2024 IEEE 20th International Conference on Automation
Science and Engineering (CASE)
August 28 - September 1, 2024. Bari, Italy

Neural Human Intent Estimator for an Adaptive Robotic Nursing Assistant

Christopher Trombley, *Student Member IEEE*, Madan Rayguru, Payman Sharafian, *Student Member IEEE*, Irina Kondaurova, Nancy Zhang, Moath Alqatamin, Sumit K. Das, and Dan O. Popa, *Senior Member IEEE*

***Abstract*—Estimation of human intent during interaction with a robot is important for maintaining safety, predictability, and performance. This paper proposes a neural, model-free, online human intent estimator to guide our Adaptive Robotic Nursing Assistant (ARNA) robot. ARNA is a service mobile manipulator designed to assist nurses and healthcare workers with patient sitting and walking tasks in hospital environments. The proposed Human Intent Estimator (HIE) is implemented as two efficient one-layer neural networks (NN) that generate reference trajectories based on the user torque inputs from the robot's handlebar. These trajectories are sent to an NN-based neuroadaptive controller (NAC) in the inner loop to generate the necessary wheel torques to follow human-guided trajectories. The proposed NN weight adaptation laws for the HIE-NAC are shown to be stable as long as the trajectories generated by the human intent dynamics are smooth and bounded, which requires the user not to change his/her intention abruptly. We tested the proposed intent estimator and controller in three different human-robot interaction experiments with 10 participants. A linear mixed-effect model was used to test the difference between the HIE-NAC scheme and a conventional admittance controller. The results show significant performance improvements by reducing jerk and tracking velocity errors during operation.**

I. Introduction

The number of robots introduced into environments with physical human collaborators has been rapidly increasing [1] [2]. Healthcare facilities are one of the frontiers where physical human-interaction (pHRI) capable robots can be employed to assist with medical needs such as monitoring patients, daily hygiene, and more. The idea of using nursing assistant robots has been a longstanding consideration for tasks such as assisting patients with lifts, transfers, and dressing [3] [4].

To help nursing staff with physically intensive duties and tasks, an Adaptive Robotic Nursing Assistant (ARNA) unit has been developed in our lab over the last decade [5]. ARNA was designed to serve in a hospital environment and perform pick-and-place tasks, guidance, tugging, and other functions. It has omnidirectional motion capabilities to suit the dynamic hospital environment and a robotic arm with seven DOFs on the top of its base, as shown in Fig.1. pHRI has long been an active research area to improve the robot's performance and maintain safety. Impedance and admittance control methods have been commonly used to control the interaction behavior during pHRI [6] [7] [8]. These control techniques, however, need to know the robot dynamics and robot-environment interaction dynamics for proper tuning. In the case of ARNA, the system dynamics is perturbed by uncertain and diverse environmental factors such as floor inclines, surfaces, and tugging heavy loads. Adaptive control methods have previously been proposed to deal with the uncertainties and parameter changes in the robot dynamics and to overcome the ambiguity in the contact environment [9]. Combining impedance control schemes with adaptive methods has been used to achieve a safe and stable operation during pHRI [10] [11] [12]. Many researchers proposed using neural networks to approximate the system nonlinearities [13] by employing offline batch training methods. Safety, operational efficiency, and task reliability during pHRI could be greatly enhanced if the robots can handle nonlinearities and infer human actions and intentions toward the robot [14].

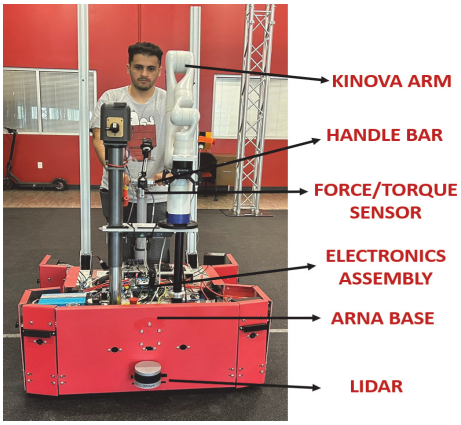

Fig. 1: Adaptive Robotic Nursing Assistant (ARNA) robot

Human intention can be deduced by utilizing various sensing modalities to measure different types of human signals, which are captured by wearable and nonwearable sensors [15], [16], [17], [18]. Previously, the human motion dynamics were modeled to estimate users' intention in [19] [20]. Hidden Markov Models (HMMs) are used in [21] to estimate people's actions and interactions with autonomous mobile robots. In [22], a Bayesian methodology was employed to estimate both human impedance and motion intention within the context of a collaborative human-robot task. Wang et al. [23] introduced an intention-driven dynamics model to deduce intentions from observed human motions, specifically in the context of robot table tennis and human activities. Mean-

This work was supported by the US National Science Foundation (NSF) through grants SCH IIS #1838808 and EPSCoR OIA #1849213. Authors are with the Louisville Automation and Robotics Research Institute, University of Louisville, KY, USA Contact: madanmohan.rayguru@louisville.edu.

while, the work [24] utilized Gaussian processes (GP) based on impedance for predicting behavior during pHRI. Their findings demonstrated improved performance compared to the standard GP model; however, this enhanced performance was highly sensitive to changes in the impedance parameters.

In our previous work [10], we designed an adaptive impedance controller with human intent estimation based on converting the applied force on the robot to a desired future position. A double integrator dynamics was exploited to approximate the human intent pushing the robot end-effector. In an additional past work [25], we designed a dual loop control scheme for the guidance pHRI application with the PR-2 robot arm. These control schemes have an inner loop based on a model-free neuroadaptive controller (NAC) to ensure that the robot follows the desired trajectories and an outer loop that predicts the human motion intent and estimates the reference trajectories for the inner loop. These Cartesian space controllers were experimentally validated on a small number of human subjects, which did not allow for statistically significant conclusions about the performance.

In our follow-up work [5], we adapted the NAC inner loop controller for ARNA's mobile base in joint space and combined it with a fixed admittance input model from the robot's handlebar force-torque sensor. Tuning the NAC in joint space was easier than in Cartesian space because joint variables have comparable dimensional values. The controller was extensively validated for performance improvements against a standard PID controller. However, it lacked the human intent estimator in the outer loop.

In this work, we combined the advantages of NN-based human-intent estimation with the inner loop NAC and made contributions in two directions: 1) We designed a pHRI scheme called HIE-NAC for our omnidirectional mobile robot base of ARNA for walking assistance; Specifically, we refined the pHRI framework based on a cascaded HIE-NAC control structure shown in Fig.2. Furthermore, we showed that the HIE scheme is stable under reasonable assumptions about the operator's force outputs. 2) We implemented the HIE-NAC scheme on our robot and performed user trials to rigorously test the effectiveness under various scenarios. The outer loop of HIE-NAC was implemented as two efficient one-layer NNs. No batch training was required, and only a few parameters needed to be adjusted to optimize the performance, including the NN size and learning rates. To ensure that the improved tracking errors and jerks of our controller are not due to random effects, we conducted experiments with 10 subjects who guided the robot along different desired paths. Data collected were statistically analyzed using a mixed effect regression model that demonstrates that our proposed scheme statistically improves a previously tuned admittance controller in terms of the smoothness and precision of guided operation.

The paper is organized as follows. The outer-loop HIE design is presented, and proof of stability is shown in section II. In section III, we describe the ARNA robot used during the experiments and present our experimental results, including the interpretation of statistics. Finally, section IV concludes the paper and discusses future work.

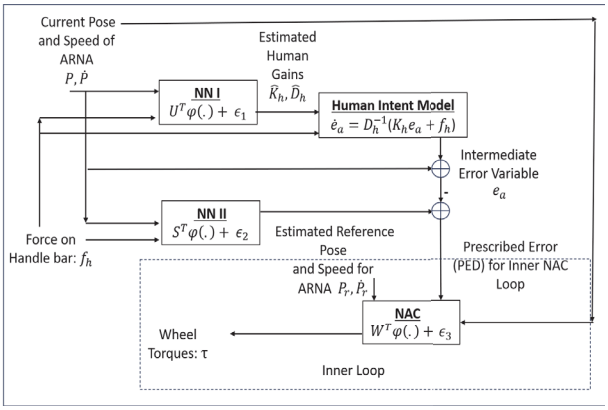

Fig. 2: ARNA Robot's HIE-NAC architecture consisting of 3 single hidden layer NNs with online updated weights

II. Control Method and Design

This section describes the technical details of the proposed pHRI system, which involves designing and implementing the outer-loop intent estimator and the inner-loop neuroadaptive controller. The outer-loop intent estimator utilizes the user's handlebar sensor data to compute a set of desired position and velocity commands for the wheel motors. These signals are passed onto the inner-loop NAC, which generates robot base motor torques and tracks the outer-loop pHRI trajectories.

A. Outer-Loop Intent Estimator Design

The objective of the HIE outer loop is to compute a set of desired position and velocity commands for the wheel motors based on the handlebar sensor data from the user. The architecture of the proposed HIE-NAC scheme is depicted in Fig. 2. We developed the HIE loop for ARNA based on the human intent model in (1). The outer HIE-NAC loop consists of two parallel single-layer neural networks. One is used to estimate the parameters (i.e., gains) of the human model, and another one is used to approximate desired trajectories for the robot to follow. It is important to note that the parameters of the human model differ from person to person; therefore, the adaptive estimation of these constants is critical to make the interaction personalized and smooth.

1) Human Transfer Function

We begin our discussion of the HIE outer loop design by describing the human intent model. In physical interactions with ARNA, the user guides the robot along desired paths by applying force to the robot's handlebar. Users change their applied forces and moments dynamically based on their perceived tracking errors and other measures of operation smoothness, such as the jerk. Given a current planar position and orientation of ARNA $P = \begin{bmatrix} x, y, \theta \end{bmatrix}^T$ and velocity $\dot{P} = \begin{bmatrix} v_x, v_y, \omega_z \end{bmatrix}^T$, the human dynamics can be modeled as a low pass filter [11]:

$$f_h = D_h \dot{e_d} + K_h e_d \tag{1}$$

where $f_h \in \mathbb{R}^{3\times 1}$ is the human force vector along the three-axis handlebar force sensor; $D_h = diag(d)$, $K_h = diag(k)$ are individual specific diagonal gain matrices containing coefficients for 3-DOF of in-plane velocities in x, y, ω directions); $e_d = P_r - P$ and $\dot{e_d} = \dot{P}_r - \dot{P}$ are the tracking errors to a reference robot position P_r. The model (1) captures this user-specific interaction through the gain matrices K_h and D_h, which are tuned by our HIE NN1. Finally, the HIE NN2 generates a robot trajectory, and the error dynamics e_d are combined into a prescribed error dynamic term and passed on to NN3 of our inner loop NAC controller.

2) Human Intent Estimator

As the desired position and velocity are known to the user but unknown to the robot, the proposed HIE scheme estimates these signals through neural network NN2 expressed as:

$$\begin{bmatrix} P_r \\ \dot{P}_r \end{bmatrix} = h(P, \dot{P}, f_h) = S^T \phi(\xi) + \epsilon_1 \tag{2}$$

where the unknown weight matrix of the NN2 is $S^T \in \mathbb{R}^{6\times 9}$, ϕ denotes a sigmoid activation function, ϵ_1 is a approximation residual, and $\xi = \begin{bmatrix} f_h^T, P^T, \dot{P^T} \end{bmatrix}^T \in \mathbb{R}^{9\times 1}$. The estimated human intent can be written as:

$$\begin{bmatrix} \hat{P}_r \\ \dot{\hat{P}}_r \end{bmatrix} = \hat{S}^T \phi(\xi) \tag{3}$$

where $\hat{S}^T$ is the estimated weight matrix. $\hat{P}_r$ and $\dot{\hat{P}}_r$ are the estimated desired position and velocity. The error dynamics can be written as:

$$\begin{bmatrix} \hat{e}_d \\ \dot{\hat{e}}_d \end{bmatrix} = \begin{bmatrix} \hat{P}_r - P \\ \dot{\hat{P}}_r - \dot{P} \end{bmatrix} \tag{4}$$

The difference between the actual and estimated errors is defined as:

$$\bar{\tilde{e}}_d = \begin{bmatrix} \tilde{e}_d \\ \dot{\tilde{e}}_d \end{bmatrix} = \begin{bmatrix} e_d \\ e_d \end{bmatrix} - \begin{bmatrix} \hat{e}_d \\ \dot{\hat{e}}_d \end{bmatrix} = \begin{bmatrix} P_r \\ \dot{P}_r \end{bmatrix} - \begin{bmatrix} \hat{P}_r \\ \dot{\hat{P}}_r \end{bmatrix} = \tilde{S}^T \phi(\xi) + \epsilon_1 \tag{5}$$

where $\tilde{S} = S - \hat{S}$ is the weight estimation error. Let's define a sliding mode error variable as: $s = \hat{e}_d - e_a$, where the filtered error e_a is determined from the approximated human dynamics:

$$f_h = \hat{D}_h \dot{e}_a + \hat{K}_h e_a = J(\hat{E}_h \odot \bar{e}_a) \tag{6}$$

where $E_h = \begin{bmatrix} k \\ d \end{bmatrix} \in \mathbb{R}^{6\times 1}$ contains the diagonal elements of $\hat{K}_h$, $\hat{D}_h$, $J = [I_6, I_6]$, $\odot$ is the Hadamard product and the vector $\bar{e}_a = \begin{bmatrix} e_a \\ \dot{e}_a \end{bmatrix} \in \mathbb{R}^{6\times 1}$. These gains k, d are estimated by neural network NN1 given in equation (9). To find the open loop error dynamics of the HIE loop, we start from the human dynamics equation, substitute the value of human force and use the definition of $\tilde{e}_d(t)$ to derive the following expression:

$$D_h(\dot{\hat{e}}_d - \dot{e}_d) = -K_h(\hat{e}_d - e_a) - K_h \tilde{e}_d - D_h \dot{\tilde{e}}_d - \tilde{K}_h e_a \tag{7}$$

where $\tilde{D}_h$ and $\tilde{K}_h$ are the estimation errors of D_h and K_h defined as:

$$\tilde{D}_h \triangleq D_h - \hat{D}_h, \tilde{K}_h \triangleq K_h - \hat{K}_h \tag{8}$$

As the human gain matrix E_h is unknown, we assume that it is a nonlinear function of ξ in the form $E_h = U^T \phi(\xi) + \epsilon_2$. It can be approximated by:

$$\hat{E}_h = \hat{U}^T \phi(\xi) \tag{9}$$

where $\hat{U} \in \mathbb{R}^{6\times 9}$ is the weight matrix of a second neural network. After using the definition of s the open loop dynamics can be written as

$$D_h \dot{s} = -K_h s - J(E_h \odot \bar{\tilde{e}}_d) - J(\tilde{E}_h \odot \bar{e}_a) \tag{10}$$

Stability Analysis: The weight matrices of two neural networks should be updated in such a way that s is bounded, which means the dynamics (10) is stable. In this regard, define a candidate Lyapunov function:

$$V = 0.5 s^T D_h s + 0.5 tr(\tilde{S}^T A^{-1} \tilde{S}) + 0.5 tr(\tilde{U}^T B^{-1} \tilde{U}) \tag{11}$$

where A, B are positive definite matrices, which can be tuned for weight updates. After some matrix manipulation, we obtain:

$$\begin{aligned} \dot{V} \leq -||s|| \Bigg(\lambda ||s|| + \kappa \begin{bmatrix} ||\tilde{S}^T||_F \\ ||\tilde{U}^T||_F \end{bmatrix}^T \begin{bmatrix} S_b \\ U_b \end{bmatrix} \\ + \begin{bmatrix} ||\tilde{S}^T||_F \\ ||\tilde{U}^T||_F \end{bmatrix}^T \begin{bmatrix} -\kappa & 0.5\phi_b^2 \\ 0.5\phi_b^2 & -\kappa \end{bmatrix} \begin{bmatrix} ||\tilde{S}^T||_F \\ ||\tilde{U}^T||_F \end{bmatrix} \Bigg), \end{aligned} \tag{12}$$

where S_b, U_b denotes the upper bound of the weight matrices, and ϕ_b is the upper bound of the activation function output.By examining equation (12), we can prove that the sliding error s is uniformly ultimately bounded in the region defined by:

$$\begin{aligned} &||s|| \geq c_1/\lambda, c_1 = \frac{\kappa^2(\kappa S_b^2 + \kappa U_b^2 + U_b S_b \phi_b^2)}{\phi_b^4 - 4\kappa^2}, \text{ or} \\ &\begin{bmatrix} ||\tilde{S}^T||_F \\ ||\tilde{U}^T||_F \end{bmatrix} > c_2, \; c_2 = \sqrt{c_1} + \frac{\kappa}{\phi_b^4 - 4\kappa^2} \frac{2\kappa S_b + U_b \phi_b^2}{2\kappa U_b + S_b \phi_b^2}. \end{aligned} \tag{13}$$

Note that the tracking error bounds can be made smaller by reducing the design constant κ. The above-described stability analysis can be formalized as the following theorem:

Theorem 1: *The HIE scheme implementing neural updates ensures that the discrepancy between reference values ($\hat{P}_r, \dot{\hat{P}}_r$) and user intended position and velocity $P_r, \dot{P}_r$ asymptotically converge to a small bound, under the following conditions: 1) the user interacts with ARNA with a bounded amount of force f_h; 2) the intended desired position P_r and velocity $\dot{P}_r$ are bounded; and 3) the NN weights in equations* (3) *and* (9) *are saturated inside compact sets.*

B. Inner-Loop Neuroadaptive Controller

The inner loop NAC learns and cancels the robot's remaining Mass and Coriolis nonlinearities of the robot and forces the current position of ARNA P towards the reference trajectory P_r. The NAC computes the wheel input torque vector τ by regulating a filtered error variable $e \triangleq P - P_r, r \triangleq \dot{e} - \rho e$ where ρ is a symmetric design matrix. In this section, we briefly introduce the NAC controller formulated through a third Neural Network. The detailed performance of the inner loop was presented in previous publications [10], [25]–[27]. Let's assume that we have a nonlinear model that describes

our robotic system, given by:

$$M\ddot{x} + C(x, \dot{x})\dot{x} + G(x) + T(x) = \tau_d \tag{14}$$

where M is the moment of inertia matrix, $C(.)$ is Coriolis Matrix, $G(x)$ is gravitational forces, $T(x)$ is cumulative uncertianties and τ_d is the input torque. By making use of the universal nonlinear function approximation property of Neural Networks (NN), we can approximate $T(x)$ as:

$$T(x) = W_n^T \sigma(V_n^T x) + \epsilon \tag{15}$$

where W_n and V_n are ideal weights, σ is the activation function vector and ϵ is the approximation error of the NN approximation such that $\|\epsilon\|_F \leq \epsilon_N$ on a compact set. Since the ideal weights are unknown, estimated weights that are dynamically tuned, $\hat{W}_n$ and $\hat{V}_n$, can be used to obtain $\hat{T}(x)$, an approximation of $T(x)$ as:

$$\hat{T}(x) = \hat{W}_n^T \sigma(\hat{V}_n^T x) \tag{16}$$

Let an approximation torque controller based on $\hat{T}(x)$ be:

$$\tau = -\hat{T(x)} + K_v r - t_r(t) \tag{17}$$

Where K_v is a positive definite matrix, and $K_v r$ is used to ensure a proportional-derivative (PD) tracking the performance of the closed loop system. Here t_r is a robustifying signal compensating for unmodeled and unstructured disturbances.

$$t_r = -K_z(\|\hat{Z}\|_F + Z_B)r \tag{18}$$

with K_z as the robustifying term gain and

$$\hat{Z} = \begin{pmatrix} \hat{W}_n & 0 \\ 0 & \hat{V}_n \end{pmatrix} \tag{19}$$

$\|.\|_F$ is the Frobeinus norm, and Z_B is a bound on the NN weights.

Using (17) and sliding-mode error dynamics of the robot and controller can be simplified to:

$$M\dot{r} = -C(x, \dot{x})r + \tilde{T}(x) + \tau_d + t_r. \tag{20}$$

The control law can ensure that, given a bounded initial reference trajectory and other feasible bounds on quantities described in the proof, the error in the following trajectories of robot states θ and derived states $\|r\|$ are asymptotically bounded. This proof presents weight tuning laws for the approximate NN weights $\hat{W}_n$ and $\hat{V}_n$ as:

$$\dot{\hat{W}}_n = A\hat{\sigma}r^T - A\hat{\sigma}^{'}\hat{V}_n x r^T - \kappa A\|r\|\hat{W}_n \tag{21}$$

$$\dot{\hat{V}}_n = Bx(\hat{\sigma}^{'}T\hat{W}_n r)^T - \kappa B\|r\|\hat{V}_n \tag{22}$$

$$\hat{\sigma}^{'} = diag\{\sigma(\hat{V}_n^T x)\}[I - diag\{\sigma(\hat{V}_n^T x\}] \tag{23}$$

In these update equations, A and B are two positive definite matrices, σ is the sigmoid activation function, and $\kappa > 0$ is a small design parameter.

III. Experimental Results

A. ARNA Robot

The Adaptive Robot Nursing Assistant (ARNA) robot is a customized mobile manipulator consisting of a base, a 7-DOF arm, and a sensorized handlebar to allow patients and nurses to physically interact. ARNA uses a four-wheel Mecanum drive and suspension system for omnidirectional movement. The wheels are arranged in a longitudinal symmetrical layout and integrated with four independently controlled servo motors [5]. The wheels are affixed at each of the robot chassis's four corners. The drivetrain provides omnidirectional mobility, enabling the robot to perform both agile translation (forward/backward, sideways) and rotation maneuvers independently and simultaneously, regardless of its initial orientation. This eliminates the need for difficult nonholonomic path maneuvering in cluttered hospital environments. High-ratio gearboxes are used to perpendicularly connect the Mecanum wheels to the servo motors. Assuming the slippage between the rollers and the ground is negligible, the inverse kinematics of the robot base can be written as [27]:

$$V_w = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \frac{1}{R} \begin{bmatrix} 1 & -1 & \frac{-(L_x+L_y)}{2} \\ 1 & 1 & \frac{(L_x+L_y)}{2} \\ 1 & 1 & \frac{-(L_x+L_y)}{2} \\ 1 & -1 & \frac{-(L_x+L_y)}{2} \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega_z \end{bmatrix} \tag{24}$$

where the vector $\begin{bmatrix} v_x, v_y, \omega_z \end{bmatrix}^T$ represents the velocity of ARNA center of rotation, $V_\omega \in \mathbb{R}$ denotes the speed vector of the wheels. The parameters R, L_x, and L_y are the wheel radius, the width, and the length of the ARNA base, respectively.

The primary pHRI interface for ARNA is a handlebar attached to the robot's rear section. This serves a dual purpose of supporting users while walking along the robot, as well as measuring the force applied to it through a 6-axis torque/force sensor and other handlebar pressure sensors. The force or torques measured on the handlebar are then sent to the real-time patched OS of our ARNA computer, which uses them to estimate the user-intended motion directions based on the NAC-HIE algorithm.

The control scheme was implemented using C++. The generated torques are sent to ARNA's four-wheeled motor drivers through the Robotic Operating System (ROS-1 Kinetic) running on an Ubuntu 16.05 operating system. The human intent estimator generates the reference velocity vector in the outer loop. Using the inverse kinematics from (1) and the reference angular velocities, the rotation vector for the four wheels is calculated and sent to the NAC controller in the inner loop. The inner loop controller is running at 300 Hz, and the outer loop HIE is running at 100 Hz.

B. Description of Experiments

Design parameters: In this section, we present experiments used to validate the proposed NAC-HIE control scheme compared to a baseline method involving a tuned conventional admittance model instead of the HIE block. Both these controllers use our neuroadaptive controller (NAC) for inner loop velocity tracking. The parameters set during experiments were as follows κ was set to 0.001, $A = \{0.04, 0.04, 0.04\}$, and $B = \{0.01, 0.01, 0.01\}$. D_h, J, and K_h were all set to the identity matrix to begin each interaction experiment. D_h, and K_h are dynamically updated by our proposed NAC-HIE method as the user is operating the robot, while J remains the identity matrix to be used in the weight update equations. For the admittance portion, a

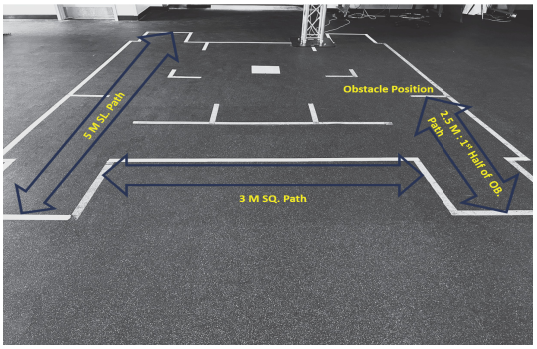

Fig. 3: Floor paths straight line (SL), square (SQ), and obstacle (OB) for evaluation of human-intent estimator and controller

second order transfer function [6]:

$$\frac{F(s)}{\dot{P}(s)} = \frac{1}{m_a s^2 + d_a s + k_a} \tag{25}$$

is implemented for reference velocity generation where the parameters of the transfer function ($k_a = 0.2, d_a = 0.03, m_a = 1.1$) are tuned by trial and error in order to minimize both velocity tracking as well as jerk for all the guiding tasks. We carried out experiments to investigate whether the proposed NAC-HIE control scheme is better than the admittance-NAC controller while following three reference trajectories (straight paths, square paths, and paths with obstacles). The performance metrics employed included velocity tracking errors and smoothness of operation represented as motion jerk. For the remainder of the paper, we refer to our conventional admittance controller as simply NAC, while our proposed human intent estimator and controller is the HIE-NAC.

Participants: 10 subjects were recruited for this study. The selection criteria for these participants are that they are at least 18 years old and have no prior experience with the robot. For each experiment, the subject was asked to follow a particular reference path: a straight line (SL), a square path (SQ), or a straight line with an obstacle (OB), depicted in Fig. 3. The University of Louisville Institutional Review Board granted approval to conduct the experiments under IRB no.18.0659.

Experiment Design and Instruments: This study employed a repeated measures design for two conditions (Conventional Controller (NAC), Proposed Controller (HIE-NAC)) with three tasks, e.g. following paths SL,SQ,OB. All experiments were run in one sitting for each participant. For a fair comparison, we administered a brief learning task before the experiment to familiarize the participants with the robot. The learning task simply consisted of the user acclimatizing themselves to the robot by pushing it around a predefined path.

The first experiment was designed to measure the performance of each method on a simple task consisting of the participant pushing the robot in a straight line for a distance of 5m. The second experiment was designed to measure the performance of each method on a more complex task involving higher dimensional motion. This task consisted of the participant pushing the robot in a 3m x 3m square trajectory. The third experiment was designed to measure the performance in a scenario where a user must navigate around an obstacle. This task consisted of the participant pushing the robot in a straight line for 2.5m, navigating around a chair in the path, and then pushing the robot another 2.5m. To measure velocity errors, we recorded the reference and actual velocity values. The reference velocity was computed using our proposed human intent estimator. The actual velocity was calculated using wheel encoders. To investigate the operation's smoothness, we used motion jerk information. Generally, jerk quantifies the rate of acceleration change. A lower jerk demonstrates a smoother motion while the user is pushing the robot. These were computed according to standard definitions in the literature [5], [25].

TABLE I: **Velocity Error Statistics.**

Meth.	Exp.	Dir.	Min	Median	Mean	Max
NAC	SL	E_x	0.025	0.050	0.055	0.100
	OB	E_x	0.040	0.050	0.062	0.100
		E_y	0.020	0.060	0.057	0.090
		E_z	0.008	0.030	0.034	0.090
	SQ	E_x	0.030	0.070	0.088	0.160
		E_y	0.006	0.070	0.064	0.120
		E_z	0.007	0.030	0.035	0.070
HIE-NAC	SL	E_x	0.020	0.035	0.051	0.140
	OB	E_x	0.030	0.040	0.051	0.090
		E_y	0.010	0.020	0.029	0.060
		E_z	0.001	0.010	0.011	0.030
	SQ	E_x	0.010	0.035	0.057	0.140
		E_y	0.002	0.020	0.027	0.060
		E_z	0.003	0.035	0.048	0.110

TABLE II: **Jerk Statistics.**

Meth.	Exp.	Dir.	Min	Median	Mean	Max
NAC	SL	J_x	9.4	72.0	81.2	207.0
	OB	J_x	62.0	106.0	124.4	217.0
		J_y	45.0	90.0	91.6	146.0
		J_z	55.0	117.0	126.8	231.0
	SQ	J_x	94.0	141.0	139.1	172.0
		J_y	6.0	148.0	145.9	241.0
		J_z	83.0	105.0	124.1	215.0
HIE-NAC	SL	J_x	7.9	13.3	36.0	112.0
	OB	J_x	4.1	6.7	7.3	12.1
		J_y	1.7	5.6	5.2	8.5
		J_z	1.8	6.3	5.4	7.3
	SQ	J_x	5.0	9.9	9.0	12.8
		J_y	4.5	7.0	7.5	10.8
		J_z	2.1	6.4	6.8	10.1

C. Descriptive Statistical Analysis

The acquired datasets were cleaned and visually checked for outliers using standard procedures such as boxplots and histograms. Any observation that was classified as a suspected outlier using the interquartile range (IQR) criterion was removed from the dataset. The statistical results are summarized in Tables I and II. E_x, E_y, E_z represent velocity errors in x, y, z coordinates, and J_x, J_y, J_z denote jerk in x,y, and rotation coordinates respectively. It can be seen from Table I that min, max, median, and mean of velocity errors for HIE-NAC are considerably smaller across all experiments and directions than for NAC. This is except for the mean value of E_z in SQ, as explained in the Discussion section. Similar trends are also observed for the jerk data, where

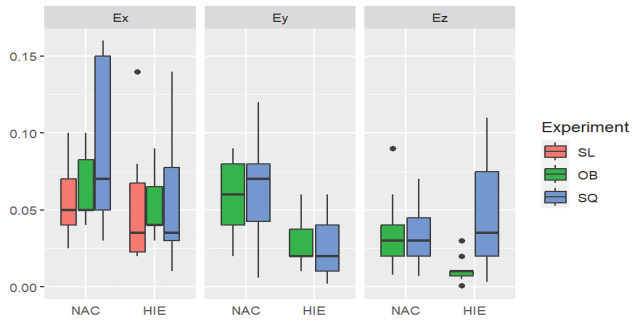

Fig. 4: Distribution of velocity errors for all subjects

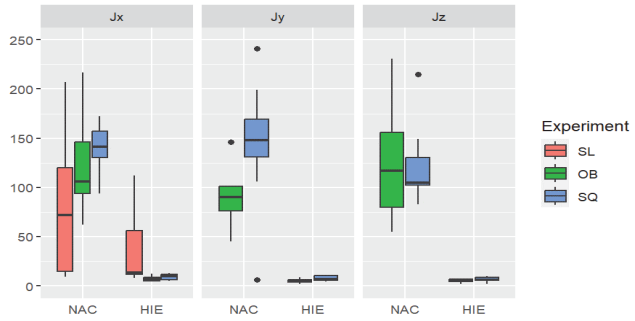

Fig. 5: Distribution of jerk for all subjects

HIE-NAC gives smoother performance than NAC for all three types of experiments. We also plotted the data from all subjects to examine the data distribution. We chose box plots to visualize the data distribution, and these plots comparing HIE-NAC vs. NAC for all experiments by direction are provided in Figures 4 and 5.

D. Mixed Effect Regression Model Analysis

Even though the results presented in the previous subsection are promising, they are not conclusive. It could be argued that the variance in performance may be due to random effects such as the individual's abilities or other influencing factors. Therefore, we fitted a mixed effect model (estimated using REML and nloptwrap optimizer) to compare velocity errors and jerks with Experiment, Method, and Directions as independent variables. The baseline corresponded to Method = NAC, Experiment = SL, and Direction = E_x, J_x. The model included ID as a random effect (intercept) term. P-values were computed using a Wald t-distribution approximation. We fitted two models, one for velocity error and another for our jerk data. The results are presented in Tables III and IV.

The results for the velocity errors mixed effect model are provided in Table III. Within this model, the estimated effect of HIE-NAC is negative and statistically non-significant ($\hat{\beta}_{HIE-NAC} = -0.006, p = 0.614$). The estimated effects of E_y and E_z are statistically significant and negative ($\hat{\beta}_{E_y} = -0.021, p < 0.001$) and ($\hat{\beta}_{E_z} = -0.03, p < 0.001$), compared to the straight line (SL). These results suggest that the average velocity errors are smaller in the direction of y and z. However, the explanatory power of the model is relatively low (conditional R^2 = 0.45, marginal R^2= 0.21), indicating that the proposed control strategy offers

TABLE III: Mixed Effects Model Results: Velocity Error Comparison

Variables	Estimate	Std. Error	Pr(t)
Intercept	0.057	0.010	0.000
OB	0.011	0.011	0.284
SQ	0.025	0.011	0.024
HIE-NAC	-0.006	0.012	0.614
E_y	-0.021	0.006	0.001
E_z	-0.030	0.006	0.000
OB:HIE	-0.014	0.014	0.303
SQ:HIE	-0.015	0.014	0.301
			$R^2 = 0.45$

TABLE IV: Mixed Effects Model Results: Jerk Comparison

Variables	Estimate	Std. Error	Pr(t)
Intercept	81.557	12.323	0.000
OB	39.209	14.780	0.009
SQ	59.021	14.609	0.000
HIE-NAC	-45.547	16.387	0.006
J_y	-3.795	8.459	0.655
J_z	-3.107	8.365	0.711
OB:HIE	-66.974	19.241	0.001
SQ:HIE	-84.951	19.016	0.000
			$R^2 = 0.73$

limited improvement in practical applications. These results are expected since the inner loop, which is responsible for generating torques for desired velocities, remains the same for both controllers.

The results in Table IV indicate the performance of our proposed controller HIE-NAC in terms of jerk. We found that the jerk mixed effect model's total explanatory power is substantial (conditional $R^2 = 0.73$, marginal $R^2 = 0.70$). The estimated effects of the experiment (OB) are statistically significant and positive ($\hat{\beta}_{OB} = 39.21, p = 0.009$). The estimated effect of the experiment (SQ) is statistically significant ($\hat{\beta}_{SQ} = 59.02, p < 0.001$), which means that the observed jerk for the square path and obstacle path are substantial and higher compared to the straight line path when ARNA is operated with the baseline NAC controller. The effect of HIE-NAC itself is significant and negative ($\hat{\beta}_{HIE-NAC} = -45.55, p = 0.006$), suggesting that the average jerk in the HIE-NAC experiment (ARNA operated with the proposed controller) is lower than the one in the conventional NAC experiment across all tasks/paths (SL, SQ, OB). The interaction terms b/w the Experiment type and HIE are also significant and negative ($\hat{\beta}_{OB} = -66.97, p = 0.001$ and $\hat{\beta}_{SQ} = -84.95, p < 0.001$). Hence, the proposed HIE controller may significantly reduce robot's jerk during an obstacle avoidance task (OB) or a square path tracking (SQ). Overall, the mixed effect model results reveal that our proposed HIE-NAC makes the operation of ARNA personalized and much smoother than the conventional admittance controller.

IV. Discussion

The proposed HIE-NAC framework demonstrates promising improvements in reducing jerk and increasing smoothness in guided robot motion. However, several limitations must be addressed. From Table I, we can observe in the SQ path that the mean value of Ez in HIE-NAC is larger than NAC. This

is because the mean can be influenced by outliers and large values, potentially skewing the result in those directions. If some users changed their direction very fast during the square path experiment, it would have affected the mean in the angular direction, which is E_z in our case. It's worth noting that the median value for HIE is smaller than for NAC, as the median is not impacted by outliers. The small sample size in our study restricts the statistical power and generalizability of the findings. Additionally, the current framework may not fully account for more complex motion scenarios and constraints. The reference for our HIE controller is generated from the human intent model, which takes the forces exerted on the force sensor as its input. If the user behaves erratically or abruptly changes his/her intention, the reference signals generated might be non-smooth and unbounded. Our current controller cannot track and adapt to these types of signals.

V. Conclusion and Future Work

This paper presents HIE-NAC, a nonlinear human intent estimator and controller framework HIE-NAC using three neural networks to estimate the desired velocity of a guided mobile manipulator based on user-applied force. The HIE includes a neural network to estimate human intent gains, which are then used by another network to compute desired positions and velocities for the robot base. The neural network weight updates are derived from a Lyapunov stability analysis, ensuring guaranteed error bounds under reasonable assumptions. A standard admittance controller maps Desired robot velocities from user-applied force-torque values. Testing against the standard NAC controller using data from 10 users shows that HIE-NAC significantly reduces jerk and increases motion smoothness. However, the small sample size limits the study's ability to detect true effects, so results should be interpreted cautiously. Future studies with larger samples are necessary to confirm these findings. Future work will optimize HIE-NAC for more complex motion scenarios and constraints and incorporate the ARNA manipulator in addition to the base.

References

[1] H. Modares, I. Ranatunga, B. AlQaudi, F. Lewis, and D. Popa, "Intelligent human–robot interaction systems using reinforcement learning and neural networks," *Trends in Control and Decision-Making for Human–Robot Collaboration Systems*, p. 153–176, 2017.

[2] V. Duchaine and C. Gosselin, "Safe, stable and intuitive control for physical human-robot interaction," in *2009 IEEE International Conference on Robotics and Automation*. IEEE Press, 2009, p. 3383–3388.

[3] M. Kangasniemi, S. Karki, N. Colley, and A. Voutilainen, "The use of robots and other automated devices in nurses' work: An integrative review," *International Journal of Nursing Practice*, 2019.

[4] N. Maalouf, A. Sidaoui, I. Elhajj, and D. Asmar, "Robotics in nursing: A scoping review," *Journal of Nursing Scholarship*, vol. 50, p. 590–600, 2018.

[5] S. Abubakar, S. K. Das, C. Robinson, M. N. Saadatzi, M. C. Logsdon, H. Mitchell, D. Chlebowy, and D. O. Popa, "Arna, a service robot for nursing assistance: System overview and user acceptability," in *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, 2020, pp. 1408–1414.

[6] N. Hogan, "Impedance control: An approach to manipulation," in *1984 American Control Conference*, 1984, p. 304–313.

[7] D. Jeong and S. Jung, "Comparison studies of two major force control algorithms for a single axis force control of a robot manipulator," in *2021 21st International Conference on Control, Automation and Systems (ICCAS)*, 2021, p. 12–15.

[8] J. Koivumäki and J. Mattila, "Stability-guaranteed impedance control of hydraulic robotic manipulators," *IEEE/ASME Transactions on Mechatronics*, vol. 22, p. 601–612, 2017.

[9] M. M. Rayguru, S. Roy, L. Yi, M. R. Elara, and S. Baldi, "Introducing switched adaptive control for self-reconfigurable mobile cleaning robots," *IEEE Transactions on Automation Science and Engineering*, pp. 1–12, 2023.

[10] I. Ranatunga, F. Lewis, D. Popa, and S. Tousif, "Adaptive admittance control for human–robot interaction using model reference design and adaptive inverse filtering," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 1, p. 278–285, 2017.

[11] H. Modares, I. Ranatunga, F. Lewis, and D. Popa, "Optimized assistive human–robot interaction using reinforcement learning," *IEEE Transactions on Cybernetics*, vol. 46, p. 655–667, 2016.

[12] Z. Li, J. Liu, Z. Huang, Y. Peng, H. Pu, and L. Ding, "Adaptive impedance control of human–robot cooperation using reinforcement learning," *IEEE Transactions on Industrial Electronics*, vol. 64, p. 8013–8022, 2017.

[13] W. He, C. Xue, X. Yu, Z. Li, and C. Yang, "Admittance-based controller design for physical human–robot interaction in the constrained task space," *IEEE Transactions on Automation Science and Engineering*, vol. 17, p. 1937–1949, 2020.

[14] N. Amirshirzad, A. Kumru, and E. Oztop, "Human adaptation to human–robot shared control," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 2, p. 126–136, 2019.

[15] J. Elfring, R. Molengraft, and M. Steinbuch, "Learning intentions for improved human motion prediction," *Robotics and Autonomous Systems*, vol. 62, p. 591–602, 2014.

[16] A. Al-Yacoub, A. Buerkle, M. Flanagan, P. Ferreira, E. Hubbard, and N. Lohse, "Effective human-robot collaboration through wearable sensors," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, 2020, p. 8–11.

[17] Y. Guo, C. Shi, and X. Yang, "Reverse psychology in trust-aware human-robot interaction," *IEEE Robotics and Automation Letters*, vol. 6, p. 4851–4858, 2021.

[18] D. Kulic and E. Croft, "Affective state estimation for human–robot interaction," *IEEE Transactions on Robotics*, vol. 23, p. 991–1000, 2007.

[19] J. Elfring, R. Molengraft, and M. Steinbuch, "Learning intentions for improved human motion prediction," in *2013 16th International Conference on Advanced Robotics (ICAR)*, 2013, p. 1–7.

[20] R. Kelley, M. Nicolescu, A. Tavakkoli, M. Nicolescu, C. King, and G. Bebis, "Understanding human intentions via hidden markov models in autonomous mobile robots," in *2008 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2008, p. 12–15.

[21] X. Yu, "Bayesian estimation of human impedance and motion intention for human–robot collaboration," *IEEE Transactions on Cybernetics*, vol. 51, p. 1822–1834, 2021.

[22] J. Medina, S. Endo, and S. Hirche, "Impedance-based gaussian processes for predicting human behavior during physical interaction," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, p. 16–21.

[23] Z. Wang, A. Boularias, K. Mülling, B. Schölkopf, and J. Peters, "Anticipatory action selection for human–robot table tennis," *Artificial Intelligence*, vol. 247, pp. 399–414, 2017.

[24] J. R. Medina, H. Börner, S. Endo, and S. Hirche, "Impedance-based gaussian processes for modeling human motor behavior in physical and non-physical interaction," *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 9, pp. 2499–2511, 2019.

[25] S. Cremer, S. Das, I. Wijayasinghe, D. Popa, and F. Lewis, "Model-free online neuroadaptive controller with intent estimation for physical human–robot interaction," *IEEE Transactions on Robotics*, vol. 36, p. 240–253, 2020.

[26] I. Ranatunga, S. Cremer, D. Popa, and F. Lewis, "Intent aware adaptive admittance control for physical human-robot interaction," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, p. 5635–5640.

[27] M. Saadatzi, S. Abubakar, S. Das, M. Saadatzi, and D. Popa, "Neuroadaptive controller for physical interaction with an omni-directional mobile nurse assistant robot," in *Volume 10: 44th Mechanisms and Robotics Conference (MR)*, 2020.