

Optimization of Merging Pedestrian Flows Based on Adaptive Dynamic Programming

Chao Jiang¹, Zhen Ni², Yi Guo¹ and Haibo He³

Abstract—Pedestrian flows in densely-populated areas may cause crowd accidents, and effective pedestrian flow regulation is highly desirable for flow optimization. In this paper, we investigate the problem of regulating two merging pedestrian flows by introducing a mobile robot moving within the flow. The pedestrian flows are regulated through dynamic human-robot interaction during their collective motion. We propose a method based on adaptive dynamic programming (ADP) to learn the optimal motion control of the robot in real time and the pedestrian outflow through the bottleneck area is maximized. Extensive simulations are performed using social force models of pedestrian motion. Simulation results show that the pedestrian outflow is significantly improved with our proposed ADP control.

I. INTRODUCTION

Modeling and control of pedestrian collective motion has received considerable research interests due to increasing demands of effective crowd regulation and evacuation guidance in public areas such as stadiums, shopping malls and train stations. Without appropriate guidance and regulation, crowd disorders, such as blocking, arise when pedestrians aggregate gradually [1], [2]. Casualties in crowd accidents have drawn considerable attention from researchers who investigate the underlying mechanism and seek for solutions to crowd safety improvement [2], [3]. In this paper, we propose a robot-assisted pedestrian flow optimization scheme using an adaptive dynamic programming (ADP)-based learning method.

Conventional approaches of pedestrian regulation primarily focused on optimal evacuation planning [1], [4], or optimal design and spatial placement of pedestrian facilities [5], [6], on the basis of the self-organization behaviors 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 improve the outflow efficiency. However, the optimal placement or geometrical parameters of static facilities vary with the changes of pedestrian flow [7], [8]. As a result, once being placed, static facilities are not easily re-configurable to adapt to real-time changes of pedestrian flows.

This work was partially supported by the US National Science Foundation under Grants CMMI-1527016, CMMI-1825709 and IIS-1838799 (Jiang and Guo), CMMI-1526835 (He) and OIA-1833005 (Ni).

¹Chao Jiang and Yi Guo are with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA Emails: cjiang6@stevens.edu, yi.guo@stevens.edu

²Zhen Ni is with the Department of Electrical Engineering and Computer Science, South Dakota State University, Brookings, SD 57007, USA Email: zhen.ni@sdstate.edu

³Haibo He is with the Department of Electrical, Computer and Biomedical Engineering, University of Rhode Island, Kingston, RI 02881, USA Email: haibohe@uri.edu

Previously, the influence of mobile robots on the motion of pedestrian flows has been studied in simulated environments [9], [10]. The results demonstrated that the pedestrian flows can be implicitly controlled through dynamic human-robot interaction (HRI) without explicit guidance. More recently, empirical studies [11], [12] were conducted to analyze the individual and collective motion behavior of pedestrians under the influence of dynamic HRI. It has been found that humans are more likely to give way to the robot to avoid collision, which can be utilized for pedestrian regulation. These works also show the applicability of utilizing HRI in replacement of static facilities. However, early works in this direction used open-loop robot control that is determined from prior knowledge of HRI characteristics under different robot motion. In our earlier work [13], we proposed passive HRI and had a mobile robot adjust its motion to regulate the pedestrian flow for desired flow velocities. The approach was verified in a uni-directional exit corridor in [13].

In this paper, we study a complex environment based on the real-world scenario presented in [2], where pedestrian flows from different directions merge together and move through a bottleneck. Compared with the uni-directional corridor scenario described in [13], the problem studied here is much more challenging as the outflow at the bottleneck is a 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. To regulate the merging pedestrian flows and achieve optimal regulation performance, we propose a robot-assisted pedestrian regulation scheme that utilizes passive HRI, and design a customized ADP learning algorithm to control the robot motion according to the pedestrian flow monitored in real time. Our proposed ADP-based robot control method is data-driven, which only takes the observation of pedestrian flows as input, and outputs optimal robot control parameters. Simulation experiments are conducted in Matlab with social force models used for pedestrian motion. The results show that significantly more people are evacuated through the bottleneck, compared with the cases of no-robot regulation and fixed robot positions or motion frequencies.

The rest of the paper is organized as follows: Section II describes the problem formulation of the merging pedestrian flows optimization with robot assistance. Section III introduces the design of our ADP-based learning algorithm for real-time robot motion control. In Section IV, we present the simulation results. The conclusion and future work are provided in Section V.

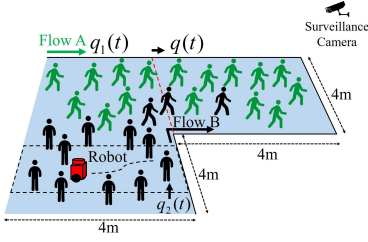


Fig. 1: The merging pedestrian flow scenario in a bottleneck area. Pedestrian outflow through the bottleneck is observed by a surveillance camera. The dashed rectangle indicates the HRI region, and the red dashed line indicates the observation line where pedestrian outflow, $q(t)$, is measured.

II. PROBLEM FORMULATION

In this section, we first introduce the merging flows scenario in a bottleneck area, and then formulate the problem of pedestrian flow regulation using HRI.

A. Environmental Setup

In this paper, the environment shown in Fig. 1 is selected for the merging flow optimization problem. The selected environment is obtained by downscaling the bottleneck area in the real-world environment studied in [2], where a crowd stampede occurred in Mina/Makkah in 2006. The size of the selected environment is 8 m by 8 m with a bottleneck width $w = 4$ m. The amount of inflow discharging into the environment is $q_1(t)$ for flow A and $q_2(t)$ for flow B, respectively. We define the HRI region as the dashed rectangle area, and the observation line as the red dashed line, where the pedestrian outflow $q(t)$ (i.e., the number of pedestrians that exit the line per second) is measured by the pre-installed surveillance camera. The left side of the observation line is the merging area of the two pedestrian flows.

A single mobile robot is deployed in the HRI region, which moves in a pre-selected trajectory to dynamically interact with the pedestrian flow. The robot's velocity is determined by a controller to be designed, which takes observed real-time pedestrian outflow as feedback. The flow B is regulated through the effect of passive HRI. The passive HRI in our particular application domain means that the robot moves in a pre-selected trajectory which is perpendicular to the pedestrian flow direction, behaving as a moving obstacle. The pedestrians adjust their paths or speed to avoid collision with the robot, and thus passively interact with the robot. Such HRI behaviors have been empirically verified in literature [11], [12]. Therefore, desired pedestrian collective motion can be achieved by optimizing the motion of the robot.

B. Robot Motion Dynamics

The robot state is defined as $\mathbf{x}_r = [x_{r1}, x_{r2}]^T \subseteq \mathbb{R}^2$, where x_{r1} and x_{r2} are the two-dimensional robot positions, respectively. To focus on the higher-level robot motion

planning problem, a single integrator is used to describe the simplified robot motion model, that is,

$$\dot{\mathbf{x}}_r = \mathbf{u}_r \quad (1)$$

where $\mathbf{u}_r = [u_{r1}, u_{r2}]^T$ denotes the vector of robot control in the directions x and y .

The robot trajectory is pre-selected to be perpendicular to the moving direction of pedestrian flow B. The selection of this trajectory is motivated by the recent simulation and empirical findings of pedestrian behavior under the influence of HRI in crossing scenarios [10]–[13]. With such interaction paths, the robot moves to create a “virtual gate” effect that controls the pedestrian traffic as the pedestrians give way to the robot or adjust their trajectories to avoid collision with the robot. The “virtual gate” can be applied in branch flows merging in an intersection so that the inflows into the merging area is regulated. We consider the flow A as the main flow, the discharge of the branch flow (i.e., flow B) into the merging area is regulated by the robot to avoid overcrowding that causes congestion at the bottleneck area (i.e., the flow merging area). The robot movement helps to maximize the pedestrian outflow through the bottleneck, and thus improves outflow efficiency.

The robot velocity is set to be zero in direction y , and set to be a sinusoid-based signal in direction x , that is,

$$\begin{bmatrix} u_{r1}(t) \\ u_{r2}(t) \end{bmatrix} = \begin{bmatrix} A_0 \Omega \sin(\Omega t) \\ 0 \end{bmatrix} \quad (2)$$

where A_0 is the parameter that determines the maximum offset of robot position from the center of channel B in direction x , Ω is the piecewise constant frequency of robot motion which can be adjusted in real time.

C. Flow Regulation Problem

To formulate the robot-assisted pedestrian regulation problem, we make the following assumptions for the environment and the robot:

Environment: A surveillance camera installed in the environment is used to observe the real-time pedestrian flow passing through the observation region. Thanks to the advanced techniques in computer vision and wireless sensor networks, real-time pedestrian crowd monitoring techniques (e.g., [14]) can be used.

Mobile Robot: The mobile robot can access the real-time flow information observed by a surveillance camera. The robot is controlled by the algorithm proposed in the paper.

We define the *merging pedestrian flow regulation problem* as minimizing the difference between the actual instantaneous pedestrian outflow $q(t)$ at the observation line and the maximum bottleneck capacity q^* during the time interval $[t_0, \infty]$. That is, the pedestrian flow regulation problem is to find a sequence of robot motion frequency $\Omega(t)$ such that

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

subject to the robot motion dynamics described in (1) and (2). Note that, since q^* is selected as the maximum bottleneck

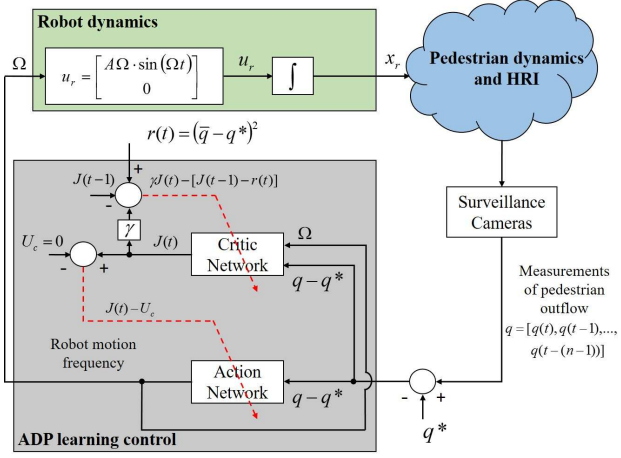


Fig. 2: The overview of the robot-assisted pedestrian regulation scheme. The ADP learning control takes the pedestrian outflow information as input and the utility $r(t)$ as feedback. The real-time robot motion frequency $\Omega(t)$ is calculated as output through a feedforward pass. The solid and dashed lines represent data flow and paths for NN parameter updating, respectively.

capacity, minimizing J indicates that the outflow through the bottleneck is maximized.

In the next section, we present the proposed ADP-based learning control algorithm that solves the optimization problem defined above.

III. ADP-BASED LEARNING CONTROL DESIGN

A. Overall Structure of Proposed Approach

The overall system diagram is depicted in Fig. 2, which is composed of robot dynamics, pedestrian dynamics and HRI, surveillance camera, and ADP learning control. The robot dynamics is defined in (1) and (2). The surveillance camera measures the pedestrian outflow in the observation region and the measured data are passed to the ADP learning control module to calculate the robot motion control. As our proposed control method is data-driven, the pedestrian dynamics and HRI are considered as a black box that is unknown to the control design (note that we use the social force model to simulate the pedestrian dynamics and HRI in our simulations, which is explained in Section IV-A). Instead, only the observed pedestrian flow information is used as input to the ADP control algorithm. Note that, unlike our earlier work [13] where the HRI region and the observation region are the same, in this paper the two regions are different without overlap (see Fig. 1). 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 propose a dedicated ADP-based learning controller using measurements of pedestrian flow as feedback. As shown in Fig. 2, the ADP control module consists of two networks, i.e., a *critic network* and an *action network*. The input of critic network includes the difference between the actual

pedestrian outflow q obtained by the surveillance camera and the desired outflow q^* , i.e. $q - q^*$, and the robot control frequency $\Omega(t)$. The output of critic network is the estimate of value function representing the total discounted future cost. The action network takes the outflow difference $q - q^*$ as input, and outputs the robot control frequency $\Omega(t)$. Note that the instantaneous observation of pedestrian outflow is very fluctuant due to the dynamic nature of pedestrian motion. Thus, we take the n most recent outflow difference in time history, i.e., $q - q^* = [q(t) - q^*, q(t-1) - q^*, \dots, q(t-(n-1)) - q^*]$ as the input of both critic and action networks to reduce the fluctuation of the instantaneous flow rate $q(t)$ for the robot control design. Both critic and action network are three-layer perceptron neural networks (NN) with one hidden layer of size N_{ch} (the number of hidden nodes) for critic network and N_{ah} for action network, respectively. The size of input layer is $n+1$ for critic network and n for action network, respectively.

B. ADP Control Design

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

$$r(t) = (\bar{q}(t) - q^*)^2 \quad (4)$$

where $\bar{q}(t)$ is the averaged n most recent measurement of $q(t)$, i.e., $\bar{q}(t) = \frac{1}{n} \sum_{k=0}^{n-1} q(t-k)$, based on the reason stated above.

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

$$R(t) = r(t) + \gamma r(t+1) + \gamma^2 r(t+2) + \dots \quad (5)$$

where γ is a discount factor for the future cost. We use a critic network in ADP design to estimate the total discounted future cost $R(t)$. Thus, the goal of this ADP module is to find a sequence of robot motion frequency $\Omega(t)$ that minimize the value function J as

$$J^*(t) = \min_{u \in \Omega} \{r(t) + \gamma J^*(t+1)\} \quad (6)$$

where the J function is formulated as the sum of discounted cost from the current time to infinite future. Introducing the discount factor $\gamma \in [0, 1]$ in the value function indicates that we are less concerned with the future cost.

According to the ADP approximation error of Bellman's equation, the error function of the critic network is defined as $e_c(t) = \gamma J(t) - [J(t-1) - r(t)]$, 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 $e_a(t) = J(t) - U_c$, where $U_c = 0$, 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, 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

Algorithm 1: ADP-Based Robot Motion Control

```
input :  $q^*$ : Desired outflow ;  
         $q(t), q(t-1), \dots, q(t-(n-1))$ : Time history of  
        pedestrian outflow measurements ;  
output: Robot control  $u_r(t)$   
1 Initialize:  $\Omega(0) = 0, A_0$  ;  
2 for  $t \leftarrow n$  to  $T_f$  do  
3   if Outflow  $q(t)$  is measured then  
4     Call ADP Learning Control module (inputs:  
        $[q(t) - q^*, q(t-1) - q^*, \dots, q(t-(n-1)) - q^*]$ );  
5     Calculate robot motion frequency  $\Omega(t)$  ;  
6   else  
7      $\Omega(t) \leftarrow \Omega(t-1)$  ;  
8   end  
9   Calculate robot control  $u_r(t)$  using (2);  
10  Robot executes control command and interacts with  
    pedestrian flow;  
11 end
```

motion frequency $\Omega(t)$ is calculated with updated weights and applied to the robot control module. The steps to update the neural network weight parameters in each network follow the same process as reported in [15]–[17].

The robot motion control algorithm is summarized in Algorithm 1. The initial robot's motion frequency $\Omega(0) = 0$ and A_0 is a design constant. When the pedestrian flow $q(t)$ is observed at time t , our ADP control algorithm is executed to calculate the robot motion frequency $\Omega(t)$, otherwise the robot motion frequency remains unchanged. Then, the robot control $u_r(t)$ at time t is calculated using (2). In general, the ADP module finishes all calculation within the sampling interval of the flow measurement because of the computational complexity in pedestrian tracking from visual observation.

IV. SIMULATION AND RESULTS

In this section, we verify and evaluate our robot-assisted pedestrian regulation method using ADP-based learning control in Matlab simulations. We first present the simulation setup and parameters, and then discuss the results of HRI characteristics obtained in open-loop pedestrian regulation simulations. Following that, we validate our ADP-based learning control algorithm for pedestrian flow optimization.

A. Simulation Setup

The simulation environment is shown in Fig. 1. The initial pedestrian speed is sampled from the Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ with mean $\mu = 2$ m/s and standard deviation $\sigma = 0.3$ m/s. The initial robot position $(x_{r1}, x_{r2}) = (0.5, 2.5)$ m, and initial robot speed is zero. The robot control parameter $A_0 = 1.5$ m.

The pedestrians' motion are simulated using the social force model and the associated parameters reported in [3] with the additional HRI term used in [13]. The HRI model is social force-based and uses parameters derived from the empirical experiments ([18]) to reproduce distinctive motion behavior of pedestrians in the presence of an interacting robot. Note that the social force model is commonly used

in robotics and physics communities to simulate pedestrian dynamics and crowd behaviors ([19], [20]). Although it may not emulate all aspects of pedestrian motion behavior, the empirical features characterizing crowd turbulence are reproduced well by the selected model as evidenced by [3]. As the focus of this paper is to use a robot to mitigate congestion and thus reduce the risk of crowd turbulence, it is adequate to use social force model for pedestrian motion simulation. Note that the social force model is only used to simulate the pedestrians' motion. Our proposed ADP control is data-driven, and does not rely on the pedestrian motion model. Our approach is applicable to different scenarios regardless the choice of pedestrian motion model and model parameters.

The sum of instantaneous inflow A and inflow B is constant, i.e., $q_1 + q_2 = 5 \text{ (m}\cdot\text{s)}^{-1}$. We change the ratio of the two inflows to create two case studies, i.e., Case 1 with inflow ratio $q_1/q_2 = 3/2$ and Case 2 with inflow ratio $q_1/q_2 = 2/3$. The two cases of inflow ratio represent different pedestrian flow conditions where inflow A is greater than inflow B and inflow A is less than inflow B, respectively. The bottleneck capacity of the environment is set as $q^* = 4 \text{ (m}\cdot\text{s)}^{-1}$, according to the simulation results of flow-density relation. The simulation time of each run is set as $T_f = 200$ s. The measurement rate of pedestrian outflow is 1 Hz, and the size of temporal history of pedestrian outflow measurement is chosen as $n = 5$.

The parameters of our ADP control is selected as follows. The number of hidden nodes in critic network and action network is set to $N_{ch} = 15$ and $N_{ah} = 12$, respectively. The learning rate $\eta_c = 0.01$ for critic network and $\eta_a = 0.01$ for action network. The discount factor $\gamma = 0.95$.

B. Open-loop Robot Control

We firstly show the simulation results of open-loop robot control, which characterize the effect of HRI on the pedestrian outflow in the bottleneck environment. We study Case 1 and Case 2 with inflow ratio $q_1/q_2 = 3/2$ and $q_1/q_2 = 2/3$, respectively. For each case, 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. We then calculate the temporal average of pedestrian outflow for each robot motion frequency Ω .

The snapshots of the open-loop simulation are shown in Fig. 3. Fig. 4 shows the average outflow vs. robot motion frequency of the two cases. One can see from the results that for each case, maximum average outflow can be obtained when the robot moves at a unique optimal motion frequency, denoted as Ω_{opt} . The optimal frequency Ω_{opt} is 0.4 rad/s and 0.7 rad/s for Case 1 and Case 2, respectively.

The open-loop robot control results demonstrate the HRI characteristics for merging flow regulation. We use the results as the "ground truth" to validate whether our ADP control algorithm can adjust the robot's motion frequency to the optimum by learning from the HRI online. In the next subsection of ADP control results, the robot motion frequencies generated by the ADP learning control will be compared with

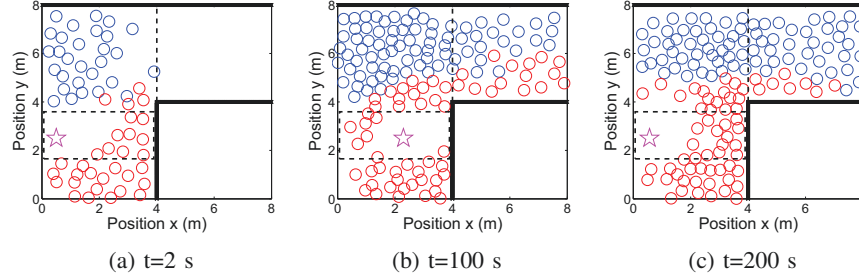


Fig. 3: Snapshots of the open-loop robot control simulation. The circles with blue and red color represent pedestrian flow A and flow B, respectively. The pink star denotes the robot. The dashed rectangle indicates the HRI region and the vertical dashed line indicates the bottleneck where outflow is measured.

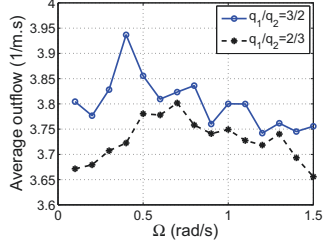


Fig. 4: The results of open-loop robot control under different ratio of pedestrian inflow.

the optimal frequencies for the two case studies presented in Fig. 4.

C. ADP Control Results

In this section, we present simulation results validating the effectiveness of our ADP control implemented in Algorithm 1 for pedestrian regulation.

1) *Case Studies of Fixed Flow:* We performed simulations of Case 1 and Case 2. From the results of HRI characteristics generated from the open-loop robot control simulation in Fig. 4, we can see that the robot should move at the frequency of 0.4 rad/s for Case 1 and 0.7 rad/s for Case 2 for maximized pedestrian outflow. The simulation results of ADP control are shown in Fig. 5 and Fig. 6 for the two cases, respectively.

For Case 1, one can see from Fig. 5a that the instantaneous outflow $q(t)$ with robot regulation (black curve) approaches the maximum bottleneck capacity q^* , while the instantaneous outflow $q(t)$ without robot regulation (red curve) is much less than q^* . Particularly, the close-up illustrates the significant decline in outflow due to crowd congestion without robot regulation, and it is prevented with robot regulation. As shown in the upper sub-figure in Fig. 5b, the robot gradually learns from the observation of pedestrian outflow, and the motion frequency converges around the optimum, $\Omega_{opt} = 0.4$ rad/s, after approximately 75 s. The red dashed line indicates the optimal values of the robot motion frequency found from the HRI characteristics shown in Fig. 4. The lower sub-figure in Fig. 5b shows the time history of robot control signal. We further compare the accumulated pedestrian outflow, $\int_0^t q(\tau) d\tau$, which is defined as the total number of pedestrians that exit the bottleneck per meter, of our ADP control with that of a) no-robot, b) randomly-chosen fixed robot position (3, 3.5) m, c) randomly-chosen fixed motion

frequency 1.2 rad/s. The results are plotted in Fig. 5c. We can see that at time $t = 200$ s, the accumulated outflow of the proposed ADP control is 741 people per meter, while the accumulated outflow without robot regulation is 656 people per meter. The accumulated outflow at $t = 200$ s is improved by 12.9% with our ADP control, compared with the no-robot case. The accumulated outflow of fixed robot position and motion frequency is 649 and 710 people per meter, respectively.

Similarly for Case 2, Fig. 6a shows the time history of instantaneous outflow $q(t)$ with and without robot regulation. As shown in the upper sub-figure of Fig. 6b, the robot motion frequency is gradually learned from the observation of pedestrian outflow, and converges around the optimum, $\Omega_{opt} = 0.7$ rad/s, after approximately 80 s. The lower sub-figure in Fig. 6b shows the time history of robot control signal. Fig. 6c shows the accumulated outflow, $\int_0^t q(\tau) d\tau$, of the proposed ADP control, compared with that of a) no-robot, b) randomly-chosen fixed robot position (3, 3.5) m, c) randomly-chosen fixed motion frequency 1.2 rad/s. One can see that the accumulated outflow of the proposed ADP control is 730 people per meter, while the accumulated outflow without robot regulation is 655 people per meter. The accumulated outflow at $t = 200$ s is improved by 11.5% with the ADP control, compared to the no-robot case. The accumulated outflow of fixed robot position and motion frequency is 689 and 706 people per meter, respectively.

We can see from the simulation results that our ADP control achieves the best regulation results of maximizing the number of people going through the bottleneck area by observing the pedestrian outflow. Moreover, it is worth noting that the performance of using a fixed robot position or motion frequency varies in different cases of inflow ratios. Thus, it is not possible to randomly select a fixed robot position or motion frequency for optimal regulation performance without the prior knowledge of pedestrian flow conditions. On the contrary, our ADP control can learn the optimal robot motion only from the real-time observation of pedestrian outflow, particularly when the pedestrian flow conditions change over time. Next, we show the online learning capability of our ADP control in changing flow case.

2) *Case Study of Changing Flow:* We further present the simulation results of changing pedestrian flow case to

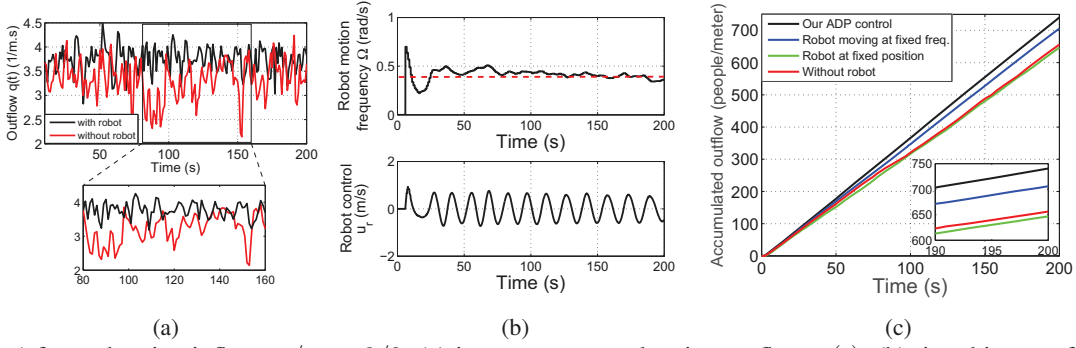


Fig. 5: Case 1 for pedestrian inflow $q_1/q_2 = 3/2$: (a) instantaneous pedestrian outflow, $q(t)$; (b) time history of robot motion frequency and robot control; (c) accumulated pedestrian outflow, $\int_0^t q(\tau)d\tau$, with different regulation approaches.

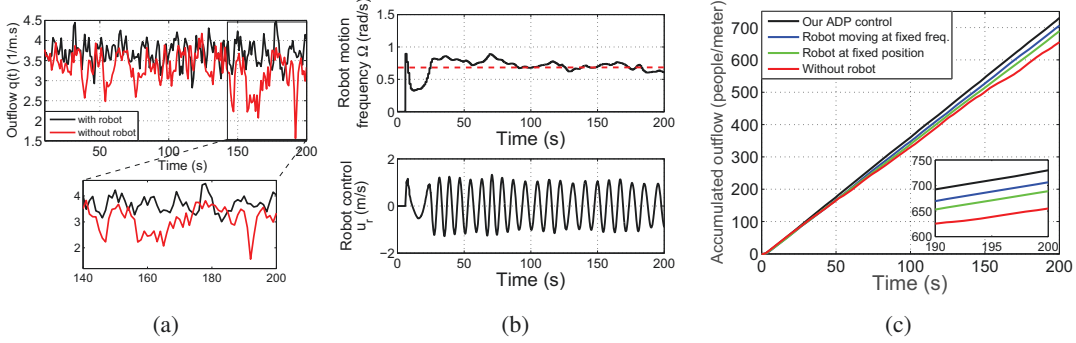


Fig. 6: Case 2 for pedestrian inflow $q_1/q_2 = 2/3$: (a) instantaneous pedestrian outflow, $q(t)$; (b) time history of robot motion frequency and robot control; (c) accumulated pedestrian outflow, $\int_0^t q(\tau)d\tau$, with different regulation approaches.

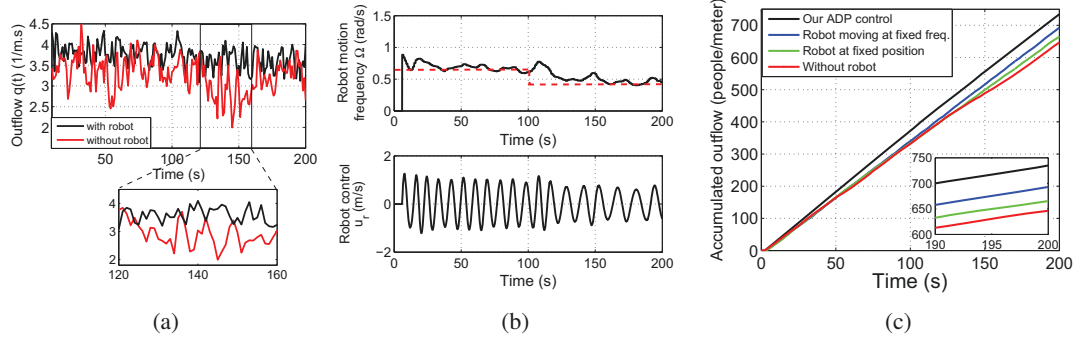


Fig. 7: Case 3 for changing pedestrian inflow, where pedestrian inflow ratio changes from $q_1/q_2 = 2/3$ to $q_1/q_2 = 3/2$ at $t = 100$ s: (a) instantaneous pedestrian outflow, $q(t)$; (b) time history of robot motion frequency and robot control; (c) accumulated pedestrian outflow, $\int_0^t q(\tau)d\tau$, with different regulation approaches.

show the online learning capability of our ADP control. We create Case 3, where the inflow ratio changes from the initial value $q_1/q_2 = 2/3$ to $q_1/q_2 = 3/2$ at $t = 100$ s. The robot needs to adaptively change its motion frequency to maximize the outflow. The simulation results of the changing flow case are shown in Fig. 7. It can be seen from Fig. 7a that compared to no-robot regulation, our ADP control achieves higher instantaneous outflow $q(t)$ in spite of the change of inflows at $t = 100$ s. Fig. 7b shows that the robot motion frequency firstly converges around the optimum $\Omega_{opt1} = 0.7$ rad/s for inflow ratio $q_1/q_2 = 2/3$, and then converges around $\Omega_{opt2} = 0.4$ rad/s after the inflow ratio changes to $q_1/q_2 = 3/2$. Fig. 7c shows the accumulated outflow, $\int_0^t q(\tau)d\tau$, of our ADP control in comparison with a) no-robot, b) randomly-chosen fixed robot position (3, 3.5)

m, c) randomly-chosen fixed motion frequency 1.2 rad/s. One can see that the accumulated outflow with our ADP control is 735 people per meter, while the accumulated outflow without robot regulation is 646 people per meter. The accumulated outflow at $t = 200$ s is improved by 13.8% with the ADP control, compared to the no-robot case. The accumulated outflow of fixed robot position and motion frequency is 662 and 687 people per meter, respectively. Thus, our ADP control is adaptive to the change of pedestrian flow and adjusts the optimal motion accordingly.

D. Statistical Results

In this subsection, we provide statistical results of the ADP control performance. In our simulation, the pedestrians' initial positions and velocities are randomly assigned from Gaussian distributions and the ADP weights are randomly

TABLE I: Statistical results over 50 runs.

Case No.	Case 1	Case 2
Success rate (%)	80	78

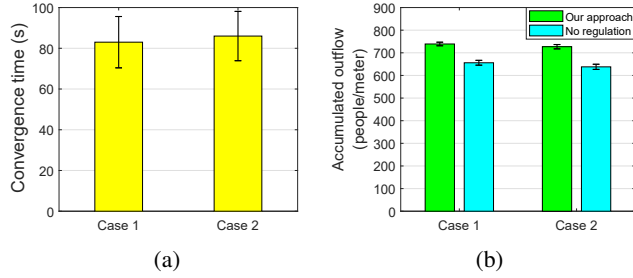


Fig. 8: Statistical results of Case 1 and Case 2 over 50 runs: (a) convergence time; (b) accumulated outflow. The error bars indicate the standard deviation.

initialized between $[-0.5, 0.5]$. We conducted 50 runs for each of Case 1 and Case 2 to obtain the statistical results. The duration of each simulation run is set to be 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 $t \in (150, 200]$. As presented in Table I, the success rate for Case 1 and Case 2 is 80% and 78%, respectively. The statistical results are shown in Fig. 8. One can see from Fig. 8a that the average convergence time is 83 s with a standard deviation of 12.6 for Case 1, and 86 s for Case 2 with a standard deviation of 12.1 s. Fig. 8b shows 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 and 9.58 for Case 1 and Case 2, respectively. In comparison, the average accumulated outflow without regulation is 656 with a standard deviation of 10.6 for Case 1, and 638 with a standard deviation of 11.4 for Case 2. It can be concluded from the results that the proposed ADP-based learning method is effective with acceptable success rate. Our ADP control can adaptively learn the optimal robot motion to regulate pedestrian flows without the prior knowledge of HRI characteristics.

V. CONCLUSION AND FUTURE WORK

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 pedestrian flows, and designed an ADP-based learning method for robot motion control. The pedestrian regulation problem was formulated as an optimal control problem and a novel ADP control algorithm was developed to solve the optimal control that adjusts robot motion parameter in real time. Simulation results demonstrated that our approach evacuates significantly more people through the bottleneck, compared with the case without robot regulation. In the future work, we will consider other ways to deploy the

robot for pedestrian regulation, and study socially-compliant aspects and multi-robot cooperation.

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 Transactions on Automation Science and Engineering*, vol. 9, no. 4, pp. 687–700, 2012.
- [2] D. Helbing, A. Johansson, and H. Z. Al-Abideen, "Dynamics of crowd disasters: An empirical study," *Physical review E*, vol. 75, no. 4, p. 046109, 2007.
- [3] W. Yu and A. Johansson, "Modeling crowd turbulence by many-particle simulations," *Physical Review E*, vol. 76, no. 4, p. 046105, 2007.
- [4] B. Tang, C. Jiang, H. He, and Y. Guo, "Human mobility modeling for robot-assisted evacuation in complex indoor environments," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 5, pp. 694–707, 2016.
- [5] D. Helbing, L. Buzna, A. Johansson, and T. Werner, "Self-organized pedestrian crowd dynamics: Experiments, simulations, and design solutions," *Transportation Science*, vol. 39, no. 1, pp. 1–24, 2005.
- [6] G. Frank and C. Dorso, "Room evacuation in the presence of an obstacle," *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 11, pp. 2135–2145, 2011.
- [7] B. Haworth, M. Usman, G. Berseth, M. Khayatkhoei, M. Kapadia, and P. Faloutsos, "Using synthetic crowds to inform building pillar placements," in *IEEE Virtual Humans and Crowds for Immersive Environments*, pp. 7–11, 2016.
- [8] Y. Zhao, M. Li, X. Lu, L. Tian, Z. Yu, K. Huang, Y. Wang, and T. Li, "Optimal layout design of obstacles for panic evacuation using differential evolution," *Physica A: Statistical Mechanics and its Applications*, vol. 465, pp. 175–194, 2017.
- [9] J. A. Kirkland and A. A. Maciejewski, "A simulation of attempts to influence crowd dynamics," in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 5, pp. 4328–4333, 2003.
- [10] K. Yamamoto and M. Okada, "Control of swarm behavior in crossing pedestrians based on temporal/spatial frequencies," *Robotics and Autonomous Systems*, vol. 61, no. 9, pp. 1036–1048, 2013.
- [11] C. Vassallo, A.-H. Olivier, P. Souères, A. Crétual, O. Stasse, and J. Pettré, "How do walkers avoid a mobile robot crossing their way?," *Gait & Posture*, vol. 51, pp. 97–103, 2017.
- [12] Z. Chen, C. Jiang, and Y. Guo, "Pedestrian-robot interaction experiments in an exit corridor," in *International Conference on Ubiquitous Robots*, pp. 29–34, 2018.
- [13] C. Jiang, Z. Ni, Y. Guo, and H. He, "Learning human-robot interaction for robot-assisted pedestrian flow optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017, doi: 10.1109/TSMC.2017.2725300.
- [14] M. Munaro, F. Basso, and E. Menegatti, "Opentrack: Open source multi-camera calibration and people tracking for RGB-D camera networks," *Robotics and Autonomous Systems*, vol. 75, pp. 525–538, 2016.
- [15] J. Si and Y.-T. Wang, "Online learning control by association and reinforcement," *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 264–276, 2001.
- [16] H. He, Z. Ni, and J. Fu, "A three-network architecture for on-line learning and optimization based on adaptive dynamic programming," *Neurocomputing*, vol. 78, no. 1, pp. 3–13, 2012.
- [17] Z. Ni, H. He, J. Wen, and X. Xu, "Goal representation heuristic dynamic programming on maze navigation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 12, pp. 2038–2050, 2013.
- [18] G. Ferrer, A. Garrell, and A. Sanfeliu, "Robot companion: A social-force based approach with human awareness-navigation in crowded environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1688–1694, 2013.
- [19] Y. Tamura, T. Fukuzawa, and H. Asama, "Smooth collision avoidance in human-robot coexisting environment," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3887–3892, 2010.
- [20] H. Kidokoro, T. Kanda, D. Bršćić, and M. Shiomi, "Simulation-based behavior planning to prevent congestion of pedestrians around a robot," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1419–1431, 2015.