relationship of foot placement, forward velocity and its change after one step are computed, as shown in Fig. 3.

By observing the symmetry of running and the influence of foot placement on the forward velocity, Raibert raised a simple algorithm to control forward velocity by the foot placement as (2)

$$x_f = \frac{\dot{x}T_s}{2} + k_{\dot{x}}(\dot{x} - \dot{x}_d), \tag{2}$$

where x_f, T_s, $\dot{x}$ and $\dot{x}_d$ are separately the foot placement, duration of stance phase, current and desired forward velocity. The item $\frac{\dot{x}T_s}{2}$ is the neutral point where the movement is symmetric and forward speed remains unchanged after one step. The item $k_{\dot{x}}(\dot{x} - \dot{x}_d)$ is a feedback to displace the foot from the neutral point to stabilize the forward speed against errors and external disturbances, and to change from one forward speed to another. More details can be found in the work of Raibert [13].

From another perspective, the Raibert's equation (2) can be considered to represent the relationship of foot placement, forward velocity and its change with a constant linear relationship as shown in Fig. 3. By adjusting the coefficients, the duration of the stance phase T_s and the gain $k_{\dot{x}}$, an appropriate plane can be found to represent the nearly linear relationship which can preserve a global stability. But Raibert's method requires the manual tuning of the coefficients thus cannot ensure a high control accuracy for all different speeds.

We are motivated to improve this control paradigm by updating the coefficients online based on the collected measurement when the robot is running. Our idea is that the linearity will be better represented around a local region of the hypersurface rather than the entire nonlinear hypersurface. It should achieve better performance, be more accurate and robust, by representing the nearly linear but actually nonlinear relationship with a set of more accurate and updated local linear surfaces than with a constant global linearization.

We have applied the linear regression analysis in this paper to estimate the correlation among variables. Regarding the foot placement as the dependent variable, and forward velocity and its change as the independent variables, the least squares method is used to estimate their relationship

$$\begin{aligned} & x_f = k_0 + k_1\dot{x} + k_2(\dot{x} - \dot{x}_d) \\ & \begin{bmatrix} k_0 & k_1 & k_2 \end{bmatrix}^T = A \cdot B \\ & A = \begin{bmatrix} I \cdot I^T & \dot{X} \cdot I^T & \Delta\dot{X} \cdot I^T \\ I \cdot \dot{X}^T & \dot{X} \cdot \dot{X}^T & \Delta\dot{X} \cdot \dot{X}^T \\ I \cdot \Delta\dot{X}^T & \dot{X} \cdot \Delta\dot{X}^T & \Delta\dot{X} \cdot \Delta\dot{X}^T \end{bmatrix}^{-1} \\ & B = \begin{bmatrix} I \cdot X_f^T \\ \dot{X} \cdot X_f^T \\ \Delta\dot{X} \cdot X_f^T \end{bmatrix} \\ & I = \begin{bmatrix} 1 & 1 \cdots 1 \end{bmatrix}_{1\times n}; \dot{X} = \begin{bmatrix} \dot{x}_1 & \dot{x}_2 \cdots \dot{x}_n \end{bmatrix} \\ & \Delta\dot{X} = \begin{bmatrix} \Delta\dot{x}_1 & \Delta\dot{x}_2 \cdots \Delta\dot{x}_n \end{bmatrix}; \Delta\dot{x}_i = \dot{x}_i - \dot{x}_{di} \\ & X_f = \begin{bmatrix} x_{f1} & x_{f2} \cdots x_{fn} \end{bmatrix}; i = 1, 2 \cdots n. \end{aligned} \tag{3}$$

In (3), a linear model with three coefficients (k_0, k_1, k_2) is used to represent the relationship and decide the foot placement x_f according to present velocity $\dot{x}$ and desired velocity $\dot{x}_d$. Compared with Raibert's equation (2), the coefficient k_0 is added into the linear model to handle the possible modeling error such as the incorrect position of center of mass or other unknown asymmetric features that needs an offset in foot placement to eliminate their influence.

For the regression analysis, n sets of data are used. Each set of data consists of three elements: forward velocity before the leg touching down the ground $\dot{x}_i$, the foot placement at touching down x_{fi}, and forward velocity change after one step $\Delta\dot{x}_i$. These data are collected when the robot is

TABLE I

SELECTIVE FILTERING OF SAMPLED DATA (PSEUDO CODE).

if $|\dot{x}_i - \dot{x}_l| < \dot{x}^{th}$ AND $|x_{fi} - x_{fl}| < x_f^{th}$
AND $|\Delta\dot{x}_i - \Delta\dot{x}_l| < \Delta\dot{x}^{th}$, $(i = 1, 2, ...n)$
    Delete data $(\dot{x}_l, x_{fl}, \Delta\dot{x}_l)$
else
    Add data $(\dot{x}_l, x_{fl}, \Delta\dot{x}_l)$ to the queue

in action, and are used to estimate the relationship online and decide where to place the foot.

To ensure that the linear model represents well the current local correlation, the latest data are used for regression analysis. One way to update the data is to use a queue that stores n sets of data, and load the latest one while delete the oldest one continuously. However, the data stored in the queue cannot be too similar, because even if the relationship of $\dot{x}$, x_f and $\dot{x}_d$ is almost constant but is still affected by other factors such as body height and attitude control. Therefore, if the sets of data used by the regression analysis are very similar, the relationship of $\dot{x}$, $\dot{x}_f$ and $\dot{x}_d$ will be downgraded, and the coupling effects of other factors will become obvious which will make the linear model incorrect and unstable. Moreover, the similar sets of data imply the sampling of the nonlinear hyperspace in a singular point. Hence, a linear model cannot be properly established to approximate the local linearity.

To avoid this, a data filter algorithm is developed and the pseudo code is presented in Table I. The data $(\dot{x}_i, x_{fi}, \Delta\dot{x}_i)$ and $(\dot{x}_l, x_{fl}, \Delta\dot{x}_l)$ are respectively the data stored in the queue and the one just collected. Define the thresholds $(\dot{x}^{th}, x_f^{th}, \Delta\dot{x}^{th})$ to determine whether or not two sets of data are the same and prevent them from storing in the queue at the same time. The value of the thresholds will affect the steady-state accuracy and the robustness of forward velocity tracking. If these thresholds are too big, the data used to estimate the relationship will be too far away from each other to form a precise local linear model. Otherwise, if too small, the filter algorithm cannot work well to keep the linear model stable. So the thresholds should be chosen carefully to keep a balance between steady-state accuracy and robustness.

The quantity n of the data used for regression analysis will also has an influence on the control performance. If n is too big, the estimated linear model will be refreshed very slowly and cannot response quickly to the change of the system. On the other hand, if n is too small, the estimated linear model will update so drastically that may result in instability.

III. ONE-LEGGED HOPPING

To evaluate the performance of our proposed foot placement control, simulations of a planar one-legged robot as Fig. 4 are studied in the ADAMS software. The parameters of this robot are listed in TABLE II. It consists of two parts: a body and a telescopic leg. There are one rotary actuator in the hip joint and one spring-damper element in the telescopic leg which can be preloaded. The body attitude is controlled by actuating the hip torque and the hopping height is controlled

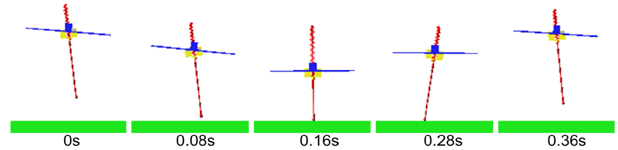

Fig. 4. Control of one-legged hopping.

TABLE II

PARAMETERS OF PLANAR ONE-LEGGED ROBOT.

Parameter	Value	Parameter	Value
Mass of Body	0.4(kg)	Spring Stiffness	80(N/m)
Inertia of Body	1.75e-3(kg m^2)	Spring Damping	0.8(Ns/m)
Mass of Leg	7.8e-3(kg)	Length of Leg	0.3(m)
Inertia of Leg	8.23e-5(kg m^2)	Spring Preload	0.07(m)

by the preload in the flight and the release of spring in the stance phase. In this study, the preload is constant and the hopping height will converge to different stable states corresponding to different forward velocities.

To implement the novel foot placement control algorithm, 6 sets of data are stored and updated in the queue for the online linear regression analysis. The limits $\dot{x}^{th}$, x_f^{th}, and $\Delta\dot{x}^{th}$ are 0.05 m/s, 0.03 m, and 0.05 m/s respectively. To compare our algorithm with Raibert's, the best control coefficients are tuned for Raibert's method as in (4). The initial data chosen for linear regression analysis are listed in TABLE III to ensure that two methods have the same control coefficients at the beginning.

$$x_f = \frac{\dot{x}T_s}{2} + k_{\dot{x}}(\dot{x} - \dot{x}_d) = 0.092\dot{x} + 0.03(\dot{x} - \dot{x}_d) \quad (4)$$

Our case study targets at controlling the robot to hop at three speeds, 0.1 m/s, 0.3m/s and 0.5m/s. The simulation results of the two methods are shown in Fig. 5. Raibert's method can keep a small stead-state error at the speed of 0.1 m/s while not at another two speeds. Our proposed method can update the coefficients as in (3) based on the collected data during hopping and keep a very small stead-state error for all the three speeds. Fig. 6 shows the update of the coefficients to eliminate the steady-state error in the new method. Our proposed foot placement control can guarantee the high accuracy and represent better the local relationship of foot placement, forward velocity and its change, compared with the fixed one provided by Raibert's method.

As mentioned in Section II, the coefficient k_0 is added to our algorithm to eliminate the possible modeling error such as unknown mass offset. To validate this, the center of mass of the body is offsetted as in Fig. 7. The control coefficients of Raibert's method and initial data of our method are the same as above. The robot is expected to hop at the speed of 0.5 m/s after 5 seconds of transition. The forward velocity control result in Fig. 8 indicates that when the unknown mass offset exists, the steady-state error of forward velocity becomes so significant for Raibert's method that the direction of travel is unavoidably reversed. However, controlled by our proposed algorithm, the robot hops first backward for some steps to prevent falling due to the unknown mass

TABLE III

INITIAL DATA FOR LINEAR REGRESSION ANALYSIS.

Num.	$\dot{x}$	$\Delta\dot{x}$	x_f	Num.	$\dot{x}$	$\Delta\dot{x}$	x_f
1	0.000	1.000	0.030	4	0.000	1.000	0.030
2	1.000	0.000	0.092	5	1.000	0.000	0.092
3	0.000	0.000	0.000	6	0.000	0.000	0.000

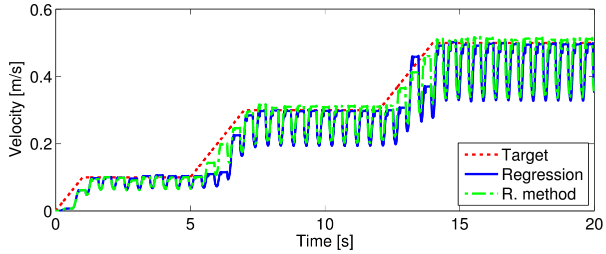

Fig. 5. Forward velocity comparison of two methods.

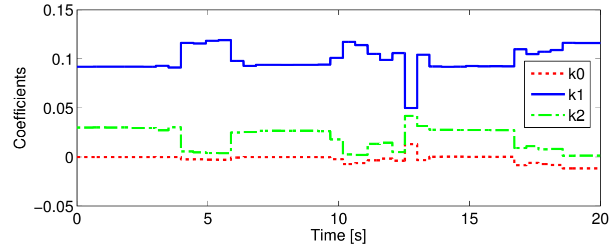

Fig. 6. Online coefficient adaptation of one-legged hopper.

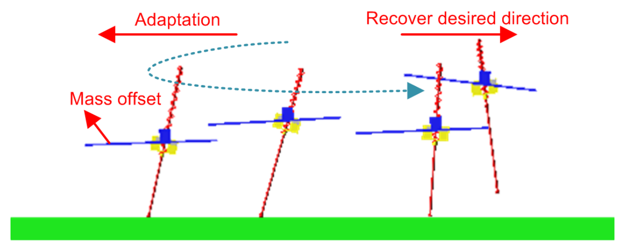

Fig. 7. Adaptive control of one-legged hopping with unknown mass offset.

offset, then adapts to the foot placement estimation and recovers its hopping direction. The forward velocity finally converges to the desired one. This process is also shown in Fig. 7. The online update of coefficients is shown in Fig. 9. The coefficient k_0 is not zero any more when the mass offset exists. The simulation result proves that our new foot placement control algorithm is more robust and adaptive to modeling error.

In Section II, the effect of the number of data samples n on the online regression analysis is also discussed. To confirm it, two simulations are performed on the robot with mass offset using 6 and 12 sets of data separately. The different responses to the unknown mass offset is shown in Fig. (10). Obviously, the result supports well the aforementioned assumption that forward velocity with more sets of data fluctuates less but takes more time to reach the desired state.

IV. EXTENSION TO BIPEDAL RUNNING AND WALKING

The key point of the foot placement control with online linear regression analysis is that the body attitude and height are controlled by independent controllers so that the foot placement can have a nearly linear relationship with forward velocity and its change which can be represented well by an estimated local linear model. This can give us more freedom to explore the diversity of the leg extension instead of behaving like a spring-damper. It can be applied to not only one-legged hoppers but also bipedal robots, for both running and walking.

To prove this, a bipedal robot is built in simulation. Its mechanical parameters are listed in TABLE IV. ϕ and θ_t marked in Fig. 1 are the knee angle and touch-down angle separately. Touch-down angle is the angle between the vertical line and the line crossing the hip joint and the end of stance leg at the first moment of touching down. Because the swing leg has less impact on the system than the stance one, the discussion about bipedal controller is focused on the stance leg.

The attitude of torso is controlled by the torque in the hip joint. Instead of using a telescopic leg with a spring-damper element as that of the one-legged robot, the bipedal robot controls its height by the knee joint. A three-order polynomial module (5) is used to plan the trajectory of the knee joint for both running and walking. After the end of stance, the leg will retract. The equivalent stiffness and damping of the knee joint are 287 Nm/rad and 5.7 Nms/rad.

$$\begin{aligned}\phi(t) &= \phi(0) + \dot{\phi}(0)t \\ &- \frac{3\phi(0) - 3\phi(T) + 2T\dot{\phi}(0) + T\dot{\phi}(T)}{T^2}t^2 \\ &+ \frac{2\phi(0) - 2\phi(T) + T\dot{\phi}(0) + T\dot{\phi}(T)}{T^3}t^3,\end{aligned} \tag{5}$$

where T is the stance time, $\phi(t)$ is the expected angle of knee joint at the time of t after touching down, $\phi(0)$ and $\dot{\phi}(0)$ are the angle and angular speed of knee joint at the first moment of touching down, and $\phi(T)$ and $\dot{\phi}(T)$ are the expected angle and angular speed at the end of the stance phase. For walking, $\phi(T) = 0$ rad, $\dot{\phi}(T) = 0$ rad/s, $T = 0.4$ s; for running, $\phi(T) = -0.4$ rad, $\dot{\phi}(T) = 0$ rad, $T = 0.3$ s. $\phi(0)$ and $\dot{\phi}(0)$ are measured at the first moment of touch-down but $\phi(0)$ will be largely affected by the motion of knee joint when the leg is swinging.

The difference between running and walking for the bipedal robot is the trajectory of the knee joint. In running, $\phi(0)$ is controlled to -1 rad for all speed. But for walking, it is controlled to a value decided by the expected velocity as the equation $\phi(0) = -0.1 - 0.4v_d$. A simple linear equation is used here to make sure that no energy is over injected by the knee motion to prevent the stance leg from lifting off the ground, and there is enough clearance to touch down with an expected touch angle. The stance time of running is a little shorter than walking, and the knee angle of running is changing in a smaller range from -1 rad to -0.4 rad. Both are designed to ensure that the robot can lift off the ground after stance which is necessary for running. The trajectories of knee joint mentioned above are the simple solutions for

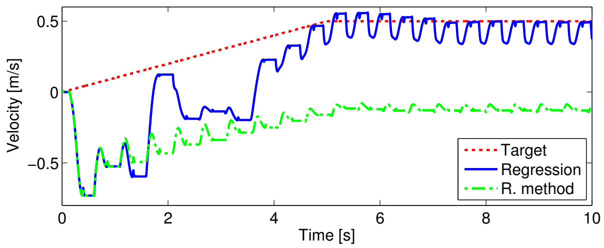

Fig. 8. Forward velocity control with mass offset.

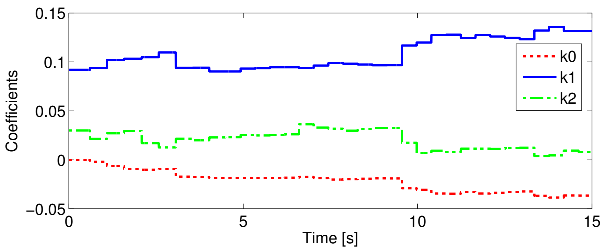

Fig. 9. Online coefficient adaptation of one-legged hopper with mass offset.

walking and running, yet effective while working with the proposed foot placement control algorithm.

For linear regression analysis, the touch-down angle θ_t represents the foot placement as a dependent variable, and forward velocity of hip joint and its change as independent variables. The distance of stance leg projected on the ground also can be used to represent the foot placement as in the one-legged robot. But touch-down angle is easier to calculate for the bipedal robot and it will be shown that our algorithm is not sensitive to which variable is used as long as the nearly linear relationship exists. The least square fitting method is used to estimate the linear relationship of (6), in which θ_t is touch-down angle, $\dot{x}$ is the velocity of hip joint before touch-down and $\dot{x}_d$ is the desired velocity.

For the data collected for regression analysis, $\dot{x}_d$ is the velocity of hip joint before touching down of next step. k_0, k_1, k_2 are the coefficients calculated from the regression analysis and a queue with 6 sets of data is used. The thresholds $\dot{x}^{th}$, θ_t^{th}, $\Delta\dot{x}^{th}$ to selectively filter data are 0.05 m/s, 0.02 rad and 0.05 m/s.

$$\theta_t = k_0 + k_1\dot{x} + k_2(\dot{x} - \dot{x}_d) \tag{6}$$

First, the bipedal robot is controlled to walk at 0.5 m/s after 5 seconds of transition. Fig. 11 is the snapshots of the successful walking. The walking velocity is tracked accurately and stably. The coefficients also converge to stable values when the velocity is stable. The walking velocity and online refreshing of the coefficients are shown in the Fig. 12 and Fig. 13.

Further, the bipedal robot is controlled to run at 1 m/s after 5 seconds of transition. Fig. 14 is the snapshots of stable running. With the same control algorithm, the robot can reach the velocity accurately and stably, although there are some velocity spikes in each step due to the large impacts between the leg and the ground during the touch-down. The

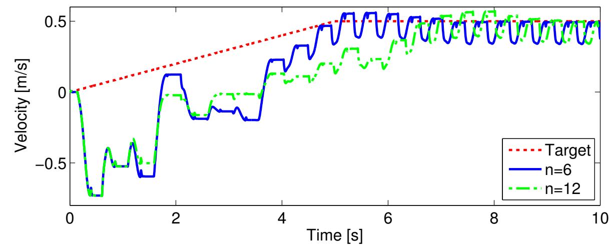

Fig. 10. Comparison between different numbers of sampled data.

TABLE IV

PARAMETERS OF PLANAR BIPEDAL ROBOT.

Part	Mass(kg)	Inertia(kg*m^2)	Length(m)
Torso	19.8	0.3	0.4
Thigh	2.8	0.02	0.25
Shin	2.5	0.02	0.25

coefficients also converge to stable values when the velocity stabilizes. The running velocity and the online update of the coefficients can be seen in the Fig. 15 and Fig. 16.

It should be noted that the leg length of the simulated biped is 0.5m, so the running speed of 1 m/s corresponds to a normalized speed of 2 leg/s. In other words, a speed of 2 m/s for a human adult, which is similar to an average speed of jogging.

V. CONCLUSION

In this paper, a unified and adaptive foot placement control algorithm is proposed for legged robots based on the online linear regression analysis. The essential is that if the body attitude and height are properly controlled, the foot placement then has a nearly linear relationship with forward velocity and its change, which can be locally represented well by an estimated linear model.

This permits more versatile control for the stance leg as long as the nearly linear relationship preserves. With very small quantity of measured data, our proposed algorithm is capable of adjusting the control parameters automatically and responding quickly to any disturbances. Its good adaptability and higher control accuracy also outperform the empirical tuning approach. The very same controller is able to produce stable hopping with accurate forward velocity tracking even with unknown mass offset, as well as stable bipedal running and walking with accurate velocity tracking.

Future research will be carried out in two aspects. First, the initial data for regression analysis here is given by some random testing data and the same for different targeted speeds. To shorten the response time and speed up the velocity convergence, reliably identified data from prior trials can be stored and recalled as the initial condition for the regression analysis. Secondly, since the foot placement has a nearly linear relationship with forward velocity and its change, it will take only several iterations to calculate a more accurate foot placement using a numerical method on a dynamical model. The foot placement provided by the linear regression analysis can be used as the initial values

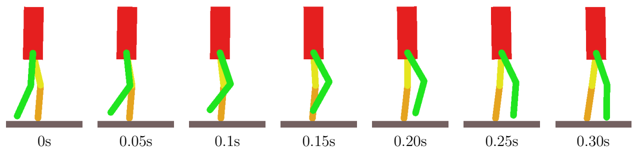

Fig. 11. Snapshots of bipedal walking.

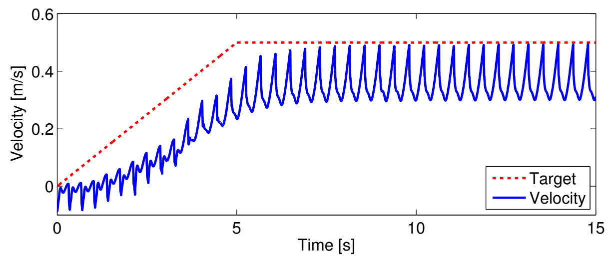

Fig. 12. Forward velocity of bipedal walking.

for the iteration. Similar methods can also be used to other control targets besides the foot placement control. All these suggested future work will lead us towards more accurate and robust locomotion control of legged robots.

Acknowledgment

This work is supported by the FP7 European project WALK-MAN (ICT 2013-10).

References

[1] M. Ahmadi and M. Buehler, “Stable control of a simulated one-legged running robot with hip and leg compliance,” *IEEE Transactions on Robotics and Automation*, vol. 13, no. 1, pp. 96–104, 1997.

[2] T. McGeer, “Passive dynamic walking,” *the international journal of robotics research*, vol. 9, no. 2, pp. 62–82, 1990.

[3] J. Pratt and G. Pratt, “Intuitive control of a planar bipedal walking robot,” in *IEEE International Conference on Robotics and Automation*, vol. 3, 1998, pp. 2014–2021.

[4] I. Havoutis, C. Semini, and D. G. Caldwell, “Virtual model control for quadrupedal trunk stabilization,” *Dynamic Walking*, 2013.

[5] M. Hutter, C. D. Remy, M. A. Hoepflinger, and R. Siegwart, “Scarleth: Design and control of a planar running robot,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 562–567.

[6] Z. Li, B. Vanderborght, N. G. Tsagarakis, and D. G. Caldwell, “Fast bipedal walk using large strides by modulating hip posture and toe-heel motion,” in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2010, pp. 13–18.

[7] P. van Zutven, D. Kostic, and H. Nijmeijer, “Foot placement for planar bipeds with point feet,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 983–988.

[8] D. L. Wight, E. G. Kubica, and D. W. Wang, “Introduction of the foot placement estimator: A dynamic measure of balance for bipedal robotics,” *Journal of computational and nonlinear dynamics*, vol. 3, no. 1, p. 011009, 2008.

[9] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, “Biped walking pattern generation by using preview control of zero-moment point,” in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, 2003, pp. 1620–1626.

[10] C. G. A. Salman Faraji, Soha Pouya and A. J. Ijspeert, “Versatile and robust 3d walking with a simulated humanoid robot (atlas) a model predictive control approach,” *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[11] Z. Li, N. G. Tsagarakis, and D. G. Caldwell, “Walking trajectory generation for humanoid robots with compliant joints: Experimentation with coman humanoid,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 836–841.

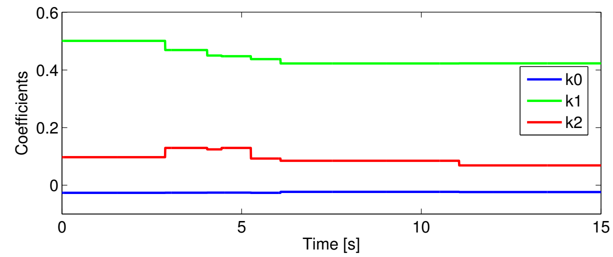

Fig. 13. Online coefficient adaptation of bipedal walking.

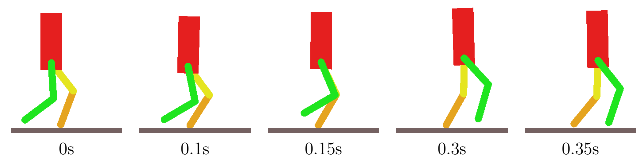

Fig. 14. Snapshots of bipedal running.

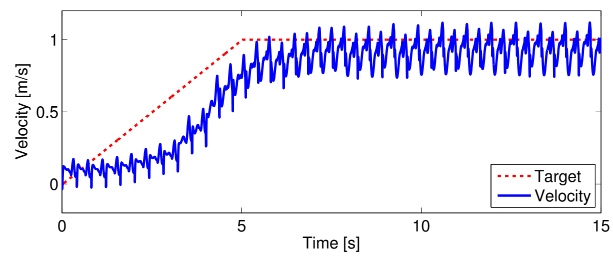

Fig. 15. Forward velocity of bipedal running.

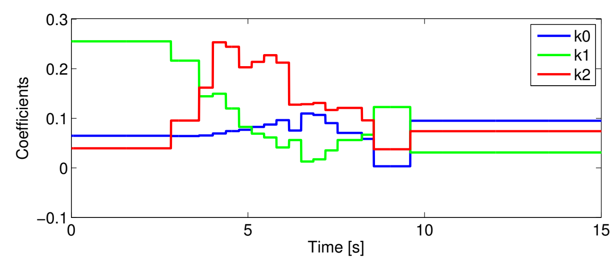

Fig. 16. Online coefficient adaptation of bipedal running.

[12] H. Geyer, A. Seyfarth, and R. Blickhan, “Compliant leg behaviour explains basic dynamics of walking and running,” *Proceedings of the Royal Society B: Biological Sciences*, vol. 273, no. 1603, pp. 2861–2867, 2006.

[13] M. H. Raibert *et al.*, *Legged robots that balance*. MIT press Cambridge, MA, 1986, vol. 3.

[14] M. Raibert, K. Blankespoor, G. Nelson, R. Playter *et al.*, “Bigdog, the rough-terrain quadruped robot,” in *Proceedings of the 17th World Congress*, vol. 17, no. 1, 2008, pp. 10 822–10 825.

[15] D. W. Marhefka, D. E. Orin, J. P. Schmiedeler, and K. J. Waldron, “Intelligent control of quadruped gallops,” *Mechatronics, IEEE/ASME Transactions on*, vol. 8, no. 4, pp. 446–456, 2003.

[16] M. H. Raibert and F. C. Wimberly, “Tabular control of balance in a dynamic legged system,” *Systems, Man and Cybernetics, IEEE Transactions on*, no. 2, pp. 334–339, 1984.

[17] R. L. Tedrake, “Applied optimal control for dynamically stable legged locomotion,” Ph.D. dissertation, Massachusetts Institute of Technology, 2004.