# Nonlinear Model Predictive Control for Mobile Medical Robot using Neural Optimization

Yingbai Hu[1], Hang Su[2], Junling Fu[2], Hamid Reza Karimi[3], *Senior Member, IEEE*, Giancarlo Ferrigno[2], *Senior Member, IEEE*, Elena De Momi[2], *Senior Member, IEEE* and Alois Knoll[1], *Senior Member, IEEE*

***Abstract*—Mobile medical robots have been widely used in various structured scenarios, such as hospital drug delivery, public area disinfection, and medical examinations. Considering the challenge of environment modeling and controller design, how to achieve the information from the human demonstration in a structured environment directly arouse our interests. Learning skills is a powerful way that can reduce the complexity of algorithm in searching space. This is especially true when naturally acquiring new skills, as mobile medical robot must learn from the interaction with a human being or the environment with limited programming effort. In this paper, a learning scheme with nonlinear model predictive control (NMPC) is proposed for mobile robot path tracking. The learning-by-imitation system consists of two levels of hierarchy: in the first level, a multi-virtual spring-dampers system is presented for imitation of the mobile robot's trajectories; and in the second level, the NMPC method is used in the motion control system. The NMPC strategy utilizes a varying-parameter one-layer projection neural network to solve an online quadratic programming optimization via iteration over a limited receding horizon. The proposed algorithm is evaluated on a mobile medical robot with an emulated trajectory in simulation and three scenarios used in the experiment.**

***Index Terms*—Imitation Learning, Multi Virtual Spring-Dampers, Model Predictive Control, Varying-Parameter One-Layer Projection Neural Network.**

## I. Introduction

Over the past decade, applications of mobile medical robots have attracted attention in our daily life. It is a powerful assistant platform that helps to reduce the workloads of operators [1]. Besides, mobile medical robots have unparalleled advantages regarding quarantine infectious disease scenarios, such as SARS and COVID-19, and thus can significantly reduce the infection rate [2]. How to control the mobile medical robot in a structured environment with a simple way and low computational complexity is a challenging issue [3].

Machine learning-based robotic system has been a popular research theme. With limited programming effort and almost no *ad hoc* modeling required, learning-by-imitation serves as a promising paradigm by which an agent learns a control policy for a complex system via observing demonstration in a dynamic environment. Learning-by-imitation consists of a demonstrating expert that provides examples of the optimal behaviors, and a learning agent reproduces similar behaviors by imitating the expert [4]. This paradigm, because of its promising prospect, has been widely used by a variety of robots such as service robots and industrial robots in our daily life [5], [6]. The development of learning methodology has become an important tool in robotics [7]. In [8], the authors proposed the Dynamic Movement Primitives (DMP) algorithm fused with machine learning methods to learn desired trajectories from the demonstration successfully. In [7], a DMP-based reinforcement learning framework is proposed for learning pick-and-place tasks to improve the robustness of robot movements. These works demonstrate the power of the motion encoding process on humanoid-like manipulators and inspire our interests in combining a better encoding algorithm with machine learning methods to further improve motor-skills learning with a mobile robot. In this paper, a novel multi virtual spring-dampers method (MVSD) consisting of two VSD systems is proposed to learn motor skills, which has an extra radial VSD system and better convergence ability to the attraction factor point. Therefore, it not only inherits the characteristics of DMP but also shows a stronger learning ability with lower mean squared error (MSE).

In [9] and [10], the robot learning paradigm is utilized for manipulators or human-like robots. However, none of these works focus on the mobile robot. Motivated by [11] and [12], we propose a novel motor-skills learning system involving demonstration of imitation learning and the motion control for mobile robots.

For a mobile robot control, the traditional nonlinear sliding mode control is proposed for quickly switching to an inverted pendulum mode with a small turning radius [13]. However, physical constraints have not been considered when using these traditional control methods, such as velocity limits and velocity incremental limits, which hampers the full exhibition of the imitation-learning power of mobile robots. In [14], linear MPC is employed for the mobile robot model considering the obstacles. However, the linear model is inadequate to describe the object model. Due to the properties of the nonlinear model predictive control (NMPC) algorithm such as input-output non-linearity along a limited horizon [15], NMPC is widely applied to path tracking of the mobile robot under kinematic constraints [16], [17]. Moreover, it makes sure that the control sequences are limited physically, which is important for the motion control of a mobile robot [18].

*This work was supported by the Human Brain Project SGA2, under the Specific Grant Agreement No. 785907. *Corresponding author: Hang Su.*

[1]Yingba Hu and Alois Knoll are with the Department of Informatics, Technical University of Munich, Munich, 85748, Germany (e-mail: yingbai.hu@tum.de, knoll@in.tum.de).

[2]Hang Su, Junling Fu, Elena De Momi and Giancarlo Ferrigno are with the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, 20133, Milano, Italy (e-mail: hang.su@polimi.it; junling.fu@polimi.it; giancarlo.ferrigno@polimi.it; elena.demomi@polimi.it).

[3]Hamid Reza Karimi is with the Department of Mechanical Engineering, Politecnico di Milano, 20133, Milano, Italy (e-mail: hamidreza.karimi@polimi.it).

However, NMPC suffers from computational complexity as it highly relies on numerical optimization. In [19], a proposed neural-dynamic MPC method prevented some major pitfalls drew by conventional numerical optimization of NMPC, with advantages of global convergence and computational simplicity. Actually, neural network methods can solve real-time optimization problem better than numerical methods under physical conditions [20], [21]. In [18] and [22], a projection neural network (PNN) method and a primal-dual neural network (PDNN) were presented, respectively. Both methods have achieved good performance in solving quadratic programming (QP) problems. However, the convergence rate cannot be further improved for some special cases.

Inspired by the aforementioned works, a novel neural network method called the varying-parameter one-layer projection neural network (VP-OneLPNN) method is proposed to solve QP problems of MPC in this paper. Compared with the PNN and PDNN methods, the proposed VP-OneLPNN model has only one layer, the least number of neurons, and the fastest convergence rate which can ensure optimal solutions for real-time QP problems.

The main objective of this paper is to find a simple and effective method for the robot's repetitive movement, especially passing through a structured environment like a human. Therefore, a learning-control based scheme is designed for a medical mobile robot, which can imitate the motor skills from human demonstration. Thus, it can reduce the complexity of the control algorithm in searching space and guarantees that a mobile robot can run in a structured environment without complicated path planning, such as moving in a hospital. The whole framework that integrates the proposed demonstration learning component and motion control method, consists of two phases: the learning phase and the motion control phase.

In the former phase, MVSD is employed to model and recreate complicated trajectories, with the nonlinear term of MVSD. In the latter phase, VP-OneLPNN based MPC method is applied to manipulate the mobile robot in line with the reproduced trajectory. The scheme of learning-based MPC for mobile robot from human demonstrations is shown in Fig. 1.

In summary, our main contribution is to develop an integrated learning-by-imitation system that includes:

1) A novel MVSD approach is proposed for trajectory learning of a mobile robot. This method has good learning performance to ensure high precision of imitation.
2) A new VP-OneLPNN algorithm is presented for QP optimization in MPC, in which all state variables can converge rapidly to the optimal one with physical constraints. Compared with the traditional methods, the proposed VP-OneLPNN model has only one layer, the least number of neurons, and the fastest convergence rate.
3) The learning system is applied to mobile robots for complex motor skills learning from demonstration, and it's suitable for human-machine interaction in the environment, such as delivering food or medicine to patients.

The remainder of this paper is organized as follows. The imitation learning scheme with our proposed MVSD model is presented in Section II. The kinematic model of the mobile robot and the NMPC system are demonstrated in Section III. The VP-OneLPNN method aims at real-time QP problem-solving in MPC is introduced in Section IV. Simulation and experimental results are conducted and discussed in Section V and Section VI, respectively. Finally, Section VII concludes this paper.

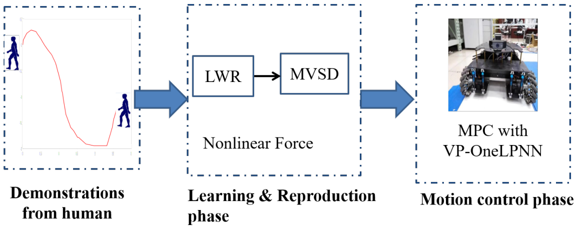


Fig. 1. The scheme of learning-based MPC for the mobile robot from human demonstrations.

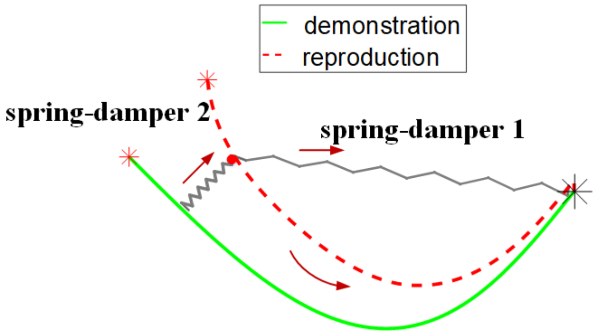


Fig. 2. The model of Multi Virtual Spring-Dampers.

## II. Imitation Learning with Multi-Virtual Spring-Dampers Scheme

MVSD is a novel second-order dynamical system involving two virtual spring-dampers (VSD) modules, defined as $vsd1$, $vsd2$. A complex dynamical system can be represented by MVSD fusion with a nonlinear forcing term. Hence, the dynamical system of MVSD is modeled as,

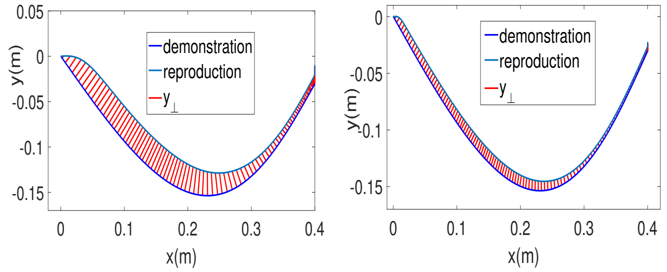
$$\ddot{y}_t = K^{p1}\left(g - y_t\right) - K^{v1}\dot{y}_t + K^{p2}(y^{\perp} - y_t) - K^{v2}\dot{y}_t + F_t \quad (1)$$

where $K^{p1}$, $K^{p2}$ and $K^{v1}$, $K^{v2}$ are the corresponding stiffness matrices, damping matrices for $vsd1$ and $vsd2$; $F_t$ is the nonlinear forcing term of a dynamical system which can be approximated with supervised learning, such as locally weighted regression (LWR); $y_t, \dot{y}_t, \ddot{y}_t$ denote the position, velocity, and acceleration generated by MVSD, respectively; $g$ is the attractor point; $y_{\perp}$ represents the minimum distance from the end-effector to the reference path. The structure of MVSD is shown in Fig. 2.

$$F_t = s_t f_t \quad (2)$$

$$\dot{s}_t = \alpha_s s_t \quad (3)$$

The equation (2), as the nonlinear item, can be employed to model complicated trajectories. The equation (3) represents the canonical system of DMP, where $\alpha_s$ is the constant value and $s_t$ is the asymptotic decay from the initial value to zero.



(a) $K^{p2} = 10$ for MVSD imitation. (b) $K^{p2} = 50$ for MVSD imitation

Fig. 3. Imitation results of different $K^{p2}$ for MVSD.

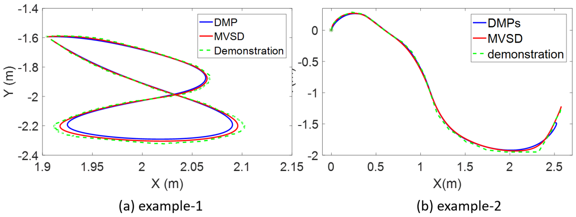


Fig. 4. Comparison between DMP and MVSD.

By multiply the canonical system, $F_t$ will converge to zero ultimately, which indicates that the MVSD model will degrade into a linear multi spring-dampers system. To evaluate the performance of the trajectory estimation, the index called Mean Squared Error (MSE) is adopted, which is denoted as:

$$MSE = \frac{1}{N_{len}} \sum_{i=1}^{N_{len}} \left(y^i_{learn} - y^i_{demon}\right)^2.$$

where $y_{learn}$ denotes the reproduction trajectory by learning; $y_{demon}$ is the collection trajectory from human demonstration; $N_{len}$ is the length of $y_{demon}$. The imitation results with different $K^p$ of MVSD are shown in Fig. 3. In Fig. 4, MSE of Example-1 for MVSD and DMP are 0.0071 and 0.013, respectively, and MSE of Example-2 for MVSD and DMP are 0.0038 and 0.0636, respectively, which indicates MVSD achieves better performance than DMP. Moreover, trajectories with MVSD can converge to the end attractor $g$ even with the different initial positions are shown in Fig. 5.

**Remark 1.** *MVSD is a novel second-order dynamical system involving two virtual spring dampers modules which has an extra radial VSD modules. There exist two forces acting on the moving object: one acting towards the goal point and the other towards the immediate trajectory point. However, DMP has only one virtual spring dampers. The learning results are shown in Fig. 4, which shows that MVSD has a lower MSE. Therefore, MVSD has a better convergence performance to the desired trajectory than DMP.*

## III. Model Predictive Control Scheme

In this part, a neural network-based MPC control scheme is presented. The proposed neural network methods can optimize the MPC problem on line for the mobile-robot system with physical constraints.

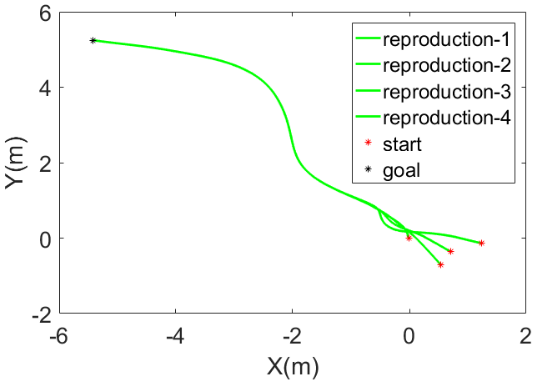


Fig. 5. Imitation learning from random initial position. The trajectory can converge to the goal $g$ from the different initial positions.

### A. Mobile Robot Control System

The kinematic model of the two-wheeled mobile robot is depicted in Fig. 6. Considering the relation between the robot velocity $v$ and the two-driving-wheel velocity $(v_l,\ v_r)$, we describe the robot velocity $v = (v_l + v_r)/2$ and the angle velocity $\omega = (v_l - v_r)/B$ respectively, where $B$ is the distance between the wheels. The kinematic model of the mobile robot is described as follows:

$$\dot{X} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} u \tag{4}$$

where $(x, y)$ is the Cartesian coordinates of the mobile robot, and $\theta$ donates the orientation angle; $u = (v, \omega)^T$ represents the control input, with the state vector $X$. Based on the representation of (4), the kinematic model of the reference trajectory can be acquired with state vector $X_r = (x_r, y_r, \theta_r)^T$ and control input $u_r = (v_r, \omega_r)^T$, which is expressed as:

$$\dot{X}_r = \begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} v_r\cos\theta_r \\ v_r\sin\theta_r \\ \omega_r \end{bmatrix} = \begin{bmatrix} \cos\theta_r & 0 \\ \sin\theta_r & 0 \\ 0 & 1 \end{bmatrix} u_r \tag{5}$$

Consequently, the kinematic errors of the mobile robot can be computed as:

$$X_e = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \tag{6}$$

where $X_e = [x_e, y_e, \theta_e]^T$. According to the error state in (6), the derivative of $X_e$ can be expressed as follows:

$$\begin{cases} \dot{x}_e = \omega y_e - v + v_r\cos\theta_e \\ \dot{y}_e = -\omega x_e + v_r\sin\theta_e \\ \dot{\theta}_e = \omega_r - \omega \end{cases} \tag{7}$$

Then, the control input is reformulated as:

$$u_e = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} v_r\cos\theta_e - v \\ \omega_r - \omega \end{bmatrix} \tag{8}$$

Substituting (8) into (7), one can rewrite the kinematic model for $X_e$ as

$$\dot{X}_e = \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} 0 & \omega & 0 \\ -\omega & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} + \begin{bmatrix} u_1 \\ v_r\sin\theta_e \\ u_2 \end{bmatrix} \tag{9}$$

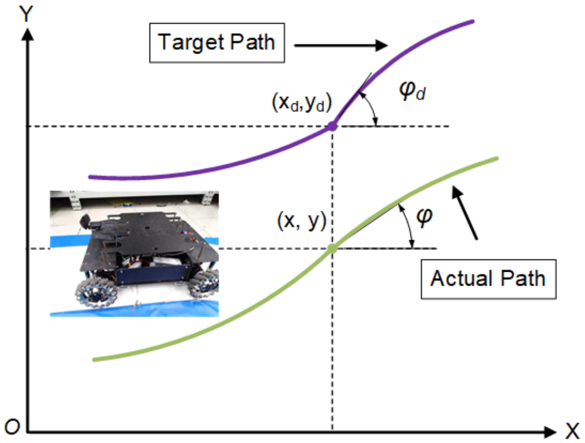


Fig. 6. The kinematic scheme of the two-wheeled mobile robot.

Reshaping the equation (9), one may obtain:

$$\dot{\bar{X}}_e = \begin{bmatrix} 0 & \omega & 0 \\ -\omega & 0 & v_r \frac{\sin\theta_e}{\theta_e} \\ 0 & 0 & 0 \end{bmatrix} \bar{X}_e + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} u_e. \quad (10)$$

where $\bar{X}_e = [x_e, y_e, \theta_e]^T$. Ultimately, a stabilization issue in the nonlinear affine system of (10) is transformed from the motion control issue with the mobile robot in (4).

### B. Nonlinear Model Predictive Control

Considered a closed-loop optimal control matter along with the finite-time domain, the kinematic equation can be defined as the following nonlinear affine system:

$$X\left(j+1\right) = \psi_1\left(X(j)\right) + \psi_2\left(X(j)\right)u(j) \quad (11)$$

where $\psi_1(\cdot)$ and $\psi_2(\cdot)$ are the continuous functions with the input vector $u$ and the state vector $X$ as follows:

$$X(j) \in R^X, j = 1, 2, \ldots, N,$$
$$u(j) \in R^U, j = 1, 2, \ldots, N_u.$$

For the nonlinear affine system (11), both $\psi_1(.)$ and $\psi_2(.)$ symbolize nonlinear continuous functions [23], [24], with initial constraints as $\psi_1(0) = 0$ and $1 \le N \le N_u$, which marks the estimation horizon.

For each sampling interval, the solution of the optimal input vector can be obtained via online optimization using a cost function, and the NMPC method will accordingly be formed through the iterative solution to the optimal control problem. Therefore, the cost function is defined as:

$$S(X,u) = \sum_{i=j}^{j+N-1} L_1\left(X(i), u(i)\right) + L_2\left(X(j+N)\right) \quad (12)$$

where $L_1(X,u)$ is the immediate cost which satisfies the $L_1(0,0) = 0$; $L_2(X(j+N))$ is the terminal cost. Besides, the cost function $S(X,u)$ can also be expressed in the following quadratic form:

$$S(j) = \sum_{i=1}^{N} \|X\left(j+i|j\right)\|_Q^2 + \sum_{i=0}^{N_u-1} \|\Delta u\left(j+i|j\right)\|_R^2 \quad (13)$$

where $X(j+i|j)$ is the following horizontal state to be estimated; the increment of the input vector is formulated as $\Delta u(j+i|j) = u(j+i|j) - u(j-1+i|j)$; both the constant parameters $R$ and $Q$ represent the designed weight matrices; $\|X\| = \sqrt{X^T X}$ denotes as its Euclidean norm based on the related vector. Then, we reshape the formulation (11):

$$\bar{X}_e\left(j+1\right) = \psi_1\left(\bar{X}_e(j)\right) + \psi_2\left(\bar{X}_e(j)\right)u_e(j) \quad (14)$$

$$\text{s.t.} \qquad u^- \le u_e(j) \le u^+ \quad (15)$$
$$\Delta u^- \le \Delta u(j) \le \Delta u^+ \quad (16)$$
$$X^- \le \bar{X}_e(j) \le X^+ \quad (17)$$

with

$$\psi_1\left(\bar{X}_e\right) = \begin{bmatrix} 1 & \omega T & 0 \\ -\omega T & 1 & v_r \frac{\sin x_3}{x_3} T \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$
$$\psi_2\left(\bar{X}_e\right) = T \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$$

where $\bar{X}_e = [x_1, x_2, x_3]^T = [x_e, y_e, \theta_e]^T$ denotes the state vector with the sampling interval $T$; the input vector is described as $u_e = [u_1, u_2]^T$ with $u_1 = v_r cos\theta_e - v$ and $u_2 = \omega_r - \omega$; $(u^-, u^+)$, $(\Delta u^-, \Delta u^+)$ and $(x^-, x^+)$ are respectively corresponding to the lower and upper boundaries of the input variable, the input increment variable, and the state variable.

Some vectors are introduced for a compact presentation of the QP problem in online optimization:

$$\bar{X} = \left[\bar{X}_e(j+1|j), ..., \bar{X}_e(j+N-1|j)\right]^T \quad (18)$$
$$\bar{u}(j) = \left[u_e(j|j), ..., u_e(j+N_u-1|j)\right]^T \quad (19)$$
$$\Delta\bar{u}(j) = \left[\Delta u_e(j|j), ..., \Delta u_e(j+N_u-1|j)\right]^T \quad (20)$$

Based on (14) and (18)–(20), the output is estimated as:

$$\bar{X}(j) = A\Delta\bar{u}(j) + \hat{\psi}_1 + \hat{\psi}_2 \quad (21)$$

where

$$A = \begin{bmatrix} \psi_2\left(\bar{X}_e(j|j-1)\right) & \cdots & 0 \\ \psi_2\left(\bar{X}_e(j+1|j-1)\right) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \psi_2\left(\bar{X}_e(j+N-1|j-1)\right) & \cdots & \psi_2\left(\bar{X}_e(j+N-1|j-1)\right) \end{bmatrix}$$

$$\hat{\psi}_1 = \begin{bmatrix} \psi_1\left(\bar{X}_e(j|j-1)\right) \\ \psi_1\left(\bar{X}_e(j+1|j-1)\right) \\ \vdots \\ \psi_1\left(\bar{X}_e(j+N-1|j-1)\right) \end{bmatrix}$$

$$\hat{\psi}_2 = \begin{bmatrix} \psi_2\left(\bar{X}_e(j|j-1)u(j-1)\right) \\ \psi_2\left(\bar{X}_e(j+1|j-1)u(j-1)\right) \\ \vdots \\ \psi_2\left(\bar{X}_e(j+N-1|j-1)u(j-1)\right) \end{bmatrix}$$

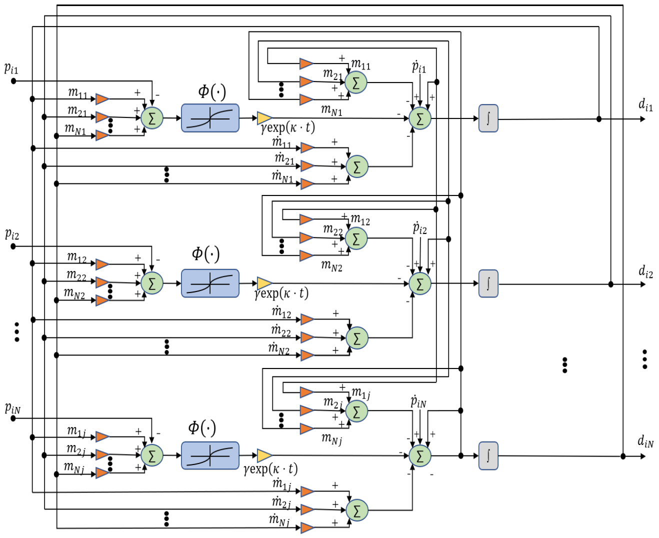


Fig. 7. Structure of VP-OneLPNN, where $p_i = (c + \xi \nabla G(d))_i$.

with $A \in R^{3N \times 2N_u}$, $\hat{\psi}_1 \in R^{3N}$ and $\hat{\psi}_2 \in R^{3N}$. Afterwards, one can redescribe the optimization problem of (13) as:

$$\min \quad \left\| A\Delta\bar{u}(j) + \hat{\psi}_1 + \hat{\psi}_2 \right\|_Q^2 + \|\Delta u(j)\|_R^2 \tag{22}$$

$$\text{s.t.} \quad \Delta\bar{u}^- \leq \Delta\bar{u}(j+1) \leq \Delta\bar{u}^+ \tag{23}$$

$$\bar{u}^- \leq \bar{u}(j-1) \leq \bar{u}^+ \tag{24}$$

$$\bar{u}^- \leq \bar{u}(j-1) + H\Delta\bar{u}(j) \leq \bar{u}^+ \tag{25}$$

$$X^- \leq \hat{\psi}_1 + \hat{\psi}_2 + A\Delta\bar{u}(j) \leq X^+ \tag{26}$$

where $H = \begin{bmatrix} I & 0 & 0 & 0 \\ I & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I & I & \cdots & I \end{bmatrix} \in R^{2N_u \times 2N_u}$.

Furthermore, a QP problem can be reformulated from the optimization issue of the equations (22)-(26):

$$\min \quad \tfrac{1}{2}\Delta\bar{u}^T M \Delta\bar{u} + c^T \Delta\bar{u} \tag{27}$$

$$\text{s.t.} \quad E\Delta\bar{u} \leq r \tag{28}$$

with

$$M = 2(A^T Q A + R) \in R^{2N_u \times 2N_u}$$

$$c = -2A^T Q(\hat{\psi}_2 + \hat{\psi}_1) \in R^{2N_u}$$

$$\mathrm{E} = \begin{bmatrix} -H \\ H \\ -A \\ A \\ I \\ -I \end{bmatrix} \in R^{n_1}, r = \begin{bmatrix} -u^- + u(j-1) \\ u^+ - u(j-1) \\ -X^- + \hat{\psi}_1 + \hat{\psi}_2 \\ X^+ - \hat{\psi}_1 - \hat{\psi}_2 \\ \Delta\bar{u}^+ \\ \Delta\bar{u}^- \end{bmatrix} \in R^{n_2}$$

where $n_1 = (6N_u + 6N) \times 2N_u$ and $n_2 = 6N_u + 6N$.

## IV. Varying-Parameter One-Layer Projection Neural Network (VP-OneLPNN)

In this section, the QP problem is supposed to be rewritten as:

$$\min \quad \tfrac{1}{2}d^T M d + c^T d \tag{29}$$

$$\text{s.t.} \quad g(d) = Ed - r \leq 0 \tag{30}$$

with $d = \Delta\bar{u}$. Considering the Lagrangian function:

$$L(d, l) = \frac{1}{2}\Delta\bar{u}^T M \Delta\bar{u} + c^T \Delta\bar{u} + l^T g(d) \tag{31}$$

According to Karush-Kuhn-Tucker (KKT) conditions in [25], if $d^*$ is a solution to (29)–(30), then there exists $l^* \in R^m$ such that $(d^*, l^*)^T$ satisfies the following conditions:

$$Md^* + c + \sum\nolimits_{i=1}^{6Nu+6N} l_i^* \nabla g_i(d^*) = 0 \tag{32}$$

$$l^* \geq 0 \tag{33}$$

$$l_i^* g_i(d^*) = 0 \tag{34}$$

The equivalent equation of (32)–(33) is expressed as [26]:

$$Md + c + \nabla G(d)\, l = 0 \tag{35}$$

$$G(d) = \sum\nolimits_{i=1}^{6Nu+6N} \max\{0, g_i(d)\}$$

Similar to [27], a penalty function is defined for the problem (29)–(30) as follows:

$$E(d) = \frac{1}{2}d^T M d + c^T d + \xi G(d) \tag{36}$$

where $\xi > 0$ is a penalty parameter. We can further obtain the gradient of (36):

$$\partial E(d) = Md + c + \xi \nabla G(d) \tag{37}$$

Actually, the residual errors in (37) exist in practice, so we define the following error function:

$$e(t) = Md + c + \xi \nabla G(d) \tag{38}$$

To ensure that the error converges to zero, a negative time derivative of (38) is defined as:

$$\dot{e}(t) = -\gamma \exp(\kappa \cdot t)\, \Phi(e(t)) \tag{39}$$

where $\Phi(e(t))$ is the active function; $\kappa$ and $\gamma$ are positive parameters that are employed to scale the convergence speed. The error in (39) is converged to zero. The active function is defined as:

$$\Phi(e_i(t)) = \frac{(1 + \exp(-\beta))(1 - \exp(-\beta e_i(t)))}{(1 - \exp(-\beta))(1 + \exp(-\beta e_i(t)))}$$

where $\beta \geq 2$ which makes $0 \leq |e_i(t)| \leq 1$.

The extended form of the (39) is expressed as:

$$\begin{aligned} &\dot{M}d + M\dot{d} + \dot{c} + \xi \nabla \dot{G}(d) \\ &\quad = -\gamma \exp(\kappa \cdot t)\, \Phi\{Md + c + \xi \nabla G(d)\} \end{aligned} \tag{40}$$

Then, a new neural network called VP-OneLPNN is proposed to solve the QP problem in (29)–(30) as follows:

$$\begin{aligned} M\dot{d} = &-\left(\dot{M}d + \dot{c} + \xi \nabla \dot{G}(d)\right) \\ &- \gamma \exp(\kappa \cdot t)\, \Phi\{Md + c + \xi \nabla G(d)\} \end{aligned} \tag{41}$$

The equation can be further reformulated in the following form:

$$\begin{aligned} \dot{d} = (I - M)\,\dot{d} &- \left(\dot{M}d + \dot{c} + \xi \nabla \dot{G}(d)\right) \\ &- \gamma \exp(\kappa \cdot t)\, \Phi\{Md + c + \xi \nabla G(d)\} \end{aligned} \tag{42}$$

The structure of VP-OneLPNN for the QP problem in MPC is shown in Fig. 7. For comparison purposes, the traditional

TABLE 1
COMPARISON OF DIFFERENT NEURAL NETWORK MODELS

| Neural network type | number of layers | number of neurons | number of connections | objective function |
|---|---|---|---|---|
| RNN [18] | 2 | $n+m$ | $n^2+2mn$ | strictly convex |
| PDNN [22] | 2 | $n+m$ | $3n^2+3mn$ | convex |
| The proposed | 1 | $n$ | $2n^2$ | convex |

gradient descent-based recurrent neural network is denoted as [28],

$$\dot{d} = -\gamma\left\{Md + c + \xi\nabla G\left(d\right)\right\} \tag{43}$$

Compared with other neural network models, the VP-OneLPNN has the fastest convergence rate and involves the least neurons and layers in the structure shown in Table.1.

**Remark 2.** *The proposed VP-OneLPNN has the simplest network structure and least neurons and layers among other models in Table.1. However, the proposed method has slower computational efficiency, because the equation in (41) involves more matrix operations, including $\dot{M}$, $\dot{l}$, etc. Noted that, the proposed method only involves matrix operations, which still satisfy the requirements of computational efficiency.*

## V. SIMULATION

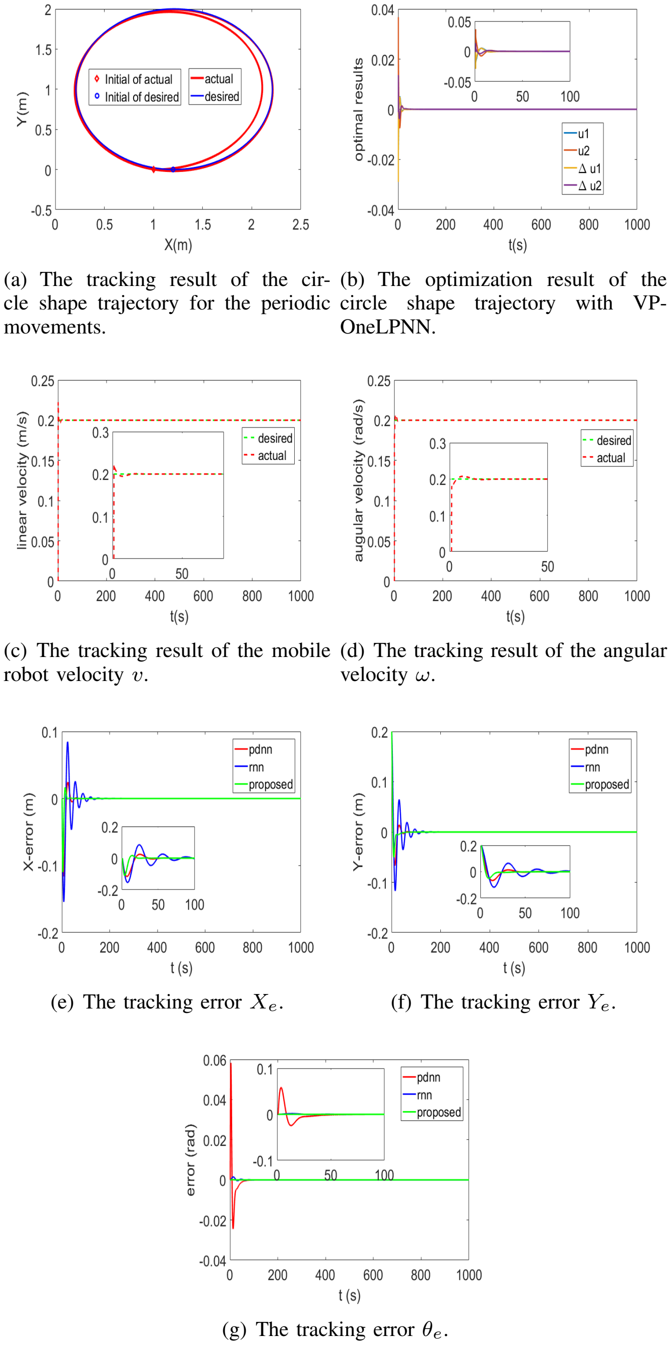
In this part, the proposed VP-OneLPNN based MPC scheme is demonstrated in simulation. To demonstrate the validity of the proposed algorithm, two different neural networks are also tested and the experimental results are compared with the proposed one. One kind of scenario is built based on a periodic-moving mobile robot with constant linear and angular velocity for the circle-shape movement. In the simulation experiment, we design the initial position and orientation of the mobile robot that is different from the desired trajectory, which is convenient for us to observe the convergence performance of the proposed method. The robot is supposed to track the trajectories with the VP-OneLPNN based MPC method in the scenario. We analyze the result in terms of tracking errors and convergence. The sampling period is $\Delta t = 0.1s$. The details of the parameters refer to Table.2.

### *A. Circle-Shape Movements with Constant Velocity for Robot*

To track the circle-shape trajectory within the periodic motion, as the task for the mobile robot, in this case, is supposed to be executed with constant linear and angular velocities. The initial position and orientation of the mobile robot are randomly chosen but different from the ideal trajectory.

The linear and angular velocities are respectively set to $v = 0.2m/s, w = 0.2rad/s$ as reference values. As for the initial position of the reference trajectory, it refers to $X_r(0) = [1.2; 0; 0]$, while that of the mobile robot denotes as $X(0) = [1; 0; 0]$.

As shown in Fig. 8(a), it's clear that the trajectories of the mobile robot rapidly converge to the reference one, despite the mobile-robot's initial positions. The optimization result of the circle-shape trajectory is shown in Fig. 8(b), where all the state variables quickly reach optimal solutions. As displayed in Figs. 8(c)–8(d), the linear and angular velocities of the



(a) The tracking result of the circle shape trajectory for the periodic movements.

(b) The optimization result of the circle shape trajectory with VP-OneLPNN.

(c) The tracking result of the mobile robot velocity $v$.

(d) The tracking result of the angular velocity $\omega$.

(e) The tracking error $X_e$.

(f) The tracking error $Y_e$.

(g) The tracking error $\theta_e$.

Fig. 8. The simulation result of the circle shape trajectory using NMPC based on VP-OneLPNN.

TABLE 2
PARAMETERS OF THE LEARNING-CONTROL SYSTEM

| Name | Parameters |
|---|---|
| MVSD | $K^{p1} = diag[50, 50], K^{v1} = diag[10, 10], K^{p2} = diag[10, 10], K^{v2} = diag[\sqrt{20}, \sqrt{20}]$ |
| MPC | $N = 3, N_u = 2, Q = I, R = 2I$ |
| VP-OneLPNN | $\kappa = 0.1, \gamma = 10^2$ |
| Dimension of the robot movements | $D = 2$ |

mobile robot can approach the respective reference ones quite well. The comparison results of the recurrent neural network, the primal-dual neural network, and our proposed method are shown in Figs. 8(e)–8(g), where all the tracking errors of the mobile robot can converge to zero, but the proposed method has the fastest convergence rate.

### B. Discussion

Successful trajectory-tracking execution of the mobile robot is guaranteed for the circle-shape trajectory scenario. The rapid convergence of the proposed method is quite significant for a mobile robot, especially for the real-time optimization of the mobile robot. The optimal solution to the trajectory-learning problem can be performed by the mobile robot regardless of variable position and orientation initializations. Besides, the effect and speed of the tracking-errors convergence are satisfactory. The effectiveness and robustness of our proposed technique can be demonstrated by the aforementioned results. Furthermore, from the comparison results, it is observed that the proposed VP-OneLPNN based MPC control scheme has an overall promising performance for real-time robot running, in terms of convergence speed, and tracking errors.

## VI. EXPERIMENTAL DEMONSTRATION

In this section, the proposed learning-by-imitation scheme is tested using real-world trajectories that are collected by the Xsens mti-310 sensor. We build three scenarios to explore the mobile-robot learning skills abilities including the trajectory generation from a demonstration by human and trajectory tracking with VP-OneLPNN based MPC. The parameter settings of MVSD and VP-OneLPNN based MPC can refer to Table.2. The experimental procedures are listed as:

- The human holds the Xsens mti-310 sensor and walk through the structured environment, where the trajectories are collected.
- Then, the motor skills are modeled and learned by MVSD.
- Finally, the robot is driven to perform the learning skills in a structured environment using neuro-dynamic optimization.

In our experiments, the tasks in different scenarios vary from simple to complex, involving to learn the straight line-shape, ’S’-shape, and ’U’-shape trajectories from human demonstration, which are shown in Fig. 9.

### A. Learning the Straight Path Tracking

In the first experiment, the mobile robot is expected to learn a simple task of a straight line-shape path tracking from demonstration, which is shown in Fig. 10. In this scenario, a human subject walks along a straight path. There are no obstacles placed on any side along the path. The human subject walks along the path repeatedly and the trajectory of the human subject is collected with an Xsens mti-310 sensor. This scenario is quite simple since the mobile robot only needs to learn how to move in the forward direction.

### B. Learning the ’S’-Shape Path Tracking

Then in the second experiment, it’s supposed for the mobile robot to learn a relatively complicated task of an ’S’-shape path tracking, as Fig. 11 exhibits. In this scenario, a human subject walks along an ’S’-shape path. We randomly set obstacles on the two sides along with the ’S’-shape path (’◇’ denotes the obstacle). The human subject is instructed to follow the path without any collision with obstacles and repeats walking along the path to obtain several trajectories. The robot’s task in this scenario is more difficult than Scenario VI-A as the mobile robot needs to learn to go forward, turn left and right.

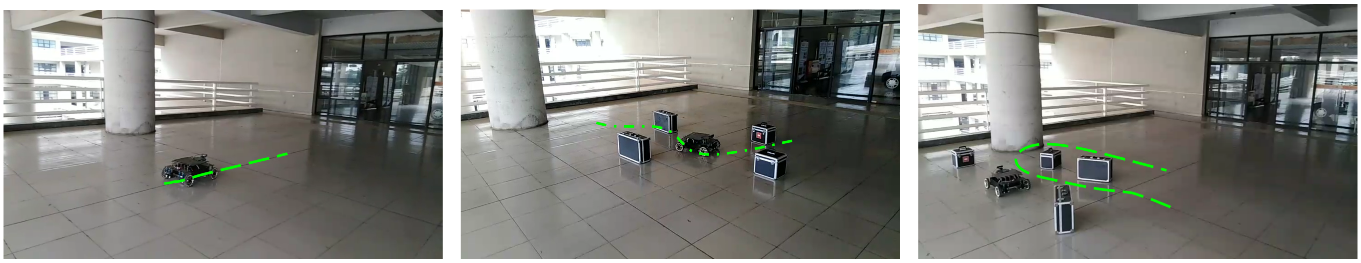
### C. Learning the ’U’-Turn Path Tracking

As for the last experiment, it is essential for the mobile robot to learn the more complicated task of a ’U’-shape path tracking based on demonstrations, as Fig. 12 shows. In this scenario, a human subject walks along with a ’U’-turn path. We randomly drop obstacles on the two sides along with the ’U’-turn path again(’◇’ denotes the obstacle). The human subject is instructed to follow the path without any collision with obstacles. The robot’s task in this scenario is much more difficult than Scenario VI-B as the mobile robot needs to learn to make a ’U’-shape turn within quite confined space.

### D. Discussion

In Scenario VI-A, the learning results are shown in Fig. 10. The trajectory reproduced is plotted with a green line, while the tracking trajectory acquired with the proposed method is in red. It can be seen that both the linear and angular velocities coincide with the desired ones in Figs. 10(b) and 10(c). Moreover, the proposed method proves to be capable to quickly converge to the global optimal solution. Fig. 10(d) displays the trajectory tracking errors in Cartesian space. It shows that all the errors are limited to a small scope.

It is not surprising that we get similar results from Scenarios VI-B even the task is more challenging, and the experimental results are recorded in Fig. 11. In Scenario VI-B, the trajectory and the mobile robot’s skills are very complex. The results of the learning-based MPC system for an ’S’-shape path tracking are shown in Figs. 11(a)–11(d). At the motion control stage,

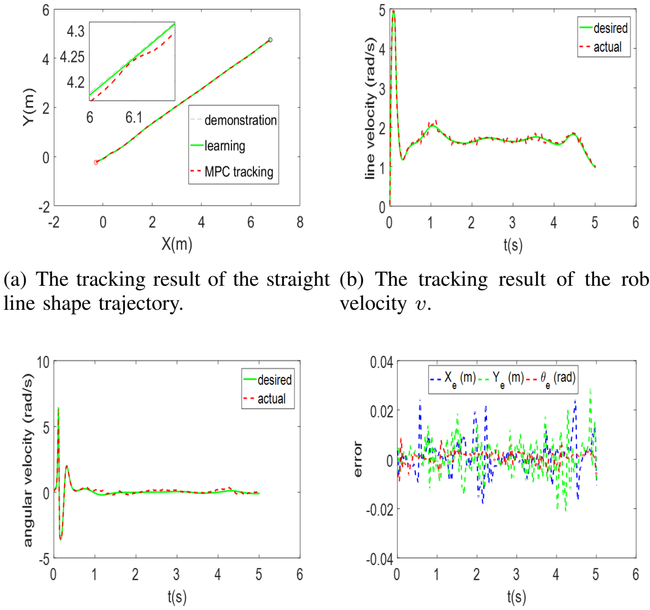


(a) Experiment-1: the line shape.

(b) Experiment-2: the 's' shape.

(c) Experiment-3: the 'u' shape.

Fig. 9. Experiment demonstration: from simple to complex scenarios.



(a) The tracking result of the straight line shape trajectory.

(b) The tracking result of the robot velocity $v$.

(c) The tracking result of the angular velocity $\omega$.

(d) The tracking errors $X_e$, $Y_e$ and $\theta_e$.

Fig. 10. The tracking result of the straight line-shape trajectory using NMPC based on VP-OneLPNN. The red '*' indicates the initial position, while the black '*' denotes the end attractor.

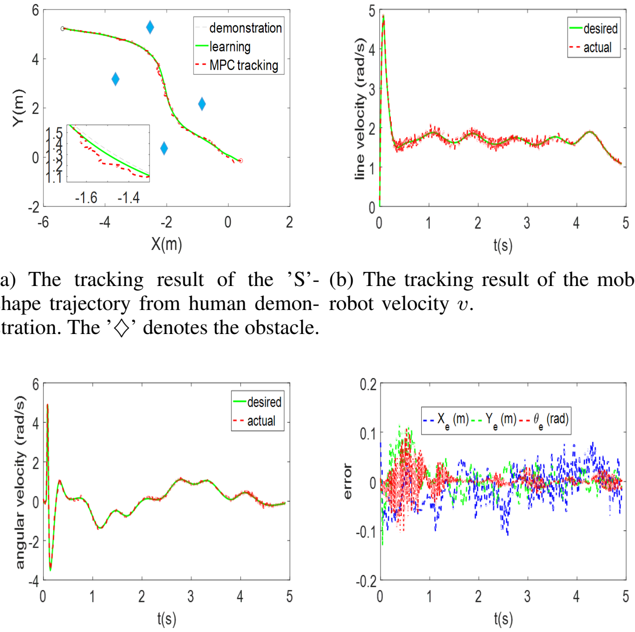
the mobile robot succeeds in crossing the path without any obstacle collision.

In Scenario VI-C, there is no obvious degradation of the overall performance compared with other scenarios. The results of the learning-based MPC system for the 'U'-turn path tracking are shown in Figs. 12(a)-12(d). Experimental results show that our learning-by-imitation system is independent of any specific scenes, and it is a powerful way to provide a mobile robot with motor skills in a natural way.

## VII. Conclusion

In this paper, the Multi-Virtual Spring-Dampers method is presented to reproduce the trajectory from the demonstration. The proposed imitation learning strategy with MVSD can promote motor-skills learning for the mobile robots by imitation learning.

For trajectory tracking, the NMPC is used to control the mobile robot along with the finite-time domain under input



(a) The tracking result of the 'S'-shape trajectory from human demonstration. The '◇' denotes the obstacle.

(b) The tracking result of the mobile robot velocity $v$.

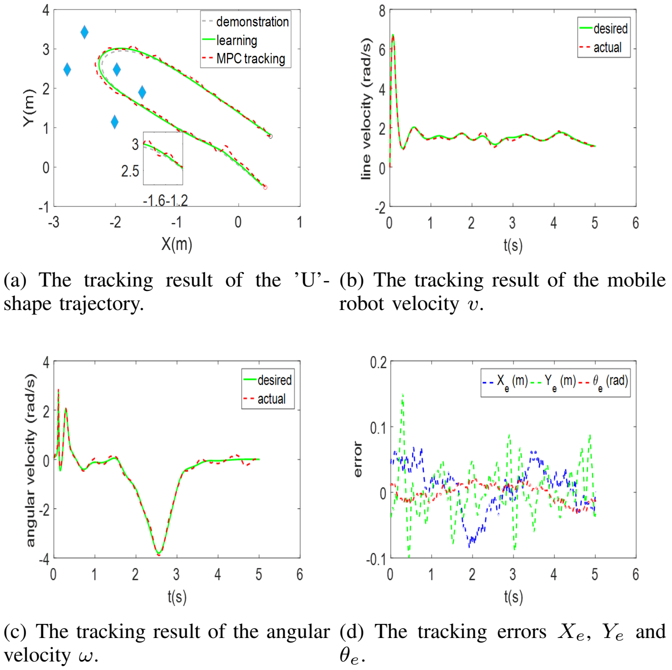
(c) The tracking result of the angular velocity $\omega$.

(d) The tracking errors $X_e$, $Y_e$ and $\theta_e$.

Fig. 11. The tracking result of the 'S'-shape trajectory using NMPC based on VP-OneLPNN. The red '*' indicates the initial position, while the black '*' denotes the end attractor.

and state limitations. The new network of VP-OneLPNN is also applied to the online QP optimization in the MPC. So, the mobile robot can cross obstacles from human demonstration and complete more complex tasks by using our method, which reduces the complexity of motion planning in a structured environment.

To test the proposed learning-by-imitation system of mobile robots, we build one challenging scenario in simulation and two real-world scenarios. The tasks of these scenarios are designed to examine the aforementioned learning abilities of motor skills. The results confirm that the mobile robot can learn the motor-skills from human demonstration with our method. To sum up, the mobile robot can successfully cross the path with no obstacle collision in all the test scenarios. Besides, small tracking errors and quick convergence to the global optimal solution emerge with the VP-OneLPNN based MPC. With the above characteristics, the proposed method is guaranteed to be a mighty and efficient solution to complex



(a) The tracking result of the 'U'-shape trajectory.

(b) The tracking result of the mobile robot velocity $v$.

(c) The tracking result of the angular velocity $\omega$.

(d) The tracking errors $X_e$, $Y_e$ and $\theta_e$.

Fig. 12. The tracking result of the 'U'-shape trajectory using NMPC based on VP-OneLPNN from human demonstration. The red '*' indicates the initial position, while the black '*' denotes the end attractor.

motion planning and control tasks.

Future works will involve different imitation learning methods, such as kernelized movement primitives (KMP) for dynamic obstacle avoidance. Additionally, we will explore the more complex model and reinforcement learning in MPC.

## References

[1] K. Zhang and Y. Shi, "Adaptive model predictive control for a class of constrained linear systems with parametric uncertainties," *Automatica*, vol. 117, p. 108974, 2020.

[2] M. Mariappan and B. Khoo, "Design and development of communication and control platform for medical tele-diagnosis robot (MTR)," *International Journal of Networks and Communications*, vol. 3, no. 1, pp. 12–20, 2013.

[3] S. G. Tzafestas, *Introduction to mobile robot control*. Elsevier, 2013.

[4] S. Chernova and A. L. Thomaz, "Robot learning from human teachers," *Synthesis Lectures on Artificial Intelligence & Machine Learning*, vol. 8, no. 3, pp. 1–121, 2014.

[5] H. Matsuzaki and G. Lindemann, "The autonomy-safety-paradox of service robotics in Europe and Japan: a comparative analysis," *Ai & Society*, vol. 31, no. 4, pp. 501–517, 2016.

[6] P. Tsarouchi, S. Makris, and G. Chryssolouris, "Human-robot interaction review and challenges on task planning and programming," *International Journal of Computer Integrated Manufacturing*, vol. 29, no. 8, pp. 916–931, 2016.

[7] F. Stulp, E. A. Theodorou, and S. Schaal, "Reinforcement learning with sequences of motion primitives for robust manipulation," *IEEE Transactions on robotics*, vol. 28, no. 6, pp. 1360–1370, 2012.

[8] S. Schaal, A. Ijspeert, and A. Billard, "Computational approaches to motor learning by imitation," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 358, no. 1431, pp. 537–47, 2003.

[9] H. Su, Y. Hu, H. R. Karimi, A. Knoll, G. Ferrigno, and E. De Momi, "Improved recurrent neural network-based manipulator control with remote center of motion constraints: Experimental results," *Neural Networks*, vol. 131, pp. 291–299, 2020.

[10] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 2, pp. 286–298, 2007.

[11] S. M. Khansari-Zadeh and A. Billard, "Bm: An iterative algorithm to learn stable non-linear dynamical systems with gaussian mixture models," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 2381–2388.

[12] A. Gams, T. Petrič, M. Do, B. Nemec, J. Morimoto, T. Asfour, and A. Ude, "Adaptation and coaching of periodic motion primitives through physical and visual interaction," *Robotics and Autonomous Systems*, vol. 75, pp. 340–351, 2016.

[13] H. Fukushima, K. Muro, and F. Matsuno, "Sliding-mode control for transformation to an inverted pendulum mode of a mobile robot with wheel-arms," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 7, pp. 4257–4266, 2014.

[14] A. Britzelmeier and M. Gerdts, "A nonsmooth newton method for linear model-predictive control in tracking tasks for a mobile robot with obstacle avoidance," *IEEE Control Systems Letters*, 2020.

[15] J. H. Lee, "Model predictive control: Review of the three decades of development," *International Journal of Control, Automation and Systems*, vol. 9, no. 3, p. 415, 2011.

[16] X. Mi, Y. Zou, S. Li, and H. R. Karimi, "Self-triggered DMPC design for cooperative multiagent systems," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 1, pp. 512–520, 2019.

[17] Y. Wei, Y. Wei, Y. Sun, H. Qi, and X.-Q. Guo, "Prediction horizons optimized nonlinear predictive control for pmsm position system," *IEEE Transactions on Industrial Electronics*, 2019.

[18] Y. Pan and J. Wang, "Model predictive control of unknown nonlinear dynamical systems based on recurrent neural networks," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 8, pp. 3089–3101, 2012.

[19] Z. Yan and J. Wang, "Model predictive control of nonlinear systems with unmodeled dynamics based on feedforward and recurrent neural networks," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 4, pp. 746–756, 2012.

[20] S. Li, L. Jin, and M. A. Mirza, "Projection neural networks for robot arm control," 2019.

[21] Y. Zhang, S. Chen, S. Li, and Z. Zhang, "Adaptive projection neural network for kinematic control of redundant manipulators with unknown physical parameters," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 6, pp. 4909–4920, 2017.

[22] C. Liu, D. Wang, Y. Zhang, and X. Meng, "Model predictive control for path following and roll stabilization of marine vessels based on neurodynamic optimization," *Ocean Engineering*, vol. 217, p. 107524, 2020.

[23] B. Jiang, H. R. Karimi, Y. Kao, and C. Gao, "A novel robust fuzzy integral sliding mode control for nonlinear semi-markovian jump t–s fuzzy systems," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 6, pp. 3594–3604, 2018.

[24] S. Aouaouda, M. Chadli, and H.-R. Karimi, "Robust static output-feedback controller design against sensor failure for vehicle dynamics," *IET Control Theory & Applications*, vol. 8, no. 9, pp. 728–737, 2014.

[25] H. C. Wu, "The karush–kuhn–tucker optimality conditions in an optimization problem with interval-valued objective function," *European Journal of Operational Research*, vol. 176, no. 1, pp. 46–59, 2007.

[26] S. Qin, X. Yang, X. Xue, and J. Song, "A one-layer recurrent neural network for pseudoconvex optimization problems with equality and inequality constraints," *IEEE transactions on cybernetics*, vol. 47, no. 10, pp. 3063–3074, 2016.

[27] G. Li, Z. Yan, and J. Wang, "A one-layer recurrent neural network for constrained nonconvex optimization," *Neural Networks*, vol. 61, pp. 10–21, 2015.

[28] J. Wang, J. Wang, and Q.-L. Han, "Neurodynamics-based model predictive control of continuous-time under-actuated mechatronic systems," *IEEE/ASME Transactions on Mechatronics*, 2020.

**YINGBAI HU** (Student Member, IEEE) received the M.Sc. degree in control engineering in South China University of Technology, Guangzhou, China, in 2017. He is currently working toward a Ph.D. degree in Computer Science as a member of the Informatics 6-Chair of Robotics, Artificial Intelligence and Real-time Systems at Technical University of Munich, Germany. His main research interests include neural network optimization and control in robotics, robot learning, surgical robotics, reinforcement learning.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIE.2020.3044776, IEEE Transactions on Industrial Electronics

**HANG SU** (Member, IEEE) received the M.Sc. degree in control theory and control engineering in South China University of Technology, Guangzhou, China, in 2015 and the Ph.D. degree in Bioengineering from Politecnico di Milano, Milano, Italy, in 2019. He participated in the EU funded project (SMARTsurg) in the field of Surgical Robotics. Dr. Hang Su is currently working in the Department of Electronics, Information and Bioengineering (DEIB) of Politecnico Di Milano. He is currently a Program Chair of IEEE ICARM 2021. He also severs as the Associate Editor for the IEEE ICRA and the IEEE IROS, and Guest Associate Editor for IEEE Robotics and Automation Letters, etc.. His main research interests include control and instrumentation in medical robotics, human-robot interaction, surgical robotics, deep learning, bilateral teleoperation, etc..

**JUNLING FU** received the B.E. degree in Mechanical Engineering from Guangzhou University and the M.S. degree in Mechanical Engineering from South China University of Technology, Guangzhou, China, in 2017 and 2020, respectively. Currently, he is pursuing a Ph.D. degree in the Department of Electronics, Information and Bioengineering (DEIB) of Politecnico di Milano, Italy. He is a member of the Neuroengineering and Medical Robotics Laboratory(NearLab). His main research interests include medical robotics, haptics, Human-Robot Interaction.

**HAMID REZA KARIMI** (Senior Member, IEEE) is currently Professor of Applied Mechanics with the Department of Mechanical Engineering, Politecnico di Milano, Milan, Italy. From 2009-2016, he has been Full Professor of Mechatronics-Control Systems at University of Agder, Norway. His current research interests include control systems and mechatronics with applications to automotive systems, robotics, vibration systems and wind energy. Dr. Karimi is currently the Editor-in-Chief, Technical Editor, and Associate Editor for several international journals such as IEEE Transactions. He is a member of Agder Academy of Science and Letters and has been awarded as the 2016-2020 Web of Science Highly Cited Researcher in Engineering, the 2020 IEEE Transactions on Circuits and Systems Guillemin-Cauer Best Paper Award, 2015 August-Wilhelm-Scheer Visiting Professorship Award, 2014 JSPS (Japan Society for the Promotion of Science) Research Award, and 2006 Alexander-von-Humboldt-Stiftung research Award, for instance.

**GIANCARLO FERRIGNO** (Senior Member, IEEE) received the M.Sc. degree in electrical engineering and the Ph.D. degree in bioengineering from the Politecnico di Milano, Milan, Italy.

He is the Founder of the Neuroengineering and Medical Robotics Laboratory with the Department of Electronics, Information and Bioengineering, Politecnico di Milano, in 2008, and a Lecturer of Medical Robotics. He is a Full Professor with the Politecnico di Milano. He has been the European Coordinator of three FP7 EU projects on ICT. Two of them, ROBOCAST (STREP 2008-2010) and ACTIVE (Integrated project 2011-2015) are in the field of the Surgical Robotics. MUNDUS (STREP 2010-2013) is in the field of Assistive and Rehabilitative Robotics. He has co-authored 20 papers (ISI Web of Knowledge) in the robotic field from 2011 to 2014. Dr. Ferrigno is working in the JWG9 ISO standard group for Surgical Robots collateral standard and organized several workshops in the surgical robotics for the last three years.

**ELENA DE MOMI** (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in biomedical engineering from the Politecnico di Milano, Milan, Italy, in 2002 and 2006, respectively.

She is an Associate Professor with the Department of Electronics, Information and Bioengineering, Politecnico di Milano. She is co-founder of the Neuroengineering and Medical Robotics Laboratory, in 2008, being responsible of the Medical Robotics section. IEEE Senior Member, she is currently Associate Editor of the Journal of Medical Robotics Research, of the International Journal of Advanced Robotic Systems, Frontiers in Robotics and AI and Medical & Biological Engineering & Computing. From 2016 she has been an Associated Editor of IEEE ICRA, IROS and BioRob and she is currently Publication Co-Chair of ICRA 2019. She is responsible for the lab course in Medical Robotics and of the course on Clinical Technology Assessment of the MSc degree in Biom. Eng. at Politecnico di Milano and she serves in the board committee of the PhD course in Bioengineering.

**ALOIS KNOLL** (Senior Member, IEEE) received the diploma (M.Sc.) degree in Electrical/Communications Engineering from the University of Stuttgart, Germany, in 1985 and his Ph.D. (summa cum laude) in Computer Science from the Technical University of Berlin, Germany, in 1988. He served on the faculty of the Computer Science department of TU Berlin until 1993. He joined the University of Bielefeld, as a full professor and the director of the research group Technical Informatics until 2001. Since 2001 he has been a professor at the Department of Informatics, TU München. He was also on the board of directors of the Central Institute of Medical Technology at TUM (IMETUM). From 2004 to 2006, he was Executive Director of the Institute of Computer Science at TUM. His research interests include cognitive, medical and sensor-based robotics, multi-agent systems, data fusion, adaptive systems, multimedia information retrieval, model-driven development of embedded systems with applications to automotive software and electric transportation, as well as simulation systems for robotics and traffic.