BVIP Guiding System with Adaptability to Individual Differences

Yibo Zhou†   Dongfei Cui†   Xiangming Dong†   Zongkai Wu*   Zhenyu Wei   Donglin Wang*

Abstract— Guiding robots can not only detect close-range obstacles like other guiding tools, but also extend its range to perceive the environment when making decisions. However, most existing works over-simplified the interaction between human agents and robots, ignoring the differences between individuals, resulting in poor experiences for different users. To solve the problem, we propose a data-driven guiding system to cope with the effect brighten by individual differences. In our guiding system, we design a Human Motion Predictor (HMP) and a Robot Dynamics Model (RDM) based on deep neural network, the time convolutional network (TCN) is verified to have the best performance, to predict differences in interaction between different human agents and robots. To train our models, we collected datasets that records the interactions from different human agents. Moreover, given the predictive information of the specific user, we propose a waypoints selector that allows the robot to naturally adapt to the user's state changes, which are mainly reflected in the walking speed. We compare the performance of our models with previous works and achieve significant performance improvements. On this basis, our guiding system demonstrated good adaptability to different human agents. Our guiding system is deployed on a real quadruped robot to verify the practicability.

I. INTRODUCTION

As for blind and visually impaired people (BVIP), traveling is necessary but difficult. There are many studies dedicated to solving the task of guiding BVIP. The solutions include intelligent white cane [1], intelligent wearables [2] and etc. Among these solutions, guiding robots can not only detect close-range obstacles like other guiding tools, but also extend its range to perceive the environment when making decisions, which contribute to its outstanding performance. Besides, with the excellent maneuverability and terrain adaptability [3], [4], quadruped robots have become an excellent chassis for guiding robots.

However, individual differences among human agents can lead to different guiding interactions. For instance, a tired person may walk slower than an energetic person when being guided with same tension(see Fig. 1). Previous researches lacked the consideration of the individual differences, resulting in stiff guiding performance. To eliminate individual differences, human agents were asked to walk slowly at the same speed. More concretely, the lack of considering

Fig. 1: Different human agents are guided by our quadruped BVIP guiding robot. Each of them has their own comfortable walking speed, and our guiding system can flexibly adapt their differences.

individual differences leads to 2 problems: 1) the motions of both the robot and the human agent after being towed are predicted inaccurately; 2) the robot's velocity cannot adapt to different human agents' velocities.

To address the problems, we propose a data-driven BVIP guiding system to cope with the effect brighten by individual differences. In our guiding system, we first use the position of the robot and camera positioning technology to locate the position of the human, and then plan the human route.

Precisely predicting different human agents' motions is the basis of motion planning in BVIP guiding task. Our experiment proved that human's velocity is related to the received tensions in past periods. Thus, we propose a Human Motion Predictor (HMP) that based on well-known neural architectures, which can precisely predict human agents' motions according to their received tensions. To train the model, we collected a dataset which records the motion data of different humans after being towed.

Next, to adapt different human agents' comfortable speeds, we propose a Waypoints Selector. The selector chooses waypoints on the planned global path as the target positions for human agents. The faster the human agent walks, the greater the spacing between the selected waypoints.

Model Predictive Control (MPC) is then used to plan the leash tensions and robot's velocity which are required to make human agents follow the selected waypoints. Besides predicting the motions of human agents, we also establish a Robot Dynamics Model (RDM) to describe robot's motion responds after being towed. A dateset that record robot's responds after receiving different tensions from each directions is collected to train the model. Our neural based models are necessary to the MPC-based motion planner.

In experiments, we compared our models with previous

Fig. 2: Overview of our BVIP guiding system.

researches. It can be concluded that our models achieved significant performance improvements. On this basis, we deployed our BVIP guiding system on a quadruped robot. The synchronization between the robot and different human agents is tested. It is demonstrated that our BVIP guiding system has good adaptability to individual differences. our contributions are summarized as follows,

- Considering the individual differences among human agents can lead to different guiding interactions. we propose the data-driven guiding system for BVIP. In our system, In our guiding system, we design a AI-based HMP and RDM to predict differences in interaction between different human agents and robots.
- We propose a waypoint selector that works with motion predictors and MPC to guide different people at a comfortable pace.
- On the experimental side, we first validate the performance improvements between our predictor and baselines, while analyzing the differences between three well-known architectures in our task, resulting in the factual results of TCN optimality. Secondly, we also applied our system to real robots to verify its practicability.

II. RELATED WORKS

A. Human Motion Prediction

The mainstream human motion predicting algorithms now is coupled approaches. They focused on modeling multi-agent interactions, which solves many problems shown by decoupled approaches, such as reciprocal dance problem [5]. Nanavati et al. [6] proposed a coupled human motion predicting model for BVIP guiding task, which was based on Markov Model. However, they ignored the impact of the tension received by human agents. They also left the individual differences among human agents out of consideration. Although the model of Chen et al. [7] considered the problems above, they over-simplified the model into linear relation, which had no time-ordered. Further more, both two models were hand-made, which leads to inaccurate performance.

Herath et al. [8] proposed a model which was based on well-known neural architectures, which predict human's position from IMU data. Their work is inspiring to our human motion predicting task.

B. Robot Dynamics Model Establishment

Modeling specific platforms accurately is usually complicated when using model-based controlling methods. Chen et al. [7] over-simplified the effect of tension on the velocity of quadruped robots into a velocity discount coefficient matrix, assuming that only the force at the current moment would cause a loss in the robot's velocity. However, the model is highly inaccurate.

We are inspired by previous works in other domains. Gillespie et al. [9] used deep neural networks to learn the nonlinear dynamics models of a soft robot, which is difficult to manually model as quadruped robots. Spielberg et al. [10] applied neural networks to modeling unknown friction in automated driving task.

C. Global Path Correction

Traditional grid-based global path planning algorithms generate paths consisting of adjacent grid points with equal distances between them, such as A* algorithm and Dijkstra algorithm. Song et al.[9] proposed an algorithm called smoothed A*, which eliminates the redundant points arise in the traditional A* algorithm. They also used interpolation to convert the discrete path into an analytical path. Our algorithm is further improved upon this foundation.

III. METHODOLOGY

A. System Framework

As shown in Fig.2, our BVIP guiding system concludes 4 subsystems: Mapping and Localization System, Global Path Planning System, Motion Planning System, and Actuator Controlling System.

Firstly, Mapping and Localization System perceives the environment. The map and the robot's position are obtained by the SLAM algorithm called Cartographer [11]. Human agent's position is obtained by the camera positioning the AprilTags [12] on the torso.

Next, Global Path Planning System links the starting point and the final target. We use the Dijkstra algorithm to plan a global path, and smooth it with a series of smoothers

TABLE I: NOMENCLATURE

Symbol	Description	Dimension
θ	yaw angle	scalar
$\mathbf{x}$	position	$\mathbb{R}^{3\times1}$
$\mathbf{v}$	velocity	$\mathbb{R}^{3\times1}$
$\mathbf{F}$	tension	$\mathbb{R}^{3\times1}$
W	sequence length of input	scalar
N	prediction horizon of MPC controller	scalar
T	the period of the controller	scalar
$\square^h$	human's state parameter	-
$\square^r$	robot's state parameter	-
$\square^*$	expectation	-
$\tilde{\square}$	prediction	-
$\square_t$	state parameter at time t	-

Fig. 3: Configuration of the guiding system. The human agent is guided by the robot through a leash.

introduced by Song et al. [13] We then select waypoints as human's target points using the waypoints selector we proposed. When selecting waypoints, HMP is used to predict human's future speed, and our selected waypoints match the speed trend.

Motion Planning System concludes 2 parts, human motion planner and robot motion planner. Human motion planner is used to plan the tension and direction required for human to follow the planned path, while robot motion planner is used to plan the walking speed and direction required to achieve this tension. Both two planers use model predictive control algorithm. In human motion planner, we use HMP to predict the optimal robot target position and leash motor target tension. To enable the robot to reach these target positions after being pulled, RDM predicts the optimal expected velocity for the robot controller in robot motion planner.

Finally, to ensure that the robot reaches the expected velocity, the Quadruped Robot Controller controls the torque of the quadruped robot's joint motors. The Leash Motor Controller is responsible for ensuring that the leash reaches the target tension by controlling the motor input current.

B. Human Motion Predictor

Precisely predicting different human agents' motions is essential. Du Toit and Burdick pointed out that the lack of explicit human motion prediction results in uncertainty explosion [14]. This phenomenon may yield the freezing robot problem described by Trautman et al. [15]

Human agent's motion is influenced by the guidance from the guiding robot. We propose to use deep neural networks to regress the sequential relationship between human's velocity and the received tension in the past. In the following paragraph, we will introduce the data collection method, the backbone architectures of our deep neural networks, and the training method of our model.

1) Data Collection: Before collecting the data, we mapped the experimental environment and pre-set several paths. We invited 10 volunteers to hold the leash which was attached to the guide robot, who guided them from the starting point to the target. All volunteers were asked to wear blindfolds, earmuffs and our positioning vest. When collecting data, we used a four wheeled robot as its movement is

smoother than quadruped robots, resulting in less oscillation and higher data quality.

Our sampling strategy was as follows: (a). The experiment assistant remotely controlled the robot to follow the planned path from the starting point to the target, and adjusted the speed to maintain a certain distance from different human agents. (b). The tension of the leash motor was randomly changed every 10 seconds, ranging from 2N to 20N. (c). The sampling frequency was set to be 50 Hz. (d). The collected data included the position of the robot $\mathbf{x}^r$, the position of the detected AprilTags, the magnitude of the tension F, and the direction of the leash $\mathbf{e}_l$.

After collecting the data, we further processed them. We obtained the relative distance of the human agent and the robot l by averaging the positions of the detected AprilTags. Human agent's positions were calculated by adding the relative positions to the robot's positions:

$$\mathbf{x}^h = \mathbf{x}^r + l\mathbf{e}_l \tag{1}$$

The sampled tension magnitudes was decomposed into orthogonal components:

$$\mathbf{F}^h = -F\mathbf{e}_l = [F_x^h, F_y^h, F_z^h]^\mathsf{T} \tag{2}$$

By taking the derivatives of human agent's positions, we obtained human agent's velocities $\mathbf{v}^h$. (d). The data was smoothed and filtered to acquire better quality. (e). The data was segmented into the form:

$$(\mathbf{v}_{k-W^h+1:k}^h, \mathbf{F}_{k-W^h+1:k}^h | \mathbf{v}_{k+1}^h) \tag{3}$$

, which means human agent's velocities and the received tensions in past W^h timesteps were set to be the input of HMP, and the velocity of next timestep were set to be the label.

2) Backbone Architecture: We use 3 well-known architecture varients for the sequence predicting task: CNN, LSTM and TCN.

HMP CNN: We used the standard convolutional network [16] to extract features.It contains 2 convolutional layers. The output channel of the convolutional layers are respectively 8 and 16, and the kernel size of each convolutional layer is 3. A fully connected layer is added at the end to regress the extracted features into a velocity vector.

Fig. 4: Overview of our Global Path Planning System. (a) shows the discrete path based on grid map planned by Dijkstra. (b) is a schematic diagram of LOPS and WRPS removing redundant points. (c) shows the interpolation. (d) is a schematic diagram of our Waypoints Selector selecting waypoints on the analytical path.

HMP LSTM: We use a stacked unidirectional LSTM architecture [17]. It contains 3 layers with 16 units each. A fully connected layer is also added at the end.

HMP TCN: Our HMP-TCN shares the same architecture with RoNIN-TCN [8] while the dimension of the input and the ouput is revised.

3) Training Method: In order to regress the predicted velocity to the real velocity we collected, we designed a loss function:

$$\widetilde{\mathbf{v}}_k^h = f_{\mathbf{HMP}}(\mathbf{v}_{k-W^h:k-1}^h, \mathbf{F}_{k-W^h:k-1}^h) \tag{4}$$

$$\widetilde{\mathbf{x}}_{k+1}^h = \widetilde{\mathbf{x}}_k^h + \widetilde{\mathbf{v}}_k^h T \tag{5}$$

$$loss = L_2(\widetilde{\mathbf{x}}^h, \mathbf{x}^h) \tag{6}$$

Moreover, we use the optimizer called Adam [18] to optimize the model parameters.

C. Robot Dynamics Model

Model-based control can generally result in superior performance. However, it is difficult to precisely describe the relationship between the quadruped robot's velocity and the tension it received. On the one hand, Quadruped robot has 12 DoF, which makes the dynamics model become complicated to build manually. On the other hand, quadruped robot has complicated motion controllers, which makes the impact of the tensions which were received in past periods are integrated, to influence its velocity. To accurately establish the model, we use deep neural networks to regress the relationship rather than designing the model manually.

1) Data Collection: Our sampling strategy was as follows: (a). Set the quadruped robot to walk at random constant speeds. (b). Applying forces of varying magnitudes, durations and directions to the quadruped robot using our force controlling device. (c). The sampling frequency was set to be 50 Hz. (d). The collected data includes the velocity of the robot $\mathbf{v}^r$, the control input of the Quadruped Robot Controller $\mathbf{u}^r$, the magnitude of the tension F, and the direction of the leash $\mathbf{e}_l$.

After the collection, we decomposed the tension into orthogonal components:

$$\mathbf{F}^r = F\mathbf{e}_l = [F_x^r, F_y^r, F_z^r]^\mathsf{T} \tag{7}$$

We also filtered the data, and segment the data into the form:

$$(\mathbf{v}_{k-W^h+1:k}^r, \mathbf{u}_{k-W^h+1:k}^r, \mathbf{F}_{k-W^h+1:k}^r | \mathbf{v}_{k+1}^r) \tag{8}$$

Our Robot Dynamics Model shares the 3 well-known architecture variants with our Human Motion Predictor, while the dimension of the input is revised.

D. Waypoints Selector

The function of Waypoints Selector is to select certain waypoints from the path generated by Dijkstra as human's target positions in next timesteps. The selected waypoints should be correlated with the speed trend of different human agents: the faster the speed of the human agent, the greater the distance between the waypoints should be.

Before selecting the waypoints, we need to transform the discrete path based on the grid map into a smooth analytical path. We used two smoothers called LOPS and WRPS which were proposed by Song et al. [13] to remove redundant points. Then, we used the spline Interpolation Path Smoother (IPS) to transform the remained path points into an analytical path.

To predict human agent's speed in next timesteps, our Neural-Based Human State Predictor is used:

$$\|\widetilde{\mathbf{v}}_k^h\| = \|f_{\mathbf{HMP}}(\mathbf{v}_{k-W^h:k-1}^h, \mathbf{F}_{k-W^h:k-1}^h)\| \tag{9}$$

We assumed that the tension remains unchanged in the future when predicting human's speed.

Finally, we use our waypoints selector to select waypoints on the smoothed analytical path as human agent's expected positions in next periods. The distances between the selected waypoints satisfy:

$$\|\mathbf{x}_{k+1}^{h*} - \mathbf{x}_k^{h*}\| = \|\widetilde{\mathbf{v}}_k^h\| T \tag{10}$$

E. Motion Planning System

1) Human Motion Planner: The function of Human Motion Planner is to use optimization algorithms to solve for the best robot position and leash tension, in order to maximize the probability of human agents reaching the expected position planned by the Waypoints Selector. MPC algorithm consists of three parts: a predictive model, feedback correction, and optimization in range of a temporal window.

The model is already explained in section III.(B). We use deep neural network to predict the relationship of human agent's motion and the received tension according to formula 4. While predicting human agent's velocity in next M^h timesteps, the prediction of the tension $\widetilde{\mathbf{F}}$ was set to be the input when $k > t$ rather than the feedback $\mathbf{F}$.

Then, human agent's position can be predicted according to formula 5, and robot's target position can be predicted according to:

$$\widetilde{\mathbf{x}}^{r*} = \widetilde{\mathbf{x}}^h + l\mathbf{e}_\theta \tag{11}$$

Next, to acquire the optimal control variables, we minimized the cost function:

$$J^h(\widetilde{\mathbf{F}}^h, \widetilde{l}, \widetilde{\theta}) = \sum_{k=0}^{M^h-1} (\|\widetilde{\mathbf{x}}_k^h - \mathbf{x}_k^{h*}\|_{\omega_1} \tag{12a}$$

$$+ \|\widetilde{\mathbf{F}}_{k+1} - \widetilde{\mathbf{F}}_k\|_{\omega_2} + \omega_3(\|\widetilde{\mathbf{F}}_{k+1}\| - \|\widetilde{\mathbf{F}}_k\|)^2 \tag{12b}$$

$$+ \omega_4(1 - \cos(\widetilde{\theta}_{k+1} - \widetilde{\theta}_k)) + \omega_5(\widetilde{l}_{k+1} - \widetilde{l}_k)^2) \tag{12c}$$

$$F_{min} < \|\widetilde{\mathbf{F}}_k\| < F_{max} \tag{12d}$$

$$l_{min} < \widetilde{l}_k < l_{max} \tag{12e}$$

$$\left\langle \widetilde{\mathbf{F}}_k, \widetilde{\mathbf{F}}_{k+1} \right\rangle \leq \varphi_{\mathbf{F}} \tag{12f}$$

$$\left\langle \mathbf{e}_{\widetilde{\mathbf{F}}_k}, \mathbf{e}_{\widetilde{\theta}_k} \right\rangle \leq \varphi_\theta \tag{12g}$$

$$\|\widetilde{\mathbf{x}}_k^h - \mathbf{x}_j^{obs}\| \geq r^{obs} \tag{12h}$$

$$\|\widetilde{\mathbf{x}}_k^{r*} - \mathbf{x}_j^{obs}\| \geq r^{obs} \tag{12i}$$

where $\|\mathbf{x}\|_\omega = \frac{1}{2}\omega\mathbf{x}^\mathsf{T}\mathbf{x}$. $\omega_1, \omega_2 \in \mathbb{R}^{3\times3}, \omega_3, \omega_4, \omega_5 \in \mathbb{R}$ are weight coefficients, $F_{min}, F_{max}, l_{min}, l_{max}, \varphi_{\mathbf{F}}, \varphi_\theta, r^{obs}$ are the the thresholds.

As for the cost function, minimizing $\sum_{k=0}^{M^h-1} \|\widetilde{\mathbf{x}}_k^h - \mathbf{x}_k^{h*}\|$ promises the predicted trajectory $\widetilde{\mathbf{x}}^h$ to approach the planned waypoints $\mathbf{x}^{h*}$; minimizing $\sum_{k=0}^{M^h-1} \|\widetilde{\mathbf{F}}_{k+1} - \widetilde{\mathbf{F}}_k\|$ and $\sum_{k=0}^{M^h-1}(\|\widetilde{\mathbf{F}}_{k+1}\| - \|\widetilde{\mathbf{F}}_k\|)$ promises the direction and magnitude of the force to change smoothly; and minimizing $\sum_{k=0}^{M^h-1}(1 - \cos(\widetilde{\theta}_{k+1} - \widetilde{\theta}_k))$ and $\sum_{k=0}^{M^h-1}(\widetilde{l}_{k+1} - \widetilde{l}_k)^2$ promises the direction and magnitude of the leash to change smoothly.

Finally, through the optimization in range of a temporal window, we obtained the optimal rope motor force $\mathbf{F}^*$, optimal relative distance l^*, and optimal relative yaw angle θ^*. Robot's target positions can be calculated according to:

$$\mathbf{x}^{r*} = \mathbf{x}^{h*} + l^*\mathbf{e}_{\theta^*} \tag{13}$$

2) Robot Motion Planner: The function of Robot Motion Planner is to plan the best control input for the Quadruped Robot Controller, so that the robot can reach the expected position planned by the Human Motion Planner.

The model is already explained in section III.(C). The model described robot's velocity after being towed:

$$\widetilde{\mathbf{v}}_k^r = f_{\mathbf{RDM}}(\mathbf{v}_{k-W^h:k-1}^r, \mathbf{u}_{k-W^h:k-1}^r, \mathbf{F}_{k-W^h:k-1}^r) \tag{14}$$

While predicting robot's velocity in next M^r timesteps, the prediction of the control input $\widetilde{\mathbf{u}}^r$ was set to be the input when $k > t$ rather than the historic value $\mathbf{u}^r$. Robot's position can be predicted according to:

$$\widetilde{\mathbf{x}}_{k+1}^r = \widetilde{\mathbf{x}}_k^r + \mathbf{v}_k^r T \tag{15}$$

Next, to acquire the optimal control variables, we minimized the cost function:

$$J^r(\widetilde{\mathbf{u}}^r) = \sum_{k=0}^{M^r-1} (\|\widetilde{\mathbf{x}}_k^r - \mathbf{x}_k^{r*}\|_{\omega_6} + \|\widetilde{\mathbf{u}}_k^r\|_{\omega_7}) \tag{16a}$$

$$\|\widetilde{\mathbf{u}}_x^r\|_\infty \leq u_{xmax}^r \tag{16b}$$

$$\|\widetilde{\mathbf{u}}_y^r\|_\infty \leq u_{ymax}^r \tag{16c}$$

$$\|\widetilde{\mathbf{u}}_\omega^r\|_\infty \leq u_{\omega max}^r \tag{16d}$$

where $\omega_6, \omega_7 \in \mathbb{R}^{3\times3}$ are weight coefficients, $u_{xmax}^r, u_{ymax}^r, u_{\omega max}^r$ are the the thresholds.

As for the cost function, minimizing $\sum_{k=0}^{M^r-1} \|\widetilde{\mathbf{x}}_k^r - \mathbf{x}_k^{r*}\|$ promises robot's predicted positions $\widetilde{\mathbf{x}}^r$ to approach the planned target positions $\mathbf{x}^{r*}$; and minimizing $\sum_{k=0}^{M^r-1} \|\widetilde{\mathbf{u}}_k^r\|$ promises the robot to walk efficiently.

Finally, through the optimization in range of a temporal window, we obtained the optimal input $\mathbf{u}^{r*}$ for the Quadruped Robot Controller.

IV. EXPERIMENTS

A. Experimental Settings

Our hardware platform is shown in Fig.5. Our sensors include a LiDAR for SLAM, a RGB camera and encoder for human localization, a force sensor for measuring leash tension, and an IMU for measuring the euler angle of the leash. The actuators include a quadruped robot and a leash motor for traction. The leash motor guides the human agent through the leash.

Robot Operating System(ROS) [19] is used for communication.

Fig. 5: Our hardware platform consists of a quadruped robot, different sensors for perceiving the environment, and actuators to execute the guiding task.

B. Results

1) Human Motion Prediction: To test the performance of our Human Motion Predictor, we compared our HMP model with the linear model proposed by Chen et al. [7] and the GeoC model proposed by Nanavati et al. [6].

As shown in TABLE II, we compare the performance of different models on our dataset using K-fold Cross-Validation. We calculate the error and standard deviation between the predicted human velocity and the ground truth velocity for each model. The result shows that our models has smaller error and standard deviation, indicating that our models is closer to the ground truth velocity and have a more stable performance. Thanks to its more advanced architecture, our HMP-TCN model performed the best among the three variants. It can be concluded that our model achieved a significant performance improvement.

TABLE II: The Comparison of Human Motion Predicting Models

Model Name		Joint Error, e	Velocity Error	
			e_x	e_y
Linear		6.153 (0.683)	7.389 (0.592)	4.917 (0.693)
GeoC		0.354 (0.074)	0.378 (0.046)	0.330 (0.057)
HMP	CNN	0.108 (0.295)	0.126 (0.261)	0.090 (0.328)
	LSTM	4.669 (0.204)	7.315 (0.125)	2.023 (0.344)
	TCN	**0.036 (0.069)**	**0.051 (0.053)**	**0.022 (0.056)**

Values are: Avg (StdDev). **Bold Font** represents best performance.

TABLE III: The Comparison of Robot Dynamics Models

Model Name		Joint Error, e	Velocity Error	
			e_x	e_y
VDCM		7.152 (0.692)	8.169 (0.398)	6.135 (0.756)
RDM	CNN	3.546 (0.264)	4.886 (0.291)	2.206 (0.276)
	LSTM	0.258 (0.127)	0.337 (0.143)	0.178 (0.111)
	TCN	**0.042 (0.059)**	**0.069 (0.061)**	**0.014 (0.049)**

Values are: Avg (StdDev). **Bold Font** represents best performance.

2) Robot Dynamics Model: To test the performance of our Robot Dynamics Model, we compared our model with the Velocity Discount Coefficient Matrix(VDCM) model proposed by Chen et al. [7]

As shown in TABLE III, we also use K-fold Cross-Validation to compare the performance of different models on the dataset. We calculate the error and standard deviation between the predicted robot velocity and the ground truth velocity for different models. Same as the results of NHMP, our models achieve good performance in both accuracy and stability, indicating that our models is effective in predicting robot's velocity.

C. Robot-Human Synchronization

To test the ability to adapt to individual differences among human agents, we monitor the speeds of both human and robot, and the relative distance between them during the guiding process.

As shown in Fig. 6, it can be seen that the robot can maintain a roughly similar speed with different human agents, and the distance between them can be kept stable during the guiding process. It can be concluded that our

BVIP guiding system can adapt well to individual differences among different human agents.

Fig. 6: Different human agents are guided by our quadruped robot equipped with our BVIP guiding system. Each of them has their own comfortable walking speed, and our robot is able to keep up with their speed.

D. Application for Real Robot

We apply the proposed guiding system to a real quadruped robot. The quadruped robot platform uses a a robot called WR1 developed by MiLAB Lab, as Fig 7. Test videos will be shown in supplementary materials, and the results show superior guiding performance in different people and at different speeds.

Fig. 7: Our BVIP guiding system has been deployed on a robot called WR1 developed by MiLAB Lab.

V. CONCLUSION

In this paper, we consider that the differences among human agents require different guiding strategies. In our proposed BVIP guiding system, we accurately model the motions of different human agents, and the different effects imposed on the robot, using AI-based HMP and RDM for sequence prediction. We compared our models with previous models and demonstrated that our models achieved outstanding performance. To further adapt the robot to individual differences, we propose the waypoints selector to correct the global path. It was shown that our BVIP robot can synchronize with the speed of different human agents and maintain a stable distance. In summary, our system

successfully addressed the adaptation problem of individual differences in the BVIP guiding task. In future work, we will improve the system so that it can adapt to individual differences without pre-collected data.

REFERENCES

[1] B. Li, J. P. Munoz, X. Rong, Q. Chen, J. Xiao, Y. Tian, A. Arditi, and M. Yousuf, "Vision-based mobile indoor assistive navigation aid for blind people," *IEEE transactions on mobile computing*, vol. 18, no. 3, pp. 702–714, 2018.

[2] V. Filipe, F. Fernandes, H. Fernandes, A. Sousa, H. Paredes, and J. Barroso, "Blind navigation support system based on microsoft kinect," *Procedia Computer Science*, vol. 14, pp. 94–101, 2012.

[3] S. Gilroy, D. Lau, L. Yang, E. Izaguirre, K. Biermayer, A. Xiao, M. Sun, A. Agrawal, J. Zeng, Z. Li *et al.*, "Autonomous navigation for quadrupedal robots with optimized jumping through constrained obstacles," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2021, pp. 2132–2139.

[4] C. Mastalli, I. Havoutis, M. Focchi, D. G. Caldwell, and C. Semini, "Motion planning for quadrupedal locomotion: Coupled planning, terrain mapping, and whole-body control," *IEEE Transactions on Robotics*, vol. 36, no. 6, pp. 1635–1648, 2020.

[5] F. Feurtey, "Simulating the collision avoidance behavior of pedestrians," *Master's thesis, University of Tokyo, Department of Electronic Engineering*, 2000.

[6] A. Nanavati, X. Z. Tan, J. Connolly, and A. Steinfeld, "Follow the robot: Modeling coupled human-robot dyads during navigation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 3836–3843.

[7] Y. Chen, Z. Xu, Z. Jian, G. Tang, Y. Yangli, A. Xiao, X. Wang, and B. Liang, "Quadruped guidance robot for the visually impaired: A comfort-based approach," *arXiv preprint arXiv:2203.03927*, 2022.

[8] S. Herath, H. Yan, and Y. Furukawa, "Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3146–3152.

[9] M. T. Gillespie, C. M. Best, E. C. Townsend, D. Wingate, and M. D. Killpack, "Learning nonlinear dynamic models of soft robots for model predictive control with neural networks," in *2018 IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE, 2018, pp. 39–45.

[10] N. A. Spielberg, M. Brown, and J. C. Gerdes, "Neural network model predictive motion control applied to automated driving with unknown friction," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 5, pp. 1934–1945, 2021.

[11] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 1271–1278.

[12] J. Wang and E. Olson, "Apriltag 2: Efficient and robust fiducial detection," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 4193–4198.

[13] R. Song, Y. Liu, and R. Bucknall, "Smoothed a* algorithm for practical unmanned surface vehicle path planning," *Applied Ocean Research*, vol. 83, pp. 9–20, 2019.

[14] N. E. Du Toit and J. W. Burdick, "Robot motion planning in dynamic, uncertain environments," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 101–115, 2011.

[15] P. Trautman, J. Ma, R. M. Murray, and A. Krause, "Robot navigation in dense human crowds: Statistical models and experimental studies of human–robot cooperation," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 335–356, 2015.

[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[19] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng *et al.*, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.