Visual Avoidance of Collision with Randomly Moving Obstacles through Approximate Reinforcement Learning

Yunfei ZHANG [1], Yanjun WANG [1], Haoxiang LANG [2], Ying WANG [3], Clarence W. DE SILVA [4]

(1. *ViWiStar Technologies Ltd.*, *Shenzhen China* 518103;
2. *Mechanical Engineering Department*, *University of Ontario Institute of Technology*, *Oshawa*, *Canada L1G 0C5.*;
3. *Department of Mechatronics Engineering at Kennesaw State University*, *Marietta*, *Georgia*, 30060, *USA.*;
4. *Department of Mechanical Engineering*, *The University of British Columbia*, *Vancouver*, *Canada V6T 1Z4*)

Abstract: In this research work, a hierarchical controller has been designed for an autonomous navigation robot to avoid unexpected moving obstacles where the state and action spaces are continuous. The proposed scheme consists of two parts: 1) a controller with a high-level approximate reinforcement learning (ARL) technique for choosing an optimal trajectory in autonomous navigation; and 2) a low-level, appearance-based visual servoing (ABVS) controller which controls and execute the motion of the robot. A novel approach for path planning and visual servoing has been proposed by the combined system framework. The characteristics of the on-board camera which is equipped on the robot is naturally suitable for conducting the reinforcement learning algorithm. Regarding the ARL controller, the computational overhead is quite low thanks to the fact that a knowledge of obstacle motion is not necessary. The developed scheme has been implemented and validated in a simulation system of obstacle avoidance. It is noted that findings of the proposed method are successfully verified by obtaining an optimal robotic plan motion strategy.

Key words: Approximate reinforcement learning; Robotic obstacle avoidance; Appearance-based visual servoing

1 Introduction

In the past decade, autonomous navigation in dynamic and unknown environments has been extensively studied by many researchers. It is widely known that obstacle avoidance is an essential and important task in robotic navigation . Typically, the robot control system generates a collision-free trajectory and drives the robot to the goal location [1]. Collisions might also be prevented by decelerating or accelerating, on encountering a moving obstacle [2].

Practically, the methodologies of obstacle avoidance in autonomous navigation can either be categorized as: model-free or model-based, which depends on whether the scheme considers a model of the work environment or not. A model-based navigation scheme requires pre-defined information of the environment and obstacles. Reference [3] developed a model-based vision system with retroactive position correction. Normally, a common characteristic of the existing methods is to adopt a three-dimensional (3D) model of the environment, which includes elements such as walls and doors, or the geometry of the path. In this research work, however, a model-free scheme is developed. And a generalized framework for obstacle avoidance is also proposed, which is appropriate and efficient for real-time autonomous navigation. Particularly, a model of the obstacle behavior or environment is not required by the developed method.

References [4] and [5] used model-free methods called deep reinforcement learning schemes to solve navigation problems. However, the system lacks a moving obstacle policy. The method of visual servo control in this research work relies on appearance-based navigation, which is mainly inspired by the work of [6] and [7]. However, their method

also did not take into account moving obstacles. Furthermore, continuous control and related optimization problem were not considered explicitly. Unlike their method, the proposed scheme utilizes the concept of appearance-based visual servo control and develops a new system framework that is effective for both indoor and outdoor scenarios. In particular, the view from the visual system is utilized to detect moving obstacles, and a new constraint called dangerous area is incorporated into the approximate reinforcement learning scheme to plan optimal trajectories.

The contributions of this research work are presented below. Through the simulation results, it is found that, obstacle avoidance in autonomous robot navigation, through the use of approximate reinforcement learning, the scheme of obstacle avoidance is suitable for dealing with optimal trajectory planning and control with continuous state and action space.

This present paper is organized as follows. Section II presents the problem definition. Section III demonstrates the system model and frame projection model. Section IV introduces approximate reinforcement learning and its application in the present problem. Simulation results are shown and discussed in Section V. Section VI concludes the research work.

2 Problem formulation

2.1 Description of the problem

This research work aims toaddress the problem of obstacle avoidance for an autonomous navigation robot, in which an initial global trajectory is given. Following the initial trajectory, it is possible that the robot might encounter unknown obstacles when initializing its path. In such scenarios, it is of great significance for the robot to be capable of avoiding initially unknown obstacles that might suddenly appear during navigation. Thus, a scheme is necessary to recalculate the local trajectory that deviates from the current trajectory, in order to reach the target location successfully. Additionally, it is assumed that the target location (local or global) is always available for the robot to reach.

2.2 Obstacle representation

It is noted that the scheme proposed in the present paper involves direct representation of an obstacle that shows up in the image frame of the on-board camera, and then the movement of the robot is controlled by using the image information in order to avoid the obstacle. Consequently, the navigation trajectory is regenerated for leading the robot to the target location. Refer to Fig.1 (a), first, the dangerous area and the safe area need to respectively be specified in the real field of robot navigation.

Then, the robot will occupy a certain area while following certain trajectory. The dangerous area may therefore be specified on this basis, which is denoted by a shadow area in Fig. 1. During the movement of the robot, If an obstacle enters this area, collision may occur. In this regard, the area outside the dangerous area is called the safe area. In order to improve the degree of safety, it is possible to expand the dangerous area with regard to the possibility of collision. In addition, these areas in the physical environment have to be mapped onto the camera image, as shown in Fig. 1(b). Section III will further discuss the mapping in details. Normally, it would be sufficient to only map the dangerous area onto the image, since the remaining area of the image frame can be considered as the safe area. If an obstacle shows up in the dangerous area, correspondingly, the leaning controller will move it out of that area, ensuring the entire obstacle is in the safe area. See Fig.2.

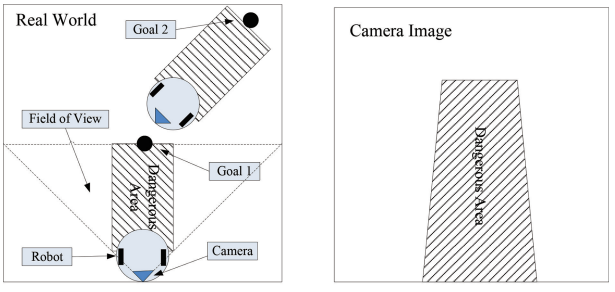


Fig. 1 Specified dangerous and safe areas according to newly detected obstacles: (a) in real world; (b) in camera image.

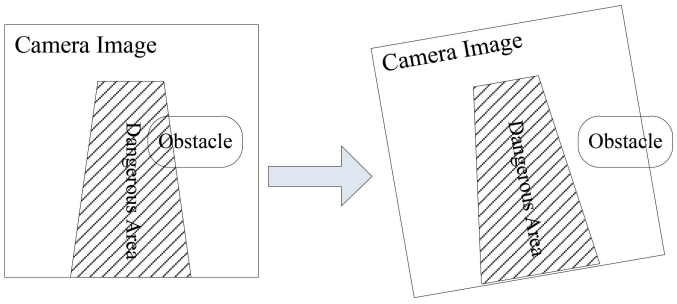


Fig. 2 Obstacle avoidance through image movement, as the robot moves.

By adopting this approach, there are three advantages in terms of obstacle avoidance. Firstly, the safe and dangerous areas can be specified according to the safety requirements. Secondly, despite the fact that the camera images are utilized to depict the method, the proposed scheme can be easily extended for other types of sensors (e.g., ultrasound). Thirdly, the designed appearance-based scheme eliminates the necessity for deriving an obstacle kinematic model, and also facilitates keeping of obstacles in the robot's field of view. The robot model and the state mapping model are found in [8].

2.3 General control scheme

Practically, the relationship between the frame of the image and the camera could be studied according to a modified pinhole camera [11], (see Fig.3). Assume that the camera plane is placed at a distance f (which is the focal length) behind the pinhole, then the intersection of the image plane and the z axis is called the principal point. The image reversal problem can therefore be avoided.

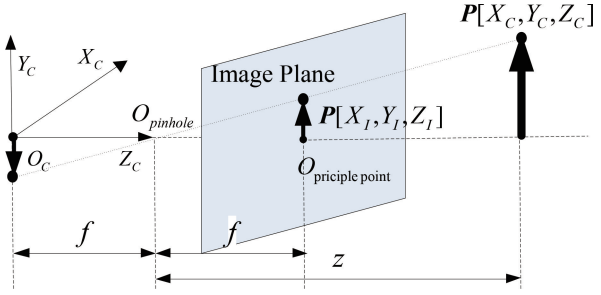


Fig. 3 Modified pinhole camera model.

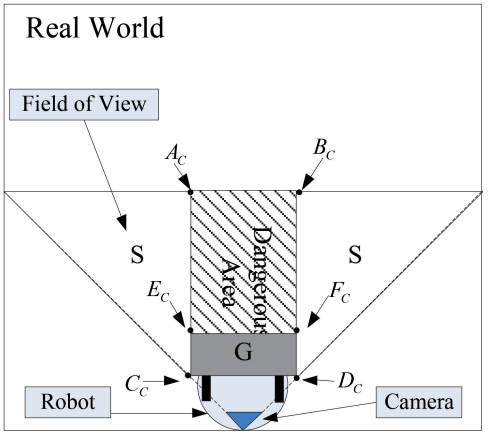
If f stands for the focal length of the given camera, while $[x_I, y_I]$ represents the coordinate of the object in the camera frame with respect to the camera frame, then on the image plane, the corresponding coordinates of the point are obtained through,

$$k\begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} = \begin{bmatrix} x_I \\ y_I \\ f \end{bmatrix}$$

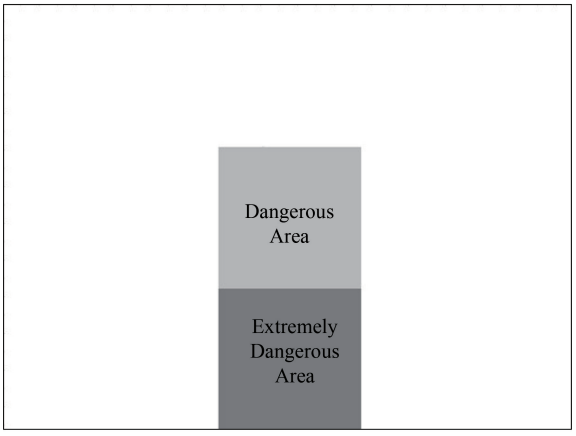
where $k = \frac{f}{z}$ and z is the depth from the camera to the object.

2.4 Dangerous area definition

This section demonstrates the scheme of separating different areas in the robot frame, and mapping them onto the image plane. Intuitively, the closer the obstacle to the robot, the more dangerous it would



(a)



(b)

Fig. 4 Definition of state area and mapping: (a) In the robot frame; (b) In the image frame.

be. The dangerous area should surround the robot and also provide enough space for the robot to move without the possibility of colliding with obstacles. When an obstacle gets closer to the dangerous area, which is detected by the system, the robot will be controlled in order to avoid it. Fig.4 illustrates this concept.

In Fig.4 (a), The dangerous area which is shown as a shadow rectangle, is defined based on the size of the physical robot operating in the real environment. The width of the shadow rectangle, which is denoted as w, equals to the diameter d of the robot, and the length l equals to $2d$. As shown in Fig.4 (b), within this dangerous area, an extremely dangerous area could be defined as a stop point for a robot once an obstacle shows up. In addition, within the dangerous area, it is noted that the empty area stands for the safe area in which the obstacles will not show up. With this definition, obstacle avoidance involves making the goal point move into the goal position and making the obstacles move out of dangerous area at the same time.

3 Approximate Reinforcement Learning in Dynamic Environment

3.1 Approximate Reinforcement Learning (ARL)

Reinforcement learning[9] is a suitable framework for optimal control with constraints, since the computation of complex Lagrange multiplier equation for constraints is not required. Reinforcement learning utilizes the solution of control constraint by setting the corresponding scope of action space. However, classic reinforcement learning requires exact representation of the value function and storing the distinct return estimates for every state-action pair (under control scenario, the pair of the state-action and Q-function $Q(s,a)$ are chosen as required). These are not possible when some of the state variables have very large (theoretically infinite) number of possible values. Therefore, it is practical to represent the Q-function approximately. In the present paper, for avoiding obstacles, a real-time optimal controller with continuous state and action will use approximate reinforcement learning.

In approximate reinforcement learning, two aspects of approximation are considered: representation and sample-based approximation. The work in [10] and [11] has proposed some approximators based on state-dependent basic functions for value functions and sampling. However, they did not consider dynamic situations. In those methods, if the environment changes, a converged set of new coefficients has to be recomputed for the approximtors. The specific approach in the present paper is based on those prior methods, but a new framework is presented to adapt to dynamic environments.

3.2 Approximate representation of the value function

The value function is a fundamental component of reinforcement learning, and is based on value iteration or policy iteration. However, the existing approaches of policy iteration; for example, LSPI, LSTD [12] and their variants, are much slower than the value iteration approaches because of the need for computing and storing high-order matrices in the iteration process. In comparison, value iteration only needs to compute and store a relatively small set of coefficients. In real-time control, computational speed is critical, and the present paper uses an approximate value function based on value iteration structure, in order to expedite the computations. In particular, a linearly parameterized approximator is used for the Q-function.

Parameterized approximation involves mapping the Q-function space into a parameter space. With an n-dimensional parameter set, vectorθ, the approximator is represented by the mapping$F:\mathbb{R}^n \to \mathbb{Q}$, where $\mathbb{R}^n$ is the parameter space and $\mathbb{Q}$ is the space of Q-function. According to this mapping, the approximate Q-function is denoted by:

$$\hat{Q}(s,a) = [F(\theta)](s,a)$$

where $[F(\theta)](s,a)$ represents the Q-function

$Q(s,a)$, approximated as $\hat{Q}(s,a)$ by mapping $F(\theta)$ at the pair of state-action (s,a). Usually linear mapping is preferred in real-time control since that the theoretical properties of the resulting approximate reinforcement learning algorithms are less complex. By employing n basis functions (BFs) $\varphi_1,\ldots,\varphi_n: S \times A \rightarrow \mathbb{R}$, together with the parameter vector θ, the approximate Q-value can be computed as:

$$F(\theta)(s,a) = \sum_{l=1}^{n} \varphi_l(s,a)\theta_l = \varphi^T(s,a)\theta$$

where, $\varphi(s,a) = [\varphi_1(s,a),\ldots,\varphi_n(s,a)]^T$ denotes the BFs vector.

Through the mapping $F(\theta)$, BFs become the primitives of the approximate reinforcement learning problem. The space of the continuous state-action pair is transferred into the BFs space in which spaces of the continuous state and action are approximated by other formations of state and action spaces with finite and relatively low dimensions. The real Q-function $Q(s,a)$ is then denoted by the designed BFs vector and the parameter vector. The objective of learning is to determine an optimal parameter vector θ^* that is able to generate the best approximation $\hat{Q}^*(s,a)$ for the value of Q-function at a specific (s,a).

4 Simulation

4.1 Simulation Setting

In this section, the combined controller is appliedto a simulated mobile robot operating in a dynamic environment. Referring to Fig. 2, the movement of the camera/robot with respect to the obstacles has an equivalent movement of the obstacles relative to the camera/robot. The image from the camera is assumed to be fixed and the pixels of the obstacles move with respect to the image. Therefore, the entire simulation tests the action strategy of the moving obstacle pixels in the fixed image.

Based on a fixed image size 640 × 480, the states and actions can be represented as:

$$s_i = \begin{bmatrix} x_{I,i} \\ y_{I,i} \end{bmatrix}, \quad a_i = \begin{bmatrix} u_{Ix,i} \\ u_{Iy,i} \end{bmatrix}$$

where, $x_{I,i} \in [-320,320]$; $y_{I,i} \in [-240,240]$; 640 and 480 correspond to the width and the height, respectively, of the image frame; and $u_{Ix,i}$, $u_{Iy,i} \in [-10,10]$ (pixel/second) correspond to the moving velocities of the feature points (obstacle pixels) along the width axis and the height axis of the image frame. Accordingly, the continuous and discrete kinematics of the feature points may be expressed as:

$$s_i = s_0 + \int_0^t a_i \mathrm{d}t, \quad s_{i+1} = s_i + a_i \Delta t$$

where, Δt denotes the discrete time interval. In the present simulation, discrete kinematics $s_{i+1} = s_i + a_i\Delta t$ is used for the samples, and $\Delta t = 1$. An integer setting of Δt is suitable in the present case, since the pixel space is a discrete space while the robot moves in a continuous world space. In other words, keeping s_i as an integer vector helps to increase execution accuracy for a mobile robot.

In terms of the present simulation, a triangular fuzzy partition is chosen as the MFs. Nevertheless, there are other types of MFs that can be used to guarantee convergence to an optimal result in approximate Q-iteration, as long as the remaining MFs take negligible values at the center of the corresponding MF fuzzy set χ_i.

4.2 Simulation Results

For 640 × 480 image frame, the entire image space is divided into 20 × 20 fuzzy grid sets for MFs, and the action space is equalized to 10 values in [-20, 20] pixel/second. The maximum speed should be less than one and half times the grid size. Otherwise, some states will be overlooked.

Assume that one moving obstacle is just one point in the image, and that the camera is always able to detect a moving obstacle that enters the dangerous area. Since the starting position of a moving obstacle will not affect the learned policy, we will randomly choose [-300, -240] as the starting position. The control objective is to move the obstacles to the goal position, which is set as [80, -240]. An optimal policy of real-time control for the entire im-

age is learned, as shown in Fig. 5.

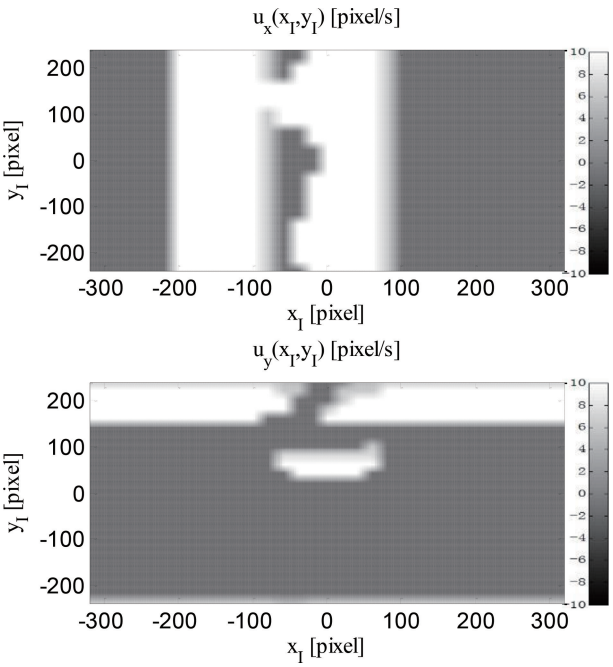


Fig. 5 Optimal policy for real-time control.

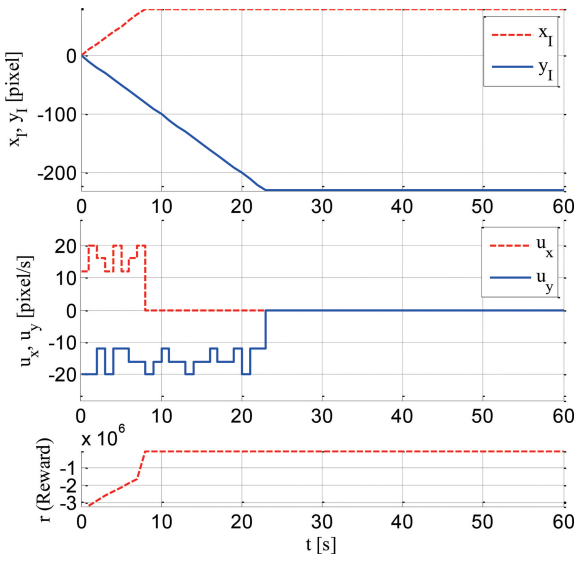


Fig. 6 State, action and reward trajectories according to the optimal policy in Fig.5.

Fig. 5 is obtained by running Algorithm 1 in MATLAB code on Inter Core i5-3320CPU@2.60Hz. It takes around 700 iterations, corresponding to about 200 seconds, to converge to the sub-optimal result (it can be termed an optimal policy, for convenience) with $\varepsilon_{QI} = 0.01$. With the learned optimal policy, the specific control trajectory for avoiding a moving obstacle detected at position [0, 0], is shown in Fig. 6.

From Fig.6, it can be seen that the position of the obstacle point is moved successfully to the goal position [80, -240]. The corresponding optimal path in the simulated image frame is shown in Fig. 7.

The optimal path, which is marked by circles in Fig. 7, shows that the obstacle detected at the starting point exactly knows what the optimal path means. Compared to the extremely dangerous area, the dangerous area can be run across the diagonal path, which is the shortest path in the dangerous area. Then the obstacle enters the extremely dangerous area where the obstacles should not be allowed to present. It is seen that the obstacle correctly chooses to go to the goal position along the side of the extremely dangerous area.

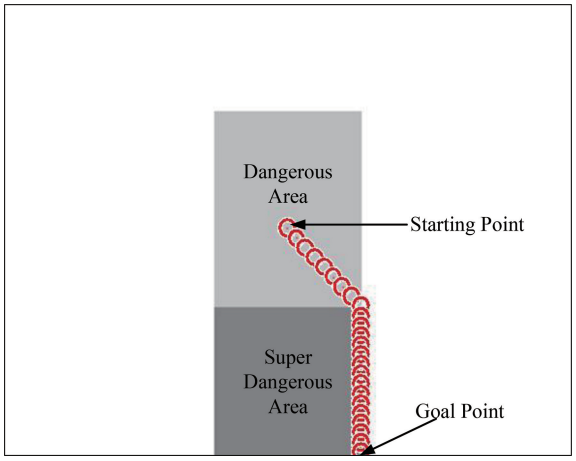


Fig. 7 Optimal path in a simulated image frame.

5 Conclusion

The present research work proposed a hierarchical controller which aims to avoid randomly moving obstacles in autonomous navigation of a robot. A high-level ARL controller was used for obtaining an optimal plan for navigation. The low-level, appearance-based visual servoing (ABVS) controller was used for controlling and executing motion of the robot. The ABVS controller enabled transferring the optimal image path to the robot path by directly ex-

ploiting the optimal policy, which obtains an optimal plan for robot motion based on the specific environment. Under a combined system framework of planning and visual servo control, the use of the learning ability of a robot in avoiding collision was a novel feature. Moreover, the presented scheme exploited the robot on-board camera whose finite field of view was naturally suitable for conducting the reinforcement learning algorithm. The simulation results showed that the proposed method successfully converged to an optimal strategy, so that the robot could also accordingly generate a proper motion plan.

ACKNOWLEDGMENT

This research work has been supported by research grants from the Natural Sciences and Engineering Research Council (NSERC) of Canada, the British Columbia Knowledge Development Fund (BCKDF), the Canada Foundation for Innovation (CFI), and the Canada Research Chair in Mechatronics and Industrial Automation held by C.W. de Silva.

References

[1] J.Minguez, F. Lamiraux and J. P. Laumond (2008) Motion planning and obstacle avoidance. *Springer Handbook of Robotics. B. Siciliano, O. Khatib (Eds.)*, Springer, pp.827-852.

[2] T. Wada, S. Doi and S. Hiraoka (2009) A deceleration control method of automobile for collision avoidance based on driver's perceptual risk. *IEEE/RSJ IROS*.

[3] F. Lamiraux, D. Bonnafous and O. Lefebvre. (2004) Reactive path deformation for nonholonomic mobile robots. *IEEE Transactions on Robotics*, 20: 967-977.

[4] Y. Zhu, R. Mottaghi, E. Kolve, J. J Lim, A. Gupta, F. F. Li, and Ali Farhadi. (2017) Target-driven visual navigation in indoor scenes using deep reinforcement learning. *IEEE International Conference on Robotics and Automation (ICRA)*, pp.3357-3364.

[5] X. Zhou, Y. Gao, L. Guan (2019) Towards Goal-Directed Navigation Through Combining Learning Based Global and Local Planners. *Sensors*.

[6] A. Cherubini and F. Chaumette (2011) Visual navigation with obstacle Avoidance. *IEEE/RSJ IROS*.

[7] A. Cherubini and F.Chaumette (2012) Visual Navigation of a Mobile Robot with Laser-based Collision Avoidance. *The International Journal of Robotics Research*, vol.0, no.0, pp.1-17.

[8] Y. Zhang, C. W. de Silva, D. Su, Y. Xue (2014) Autonomous robot navigation with self-learning for collision avoidance with randomly moving obstacles. *IEEE* 2014 9*th International Conference on Computer Science & Education*, pp. 22-24 Aug.

[9] R. S. Sutton and A. G.Barto. (1998) *Reinforcement Learning: An Introduction*. MIT Press, Cambridge.

[10] L. Bus, oniu, D. Ernst, B. De Schutter, and R. Babůska. (2010) Approximate dynamic programming with a fuzzy parameterization. *Automatica*, vol. 46, no.5, pp.804-814.

[11] D. P.Bertsekas, (2012) *Dynamic Programming and Optimal Control* (Chapter 6, volume 2).

[12] L.Bus, oniu, D. Ernst, B. De Schutter, and R. Babů ska. (2010) Online least squares policy iteration for reinforcement learning control. *Proceedings American Control Conference, Baltimore*, pp. 486-491.

Authors' Biographies

Yunfei ZHANG received his B.S. degree in Automation from Qingdao University of Science and Technology in 2006, and M.S. degree in Automotive Engineering from Shanghai Jiao Tong University, China, in 2010. He finished his Ph.D. degree the Department of Mechanical Engineering at the University of British Columbia, Canada in 2015. He is currently working on robotic technologies for caring elderly people, as an ambitious start up entrepreneur. His main research interests include deep reinforcement learning and control, decision making, robotics, and autonomous driving.

Yanjun WANG obtained his PhD degree in the Department of Mechanical Engineering, The University of British Columbia, Vancouver, Canada in 2014. He received his Master's degree in Automotive Engineering from Shanghai Jiao Tong University, Shanghai, China in 2009, and his Bachelor's degree in Automotive Engineering from Nanjing University of Aeronautics & Astronau-

tics, Nanjing, Jiangsu, China in 2006. Presently he works on compliance control of robotics in ViWiSTAR Technologies Ltd. His research interests are in robot dynamics modelling and control, system identification, automotive powertrain system modeling, control and optimization, and soft computing.

Haoxiang LANG received his M. Sc. and Ph.D. degrees from the Department of Mechanical Engineering, The University of British Columbia, Vancouver, Canada in 2008 and 2012, respectively; and the Bachelor's degree from Ningbo University in 2003. He worked in the Motorola Cellular Equipment Company as a software engineering from January 2004 to September 2005. He was with the Industrial Automation Laboratory as a postdoctoral research fellow and the Lab Manager from November 2012 to December 2014. Haoxiang Lang is currently an Assistant Professor of Mechanical Engineering and the Director of the GRASP (General Robotics and Autonomous Systems and Processes) at University of Ontario Institute of Technology (UOIT). His research and development areas are Mechatronics, Robotics, and Artificial Intelligence.

Ying WANG received his Ph.D. degree in Robotics and Mechatronics from The University of British Columbia (UBC), Vancouver, BC, Canada in 2008. He also received his Master's degree (1999) and the Bachelor's degree (1991) from Shanghai Jiao Tong University, China. He is now an Associate Professor in the Department of Mechatronics Engineering at Kennesaw State University, Marietta, Georgia, 30060, USA. Dr. Wang's research interests are in the areas of robotics, controls, mechatronics and machine learning.

Clarence W. de Silva received Ph. D. degrees from Massachusetts Institute of Technology, Cambridge, USA, in 1978, and the University of Cambridge, Cambridge, U. K., in 1998, the Honorary D. Eng. degree from the University of Waterloo, Waterloo, ON, Canada, in 2008, and the higher doctorate (Sc. D.) from the University of Cambridge (2020). He has been a Professor of Mechanical Engineering and NSERC-BC Packers Chair in Industrial Automation at the University of British Columbia, Vancouver, BC, Canada, since 1988, and the Senior Canada Research Chair in Mechatronics and Industrial Automation. He has authored 25 books and more than 550 papers, approximately half of which are in journals. His recent books published by Taylor & Francis/CRC are: *Modeling of Dynamic Systems—with Engineering Applications* (2018); *Sensor Systems* (2017); *Sensors and Actuators—Engineering System Instrumentation*, 2nd *edition* (2016); *Mechanics of Materials* (2014); Mechatronics—A Foundation Course (2010); Modeling and Control of Engineering Systems (2009); Sensors and Actuators— Control System Instrumentation (2007); VIBRATION—Fundamentals and Practice (2nd ed., 2007); Mechatronics—An Integrated Approach (2005); and by Addison Wesley: Soft Computing and Intelligent Systems Design—Theory, Tools, and Applications (with F. Karray, 2004). Prof. de Silva is a Fellow of: The Institute of Electrical and Electronics Engineers (IEEE), American Society of Mechanical Engineers (ASME), the Canadian Academy of Engineering, and the Royal Society of Canada.