Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Balancing a Legged Robot Using State-Dependent Riccati Equation Control

E. Najafi [*,**] **G.A.D. Lopes** [*] **R. Babuška** [*]

[*] *Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands.*
{e.najafi; g.a.delgadolopes; r.babuska}@tudelft.nl
[**] *Department of Mechatronics Engineering, Faculty of New Sciences and Technologies, University of Tehran, Iran.*

Abstract: In this paper, we propose a nonlinear control approach for balancing underactuated legged robots. For the balancing task, the robot is modeled as a generalized version of a Segway. The control design is based on the State-Dependent Riccati Equation (SDRE) approach. The domain of attraction of the SDRE controller is compared to the domain of attraction of a linear quadratic controller. Using a simulation example of a four-legged robot balancing on its hind legs, we show that the SDRE controller gives a reasonably large domain of attraction, even with realistic level constraints on the control input, while the linear quadratic controller is unable to stabilize the system.

Keywords: State-dependent Riccati equation control, Linear quadratic control, Legged robots, Domain of attraction.

1. INTRODUCTION

Designing motion controllers for ground robots that need to traverse unstructured terrains is a challenging problem. For example, a legged robot should be able to execute different tasks such as walking, jumping, running and balancing. As legged robots are efficient in navigating various terrains, legged locomotion has been the subject of extensive research, see, for instance, (Saranli et al., 2001; Raibert et al., 2008; Kalakrishnan et al., 2011). In this paper, we address stabilizing of a four-legged RHex-type robot (Weingarten et al., 2004), called RQuad (Lopes et al., 2009) in its upright position. This task is relevant to traversing different terrains and climbing large obstacles. In addition, the robot can observe its environment standing on its hind legs. The robot is equipped with rotating legs of semicircular shape, see Fig. 1. During the balancing task, the hind legs of the RQuad robot roll over the ground in a way similar to wheels rotating around a pivot point placed off the wheel center, as schematically shown in Fig. 2. In this sense, the task can be regarded as a more complex variant of Segway balancing.

The control of the Segway, wheeled inverted pendulums and ballbots, which are all similar platforms, has been investigated in the literature (Grasser et al., 2002; Nagarajan et al., 2012). It has been shown that although the Segway is a nonlinear underactuated system, it can be controlled by linear controllers (Pathak et al., 2005). However, when the pivot point moves from the wheel's center toward the wheel's rim (Fig. 2), the nonlinearity of the system becomes more severe. As a result, the linear controller's domain of attraction (DA) [1] shrinks quickly and so linear control is not adequate for the legged robot. Therefore, one has to resort to nonlinear controllers in order to control the robot in a sufficient range of its configuration space.

[1] The DA of a controller is a set of initial states such that each trajectory starting from the set, finally converges to the controller's desired goal.

Fig. 1. Four-legged robot RQuad developed at Delft Center for System and Control, TU Delft.

In this paper, we investigate the nonlinear counterpart of the linear quadratic regulator (LQR), which is called *state-dependent Riccati equation* (SDRE) controller (Cloutier, 1997). This approach has been successfully applied to various nonlinear systems, see, for example, (Dang and Lewis, 2005) where an inverted pendulum is controlled using SDRE controller.

The paper is organized as follows. A brief review of the Segway model and the legged robot is presented in Section 2. In Section 3, a standard LQR is designed to stabilize the legged robot in the balancing position. In addition, the performance of the LQR in terms of its DA is investigated. Section 4, first, reviews the main concepts of the SDRE controller. Then, an SDRE controller is designed to stabilize the robot in its upright position. Moreover, the DA of the SDRE controller is compared with the DA of the LQR. Finally, Section 5 concludes the paper.

2. PHYSICAL MODEL OF BALANCED LEGGED ROBOT

In order to design an SDRE controller, which is a model-based controller, we need to develop a mathematical model of the balancing robot in the sagittal plane. Denote by $q = [\alpha\ \phi]^T$ the vector of the configuration variables, with α the angle between the body and the vertical axis and ϕ the wheel angle, as illustrated in Fig. 2. Let c be the ratio of the pivot point distance from the wheel's center to the wheel's radius. Parameter c can be set to a value between zero and one to make a gradual transition from the wheel to a semi-circular leg. As c increases, the nonlinearity of the system becomes more severe.

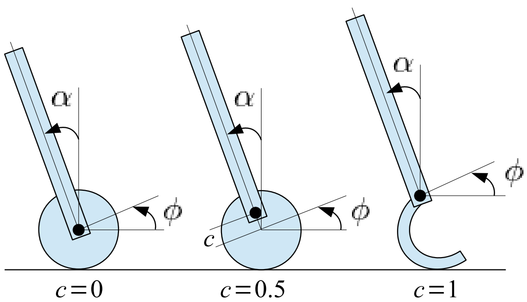

Fig. 2. The transition from balancing a wheeled robot (Segway) to balancing a RHex-type legged robot. Parameter c denotes the ratio of the pivot point distance from the wheel's center to the wheel's radius.

Using Euler-Lagrange equations, the following model is obtained:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = B(q)\tau + F_{ext} \tag{1}$$

with the inertia matrix $M(q) \in \mathbb{R}^{n\times n}$, the Coriolis and centrifugal forces matrix $C(q,\dot{q}) \in \mathbb{R}^{n\times n}$, the gravity matrix $G(q) \in \mathbb{R}^n$, the input vector $B(q) \in \mathbb{R}^n$ and the vector of the external forces $F_{ext} \in \mathbb{R}^n$. These matrices are given by:

$$\begin{aligned}
M(q) &= \begin{bmatrix} m_{\mathrm{b}}l^2 + I_{\mathrm{b}} & m_{\mathrm{b}}rl\cos(\alpha) + M_1 \\ m_{\mathrm{b}}rl\cos(\alpha) + M_1 & (m_{\mathrm{w}} + m_{\mathrm{b}})r^2 + I_{\mathrm{w}} + M_2 \end{bmatrix} \\
C(q,\dot{q}) &= \begin{bmatrix} 0 & C_1 \\ -m_{\mathrm{b}}rl\dot{\alpha}\sin(\alpha) + C_2 & C_3 \end{bmatrix} \\
G(q) &= \begin{bmatrix} -m_{\mathrm{b}}lg\sin(\alpha) \\ G_1 \end{bmatrix} \\
B(q) &= [-2\ \ 2]^T \\
F_{ext} &= \begin{bmatrix} 2b_j(\dot{\phi} - \dot{\alpha}) & -2b_g - 2b_j(\dot{\phi} - \dot{\alpha}) \end{bmatrix}^T
\end{aligned} \tag{2}$$

with

$$\begin{aligned}
M_1 &= cm_{\mathrm{b}}rl\cos(\alpha - \phi), & M_2 &= c^2m_{\mathrm{b}}r^2 + 2cm_{\mathrm{b}}r^2\cos(\alpha), \\
C_1 &= cm_{\mathrm{b}}rl\dot{\phi}\sin(\alpha - \phi), & C_2 &= -cm_{\mathrm{b}}rl\dot{\alpha}\sin(\alpha - \phi), \\
C_3 &= -cm_{\mathrm{b}}r^2\dot{\phi}\sin(\phi), & G_1 &= -cm_{\mathrm{b}}rg\sin(\phi).
\end{aligned} \tag{3}$$

The symbols are explained in Table 1. It also lists the values of the RQuad's physical parameters. These values were found partly by measuring and partly estimated using nonlinear system identification.

While the RQuad robot stands on its hind legs, the motors of both legs are controlled with the same control input. This fact is modeled by the matrix $B(q)$ which doubles the torque value that one motor exerts on a leg. The torque τ delivered by the motors is given by the following model of the DC motor:

$$\begin{aligned}
\tau &= N_{\mathrm{g}}\tau_{\mathrm{m}} \\
\tau_{\mathrm{m}} &= \frac{K_{\mathrm{t}}}{R_{\mathrm{m}}}(V - K_{\mathrm{e}}\dot{\theta}_{\mathrm{m}}) \\
\dot{\theta}_{\mathrm{m}} &= N_{\mathrm{g}}(\dot{\phi} - \dot{\alpha}) \\
V &= N_s v.
\end{aligned} \tag{4}$$

The subscript m refers to the motor variables and v is the control input normalized in the range $[-1, 1]$. The matrix F_{ext} includes the friction force in the pivot joint and the rolling friction force due to the contact between the legs and the ground. We assume that the legs do not slip. Additionally, though the RQuad's legs are made of a flexible material, in the model they are considered as rigid bodies.

Table 1. Physical parameters of the RQuad robot.

Physical parameter	**Symbol**	**Value**	**Unit**
Body inertia	I_{b}	$2.60 \cdot 10^{-2}$	$\mathrm{Kgm^2}$
Body mass	m_{b}	2.55	Kg
Wheel inertia	I_{w}	$4.54 \cdot 10^{-4}$	$\mathrm{Kgm^2}$
Wheel mass	m_{w}	$3 \cdot 10^{-1}$	Kg
Gravity	g	9.81	$\mathrm{ms^{-2}}$
Body half length	l	$13.8 \cdot 10^{-2}$	m
Wheel radius	r	$5.5 \cdot 10^{-2}$	m
Rolling friction	b_g	$6.5 \cdot 10^{-3}$	Nms
Joint friction	b_j	$1 \cdot 10^{-4}$	Nms
Torque constant	K_{t}	$3.15 \cdot 10^{-2}$	$\mathrm{NmA^{-1}}$
Back EMF	K_{e}	$3.15 \cdot 10^{-2}$	$\mathrm{NmA^{-1}}$
Rotor resistance	R_{m}	5.21	Ω
Gear ratio	N_{g}	35	–
Input gain	N_s	25	V

Note that by setting $c = 0$ functions M_i, C_i and G_i in equations (3) become equal to zero and we obtain the standard Segway model.

3. LQR AND DOMAIN OF ATTRACTION ANALYSIS

We first investigate the equilibrium manifolds of the Segway and the legged robot. Then, an LQR is designed to stabilize the RQuad robot in the balancing position. In order to study the performance of the controller, its domain of attraction is investigated for different values of parameter c. It is well known in the literature that the DA of a nonlinear controller is a complex set, see, e.g., (Vannelli and Vidyasagar, 1985; Chiang et al., 1988). Generally, an analytical representation is only found for trivial systems as discussed by Genesio et al. (1985). In this paper, we simulate the control system with various initial conditions to approximate the DA of the controller.

3.1 Equilibrium Manifold

For $c = 0$, the pivot is in the wheel's center and the legged robot is equivalent to an ordinary Segway. To stabilize the Segway in the upright position, the angle α between the

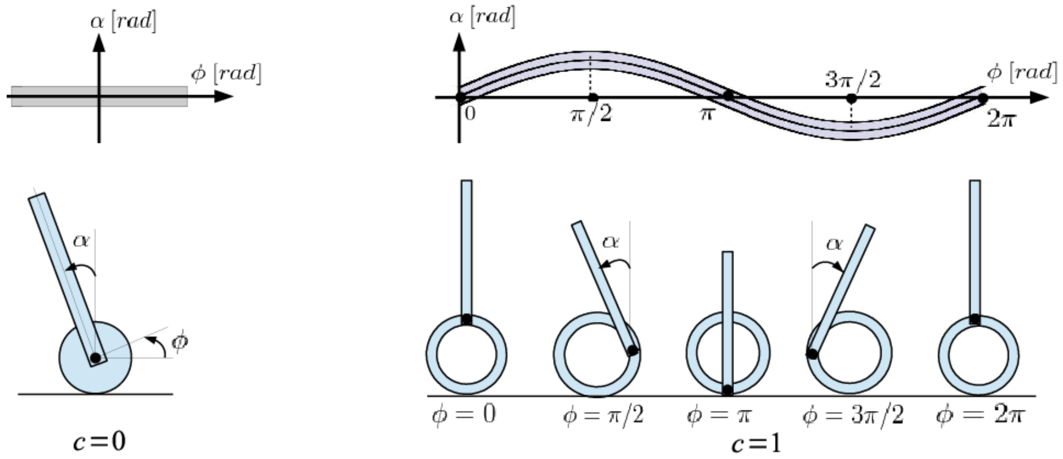

Fig. 3. The equilibrium manifold of the robot when $c = 0$ (Segway) and when $c = 1$ (legged robot).

body and the vertical axis has to be zero, but the wheel angle ϕ can take any value. The equilibrium manifold of the Segway is shown at the left side of Fig. 3. This manifold illustrates that the equilibrium points of the Segway lie at the zero value of the body angle.

If the pivot point moves from the wheel's center to the wheel's rim, the position of the equilibrium point will depend on the robot's configuration. The robot will be stable if the centers of mass of the body and the wheel lie exactly above each other and the angular velocities are zero as well, see the right side of Fig. 3. The goal of controller is to stabilize the robot in one of the equilibrium points (not necessarily at the zero angles).

3.2 Linear Quadratic Controller

We first design an LQR to stabilize the RQuad robot in the balancing position. However, we expect the performance of the linear controllers to be limited as parameter c increases. The main purpose of this section is to analyze the effect of c on the controller's performance with respect to its DA.

Define the state vector of the system as:

$$x = \left[\alpha \ \phi \ \dot{\alpha} \ \dot{\phi}\right]^T \tag{5}$$

where $\dot{\alpha}$ and $\dot{\phi}$ are the angular velocities of the body and the wheel, respectively. The equations of motion are written in the general input-affine nonlinear state-space form:

$$\dot{x} = f(x) + g(x)u. \tag{6}$$

To design a linear controller, such as LQR, functions $f(x)$ and $g(x)$ have to be linearized around the operating point of the system (chosen at $x = 0$ here), yielding:

$$\dot{x} = Ax + Bu. \tag{7}$$

Given matrices A and B, one can now design an LQR described by:

$$u = -(R^{-1}B^T P)x = -Kx, \tag{8}$$

where R is a positive definite matrix and P is the solution of the algebraic Riccati equation:

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \tag{9}$$

whit Q is a positive definite matrix.

3.3 LQR: Domain of Attraction Analysis

First an LQR is designed based on the linearized state space model. The value of parameter c determines the position of the pivot joint. In order to investigate the performance of the controller, we compute its DA for different values of c, using numerical simulation. To compute the DA, the controller is applied to stabilize the robot starting at different initial states. The initial value of the body angle ranges from $-\pi/2$ rad to $+\pi/2$ rad. Similarly, the initial value of the wheel angle is defined within the range of -30 rad to $+30$ rad. These ranges are discretized at 100 samples. The discretization step of α and ϕ therefore equals to 0.031 rad and 0.6 rad, respectively.

Fig. 4 illustrates how the DA of the LQR changes regarding to parameter c. Observe that the DA shrinks as the nonlinearity increases. When c changes from 0 to 0.25, the DA shrinks slightly, but for c increased to 0.75 the DA becomes very small. In conclusion, as the system's nonlinearity becomes more severe, the LQR controller cannot stabilize the system properly. Therefore, we implement the SDRE controller, which is a nonlinear counterpart of LQR.

4. SDRE CONTROLLER AND DOMAIN OF ATTRACTION ANALYSIS

This section first reviews the main concepts of the SDRE method. Then, an SDRE controller is designed to stabilize the RQuad robot in the balancing position. In addition, the DA of the controller is plotted for different values of parameter c. Finally, the result achieved for both the LQR and the SDRE controller are compared with respect to the nonlinearity of the system.

4.1 SDRE Controller

State-Dependent Riccati Equation controller is a nonlinear counterpart of the LQR, see (Çimen, 2008). This method has recently become popular as it provides an algorithm for designing nonlinear state feedback controllers. The SDRE based controllers accept nonlinearity for the state space model. They also offer the flexibility of using state-dependent design matrices $Q(x)$ and $R(x)$. SDRE method

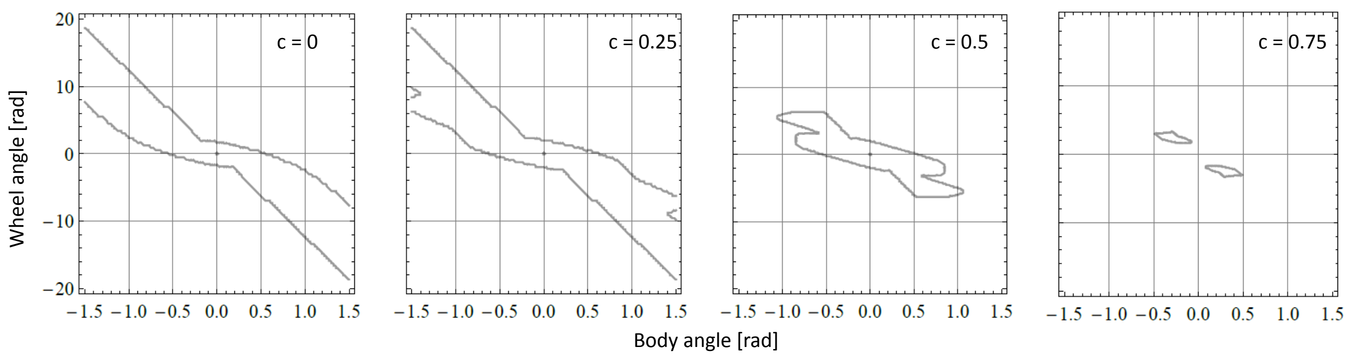

Fig. 4. The domain of attraction of LQR for different values of parameter c.

has been developed for infinite horizon suboptimal problems in nonlinear systems that are autonomous and observable. The state-space equations are input-affine in the form of (6) and the task is to minimize the performance index defined as:

$$J = \frac{1}{2}\int_{t_0}^{\infty} \left(x^T Q(x)x + u^T R(x)u\right)dt, \tag{10}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $f(x) \in \mathbb{R}^n$, $g(x) \in \mathbb{R}^k$, $Q(x) \in \mathbb{R}^{n\times n}$ and $R(x) \in \mathbb{R}^{m\times m}$, with the assumptions that $f(0) = 0$ and $g(x) \neq 0$, $\forall x$. To apply the SDRE method, the nonlinear system defined in (6) needs to be represented in a state-dependent linear-like structure:

$$\dot{x} = A(x)x + B(x)u, \tag{11}$$

where $A(x)$ and $B(x)$ are the state-dependent matrices. Under the assumption $f(0) = 0$ and $f(x) \in \mathbb{R}^n$, the state-dependent matrix $A(x) : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ can be found by mathematical manipulation; however, $A(x)$ is not unique. This fact offers some flexibility in the control design. In any case, $A(x)$ has to be selected such that the full state measurement for the system is available and the pair $(A(x), B(x))$ is point-wise controllable (Dang and Lewis, 2005).

The main concept of computing the control law in SDRE is solving the algebraic Riccati equation:

$$\begin{aligned} A(x)^T P(x) + P(x)A(x) - P(x)B(x)R(x)^{-1} \\ B(x)^T P(x) + Q(x) = 0 \end{aligned} \tag{12}$$

to obtain $P(x) \geq 0$, which is then applied in the feedback control law given as:

$$U = -K(x)x \tag{13}$$

with $K(x) = R(x)^{-1}B(x)^T P(x)$. The main difference between an SDRE controller and LQR is the fact that, in the SDRE controller the matrices $A(x)$ and $B(x)$ and also the weighting matrices $Q(x)$ and $R(x)$ are state-dependent. The SDRE method provides flexibility in choosing the weighting matrices which can lead to optimize the control accuracy and the control effort. However, the obtained controller is not necessarily optimal with respect to the performance index (10) (Çimen, 2010). Additionally, the closed-loop matrix $A_{CL}(x)$ which is computed as:

$$A_{CL}(x) = A(x) - B(x)K(x) \tag{14}$$

is point-wise Hurwitz. Thus, the origin of the closed-loop system is locally asymptotic stable, see (Erdem and Alleyne, 2004).

4.2 SDRE: Domain of Attraction Analysis

In order to design an SDRE controller, the general input-affine nonlinear form of the system, see (6), has to be written in the state-dependent linear form (11). Substituting equations (2) in (6) yields:

$$\dot{x} = \begin{bmatrix} 0 & I \\ -M^{-1}G_s & -M^{-1}C \end{bmatrix} x + \begin{bmatrix} 0 \\ M^{-1}B \end{bmatrix} u, \tag{15}$$

where C is an identity matrix and $G(q) = G_s q$. There are different options for matrix G_s that can respect both equation (6) and (15). Exploiting this flexibility, one can put emphasize on a specific state compared to the other states. For example, in the case of the RQuad balancing, the emphasize is on the body angle, due to the significant role of the angle α in the stabilization. Matrix G_s is therefore given by:

$$G_s = \begin{bmatrix} -m_b lg\dfrac{\sin(\alpha)}{\alpha} & 0 \\ -cm_b rg\dfrac{\sin(\phi)}{\alpha} & 0 \end{bmatrix}. \tag{16}$$

By substituting equations (2) and (16) in (15) and doing some manipulation, the state-dependent matrices of the RQuad robot are derived. These equations for the case $c = 0$ (just to simplify the presentation) are described as follows:

$$A(x) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ a_{31} & 0 & a_{33} & 0 \\ a_{41} & 0 & a_{43} & 0 \end{bmatrix} \tag{17}$$

$$B(x) = \begin{bmatrix} 0 \\ 0 \\ K\left(I_w + m_b rl\cos(\alpha) + r^2(I_b + m_b)\right) \\ K\left(I_b + m_b l^2 + m_b rl\cos(\alpha)\right) \end{bmatrix} \tag{18}$$

with

$$\begin{aligned} a_{31} &= (K/\alpha)\Big(m_b lg\big(I_w + r^2(m_w + m_b)\big)\sin(\alpha)\Big) \\ a_{33} &= K\big(-m_b^2 r^2 l^2 \sin(\alpha)\cos(\alpha)\dot{\alpha}\big) \\ a_{41} &= (K/\alpha)\big(-m_b^2 rl^2 g\sin(\alpha)\cos(\alpha)\big) \\ a_{43} &= K\big(m_b rl(I_b + m_b l^2)\sin(\alpha)\dot{\alpha}\big) \\ K &= \frac{1}{(I_b + m_b l^2)\big(I_w + r^2(m_w + m_b)\big) - m_b^2 r^2 l^2\cos^2(\alpha)}. \end{aligned} \tag{19}$$

When the state-dependent matrices are obtained, one can compute the control input (13).

In order to assess the performance of the controller in terms of its DA, we repeat the same simulations as for the LQR. The value of c increases from 0 to 1 and the controller's DA is computed. Fig. 5 illustrates the DA of the SDRE controller just for the values 0.75 and 1 of c, where there is a considerable improvement compared to LQR.

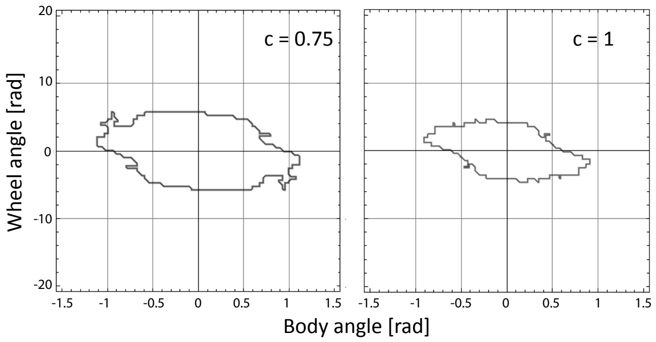

Fig. 5. The domain of attraction of SDRE controller for two different values of parameter c.

Although the DA of the SDRE controller becomes smaller as the nonlinearity becomes more severe, it is larger than the DA of the LQR, both for low and high values of parameter c. This is shown in Fig. 6 which plots for different values of parameter c the maximum absolute value of the initial body angle $\alpha(0)$, while $\phi(0) = 0$, for which the controller still can stabilize the robot. The SDRE controller is considerably more effective than the LQR, especially when c approaches 1 (the pivot joint lies on the wheel's rim and the wheeled robot changes to RQuad).

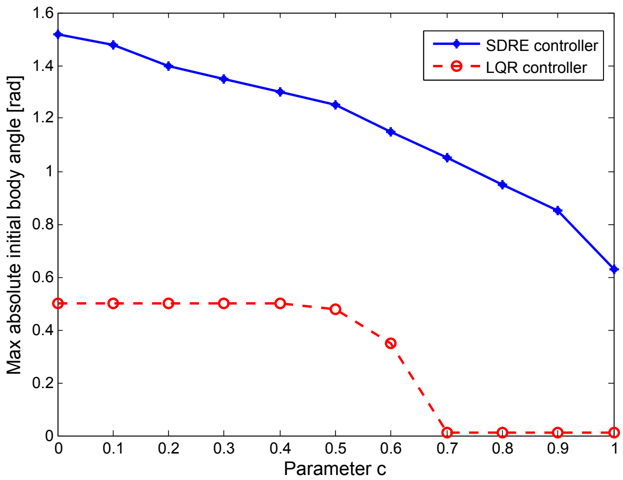

Fig. 6. Maximum absolute initial value of the body angle, when the wheel angle is zero, such that the controller still can balance the legged robot.

5. DISCUSSION AND CONCLUSIONS

We have compared nonlinear state-dependent Riccati equation control and linear LQR in the task of balancing a RHex-type legged robot on its hind legs. The LQR's domain of attraction is significantly smaller than that of the SDRE controller and the LQR cannot stabilize the robot. The most challenging part of the SDRE controller design is the computation of the state-dependent matrices from the model equations. Once the matrices are obtained, an algebraic Riccati equation is solved for computing the control input.

Besides using SDRE controller, one can apply methods that use a collection of controllers designed for specific parts of the state space. For example, sequential composition (Burridge et al., 1999) is an effective approach to address complex dynamical systems. In sequential composition, a collection of simple local controllers with possibly small domains of attraction are executed consecutively. LQR-tree is another method which implements the DAs of a series of local linear quadratic regulators (LQR) to create a tree (path) from the initial state to the desired goal through the state space (Tedrake et al., 2010). In addition, a learning strategy has been recently introduced by Najafi et al. (2013) which is an augmentation of the traditional sequential composition by learning new controllers on a need basis in runtime.

In composition based methods, the performance of each local controller, particularly its DA, plays an important role in the control synthesis. This paper showed that using the SDRE method can effectively enlarge the DAs of local controllers.

ACKNOWLEDGEMENTS

The first author gratefully acknowledges the support of the Government of the Islamic Republic of Iran, Ministry of Science, Research and Technology.

REFERENCES

Burridge, R.R., Rizzi, A.A., and Koditschek, D.E. (1999). Sequential composition of dynamically dexterous robot behaviors. *The International Journal of Robotics Research*, 18(6), 534–555.

Chiang, H.D., Hirsch, M.W., and Wu, F.F. (1988). Stability regions of nonlinear autonomous dynamical systems. *Automatic Control, IEEE Transactions on*, 33(1), 16–27.

Çimen, T. (2008). State-dependent Riccati equation (SDRE) control: a survey. In *Proceedings of the 17th World Congress, The International Federation of Automatic Control*, 6–11.

Çimen, T. (2010). Systematic and effective design of nonlinear feedback controllers via the state-dependent Riccati equation (SDRE) method. *Annual Reviews in Control*, 34(1), 32–51.

Cloutier, J.R. (1997). State-dependent Riccati equation techniques: an overview. In *American Control Conference, 1997. Proceedings of the 1997*, volume 2, 932–936. IEEE.

Dang, P. and Lewis, F.L. (2005). Controller for swing-up and balance of single inverted pendulum using SDRE-based solution. In *Industrial Electronics Society, 2005. IECON 2005. 31st Annual Conference of IEEE*, 6–pp. IEEE.

Erdem, E.B. and Alleyne, A.G. (2004). Design of a class of nonlinear controllers via state dependent Riccati equations. *Control Systems Technology, IEEE Transactions on*, 12(1), 133–137.

Genesio, R., Tartaglia, M., and Vicino, A. (1985). On the estimation of asymptotic stability regions: State of the art and new proposals. *Automatic Control, IEEE Transactions on*, 30(8), 747–755.

Grasser, F., D'Arrigo, A., Colombi, S., and Rufer, A.C. (2002). JOE: a mobile, inverted pendulum. *Industrial Electronics, IEEE Transactions on*, 49(1), 107–114.

Kalakrishnan, M., Buchli, J., Pastor, P., Mistry, M., and Schaal, S. (2011). Learning, planning, and control for quadruped locomotion over challenging terrain. *The International Journal of Robotics Research*, 30(2), 236–258.

Lopes, G., Babuska, R., De Schutter, B., and van den Boom, A. (2009). Switching max-plus models for legged locomotion. In *Robotics and Biomimetics (ROBIO), 2009 IEEE International Conference on*, 221–226. IEEE.

Nagarajan, U., Kantor, G., and Hollis, R. (2012). Integrated planning and control for graceful navigation of shape-accelerated underactuated balancing mobile robots. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 136–141. IEEE.

Najafi, E., Lopes, G.A., and Babuska, R. (2013). Reinforcement learning for sequential composition control. In *52nd IEEE Conference on Decision and Control*, 7265–7270. IEEE.

Pathak, K., Franch, J., and Agrawal, S.K. (2005). Velocity and position control of a wheeled inverted pendulum by partial feedback linearization. *Robotics, IEEE Transactions on*, 21(3), 505–513.

Raibert, M., Blankespoor, K., Nelson, G., Playter, R., et al. (2008). Bigdog, the rough-terrain quadruped robot. In *Proceedings of the 17th World Congress*, 10823–10825.

Saranli, U., Buehler, M., and Koditschek, D.E. (2001). Rhex: A simple and highly mobile hexapod robot. *The International Journal of Robotics Research*, 20(7), 616–631.

Tedrake, R., Manchester, I.R., Tobenkin, M., and Roberts, J.W. (2010). LQR-trees: Feedback motion planning via sums-of-squares verification. *The International Journal of Robotics Research*, 29(8), 1038–1052.

Vannelli, A. and Vidyasagar, M. (1985). Maximal Lyapunov functions and domains of attraction for autonomous nonlinear systems. *Automatica*, 21(1), 69–80.

Weingarten, J.D., Lopes, G.A., Buehler, M., Groff, R.E., and Koditschek, D.E. (2004). Automated gait adaptation for legged robots. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 3, 2153–2158. IEEE.