COMETa 2024
7th INTERNATIONAL SCIENTIFIC CONFERENCE
14th - 16th November 2024
Jahorina, B&H, Republic of Srpska
University of East Sarajevo
Faculty of Mechanical Engineering
Conference on Mechanical Engineering Technologies and Applications

OPTIMAL CONTROL OF A TWO-WHEELED SELF-BALANCING MOBILE ROBOT BASED ON ADAPTIVE DYNAMIC PROGRAMMING

Vladimir Stojanović[1], Vladimir Djordjević[2] Ljubiša Dubonjić[3], Saša Prodanović[4]

Abstract: This paper considers optimal tracking control for a two-wheeled self-balancing mobile robot with unknown dynamics. The aim is to achieve asymptotic tracking and disturbance rejection by minimizing some predefined performance index. Through the combination of adaptive dynamic programming (ADP) and internal model principle, an approximate optimal controller is iteratively learned online using measurable input/output data. Unmeasurable states are also reconstructed from input/output data. The discrete-time algebraic Riccati equation is iteratively solved by ADP approach. Simulation results demonstrate the feasibility and effectiveness of the proposed approach.

Key words: Adaptive dynamic programming, Adaptive optimal control, two-wheeled self-balancing mobile robot, Unknown dynamics, Algebraic Riccati equation

1 INTRODUCTION

A self-balancing mobile robot is a typical robot that has potential applications in many fields, such as transportation and research. Control of a self-balancing robot is attracting a lot of attention, both in academia and industry. Namely, the self-balancing robot is inherently unstable, high-order, multivariable, nonlinear and strongly coupled system, and at the same time represents an underactuated mechanical system. For such an underactuated system, which has fewer control inputs than generalized coordinates, it is necessary to indirectly control non-activated generalized coordinates, via dynamic coupling. Underactuation, although resulting in fewer actuators and thus helping to reduce manufacturing costs and failure rates, presents challenges for

[1] Vladimir Stojanović, University of Kragujevac, Faculty of Mechanical and Civil Engineering, 36000 Kraljevo, Serbia, stojanovic.v@mfkv.kg.ac.rs (CA)
[2] Vladimir Djordjević, University of Kragujevac, Faculty of Mechanical and Civil Engineering, 36000 Kraljevo, Serbia, djordjevic.v@mfkv.kg.ac.rs
[3] Ljubiša Dubonjić, University of Kragujevac, Faculty of Mechanical and Civil Engineering, 36000 Kraljevo, Serbia, dubonjic.lj@mfkv.kg.ac.rs
[4] Saša Prodanović, University of East Sarajevo, Faculty of Mechanical Engineering, East Sarajevo, B&H, sasa.prodanovic@ues.rs.ba

controller design. Moreover, unlike simpler systems, such as the inverse pendulum on a trolley, which are limited to a guided path, the self-balancing robot moves along its own path while balancing the pendulum. One of the difficulties in controlling a self-balancing robot is to simultaneously control its linear motion, tilt motion, and yaw.

Adaptive dynamic programming (ADP) offers a viable and effective approach to achieving optimal control performance using standard or intelligent control methods. It integrates the concepts of dynamic programming with neural networks, attempting to fix optimal control issues in dynamic programming problems utilizing the approximation feature of neural networks [1]. Recent years the ADP also has been extended and applied to many different areas, such as robots, spacecraft and so on [2].

When the state of a system is not immediately quantifiable and the system matrices are unknown, it is appropriate to use output feedback-based ADP. [3] proposes an output feedback ADP approach for discrete-time linear systems. Measurable input/output data are utilized to characterize the state of the discretized model, followed by policy iteration (PI) and value iteration (VI) to produce the optimal control policy. However, in order to obtain a unique solution in each iteration step, some exploration noise must be incorporated, which may affect the accuracy of results.

Direct measurement of system states is frequently too costly. Self-applied state estimation approaches presume that the system parameters remain constant. In the real world, these criteria cannot be known. It is also recognized that complex systems' dynamic behavior may be characterized by a linear stochastic state-space model with online estimated dynamics [4]. The precise understanding of system characteristics and states is critical for the successful implementation of various control strategies.

In this study, the continuous-time linear plant is discretized for simpler practical implementation, and the optimum control problem is then addressed. A self-balancing mobile robot with uncertain dynamics is controlled using an adaptive optimal output feedback technique for the discretized model. Simulation findings show that the suggested control strategy is valid and effective, with exploration noise having no influence on the correctness of the solution of the discrete Riccati equation.

2 DESCRIPTION OF A TWO-WHEELED SELF-BALANCING MOBILE ROBOT

A self-balancing robot is a multivariable underactuated mechanical system [5]. Underactuated mechanical systems have a larger number of generalized coordinates than the number of actuators. In these systems, the generalized coordinates are controlled indirectly, through their interconnection. Also, a self-balancing robot is a high-order nonlinear and unstable system. The mobile robot consists of a chassis to which are attached wheels driven by electric motors, see Fig. 1.

The robot has three degrees of freedom (generalized coordinates):

1) tilt angle θ (rotation around the Z-axis)
2) linear motion along the X-axis
3) yaw angle δ (rotation around the Y-axis)

These three generalized coordinates and their corresponding velocities fully describe the dynamics of the robot and represent the elements of the state vector.

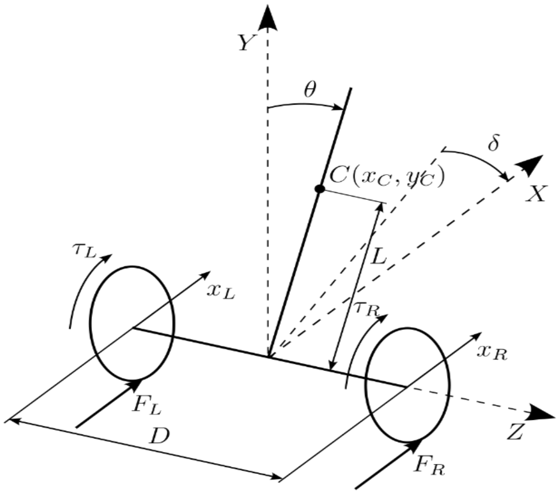

Figure 1. *A self-balancing mobile robot*

The robot is controlled by two electric motors that drive the respective wheels with torques τ_L and τ_R. Other parameters of the robot are: M [kg] - Mass of the chassis; m [kg] - Mass of each wheel; R [m] - Radius of the wheel; D [m] - Distance between the two wheels; L [m] - Distance between the center of gravity of the robot and the Z-axis; J_y [kg·m2] - Moment of inertia of the chassis with respect to the Y-axis; J_z [kg·m2] - Moment of inertia of the chassis with respect to the Z-axis; J_ω [kg·m2] - Moment of inertia of the wheel with respect to the Z-axis; x_L,x_R [m] - Displacements of the left and right wheels; x_C,y_C [m] - The position of the center of gravity of the robot; X_L, X_R [N] Interacting forces between the wheels and the chassis on the X-axis; Y_L, Y_R [N] Interacting forces between the wheels and the chassis on the Y-axis; F_L, F_R [N] - Frictions between the wheels and the ground surface; τ_L,τ_R [N·m] - Torques generated from the left and right actuators; θ_L, θ_R [rad] - Rotational angles of the left and right wheels. The dynamics of the wheels is defined by the sum of forces:

$$m\ddot{x}_L = F_L - X_L \,, \tag{1}$$

$$m\ddot{x}_R = F_R - X_R \,, \tag{2}$$

and sums of moments

$$J_\omega \ddot{\theta}_L = \tau_L - F_L R \,, \tag{3}$$

$$J_\omega \ddot{\theta}_R = \tau_R - F_R R \,. \tag{4}$$

The dynamics of the chassis is described by the sums of the forces for the x and y axes, as well as the sums of the moments for the z and y axes as follows:

$$M\ddot{x}_C = X_L + X_R \,,\ M\ddot{y}_C = Y_L + Y_R - Mg \,,\ J_y \ddot{\delta} = \frac{D}{2}\left(X_L - X_R\right) \tag{5}$$

$$J_z\ddot{\theta} = (Y_L + Y_R)L\sin(\theta) - (X_L + X_R)L\cos(\theta) - (\tau_L + \tau_R) \tag{6}$$

If we assume no wheel slip, then $x_L = R\theta_L$ and $x_R = R\theta_R$. Based on this, we can solve equations (1)-(4) and eliminate from them the angles of rotation of the wheels θ_L and θ_R, as well as the friction forces on the wheels F_L and F_R. As a result, we get the following equations that describe the dynamics of the left and right wheel:

$$\left(M + \frac{J_\omega}{R^2}\right)\ddot{x}_L = \frac{\tau_L}{R} - X_L, \quad \left(M + \frac{J_\omega}{R^2}\right)\ddot{x}_R = \frac{\tau_R}{R} - X_R. \tag{7}$$

The relationship between yaw angle δ of the robot and the linear motion of the wheels x_L and x_R is given by the expression:

$$D\delta = x_L - x_R. \tag{8}$$

The center of mass of the robot is determined by coordinates

$$x_C = x + L\sin(\theta), \ y_C = L\cos(\theta) \tag{9}$$

where $x = \frac{1}{2}(x_L + x_R)$. Based on the above equations, we can obtain the nonlinear equations of the system

$$\ddot{x}\left(M + 2m + \frac{2J_\omega}{R^2}\right) + ML\left(\ddot{\theta}\cos(\theta) - \dot{\theta}^2\sin(\theta)\right) = \frac{1}{R}(\tau_L + \tau_R), \tag{10}$$

$$\left(\frac{2J_y}{D} + \frac{DJ_\omega}{R^2} + Dm\right)\ddot{\delta} = \frac{1}{R}(\tau_L - \tau_R), \tag{11}$$

$$\begin{aligned} J_z\ddot{\theta} = 2\ddot{x}L\left(m + \frac{J_\omega}{R^2}\right)\cos(\theta) + MgL\sin(\theta) - ML^2\ddot{\theta}\sin^2(\theta) - \\ -ML^2\dot{\theta}^2\sin(\theta)\cos(\theta) - \left(1 + \frac{L\cos(\theta)}{R}\right)(\tau_L + \tau_R). \end{aligned} \tag{12}$$

Linearizing these nonlinear equations around the operating point $\theta = 0$, we obtain the following linear state space model of the self-balancing mobile robot

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\delta} \\ \ddot{\delta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{23} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & a_{43} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \\ \delta \\ \dot{\delta} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ b_{21} & b_{22} \\ 0 & 0 \\ b_{41} & b_{42} \\ 0 & 0 \\ b_{61} & b_{62} \end{bmatrix}, \tag{13}$$

$$y=\begin{bmatrix}1&0&0&0&0&0\\0&0&1&0&0&0\\0&0&0&0&1&0\end{bmatrix}\begin{bmatrix}x & \dot{x} & \theta & \dot{\theta} & \delta & \dot{\delta}\end{bmatrix}^T, \tag{14}$$

where $\begin{bmatrix}x & \dot{x} & \theta & \dot{\theta} & \delta & \dot{\delta}\end{bmatrix}^T$ is a state vector, $\begin{bmatrix}\tau_L & \tau_R\end{bmatrix}^T$ is an input vector, $\begin{bmatrix}x & \theta & \delta\end{bmatrix}^T$ is an output vector, while the model parameters are defined as

$$a_{23}=\frac{-M^2L^2g}{MJ_z+2\left(J_z+ML^2\right)\left(m+J_\omega/R^2\right)},\ a_{43}=\frac{M^2gL+2MgL\left(m+J_\omega/R^2\right)}{MJ_z+2\left(J_z+ML^2\right)\left(m+J_\omega/R^2\right)},$$

$$b_{21}=b_{22}=\frac{\left(J_z+ML^2\right)/R+ML}{MJ_z+2\left(J_z+ML^2\right)\left(m+J_\omega/R^2\right)},\ b_{41}=b_{42}=\frac{-(R+L)M/R-2\left(m+J_\omega/R^2\right)}{MJ_z+2\left(J_z+ML^2\right)\left(m+J_\omega/R^2\right)}$$

$$b_{61}=-b_{62}=\frac{D/2R}{J_y+D^2/(2R)\left(mR+J_\omega/R\right)}.$$

3 OPTIMAL PROBLEM FORMULATION

For practical application in a self-balancing mobile robot's control system, we shall explore the discretized system represented by

$$x_{k+1}=A_dx_k+B_du_k \tag{15}$$

$$y_k=Cx_k \tag{16}$$

in which $A_d=e^{Ah}$, $B_d=\int_0^h\left(e^{A\tau}d\tau\right)B$ and $h>0$ is the sampling period, assuming $\omega_h=2\pi/h$ is non-pathological sampling frequency [6]. In other words, there are no two eigenvalues of A with equal real and imaginary components that vary by an integral multiple of ω_h. The state, input, and output vector at the sampled instant kh are x_k, u_k, y_k, respectively. Then, both $\left(A_d,C\right)$ and $\left(A_d,Q^{1/2}C\right)$ are observable and $\left(A_d,B_d\right)$ is controllable. Cost for (15)-(16) is:

$$J_d(x_0)=\sum_{j=0}^{\infty}y_j^TQy_j+u_j^TRu_j \tag{17}$$

The optimal control law minimizing (un) is

$$u_k=-K_d^*x_k \tag{18}$$

where discrete optimal feedback gain matrix is $K_d^* = \left(R + B_d^T P_d^* B_d\right)^{-1} B_d^T P_d^* A_d$, and P_d^* is the unique symmetric positive definite solution to

$$A_d^T P_d^* A_d - P_d^* + C^T QC - A_d^T P_d^* B_d K_d^* = 0 \tag{19}$$

This well-known optimal control design approach has so far been limited to low-order simple linear systems. In reality, for high order, large scale systems, it is frequently challenging to directly solve P_d^* from (19), which is nonlinear in P_d. Nonetheless, numerous effective techniques have been created to numerically estimate the solution of (19). A certain algorithm has been established by Hewer [7]. By iteratively solving the Lyapunov equation

$$\left(A_d - B_d K_j\right)^T P_j \left(A_d - B_d K_j\right) + C^T QC + K_j^T R K_j = 0 \tag{20}$$

which is linear in P_j, and updating K_j by

$$K_j = \left(R + B_d^T P_{j-1} B_d\right)^{-1} B_d^T P_{j-1} A_d \tag{21}$$

the solution to the nonlinear equation (19) is numerically approximated. It has been concluded that sequences $\left\{P_j\right\}_{j=0}^{\infty}$ and $\left\{K_j\right\}_{j=0}^{\infty}$ computed from this algorithm converge to P_d^* and K_d^*, respectively. Moreover, for $j = 0,1,\dots$, $A_d - B_d K_j$ is a Schur matrix.

It should be emphasized that Hewer's approach is a model-based policy iteration (PI) algorithm that cannot be used when all of the system matrices are undetermined, as it is an offline technique that relies on system parameters. To implement it online, we will create an adaptive optimal control technique for the discretized system (15)-(16) using output feedback that does not need to know the system matrices. The online output feedback learning technique can now be formulated based on (20)-(21).

$$x_{k+1} = A_j x_k + B_d \left(K_j x_k + u_k\right) \tag{22}$$

where $A_j = A_d - B_d K_j$. Letting $\bar{K}_j = K_j \Theta$ and $\bar{P}_j = \Theta^T P_j \Theta$, from (20) and (22) it follows that,

$$z_{k+1}^T \bar{P}_j z_{k+1} - z_k^T \bar{P}_j z_k = \phi_\kappa^1 vec(\bar{H}_j^1) + \phi_\kappa^2 vec(\bar{H}_j^2) - \left(y_k^T Q y_k + z_k^T \bar{K}_j^T R \bar{K}_j z_k\right) \tag{23}$$

in which $\bar{H}_j^1 = B_d^T \bar{P}_j B_d$, $\bar{H}_j^2 = B_d^T \bar{P}_j A_d \Theta$, $\phi_\kappa^1 = u_k^T \otimes u_k^T - (z_k^T \otimes z_k^T)(\bar{K}_j^T \otimes \bar{K}_j^T)$,
$\phi_\kappa^2 = 2\left[(z_k^T \otimes z_k^T)(I_q \otimes \bar{K}_j^T) + (z_k^T \otimes u_k^T)\right]$.

This assumption is related to the condition of persistent excitation according to adaptive control theory [8]. Then, $\bar{K}_{j+1}$ can be calculated as $\bar{K}_{j+1} = \left(R + \bar{H}_j^1\right)^{-1} \bar{H}_j^2$.

Here, (23) is called policy evaluation, which is used to uniquely solve $\bar{P}_j$, and $\bar{K}_{j+1}$ is policy improvement (PI).

4 SIMULATION RESULTS

In this section, we conduct simulations on the two-whelled self-balancing mobile robot to show the effectiveness of the output-feedback ADP control algorithm in the case with unknown system matrices and unmeasurable states. Through iterative calculation, the approximated optimal control gain and performance index for the discrete-time system can be obtained. Furthermore, the discrete control policy is implemented on the continuous plant by zero-order holder. Adopted sampling time is $h=0.1s$. Robot parameters which are used in the simulations are: M=21 kg, m=0.42 kg, R=0.106 m, D=0.44 m, L=0.3 m, J_y=0.3388 kgm^2, J_z=0.63 kgm^2, J_w=2.4·10^{-3} kgm^2.

Weight matrices Q and R are unit matrices, while for learning purposes, in a period of 4 s, optimal exploration noise in the form of sums of sinusoids was used:

$$e_i(t)=\sum_{j=1}^{6}\sin\omega_{ij}(t)\,, \tag{24}$$

where the optimal frequencies for both inputs are

$$\omega=\begin{bmatrix} -4.65 & 15.44 & 17.90 & 44.51 & 20.92 & -38.06 \\ 39.09 & -16.58 & 19.87 & -30.21 & -46.94 & 24.40 \end{bmatrix}. \tag{25}$$

Also, in simulations, the initial value of the state vector was taken as $x_0=\begin{bmatrix}0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1\end{bmatrix}^T$, while the convergence threshold is $\varepsilon=10^{-10}$. The input/output data are collected from 0.8 to 4 seconds, and the PI is started from t = 4s. The inputs and outputs of the robot controlled by the ADP-based controller are shown in Fig. 2.

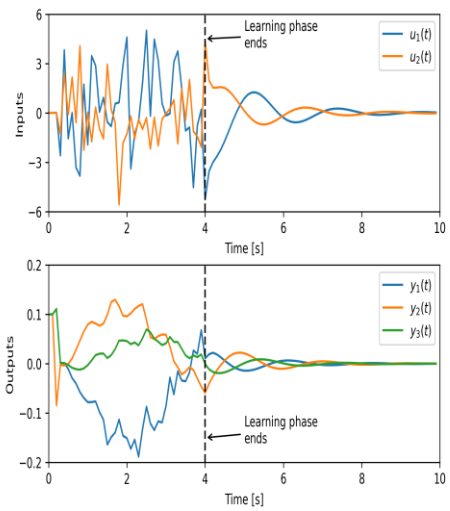

Figure 2. *Input and output signals of the robot*

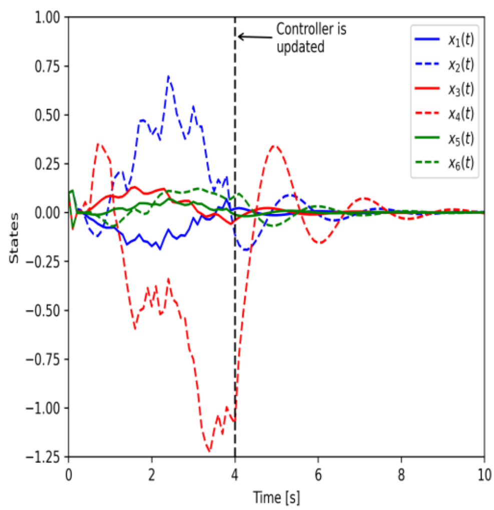

Figure 3. *Trajectory of states*

Fig. 3 shows the state trajectories of the mobile robot. From this figure it can be seen that the states converge to the equilibrium state quickly after the controller is updated. Fig. 4 shows the convergence of the approximate values of the matrices $\bar{P}_j$ and $\overline{K}_j$ towards their optimal values $\bar{P}^*$ and $\overline{K}^*$.

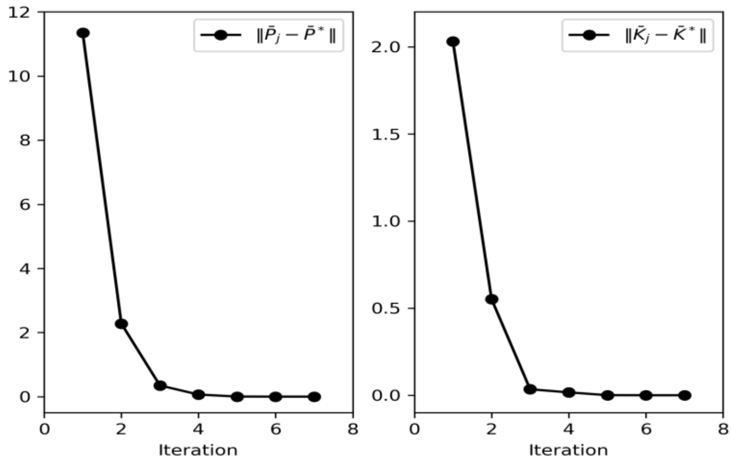

Figure 4. Convergence of the approximated matrices $\bar{P}_j$ and $\overline{K}_j$

5 CONCLUSIONS

This paper considers ADP-based optimal controller design for two-whelled self-balancing mobile robot with entirely unknown dynamics. The applied sampled-data adaptive optimal control technique based on the discretized model and output feedback has been proved to be a beneficial tool in these situations. It should be emphasized that exploration noise has no effect on the correctness of the solution to the discrete Riccati equation. The simulation results suggest that the proposed control strategy is valid and effective.

ACKNOWLEDGEMENTS

This research was partly funded by the Serbian Ministry of Science, Technological Development and Innovation from the grant 451-03-65/2024-03/200108, and the Ministry of Civil Affairs of Bosnia and Herzegovina, from the grant Support to technical culture and innovation in Bosnia and Herzegovina.

REFERENCES

[1] Tao, B. and Jiang, Z.P. (2016). Value iteration and adaptive dynamic programming for data-driven adaptive optimal control design, Automatica, 71, pp. 348-360.

[2] Roozegar, M., Mahjoob, M.J. and Jahromi, M. (2016). Optimal motion planning and control of a nonholonomic spherical robot using dynamic programming approach: simulation and experimental results, Mechatronics, 39, pp. 174-184.

[3] Lewis, F.L. and Vamvoudakis, K.G. (2011). Reinforcement learning for partially observable dynamic processes: adaptive dynamic programming using measured output data, IEEE Trans. Syst. Man Cybern. B, Cybern, 41(1), pp. 14-25.

[4] Rodriguez, M.T. and Banks, S.P. (2010) Linear, Time-Varying Approximations to Nonlinear Dynamical Systems, Berlin Springer (Germany).

[5] Guo, L., Rizvi, S.A.A. and Lin. Z. (2021). Optimal control of a two-wheeled self-balancing robot by reinforcement learning, International Journal of Robust and Nonlinear Control, 31(6), pp. 1885–1904

[6] Chen, T. and Francis, B.A. (1995). Optimal Sampled-data Control Systems, New York (USA) Springer-Verlag.

[7] Hewer, G. (1971). An iterative technique for the computation of the steady state gains for the discrete optimal regulator, IEEE Transactions on Automatic Control, 16(4), pp. 382–384.

[8] Astrom, K.J. and Wittenmark, B. (1994). Adaptive control, Addison-Wesley Longman Publishing Co.