Pedestrian Flow Optimization to Reduce the Risk of Crowd Disasters Through Human–Robot Interaction

Chao Jiang, *Student Member, IEEE*, Zhen Ni, *Member, IEEE*, Yi Guo, *Senior Member, IEEE*, and Haibo He, *Fellow, IEEE*

***Abstract*—Pedestrian flow in densely populated or congested areas usually presents irregular or turbulent motion state due to competitive behaviors of individual pedestrians, which reduces flow efficiency and raises the risk of crowd accidents. Effective pedestrian flow regulation strategies are highly valuable for flow optimization. Existing studies seek for optimal design of indoor architectural features and spatial placement of pedestrian facilities for the purpose of flow optimization. However, once placed, the stationary facilities are not adaptive to real-time flow changes. In this paper, we investigate the problem of regulating two merging pedestrian flows in a bottleneck area using a mobile robot moving among the pedestrian flows. The pedestrian flows are regulated through dynamic human-robot interaction (HRI) during their collective motion. We adopt an adaptive dynamic programming (ADP) method to learn the optimal motion parameters of the robot in real time, and the resulting outflow through the bottleneck is maximized with the crowd pressure reduced to avoid potential crowd disasters. The proposed algorithm is a data-driven approach that only uses camera observation of pedestrian flows without explicit models of pedestrian dynamics and HRI. Extensive simulation studies are performed in both MATLAB and a robotic simulator to verify the proposed approach and evaluate the performances.**

***Index Terms*—Pedestrian flow regulation, human-robot interaction, learning-based optimal control, and pedestrian crowd pressure.**

I. Introduction

MODELING and control of pedestrian collective motion behavior have received considerable research interest due to the increasing demand of effective pedestrian flow regulation and evacuation in public areas such as stadiums, shopping malls, and train stations. Without appropriate guidance and regulation, crowd disorder such as blocking [1], and irregular and turbulent pedestrian flow [2], [3] arises when pedestrians aggregate gradually. Particularly, crowd disorder may evolve into crowd accident such as stampedes under emergency circumstances due to competitive behavior of individual pedestrians. Therefore, investigations on pedestrian flow regulation strategies are of great importance for public crowd safety. The focuses of the existing work are primarily on either optimal evacuation planning [1], [4] or optimal architecture design and spatial placement of facilities [5]–[9] based on self-organization behavior of pedestrian collective motion. For instance, the study in [5] suggested that properly placing obstacles in front of an exit could mitigate crowd congestion and thus improves outflow efficiency. However, the optimal design of stationary facilities' geometry parameters vary with the changes of pedestrian flows [10]. As a result, stationary facilities are not adaptive to real-time changes of pedestrian flows as they may not be easily reconfigurable once being placed.

Most recently, the studies on human-robot interaction (HRI) has received remarkable attention in the applications of social robots such as human-aware navigation [11]–[14] and guidance-providing [15]–[17], where the robots are able to interact with pedestrians. Motivated by the shepherding behavior observed in animal flocks, where the collective motion of a group of agents is controlled by a limited number of external agents through repulsive or attractive interacting force [18], [19], mobile robots have been used to control or guide human groups [20]–[24]. The focuses of these work are on explicit guidance of robots without particular consideration of HRI. Previous studies have also suggested that mobile robots can influence the motion of a pedestrian crowd in a manner that no explicit guidance is required [25]–[28]. The pedestrian flows are implicitly controlled through the dynamic interaction between pedestrians and robots deployed in the pedestrian flows. These work have enlightened the development of new methods for pedestrian flow regulation and optimization, which use assistive mobile robots in place of stationary facilities. Our earlier work [29], [30] investigated the pedestrian flow regulation problem in a uni-directional exit corridor using a mobile robot, which learns passive human-robot interaction in real time and adjusts its motion accordingly to attain desired collective flow performance.

In this paper, we study a complex environment based on the real-world scenario presented in [3], [31], where pedestrian flows from two perpendicular directions merge together and

Manuscript received February 7, 2019; revised May 5, 2019 and June 24, 2019; accepted July 13, 2019. The work of C. Jiang and Y. Guo was supported by the U.S. National Science Foundation under Grant CMMI-1527016, Grant CMMI-1825709, and Grant IIS-1838799. The work of Z. Ni was supported by the U.S. National Science Foundation under Grant OIA-1833005. The work of H. He was supported by the U.S. National Science Foundation under Grant CMMI-1526835. *(Corresponding author: Yi Guo.)*

C. Jiang and Y. Guo are with the Stevens Institute of Technology, Hoboken, NJ 07030 USA (e-mail: cjiang6@stevens.edu; yi.guo@stevens.edu).

Z. Ni is with the South Dakota State University, Brookings, SD 57007 USA (e-mail: zhen.ni@sdstate.edu).

H. He is with the University of Rhode Island, Kingstown, RI 02881 USA (e-mail: haibohe@uri.edu).

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors. In the supplementary PDF file, the authors provide the details of the social force model, the definition of pedestrian flow variables, the details of the ADP learning control, additional robotic evaluation, and discussions.

Digital Object Identifier 10.1109/TETCI.2019.2930249

move through a bottleneck. Our goal is to regulate the merging pedestrian flows to achieve efficient pedestrian outflow through the bottleneck. In addition, this work also takes into account the crowd pressure [3], the quantity that measures the critical crowd condition that may evolve into crowd accidents. To regulate the merging pedestrian flows and achieve optimal flow performance, we propose a robot-assisted pedestrian regulation scheme that utilizes dynamic HRI and design a customized adaptive dynamic programming (ADP) learning control to tune the robot motion parameters in real time. Simulations have been conducted in Matlab and in a robotic simulator to validate our algorithm, and the results show successful flow regulation with significant outflow improvement, compared with the results without robot assistance.

The contributions of the paper are twofold. First, we propose to use a dynamically interacting robot to reduce crowd pressure build-up and to maximize merging pedestrian outflow through a bottleneck. Second, we design learning-based feedback control for the robot and optimize robot motion parameters online. Comparing existing work, the results presented in [25], [26] explored possible ways of deploying mobile robots to control pedestrian flows. As the preliminary attempts to solve pedestrian regulation problem using robots, these work didn't provide systematic methods for optimal control of robot motion. The work in [27], [28] analyzed the characteristics of HRI in the scenario of crossing pedestrian flows, and developed a robot motion control approach to regulate the flows for congestion reduction. However, the robot motion control approach developed therein requires knowledge of HRI characteristics under different robot motion frequencies, which makes their approach not applicable when such knowledge is not known a priori. In contrast to the approaches in [27], [28], our algorithm learns the optimal robot motion control parameter through the online learning of ADP that uses the observation of pedestrian flow only. Thus, our approach can be applied to online regulation without prior knowledge of pedestrian flow conditions and HRI characteristics. Compared with the simple scenario of uni-directional corridor presented in our previous papers [29], [30], the problem studied here is more challenging as the outflow at the bottleneck is the combined result of the behavior of merging flows and the capacity of the bottleneck. Crowd disasters are much more likely to occur in merging flow situations rather than in a uni-directional corridor environment. Compared with our recent work [32] where deep reinforcement learning was presented for end-to-end control of pedestrian flows, the ADP method proposed in this paper has much better real time online learning capability and avoids extensive offline training used by deep neural networks. Some preliminary results of this paper appeared in a conference paper [33], and substantial extensions are added in this paper including Sections II, III, VII, and the supplementary materials.

The rest of the paper is organized as follows: Section II reviews related work on robot-assisted evacuation and flow control approaches. Section III introduces the environmental setup and the motivation of the selected environment. Section IV describes the problem formulation of the merging pedestrian flow regulation with robot assistance. Section V presents the design of the ADP-based learning algorithm for real-time robot motion control. In Section VI and VII, we present the simulation results in both Matlab and PedSim platforms, respectively. We conclude our work in Section VIII.

II. Related Work

According to different ways that robots assist human crowds, existing work can be categorized into two types, *robot-assisted evacuation* and *robot-assisted flow control*. We discuss each of the two types of work in this section.

A. Robot-Assisted Evacuation

In this category, robots perform as mobile guiding agents which are able to access the global information of the egress location and the optimal evacuation route, and propagate them among evacuees. In [34], a team of autonomous mobile robots equipped with directional audio beacon was dynamically deployed in an office building in emergency situations. Each robot in the team served as a signal beacon which was activated after it reached the desired location. The deployment of audio beacon using multiple robots was formulated as an optimal task assignment problem and the evacuation algorithm was developed based on multi-robot task allocation. The work in [35] proposed an evacuation route discovery method, in which the robots deployed a network of sensor nodes while exploring the evacuation area. The evacuation routes were planned on the fly using distributed route discovery and existing exploration algorithms combined. However, the work primarily aims to find the shortest evacuation route, the crowd congestion issue in realistic evacuation process was not taken into consideration from a macroscopic perspective.

In [4], a robot guided evacuation scheme was proposed, in which the robot redirected a group of evacuees from congested exits to a less occupied one so as to accelerate the evacuation process. The environment was modelled as Cellular Automata (CA) workspace with an embedded dynamic potential field where exits and robots were considered as attractive sources. Simulations on crowd evacuation with and without robot intervention were conducted offline, the results of which provide guidance for robot trajectory planning in real-world evacuation experiment. In [36], a crowd panic model that governs the panic propagation among evacuees was developed and combined with an existing social force model to simulate realistic human behavior during evacuation. Then, a robot-assisted evacuation planning scheme was designed to improve evacuation efficiency. Specifically, evacuees were provided by the robot with evacuation guidance which was determined by an exit selection algorithm that fuses global information of evacuee flows to select an optimal exit with minimum escape time.

The focuses of aforementioned work are evacuation planning, where explicit robot guidance is provided to evacuees. However, little attention was paid to the effect of HRI in those work. Also, the work on egress route planning and exit selection considers the pedestrian crowd from a macroscopic model perspective, local crowd self-organization phenomena such as the clogging effect at exits were not particularly considered in evacuation planning.

B. Robot-Assisted Flow Control

It is known that modification of pedestrian facilities can increase efficiency and safety [5]. For example, adding "obstacles" can stabilize flow patterns and make the flow more fluid; adding zigzag-shaped geometries and columns can reduce pressure in panicked crowds [5]. However, modification of infrastructure is often expensive and not easily re-configurable in real time. Inspired by these studies of pedestrian flow control using stationary architecture facilities, attempts have been made to explore how pedestrian motion can be controlled by introducing autonomous mobile robot in pedestrian flows [30], taking advantage of human-robot interactions. Instead of placing stationary architecture facilities in the environment, these studies aim to design appropriate robot motion to optimize pedestrian flows.

In [25], the interaction between a robot and the human was modeled as a cohesive social force imposed on an individual person, which is embedded in the existing social force model to describe human motion dynamics. Two types of robot motion behaviors that could change pedestrian flow behavior were investigated, i.e., a group of robots maneuvered to move in different designate patterns or remain stationary with designate formation, and a group of robots with mutual social attraction moving within the pedestrian flow. The effect of the mobile robot on the average flow velocity was evaluated in two scenarios, and the results showed that desired pedestrian motion behavior can be obtained by HRI. The follow-up work in [26] presents the optimization of robot motion and the parameters of the HRI model using a genetic algorithm to improve pedestrian flow efficiency. The performance with optimized HRI model parameters was evaluated in a corridor scenario therein.

The work in [28] presents a new approach of controlling crossing pedestrian flows using mobile robot guides that move within the pedestrian flows to solve the flow congestion problem at intersections. The pedestrian dynamics was modeled by the continuum fluid model proposed in [27]. Taking advantage of the dynamic interaction between the robot and pedestrians, the robot in pedestrian flows helps to create diagonal stripe pattern in the crossing flows as pedestrians adjust their path to avoid collision with the robot. The swarm behavior of varying pedestrian flows were controlled by adjusting the frequency of robot motion to maximize the average flow velocity. However, the continuum fluid model-based method used therein does not consider the heterogeneity of individual pedestrian dynamics that may cause clogging and pressure build-up, and the robot control was designed offline based on different open-loop control performances.

In our earlier work [29], [30], the pedestrian flow optimization based on passive HRI was studied. The slow-down effect of the passive HRI on the pedestrian flows was utilized to regulate the flow velocity, and was verified in simulated experiments using social force models. Rigorous theoretical proof was provided for the convergence analysis of ADP-based control [30]. Our recent work [37] reported the empirical study of the effect of passive HRI in a uni-directional exit corridor, where both individual and collective motion of the pedestrians under the influence of robot motion was analyzed. The results of the empirical study are qualitatively consistent with the simulated pedestrian motion behavior in [30], and justify the passive HRI and its applicability in robot-assisted pedestrian regulation.

Motivated by the studies of real-world crowd accidents, our work aims to exploit a mobile robot in pedestrian flow regulation in a complex scenario with merging pedestrian flows where the crowd pressure may build up to cause crowd disasters. Unlike existing work on robot-assisted evacuation via explicit robot guidance, our approach utilizes passive HRI for the optimization of pedestrian outflow through the bottleneck and the avoidance of crowd pressure build-up. The passive HRI in our problem setup means that the robot moves in a controlled motion, and the pedestrians around the robot adapt their motions to avoid potential collisions with the robot. The collision avoidance behavior of pedestrians is a passive reaction to an approaching robot. In such a way, the robot motion affects the collective pedestrian flows through HRI. We propose to use the ADP-based method to learn HRI and tune robot motion control parameters online. Our proposed control method only takes the real-time measurement of pedestrian flows as feedback to adjust the robot motion. In the next section, we present the problem formulation of our robot-assisted merging pedestrian flow regulation.

III. The Environment and System Setup

The environment selection of this paper is inspired by the crowd stampede incident occurred in Mina/Makkah on January 12, 2006 [38], where tens of thousands of pilgrims moved toward the Jamarat Bridge to perform the stoning ritual. The stampede incident started at the entrance of the Jamarat Bridge where pilgrim pedestrians from different directions merge together. At least 345 pilgrims were killed and around 1000 were injured in the deadly stampede. The large casualties in such crowd accident have drawn considerable attention on crowd safety improvement from researchers who hope to investigate the underlying mechanism and possible solutions [3], [31], [39]. Specifically, Helbing, *et al.* [3] and Johansson, *et al.* [39] presented empirical studies on this particular crowd stampede. To quantitatively explain how critical crowd conditions originated and evolved into a crowd disaster in this incident, the authors analyzed the local crowd density, speeds and flows from the video recording of the accident site using the pedestrian tracking algorithm they developed. They observed the transition from a smooth pedestrian flow to a turbulent flow prior to the occurrence of the stampede. Following the work [3], Yu and Johansson [31] proposed a social force-based model to simulate the turbulent crowd movement for a crowd going through a bottleneck, and verified that the model can reproduce observed empirical features characterizing crowd turbulence during the annual Muslim pilgrimage.

In this paper, we select the environment shown in Fig. 1, which is obtained by down-scaling the environment presented in [31] to study the real-world crowd disaster mentioned above. The environment represents a typical structure where complex pedestrian motion behaviors take place due to merging flows and the bottleneck. We conduct numerical simulations using

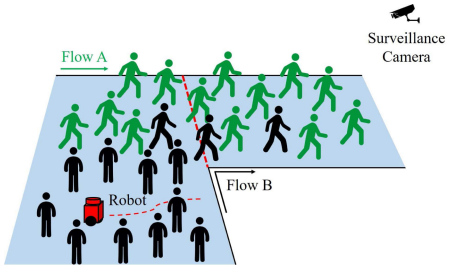


Fig. 1. The schematic diagram of merging pedestrian flows at the bottleneck area. Pedestrian outflow through the bottleneck is observed by the surveillance camera pre-installed in the environment.

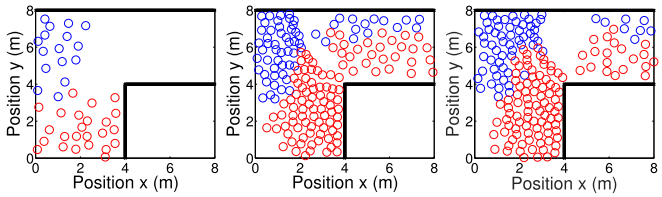


Fig. 2. Clogging effect observed in merging flows without regulation at a bottleneck area. The blue and red circles represent two pedestrian flows.

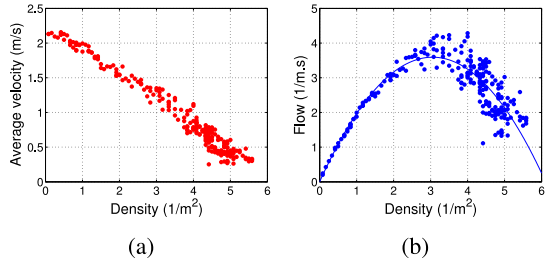


Fig. 3. The fundamental diagram of merging flows at the bottleneck area: (a) average velocity vs. density; (b) flow vs. density. The solid parabolic curve represents the fitting curve to the data.

the social force model (described in Section I of the supplementary material with flow variables defined in Section II of the supplementary material) in order to plot the fundamental diagrams of the uncontrolled merging flows. The time sequence of snapshots, as shown in Fig. 2, illustrate the process of two pedestrian flows merging and moving through a bottleneck. One can see that, as pedestrians aggregate in the bottleneck area, the crowd clogging starts at $t = 72$ s and evolves into a turbulent flow at $t = 78$ s. We analyze the relationships of the velocity-density and flow-density, and plot the fundamental diagram of the pedestrian flow in Fig. 3. Fig. 3a shows that the average flow velocity decreases monotonically from the free flow velocity as the flow density increases. Fig. 3b shows a parabolic flow-density curve, where the flow grows initially with the increase of density until the flow reaches the maximum value, i.e., 4 $(\text{m}\cdot\text{s})^{-1}$, at the density around 3.5 m^{-2}. Then crowd congestion takes place and the flow decreases drastically when the density is higher than 3.5 m^{-2}. Note that Figs. 2 and 3 are plotted using our simulated data, which is consistent

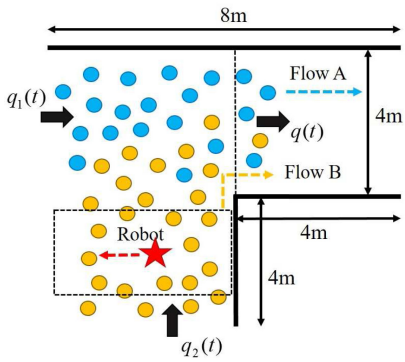


Fig. 4. The environmental setup. The dashed rectangle indicates the HRI region and the vertical dashed line indicates the observation region where pedestrian outflow is measured.

to the existing literature, e.g., the Greenshield's model in [40] which describes a linear relationship between traffic density and velocity, and a parabolic relationship between traffic density and flow.

Considering the importance of pedestrian flow regulation in the scenario discussed above, we choose such an environment for our proposed robot-assisted pedestrian regulation problem, and explore robot motion control in merging pedestrian flow regulation in order to maximize outflow and prevent potential crowd disasters.

IV. Problem Formulation

The environment shown in Fig. 4 is 8 m by 8 m with a bottleneck width $w = 4$ m. The amount of inflow discharging into the environment is denoted as $q_1(t)$ for flow A and $q_2(t)$ for flow B, respectively. We define *the HRI region* as the dashed rectangle, and the observation line as the vertical dashed line where *the pedestrian outflow* $q(t)$ (i.e., the number of pedestrians passed the line per meter per second) is measured by the pre-installed surveillance cameras. The left side of the observation line is the merging area of two pedestrian flows. A single interacting mobile robot is deployed in the HRI region, which moves in a pre-designated trajectory to dynamically interact with the pedestrian flow. The robot's velocity is determined by the proposed learning-based controller that takes observed real-time pedestrian outflow as feedback. The flow B is regulated through the effect of passive HRI.

As mentioned in Section II, we utilize passive HRI to regulate merging pedestrian flows, where the robot moves in a controlled motion, and the pedestrians around the robot adapt their motion to avoid potential collisions with the robot. Thus, the motion of the robot affects the collective pedestrian flows through passive HRI. Note that the effect of passive HRI on pedestrian flows was originally proposed in [27], [28] by Yamamoto and Okada, and then studied in our work [29], [30], [37]. In particular, real HRI experiments were conducted in [37] to validate the passive HRI effect on pedestrian flows and to show the consistency with simulations using social force models.

A. Robot Motion Dynamics

The robot state is defined as $\boldsymbol{x}_r = [x_r^x, x_r^y]^T \in \mathbb{R}^2$, where x_r^x and x_r^y represent the two-dimensional robot positions in the directions x and y, respectively. To focus on the higher-level robot motion planning problem, a single integrator model is used to describe the simplified robot motion model, that is,

$$\dot{\boldsymbol{x}}_r = \boldsymbol{u}_r \tag{1}$$

where $\boldsymbol{u}_r = [u_r^x, u_r^y]^T$ is the control input of the robot in the directions x and y, respectively.

The robot trajectory is pre-designed to be perpendicular to the moving direction of pedestrian flow B such that the robot behaves to mimic a "virtual gate" effect due to the repulsive effect between robot and pedestrians [30]. We consider flow A as the main flow. The branch flow (flow B) is regulated by the robot in order to avoid overcrowding that causes congestion at the bottleneck area. The robot movement helps to maximize the pedestrian outflow and meanwhile prevent crowd pressure build-up. The faster the robot moves, less people from flow B getting through and merging into the bottleneck area.

The robot velocity is set to zero in direction y, and the robot velocity is set to be sinusoidal in direction x, that is,

$$u_r^x(t) = A_0 \Omega \sin(\Omega t) \tag{2a}$$

$$u_r^y(t) = 0 \tag{2b}$$

where A_0 is the maximum displacement of robot position from the center of the branch corridor along direction x, Ω is the piecewise constant robot motion frequency that can be adjusted online.

B. Flow Optimization Problem

To formulate the robot-assisted flow optimization problem, the following assumptions are made for the environment and the robot:

Environment: Surveillance cameras are installed in the environment, which are used to observe the real-time pedestrian flow passing through the observation line. Thanks to the advance of computer vision and wireless sensor network techniques, real-time pedestrian crowd monitoring and analysis are available as reported in the literature such as [41], [42].

Mobile Robot: The mobile robot is able to access the real-time crowd information obtained from the surveillance cameras.

The merging pedestrian flow optimization problem is formulated as finding a sequence of robot motion frequency $\Omega(t)$ that minimize the value function

$$\underset{\Omega(t)}{\text{minimize}} \quad J = \int_{t_0}^{\infty} (q(t) - q^*)^2 \, dt \tag{3}$$

subject to the robot motion dynamics described in (1), where q^* is the bottleneck flow capacity (that can be pre-determined based on the infrastructure such as the width of the bottleneck). Minimizing J indicates that the outflow through the bottleneck is maximized over time. In the next section, we present our ADP-based learning control that solves this optimization problem.

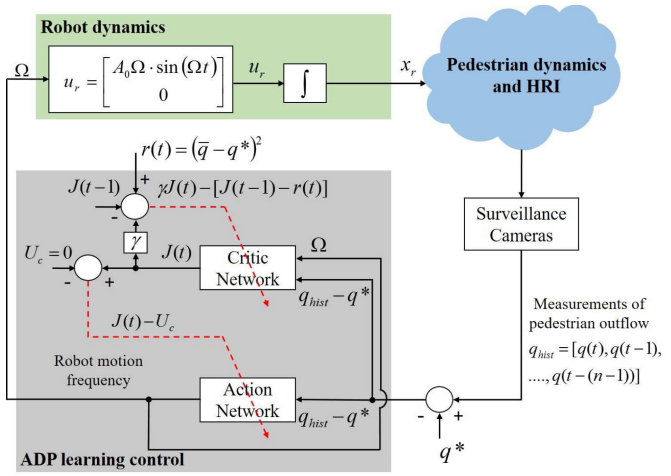


Fig. 5. The overall diagram of the robot-assisted pedestrian flow optimization. The ADP control block uses the measured camera data as inputs, observes the utility function $r(t)$ accordingly, and outputs robot motion frequency $\Omega(t)$ in real time. The solid lines represent signal flow, and the dashed line are the paths for parameter tuning.

V. ADP-Based Learning Control Design

In this section, we first present the overview of the robot-assisted pedestrian regulation system, and then introduce the design of the proposed ADP-based learning algorithm for robot motion control, which is followed by discussions on online implementation.

A. Overall Structure of Proposed Approach

The overall system diagram is depicted in Fig. 5, which is composed of the robot dynamics, the pedestrian dynamics and HRI, the surveillance cameras, and the ADP control. The model of the robot dynamics is defined in (1). The surveillance cameras measure the pedestrian outflow in the observation region and the measured data is fed into the ADP learning control module to control the robot motion. As this proposed control method is a data-driven approach, the pedestrian dynamics and HRI are considered as a black box that is unknown to the control design. Instead, only the observed pedestrian flow information is used as input for the ADP control algorithm. Note that, unlike our previous work [30] where the HRI region and the observation region are the same, in this paper the two regions are different without overlap. This makes the problem more challenging as the observed pedestrian flow is the result of both HRI and merging flow effects.

To solve the robot-assisted flow optimization problem, we design a dedicated ADP-based learning controller using measurement of pedestrian flow as feedback. As shown in Fig. 5, the ADP control module consists of two networks, i.e., a *critic network* and an *action network*. The critic network is used to approximate the value function J, and the action network is used to generate the robot motion frequency $\Omega(t)$. Both networks are implemented with multi-layer perceptron neural networks [43]. Due to the dynamic nature of pedestrian motion, the instantaneous observations of pedestrian outflow are very fluctuant. To reduce the effect of the fluctuation for the robot control design,

the average of n most recent measurements of pedestrian outflow in time history

$$\bar{q}(t) = \frac{1}{n}\sum_{k=0}^{n-1} q(t-k) \tag{4}$$

is used to calculate the utility function. The time history of outflow difference, i.e., $\boldsymbol{q}_{hist} - q^* = [q(t) - q^*, q(t-1) - q^*, ..., q(t-(n-1)) - q^*]$, is fed into the critic network and the action network as inputs. The critic network also uses the robot motion frequency as input in addition to the time history of outflow difference.

B. Design of the Utility and Objective Functions

In our ADP-based learning control design, the utility function is chosen as

$$r(t) = \left(\bar{q}(t) - q^*\right)^2 \tag{5}$$

where $\bar{q}(t)$ is the averaged n most recent measurement of outflow defined in (4).

The summation of the utility function $r(t)$ from current time instance t to the infinite future is defined as

$$R(t) = r(t) + r(t+1) + r(t+2) + \cdots \tag{6}$$

The role of the critic network in the ADP design is to estimate the total future cost $R(t)$. The optimal point of $R(t)$ is exactly the same as the value function J in (3) if $t = t_0$. Thus, the goal of this ADP module is to seek a sequence of robot motion frequency $\Omega(t)$ to minimize the value function J as

$$J^*(t) = \min_{u\in\Omega}\{r(t) + \gamma J^*(t+1)\} \tag{7}$$

where γ is a discount factor for the future cost function, and the J function is regarded as the summation of discounted cost from the current time to the infinite horizon future. The discount factor γ, which is usually in the range of $[0, 1]$, indicates that when we map any future cost to the current time instance, it should be discounted.

According to the ADP approximation error of Bellman's equation, the error function of the critic network is defined as the temporal-difference (TD) error, $e_c(t) = \gamma J(t) - [J(t-1) - r(t)]$, and the corresponding objective function is $E_c = \frac{1}{2}e_c^2(t)$. For the action network, the error function is defined as $e_a(t) = J(t) - U_c$, and the corresponding objective function is $E_a = \frac{1}{2}e_a^2(t)$. Once the outflow measurement is available, the ADP learning control module is executed to calculate the value function $J(t)$ and the robot frequency control $\Omega(t)$ by the critic network and the action network with initial weights, respectively. Then the weights in both networks are updated using the gradient descent (back-propagation) algorithm to minimize the objective function $E_c(t)$ and $E_a(t)$ till the maximum iteration or the error threshold is reached. Then the robot motion frequency $\Omega(t)$ is calculated with updated weights and returned to the robot control module. The steps to update the neural network weight parameters in each network follow the same process as reported in [43]–[45]. The details of the ADP learning control algorithm is presented in Section III of the supplementary material.

Algorithm 1: Robot Motion Control.

input : q^*: desired outflow;
$q(t), q(t-1), ..., q(t-(n-1))$: time history of pedestrian outflow measurements ;
output: robot motion control $\boldsymbol{u}_r(t)$

```
initialize: Ω(0) = 0, A_0 ;
for t ← n to T_f do
    if flow measurement q(t) is available then
        call ADP Learning Control module (inputs:
        [q(t) − q*, q(t−1) − q*, ..., q(t−(n−1)) − q*]);
        update robot motion frequency Ω(t) ;
    else
        Ω(t) ← Ω(t−1) ;
    end
    calculate robot control u_r(t) using (2);
end
```

Remark 1: The fundamental principle of the ADP method in this paper is based on the action-dependent heuristic dynamic programming (ADHDP) design presented in [46], and it shares similar actor-critic architecture with other actor-critic reinforcement learning algorithms (e.g., [47], [48]). Applying the general actor-critic framework to particular application problems will lead to various actor-critic algorithms with specific designs. Our ADP method is one type of actor-critic implementation in the sense that the critic estimates the action-value function and the one-step backup TD learning is used for policy evaluation.

Remark 2: The ADP method provides a general framework solving optimal control problems forward in time using only system data measured online, and its adaptive learning capability makes it efficient to solve our formulated pedestrian flow optimization problem. With the actor-critic architecture, applying the general learning control framework within the problem domain is non-trivial. A variety of issues are crucial to the success of our proposed ADP method, including: 1) appropriately formulating the problem, particularly the choice of state measurement as the input of the critic and action networks; 2) defining the utility function that captures the optimality of the problem domain; 3) designing the training strategies and parameters; 4) designing the neural network structure for the critic and actor implementation. Overall, this paper provides an innovative design of an ADP method targeting on emerging robotic applications, and demonstrates a success development of computational intelligence methodology that efficiently integrates the problem domain knowledge.

C. The Algorithm and Online Implementation

The robot motion control algorithm is summarized in Algorithm 1. The robot's motion parameters are initialized as $\Omega(0) = 0$ and A_0 is a given constant. If the pedestrian flow measurement $q(t)$ is available at time t, the ADP Learning Control module is executed to update robot motion frequency $\Omega(t)$, otherwise the robot motion frequency remains unchanged. Then, the current robot velocity control $\boldsymbol{u}_r(t)$ is updated using (2). Note that in general, the updating frequency of the robot

motion control is faster than the availability of the measurement data from the camera due to complexity in pedestrian tracking and image processing.

To apply the proposed ADP control algorithm online, we randomly initialize the parameters of the ADP networks, that is, the weight parameters of the action/critic networks are randomly initialized. The ADP controller needs some time to learn the mapping between the measurement of outflow and the robot motion by adjusting the weight parameters, and then the weight parameters converge to the optimal values with the control goal met. As demonstrated in the performance evaluation in the next section, the convergence time of the learning algorithm is fast and satisfies the practical requirement. Also, our simulation results (presented in the next section) show that the algorithm adapts to the changing flow conditions (such as different ratios of the two merging flows), and converges to different optimal control outputs.

VI. Simulation and Performance Evaluation

In this section, we verify the proposed robot-assisted pedestrian flow regulation approach and the ADP-based learning control in Matlab simulations. We first describe the simulation setup and parameters, and then present the results of open-loop pedestrian regulation to provide HRI characteristics. We then validate and evaluate the ADP-based learning algorithm for pedestrian flow optimization in extensive case studies.

A. Simulation Setup

The simulation environment is chosen as shown in Fig. 4. The initial speeds of the pedestrians in both flows are set to be Gaussian distributed with mean $\mu = 2$ m/s and standard deviation $\sigma = 0.3$ m/s, denoted as $\mathcal{N}(\mu, \sigma^2)$. The robot is initially placed at $(x_r^x, x_r^y) = (0.5, 2.5)$ m with zero initial speed. We set the robot control parameter as $A_0 = 1.5$ m so that the robot can regulate pedestrians across the entire corridor width without colliding with the boundaries. The robot initial position in x direction is $x_r^x = 0.5$ m. Note that the robot initial position only affects the transient period of the control process. The algorithm will converge to the optimum regardless of the choice of robot initial position. The sum of instantaneous inflow A, q_1, and inflow B, q_2, is set to be constant, i.e., $q_1 + q_2 = 5$ (m·s)$^{-1}$. We vary the ratio of the two inflows to create two case studies, i.e., Case 1 where the ratio of two inflows is $q_1/q_2 = 3/2$ and Case 2 where the ratio of two inflows is $q_1/q_2 = 2/3$. The two cases of inflow ratio represent different pedestrian flow conditions that inflow A is greater than inflow B, and inflow A is less than inflow B, respectively. The desired flow is set as $q^* = 4$ (m·s)$^{-1}$ according to the bottleneck capacity, and is also consistent with the simulation result shown in Fig. 3b. The duration of each simulation run is set as $T_f = 200$ s. The size of the time history of pedestrian outflow measurement is set as $n = 5$.

The pedestrians' motion is simulated using the social force models reported in [5], [31] with the embedded HRI term used in [29], [30], [49]. The details of the social force model is presented in Section I of the supplementary material, in which the parameters are summarized in Table I. Note that it was reported in [31] that the empirical features characterizing crowd turbulence are reproduced well by this simulated model and such a model demonstrates the crowd pressure build-up leading to crowd disasters.

The critic and action networks in the ADP-based control algorithm are both three-layer networks. The parameters used in the ADP-based control algorithm are summarized in Table II of the supplementary material.

B. Open-Loop Robot Control for HRI Characteristics

Before validating our ADP-based learning control algorithm, we first conduct open-loop robot control simulations to characterize the effect of HRI on the pedestrian outflow at the bottleneck. In the open-loop simulation, the robot is controlled to move at a set of constant motion frequencies, Ω, ranging from 0.1 rad/s to 1.5 rad/s with an increment of 0.1 rad/s, given a constant ratio of pedestrian inflows q_1 and q_2. We then plot the pedestrian outflow, $q(t)$, for each robot motion frequency Ω.

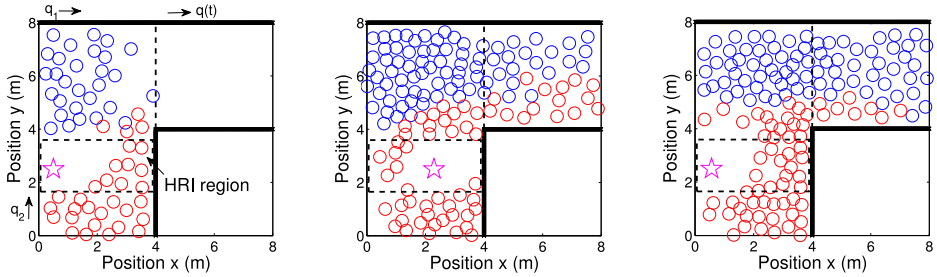
The temporal sequences of the open-loop simulations are shown in Fig. 6. Fig. 7 shows the outflow, $q(t)$, vs. robot motion frequency, Ω, of the two cases, where the ratio of the two flows varies as defined in Section VI-A. One can see from the results that for each case a maximum of average outflow can be obtained when the robot moves at a unique optimal motion frequency, Ω_{opt}. The optimal frequency Ω_{opt} is 0.4 rad/s and 0.7 rad/s for Case 1 and Case 2, respectively. Correspondingly, when the branch flow q_2 (flow B) is smaller in Case 1, the robot moves slower; and if q_2 grows bigger in Case 2, the robot moves faster to have less people merge into the bottleneck for efficient traffic control.

The simulation results of open-loop robot control present the quantitative HRI characteristics for merging pedestrian flow regulation. The results will be used as the "ground truth" of the optimization goal to validate whether our proposed ADP-based learning control algorithm can adjust the robot's motion frequency to the optimal value by learning from HRI online. In the next subsection, the optimal robot motion frequency for the two cases obtained in Fig. 7 will be compared with the frequency generated online from the ADP learning control.

C. ADP Control Performance

We have conducted extensive Matlab simulations to validate our developed algorithm and to evaluate the performances. Due to space limitation, we show a few representative cases in this subsection. We first present the case studies with fixed pedestrian flows, i.e., Cases 1 and 2 defined in Section VI-A. We then evaluate the online learning capability when the pedestrian flow condition changes from Case 2 to Case 1. After that, we evaluate the crowd pressure with and without robot to demonstrate that our developed algorithm is able to reduce the crowd pressure to avoid potential crowd diasters. Last, we show the statistical results of the online learning-based control.

1) Fixed Flow Case Studies: We implemented Algorithm 1 and performed simulations for Cases 1 and 2 where the pedestrian inflow ratio is set to be $3/2$ and $2/3$, respectively. The simulation results of our ADP control are shown in Fig. 8 and



(a) t=2s (b) t=100s (c) t=200s

Fig. 6. Snapshots of the open-loop simulation. The blue and red circles represent pedestrian flow A and flow B, respectively, and the pink star represents the robot. The dashed rectangle indicates the HRI region and the vertical dashed line indicates the bottleneck where outflow is measured.

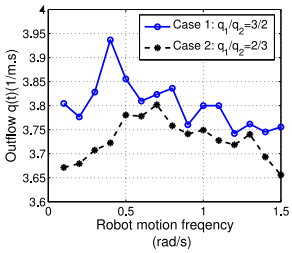
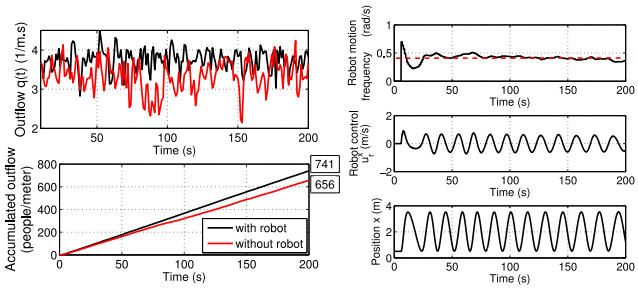
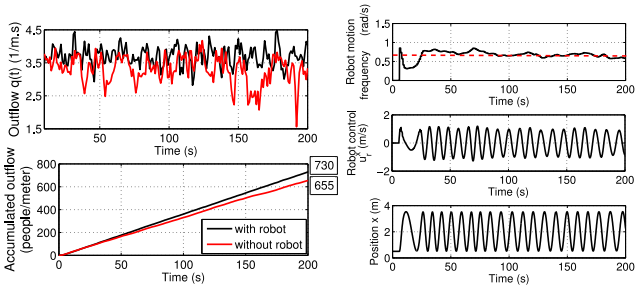


Fig. 7. The results of open-loop control under different pedestrian flow conditions.



(a) (b)

Fig. 8. Case 1 for pedestrian inflow $q_1/q_2 = 3/2$: (a) instantaneous pedestrian outflow $q(t)$, and accumulated pedestrian outflow $\int_{t_0}^{t} q(\tau)d\tau$; (b) time history of robot motion frequency $\Omega(t)$, robot control $u_r^x(t)$, and robot position $x_r^x(t)$. The red dotted line in (b) denotes the optimal frequency.



(a) (b)

Fig. 9. Case 2 for pedestrian inflow $q_1/q_2 = 2/3$: (a) instantaneous pedestrian outflow $q(t)$, and accumulated pedestrian outflow $\int_{t_0}^{t} q(\tau)d\tau$; (b) time history of robot motion frequency $\Omega(t)$, robot control $u_r^x(t)$, and robot position $x_r^x(t)$. The red dotted line in (b) denotes the optimal frequency.

Fig. 9 for Cases 1 and 2, respectively. Note that we want to confirm that our ADP control algorithm returns the optimal motion frequency for the robot, which is 0.4 rad/s for Case 1, and 0.7 rad/s for Case 2, as demonstrated by the open-loop robot control simulation in Fig. 7.

In Case 1, as shown in the first sub-figure of Fig. 8b, the robot gradually learns from the observation of pedestrian outflow, and the motion frequency converges after approximately 75 s around the optimal value, $\Omega_{opt} = 0.4$ rad/s, as indicated by the red dashed line in the figure. The second sub-figure of Fig. 8b shows the time history of robot control signal, and the third sub-figure shows the time history of robot position in direction x. Correspondingly, one can see from the first sub-figure of Fig. 8a that the instantaneous outflow, $q(t)$, approaches the maximum bottleneck capacity, $q^* = 4$ (m·s)$^{-1}$, with robot-assisted regulation (shown as the black curve), while the instantaneous outflow is much less without robot-assisted regulation (shown as the red curve). In addition, without robot, the flow drops significantly due to the clogging effect when pedestrian flows aggregate in the merging area, and the robot running our control algorithm can keep the flow smooth. The time history of the accumulated pedestrian outflow, $\int_{t_0}^{t} q(\tau)d\tau$, is shown in the second sub-figure of Fig. 8a . Comparing the accumulated outflow results with robot (black curve) and without robot (red curve), we can see that the accumulated outflow is 656 people/meter without robot-assisted regulation and 741 people/meter with robot-assisted regulation at time $t = 200$ s, which shows that the accumulated outflow is improved by 12.9% with robot-assisted regulation.

Similarly, as shown in the first sub-figure of Fig. 9b for Case 2, the robot motion frequency is gradually learned from the observation of pedestrian outflow, and converges around the optimal value, $\Omega_{opt} = 0.7$ rad/s, after approximately 80 s. The second sub-figure in Fig. 9b shows the time history of robot control signal. The first sub-figure of Fig. 9a shows the time history of instantaneous outflow, $q(t)$, with and without robot-assisted regulation. One can observe the sharp declines of instantaneous outflow without robot in Fig. 9a . It can be seen from the second sub-figure of Fig. 9a that the accumulated

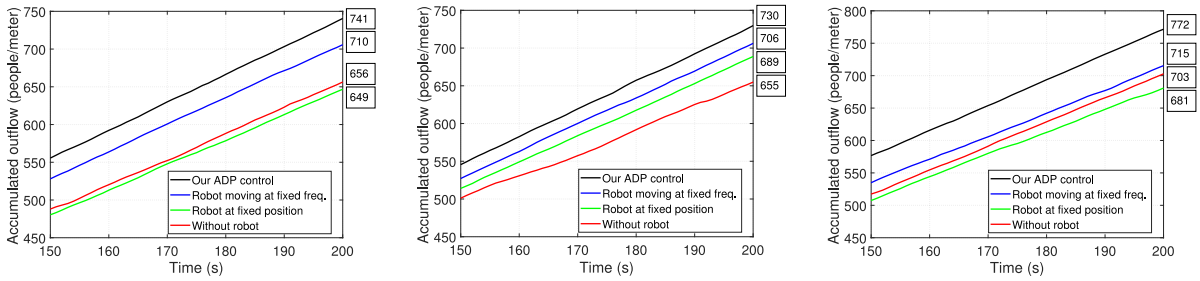


Fig. 10. Comparison of our proposed ADP control with ad-hoc robot behaviors: 1) robot moving at a fixed frequency $\Omega = 1.2$, shown in blue curves; 2) robot staying at the fixed position (3, 3.5), shown in green; and 3) without robot regulation, shown in red. Three different inflow conditions are shown as: (a) $q_1/q_2 = 3/2$; (b) $q_1/q_2 = 2/3$; (c) $q_1/q_2 = 4/1$.

outflow, $\int_{t_0}^{t} q(\tau)d\tau$, is 655 people/meter without robot-assisted regulation and 730 people/meter with robot-assisted regulation at time $t = 200$ s, which shows an improvement of 11.5% with robot-assisted regulation.

The above simulations verify that our ADP control algorithm converges to the optimal robot motion frequencies for both Cases 1 and 2. The performance shows that the robot can regulate the pedestrian flow smoothly and avoid significant declines of the pedestrian outflow after the merging area. The convergence time is below 100 s, which meets the practical requirement of pedestrian evacuation. Accordingly, the temporally accumulated flow is increased about 10% after 200 s.

2) Performance Comparison With Ad-hoc Robot Behaviors: We further compare the accumulate outflow with 1) our ADP algorithm, 2) robot moving at a fixed frequency 1.2 rad/s, 3) robot staying at a fixed position $(3, 3.5)$ m, and 4) no robot. Note that the fixed position and fixed motion frequency are randomly selected to represent ad-hoc robot behaviors for comparison purposes. The accumulate outflow results of three inflow ratios, $q_1/q_2 = 3/2$, $q_1/q_2 = 2/3$, and $q_1/q_2 = 4/1$, are plotted in Fig. 10. Comparing the performance of our proposed ADP control with other ad-hoc robot behaviors, we can see that 1) our algorithm achieves best performances under different inflow conditions; 2) the performances of ad-hoc robot behaviors vary under different inflow conditions. For example, compared with the no-robot case, adding a robot moving at a fixed frequency improves the pedestrian flows in all three different flow condition cases, although the degree of improvement varies in different cases as shown in the figure. On the other hand, adding a robot at a fixed position improves accumulated flow in one case (b), but does not help at all in the other two cases (a) and (c), which is understandable in the sense that an inappropriately placed obstacle may impede and harm the flow traffic. From this figure, it clearly shows the advantage of our proposed ADP control, which can find the best motion frequency in different flow conditions. Using a pre-determined robot motion frequency or a fixed robot position may not improve pedestrian flows when the flow condition changes. To further demonstrate the dynamic reconfigurability of our proposed algorithm, we show the online learning capability with changing flows in the next subsection.

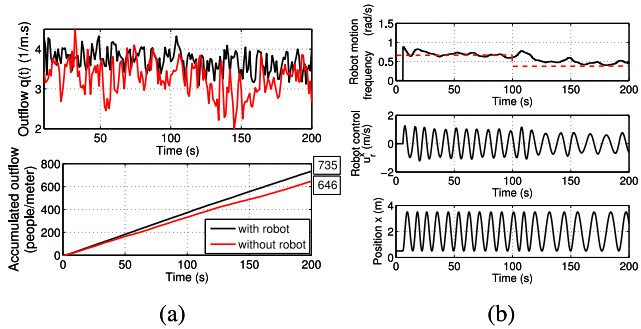


Fig. 11. Changing pedestrian inflow, where pedestrian inflow $q_1/q_2 = 2/3$ changes to $q_1/q_2 = 3/2$ at $t = 100$ s: (a) instantaneous pedestrian outflow $q(t)$, and accumulated pedestrian outflow $\int_{t_0}^{t} q(\tau)d\tau$; (b) time history of robot motion frequency $\Omega(t)$, robot control $u_r^x(t)$, and robot position $x_r^x(t)$. The red dotted line in (b) denotes the optimal frequency.

3) Online Learning With Changing Pedestrian Flows: To further demonstrate the online learning capability, we show the performance of the ADP-based learning control with changing pedestrian inflow conditions, that is, the inflow ratio of flow A and flow B is initially set to be Case 2 with $q_1/q_2 = 2/3$, then it changes to be Case 1 with $q_1/q_2 = 3/2$ at $t = 100$ s. This case represents the scenario where the main inflow (i.e., flow A) is initially less than the branch inflow (i.e., flow B), and then it gets more than the branch inflow as time goes by. The robot is expected to adjust its motion accordingly to maximize the outflow at the bottleneck regardless of the changes of two inflows.

The simulation results for the changing flow case is shown in Fig. 11. One can see from the first sub-figure in Fig. 11b that the robot motion frequency settles around $\Omega_{opt1} = 0.7$ rad/s after about 65 s. When the inflows changes at 100 s, the robot adjusts its motion frequency again, which converges around $\Omega_{opt2} = 0.4$ rad/s after about 160 s. Fig. 11a shows the time history of instantaneous outflow and accumulated outflow. From the first sub-figure of Fig. 11a, one can observe the sharp declines of instantaneous outflow due to congestion without robot-assisted regulation. In comparison, the instantaneous outflow with robot-assisted regulation is smoother. The second sub-figure of

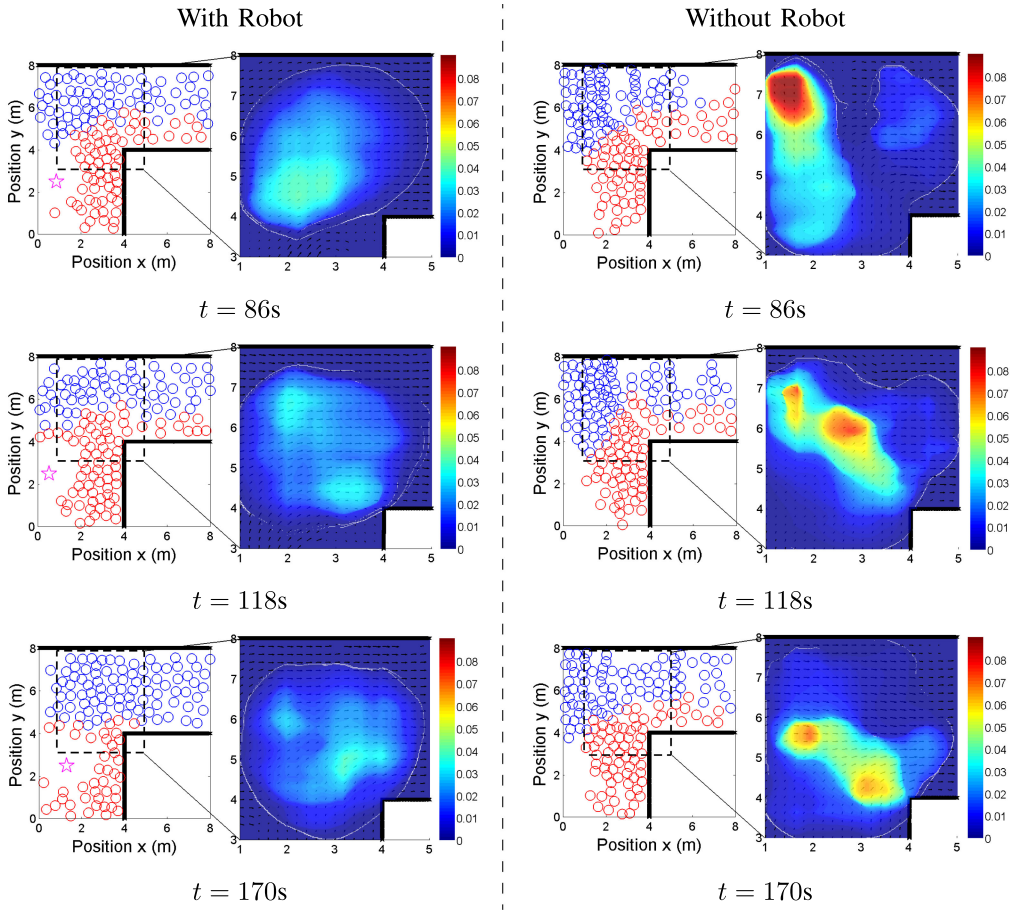


Fig. 12. Comparison of the heat map of crowd pressure in the merging area with and without robot-assisted regulation, which indicates that the crowd pressure is reduced to alleviate the risk of crowd accidents. The color bar indicates the scale of crowd pressure $P(\boldsymbol{x}, t)$ in the unit of (s^{-2}).

Fig. 11a shows that at time $t = 200$ s, the accumulated outflow is 646 people/meter without robot-assisted regulation, while the accumulated outflow is 735 people/meter with robot-assisted regulation. The accumulated outflow is improved by 13.8% with robot-assisted regulation.

4) Crowd Pressure in Merging Flows: The "crowd pressure" was proposed in [3] to measure the degree of congestion in pedestrian flows, and it is defined in (II.5) of the supplementary material as the product of flow density and velocity variance. Crowd pressure is a key quantity indicating the risk of crowd stampede as congestion builds up. As the crowd pressure is correlated to the smoothness of merging flow at the bottleneck area, improving the overall outflow efficiency by using a robot also helps to avoid crowd pressure accumulation at the same time. We compare the simulation results of the crowd pressure with and without robot in this subsection.

We plot the crowd pressure, $P(\boldsymbol{x}, t)$, as defined in (II.6) of the supplementary material, for the rectangular area with the x-coordinate from 1 m to 5 m, and the y-coordinate from 3 m to 8 m that covers the merging flow area. Fig. 12 show the spatial distribution of local crowd pressure calculated by (II.6) at the time $t = 86$ s, $t = 118$ s and $t = 170$ s for the case with and without robot, respectively. One can see that, in the case without robot, the local crowd pressure builds up and can reaches up to 0.088 s^{-2} at the red-colored locations where the most turbulent pedestrian motion is observed due to congestion. On the contrary, in the case with robot, the local crowd pressure stays at low levels and thus the motion of pedestrian flow is smoother at the bottleneck area.

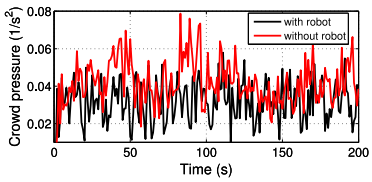


Fig. 13. Temporal evolution of spatially averaged crowd pressure in the merging area, $P(t)$.

Fig. 13 shows the time history of the spatially averaged crowd pressure, $P(t)$, defined in (II.5) of the supplementary material, for Case 1 of the simulation. One can see that the crowd pressure without robot is higher than that with robot on average. Particularly, without robot, the crowd pressure reaches peak values at times, e.g., $t = 86$ s, $t = 118$ s and $t = 170$ s, which indicates potential crowd disasters. With robot-assisted regulation, the peak values are significantly reduced by 26.5% from 0.079 s^{-2} to 0.058 s^{-2}.

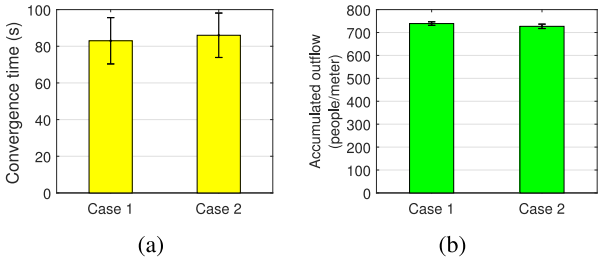


Fig. 14. Statistical results of Case 1 and Case 2 over 50 runs: (a) average convergence time; (b) average accumulated outflow. The error bars indicate the standard deviation.

5) Statistical Results: In this subsection, we provide statistical results of our ADP control algorithm. In our simulations, each pedestrian's initial position and velocity are randomly assigned from Gaussian distributions and the ADP weights are randomly initialized between $[-0.5,0.5]$. We present the statistical results of 50 runs for Case 1 and Case 2, respectively. Extensive simulations are performed to obtain the statistical results. We present the results of 50 simulated experiments as they are representative to demonstrate the performance of our algorithm. The duration of each simulation run is set as 200 s. We consider a run successful if the output of the ADP function, i.e. robot motion frequency $\Omega(t)$, converges to the optimal value Ω_{opt} within 200 s in the sense that the average error between $\Omega(t)$ and Ω_{opt} over the most recent 50 s is smaller than 0.01, i.e., $\sum_t[|\Omega(t)-\Omega_{opt}|^2]/50 < 0.01$, for $t \in (150, 200]$. The success rates for Case 1 and Case 2 are 80% and 78%, respectively. The statistical results are presented in Fig. 14. As shown in Fig. 14a, it takes on average 83 s for Case 1 and 86 s for Case 2 to converge, respectively. The standard deviation of convergence time is 12.6 for Case 1, and 12.1 for Case 2. One can see from Fig. 14b that the average accumulated outflow in 200 s is 739 people per meter for Case 1 and 727 people per meter for Case 2. The standard deviation of accumulated outflow is 7.81 for Case 1, and 9.58 for Case 2. It can be concluded from the results that the proposed ADP-based learning method is effective with acceptable successful rate.

VII. Robotic Evaluation Results

In this section, we present robotic simulation results to demonstrate that the developed ADP control algorithm applies to real robot platforms and the performances are independent of pedestrian simulators. For this purpose, we use the open source pedestrian simulator PedSim in Robot Operating System (ROS) to simulate the pedestrian populated environment [50]. PedSim is a microscopic pedestrian crowd simulator that implements the social force model and renders interface for 3D visualization to emulate realistic pedestrian crowds in real-world scenarios. We also implemented our ADP algorithm and the robot motion control in a ROS program which can be readily transferred on a real robot to conduct real-world experiments. The ROS package of the PedSim simulator is available in [51].

A. Robotic Simulation Platforms

The experimental platform includes ROS Indigo with the 3D visualization tool Rviz and the ROS implementation of PedSim simulator. The PedSim simulator simulates the motion of the pedestrians and the robot with embedded motion models. The ROS program that implements our ADP algorithm takes as input the observation of current outflow from the PedSim simulator, and calculates the robot velocity control in real time. The robot control signals are then fed into the PedSim simulator to control the robot's motion. The visualization of the simulation is done by Rviz. The data exchange among different modules are managed by the ROS communication mechanism. The experiments were conducted on a desktop computer with an Intel Core™ i7-6700 3.40 GHz CPU and 16 GB RAM in Ubuntu 14.04 operating system.

The motion of the pedestrians in the simulator is governed by the social force model presented in the supplementary material, and we further tune the parameters of the social force model to acquire more realistic pedestrian motion behaviors. The parameters modified in Table I are as follows: maximum pedestrian interaction strength $A_{ij} = 8$; pedestrian radius $r_i = 0.3$; effective range of HRI $B_{ir} = 0.6$. Note that due to different implementation mechanisms (such as solving the differential equations of pedestrian dynamics) in the PedSim and Matlab environments, the pedestrian behaviors are slightly different even with similar social force models.

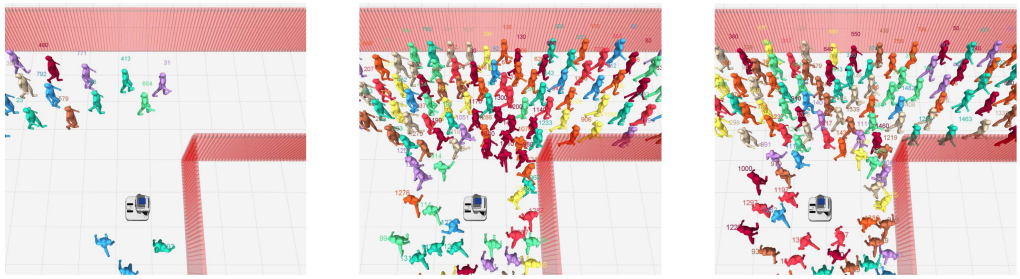
In the next subsection and Section IV in the supplementary material, we present two different intersection environments and validate our proposed algorithm in these environments.

B. Case Study

In this case, the merging flow scenario in the environment shown in Fig. 4 was created. The number of pedestrians in flow A and flow B are set to be 600 each. The pedestrians in two flows were randomly initialized in the entry regions with initial speed following a Gaussian distribution of $\mathcal{N}(2, 0.3^2)$. The inflow of flow A and flow B that discharge into the environment are both about 1.25 $(\text{m}\cdot\text{s})^{-1}$. The robot was initialized at $(2.6, 2.5)$ m with zero initial speed.

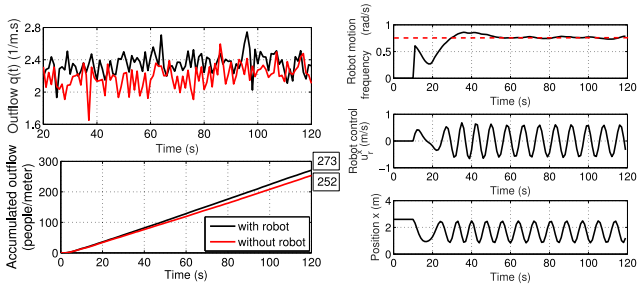
We have performed similar case studies as presented in Section VI, and obtained consistent results as in the Matlab simulations. We only present one case with pedestrian inflow ratio $q_1/q_2 = 1$ due to space limitation.

The snapshots of merging pedestrian flows with robot-assisted regulation are shown in Fig. 15. Fig. 16 shows the numerical results of the performance in PedSim. From the time history of robot motion frequency $\Omega(t)$ shown in the first sub-figure in Fig. 16b , one can see that the robot motion frequency converges around 0.78 rad/s after about 60 s. The second and third sub-figures in Fig. 16b show the time history of robot velocity and position in direction x, respectively. It can be seen from the first sub-figure in Fig. 16a that the instantaneous outflow with robot-assisted regulation is improved, compared with the result without robot-assisted regulation. The second sub-figure in Fig. 16a shows that at $t = 120$ s, the accumulated outflow



(a) t=5 s (b) t=55 s (c) t=90 s

Fig. 15. Snapshots of robotic experimental case study with robot-assisted regulation.



(a) (b)

Fig. 16. Robotic experiment results for environment case study 1: (a) instantaneous pedestrian outflow $q(t)$, and accumulated pedestrian outflow $\int_{t_0}^{t} q(\tau)d\tau$; (b) time history of robot motion frequency $\Omega(t)$, robot control $u_r^x(t)$, and robot position $x_r^x(t)$. The red dotted line in (b) denotes the converged frequency.

with robot is 273 people/meter, while the accumulated outflow without robot is 252 people/meter. The improvement is 8.3% after 120 s.

To verify the generalizability of our approach in different intersection environments, we further study another merging flow scenario and the results are presented in the supplementary material. The two environment case studies demonstrate that the algorithm applies to different merging flow situations. In general, the proposed method is scalable to multiple merging flows as long as there exists optimal solutions to robot motion control achieving maximum pedestrian outflow using a single robot. Indeed, our proposed method does not rely on any modeling methods of merging flow dynamics and the environment, but uses the observed outflow measurement to adjust the robot motion speed through online learning that is facilitated by the critic and action neural networks in the proposed ADP architecture.

Computational time: We also calculate the computational time spent by the ADP learning control module to output the control parameters after receiving the flow measurement data in one execution, denoted as C_i. The average computational time over 100 executions, i.e., $(\sum_{i=1}^{100} C_i)/100$, is 6.8 ms.

VIII. Conclusion

In this paper, we investigated the merging pedestrian flow regulation problem in a bottleneck environment. We proposed to use a mobile robot that dynamically interacts with the pedestrian flow, and designed an ADP-based learning method for robot motion control. The pedestrian regulation problem was formulated as an optimal control problem, and a customized ADP approach was designed to solve the formulated optimal control that adjusts robot motion parameters online. Simulation results in both Matlab and the robotic simulator demonstrated that our approach can regulate pedestrian flows to optimize outflow by online learning from the real-time observation of the pedestrian flow, and the critical crowd pressure is reduced to prevent potential crowd disasters.

References

[1] P. B. Luh, C. T. Wilkie, S.-C. Chang, K. L. Marsh, and N. Olderman, "Modeling and optimization of building emergency evacuation considering blocking effects on crowd movement," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 4, pp. 687–700, Oct. 2012.

[2] D. Helbing, I. Farkas, and T. Vicsek, "Simulating dynamical features of escape panic," *Nature*, vol. 407, no. 6803, pp. 487–490, 2000.

[3] D. Helbing, A. Johansson, and H. Z. Al-Abideen, "Dynamics of crowd disasters: An empirical study," *Phys. Rev. E*, vol. 75, no. 4, 2007, Art. no. 046109.

[4] E. Boukas, I. Kostavelis, A. Gasteratos, and G. Sirakoulis, "Robot guided crowd evacuation," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 2, pp. 739–751, Apr. 2014.

[5] D. Helbing, L. Buzna, A. Johansson, and T. Werner, "Self-organized pedestrian crowd dynamics: Experiments, simulations, and design solutions," *Transp. Sci.*, vol. 39, no. 1, pp. 1–24, 2005.

[6] M. Okada, Y. Motegi, and K. Yamamoto, "Human swarm modeling in exhibition space and space design," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2011, pp. 5021–5026.

[7] S. Rodriguez, Y. Zhang, N. Gans, and N. M. Amato, "Optimizing aspects of pedestrian traffic in building designs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 1327–1334.

[8] A. Johansson and D. Helbing, "Pedestrian flow optimization with a genetic algorithm based on boolean grids," in *Pedestrian and Evacuation Dynamics 2005*. Berlin, Germany: Springer, 2007, pp. 267–272.

[9] G. Frank and C. Dorso, "Room evacuation in the presence of an obstacle," *Physica A, Statist. Mech. Appl.*, vol. 390, no. 11, pp. 2135–2145, 2011.

[10] Y. Zhao *et al.*, "Optimal layout design of obstacles for panic evacuation using differential evolution," *Physica A, Statist. Mech. Appl.*, vol. 465, pp. 175–194, 2017.

[11] M. Bennewitz, W. Burgard, G. Cielniak, and S. Thrun, "Learning motion patterns of people for compliant robot motion," *Int. J. Robot. Res.*, vol. 24, no. 1, pp. 31–48, 2005.

[12] C.-P. Lam, C.-T. Chou, K.-H. Chiang, and L.-C. Fu, "Human-centered robot navigation towards a harmoniously human–robot coexisting environment," *IEEE Trans. Robot.*, vol. 27, no. 1, pp. 99–112, Feb. 2011.

[13] H. Kidokoro, T. Kanda, D. Brščić, and M. Shiomi, "Simulation-based behavior planning to prevent congestion of pedestrians around a robot," *IEEE Trans. Robot.*, vol. 31, no. 6, pp. 1419–1431, Dec. 2015.

[14] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard, "Socially compliant mobile robot navigation via inverse reinforcement learning," *Int. J. Robot. Res.*, vol. 35, no. 11, pp. 1289–1307, 2016.
[15] S. Satake, T. Kanda, D. F. Glas, M. Imai, H. Ishiguro, and N. Hagita, "A robot that approaches pedestrians," *IEEE Trans. Robot.*, vol. 29, no. 2, pp. 508–524, Apr. 2013.
[16] P. Liu, D. F. Glas, T. Kanda, and H. Ishiguro, "Data-driven HRI: Learning social behaviors by example from human–human interaction," *IEEE Trans. Robot.*, vol. 32, no. 4, pp. 988–1008, Aug. 2016.
[17] D. Brščić, T. Ikeda, and T. Kanda, "Do you need help? A robot providing information to people who behave atypically," *IEEE Trans. Robot.*, vol. 33, no. 2, pp. 500–506, Apr. 2017.
[18] J.-M. Lien, B. Bayazit, R. T. Sowell, S. Rodriguez, and N. M. Amato, "Shepherding behaviors," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2004, vol. 4, pp. 4159–4164.
[19] D. Strömbom *et al.*, "Solving the shepherding problem: Heuristics for herding autonomous, interacting agents," *J. Roy. Soc. Interface*, vol. 11, no. 100, 2014, Art. no. 20140719.
[20] J.-M. Lien, S. Rodriguez, J.-P. Malric, and N. M. Amato, "Shepherding behaviors with multiple shepherds," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2005, pp. 3402–3407.
[21] E. A. Martinez-Garcia, O. Akihisa, and S. Yuta, "Crowding and guiding groups of humans by teams of mobile robots," in *Proc. IEEE Workshop Adv. Robot. Social Impacts*, 2005, pp. 91–96.
[22] C. Vo, J. F. Harrison, and J.-M. Lien, "Behavior-based motion planning for group control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2009, pp. 3768–3773.
[23] A. Garrell, A. Sanfeliu, and F. Moreno-Noguer, "Discrete time motion model for guiding people in urban areas using multiple robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2009, pp. 486–491.
[24] A. Garrell and A. Sanfeliu, "Cooperative social robots to accompany groups of people," *Int. J. Robot. Res.*, vol. 31, no. 13, pp. 1675–1701, 2012.
[25] J. A. Kirkland and A. A. Maciejewski, "A simulation of attempts to influence crowd dynamics," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, 2003, vol. 5, pp. 4328–4333.
[26] B. D. Eldridge and A. A. Maciejewski, "Using genetic algorithms to optimize social robot behavior for improved pedestrian flow," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, 2005, vol. 1, pp. 524–529.
[27] K. Yamamoto and M. Okada, "Continuum model of crossing pedestrian flows and swarm control based on temporal/spatial frequency," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 3352–3357.
[28] K. Yamamoto and M. Okada, "Control of swarm behavior in crossing pedestrians based on temporal/spatial frequencies," *Robot. Auton. Syst.*, vol. 61, no. 9, pp. 1036–1048, 2013.
[29] C. Jiang, Z. Ni, Y. Guo, and H. He, "Robot-assisted pedestrian regulation in an exit corridor," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 815–822.
[30] C. Jiang, Z. Ni, Y. Guo, and H. He, "Learning human–robot interaction for robot-assisted pedestrian flow optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 4, pp. 797–813, Apr. 2019.
[31] W. Yu and A. Johansson, "Modeling crowd turbulence by many-particle simulations," *Phys. Rev. E*, vol. 76, no. 4, 2007, Art. no. 046105.
[32] Z. Wan, C. Jiang, M. Fahad, Z. Ni, Y. Guo, and H. He, "Robot-assisted pedestrian regulation based on deep reinforcement learning," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2018.2878977.
[33] C. Jiang, Z. Ni, Y. Guo, and H. He, "Optimization of merging pedestrian flows based on adaptive dynamic programming," in *Proc. Amer. Control Conf.*, 2019, pp. 2626–2632.
[34] D. A. Shell and M. J. Matarić, "Insights toward robot-assisted evacuation," *Adv. Robot.*, vol. 19, no. 8, pp. 797–818, 2005.
[35] E. Ferranti and N. Trigoni, "Robot-assisted discovery of evacuation routes in emergency scenarios," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2008, pp. 2824–2830.
[36] B. Tang, C. Jiang, H. He, and Y. Guo, "Human mobility modeling for robot-assisted evacuation in complex indoor environments," *IEEE Trans. Human Mach. Syst.*, vol. 46, no. 5, pp. 694–707, Oct. 2016.
[37] Z. Chen, C. Jiang, and Y. Guo, "Pedestrian-robot interaction experiments in an exit corridor," in *Proc. Int. Conf. Ubiquitous Robots*, 2018, pp. 29–34.
[38] Wikipedia, "2006 Hajj stampede," 2018. [Online]. Available: https://en.wikipedia.org/wiki/2006_Hajj_stampede. Accessed: Apr. 20, 2018.
[39] A. Johansson, D. Helbing, H. Z. Al-Abideen, and S. Al-Bosta, "From crowd dynamics to crowd safety: A video-based analysis," *Adv. Complex Syst.*, vol. 11, no. 4, pp. 497–527, 2008.
[40] B. Greenshields *et al.*, "A study of traffic capacity," in *Highway Research Board Proceedings*, vol. 1935. Washington, DC, USA: National Research Council, Highway Research Board, 1935.
[41] M. Boltes, J. Zhang, A. Seyfried, and B. Steffen, "T-junction: Experiments, trajectory collection, and analysis," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2011, pp. 158–165.
[42] M. Munaro, F. Basso, and E. Menegatti, "Openptrack: Open source multi-camera calibration and people tracking for RGB-D camera networks," *Robot. Auton. Syst.*, vol. 75, pp. 525–538, 2016.
[43] J. Si and Y.-T. Wang, "Online learning control by association and reinforcement," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 264–276, Mar. 2001.
[44] Z. Ni, H. He, and J. Wen, "Adaptive learning in tracking control based on the dual critic network design," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 6, pp. 913–928, Jun. 2013.
[45] Z. Ni, H. He, J. Wen, and X. Xu, "Goal representation heuristic dynamic programming on maze navigation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 12, pp. 2038–2050, Dec. 2013.
[46] D. V. Prokhorov and D. C. Wunsch, "Adaptive critic designs," *IEEE Trans. Neural Netw.*, vol. 8, no. 5, pp. 997–1007, Sep. 1997.
[47] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 1057–1063.
[48] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2 ed. Cambridge, MA, USA: MIT Press, 2018.
[49] G. Ferrer, A. Garrell, and A. Sanfeliu, "Robot companion: A social-force based approach with human awareness-navigation in crowded environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 1688–1694.
[50] B. Okal and K. O. Arras, "Towards group-level social activity recognition for mobile robots," in *Proc. IROS Assistance Service Robot. Human Environ. Workshop*, 2014, pp. 1–6.
[51] S. R. L. at University of Freiburg, "ROS packages for pedSim (pedestrian simulator) based on social force model," 2017. [Online]. Available: https://github.com/srl-freiburg/pedsim_ros. Accessed: May 1, 2017.
[52] I.-B. Jeong, W.-R. Ko, G.-M. Park, D.-H. Kim, Y.-H. Yoo, and J.-H. Kim, "Task intelligence of robots: Neural model-based mechanism of thought and online motion planning," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 1, no. 1, pp. 41–50, Feb. 2017.
[53] B. Peng, J. MacGlashan, R. Loftin, M. L. Littman, D. L. Roberts, and M. E. Taylor, "Curriculum design for machine learners in sequential decision tasks," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 4, pp. 268–277, Aug. 2018.

Chao Jiang (S'13) received the B.S. degree in measuring and control technology and instrument from Chongqing University, Chongqing, China, in 2009, and the Ph.D. degree in electrical engineering from the Stevens Institute of Technology, Hoboken, NJ, USA, in 2019. He worked as a Research Assistant with the Key Laboratory of Optoelectronic Technology and Systems (Chongqing University), Ministry of Education, China, from 2009 to 2012. He will join the Department of Electrical and Computer Engineering, University of Wyoming, Laramie, WY, USA, as an Assistant Professor in Fall 2019. His current research interests include autonomous robots, human–robot interaction, robotic learning, deep reinforcement learning, and multi-robot systems. He is the recipient of the Innovation and Entrepreneurship Doctoral Fellowship (2012–2016), and the Outstanding Ph.D. Dissertation Award in Electrical Engineering (2019) at Stevens Institute of Technology.

Zhen Ni (M'15) received the B.S. degree in control science and engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2010, and the Ph.D. degree in electrical, computer and biomedical engineering from the University of Rhode Island, Kingston, RI, USA, in 2015. He will join Florida Atlantic University, Boca Raton, in Fall 2019. Before that, he worked as an Assistant Professor with the Department of Electrical Engineering and Computer Science, South Dakota State University, Brookings, SD, USA, from 2015 to 2019. His current research interests include computational intelligence, reinforcement learning, adaptive dynamic programming, and its engineering applications.

Prof. Ni is a recipient of the prestigious IEEE Computational Intelligence Society (CIS) Outstanding Ph.D. Dissertation Award (2020), the International Neural Networks Society (INNS) Aharon Katzir Young Investigator Award (2019), the URI Excellence in Doctoral Research Award (2016), and the Chinese Government Award for Outstanding Students Abroad (2014). He has been actively involved in numerous conference and workshop organization committees in the society. Since 2018, he has been an Associate Editor for the IEEE Computational Intelligence Magazine, since 2019, he has been an Associate Editor for the IEEE TRANSACTIONS OF NEURAL NETWORKS AND LEARNING SYSTEMS (2019), and a Guest Editor for *IET Cyber-Physical Systems: Theory and Applications* (2017–2018).

Yi Guo (SM'04) received the B.S. and M.S. degrees from the Xi'an University of Technology, Xi'an, China, in 1992 and 1995, respectively, and the Ph.D. degree from the University of Sydney, Sydney, NSW, Australia, in 1999, all in electrical engineering. She was a Postdoctoral Research Fellow with the Oak Ridge National Laboratory, from 2000 to 2002, and a Visiting Assistant Professor with the University of Central Florida, Orlando, FL, USA, from 2002 to 2005. In 2005, she joined the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, USA, where she is currently a Professor. Her main research interests include autonomous mobile robotics, distributed sensor networks, and nonlinear control systems. She has authored or coauhored more than 100 peer-reviewed journal and conference papers, authored the book entitled *Distributed Cooperative Control: Emerging Applications* (Wiley, 2017), and edited a book on *Selected Topics in Micro/Nano-Robotics for Biomedical Applications* (Springer, 2013). She currently serves on the editorial boards of several journals including the IEEE Robotics and Automation Magazine and IEEE/ASME Transactions on Mechatronics. She served in Organizing Committees of IEEE International Conference on Robotics and Automation (2015, 2014, 2008, and 2006).

Haibo He (SM'11–F'18) received the B.S. and M.S. degrees in electrical engineering from the Huazhong University of Science and Technology, Wuhan, China, in 1999 and 2002, respectively, and the Ph.D. degree in electrical engineering from Ohio University in 2006. He is currently the Robert Haas Endowed Chair Professor with the Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, Kingstown, RI, USA. His current research interests include computational intelligence, machine learning, data mining, and various applications. He has authored or coauthored one sole-author research book (Wiley), edited one book (Wiley-IEEE), and six conference proceedings (Springer), and authored and coauthored more than 250 peer-reviewed journal and conference papers. He was the General Chair of the IEEE Symposium Series on Computational Intelligence (SSCI 2014). He was the recipient of the IEEE International Conference on Communications Best Paper Award (2014), IEEE CIS Outstanding Early Career Award (2014), K. C. Wong Research Award, Chinese Academy of Sciences (2012), U.S. National Science Foundation CAREER Award (2011), and Providence Business News "Rising Star Innovator" Award (2011). He is currently the Editor-in-Chief of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS.