Proceedings of the 2nd International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/67/2024MA0061

A master-slave teleoperation system control model based on model predictive control with fast dynamic response

Yixin Song

Nanjing University of Posts and Telecommunications, Nanjing, China

13920035272@163.com

Abstract. Robot technology is advancing with constantly expanding applications. Teleoperation system demonstrates its potential to extend human capabilities in unstructured and hazardous environments as an important branch of robotics. However, there are still issues involving insufficient position tracking accuracy and dynamic response capabilities. This article proposes a master-slave teleoperation system control model based on model predictive control (MPC) with a real-time dynamic adaptive parameter adjustment algorithm. We establish a model based on the integral barrier Lyapunov function and construct a dynamic optimization model based on MPC. Then we carry out rolling optimization and feedback correction at each moment to feed the predicted values generated by MPC back to correct the actual tracking trajectories in real-time. Additionally, we utilize an adaptive neural network and robust term compensation to adjust dynamic parameters and eliminate the effects of time delays, external disturbances, and model uncertainties. Theoretical simulation experiments are conducted on the Simulink platform, analyzing position tracking error and dynamic response capabilities of the master-slave robot respectively to verify the proposed model. The results indicate a reduction in the position tracking errors of the master with 10% and slave with 11%, while also showing an improvement in their dynamic response capabilities. The speed error ranges of the master and slave are shortened with 6% and 7% respectively.

Keywords: Teleoperation system, model predictive control, tracking accuracy, dynamic response, adaptive parameters.

1. Introduction

Robot technology has been applied to all walks of life, becoming an integral part of individual lives. Teleoperation robot system forms an important branch that is used to extend human perception and operation capabilities in remote, unstructured and dangerous environments [1]-[2]. A typical bilateral teleoperation system consists of five parts - human operator, master robot, communication channel, slave robot, and external environment [3]-[5]. In this system, the master operator sends instructions to the slave robot through the information transmission channel. The slave robot then performs tasks and sends feedback to the master, enabling master-slave synchronization and collaborative work. However, this process with communication delays and external environmental disturbances leads to insufficient accuracy and real-time performance of the master-slave trajectory tracking thereby bringing security risks. Therefore, an accurate and responsive teleoperation system model is required to confront such challenges.

© 2024 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/67/2024MA0061

However, current teleoperation systems are applied with insufficient master-slave position tracking accuracy and dynamic response capabilities. Firstly, maintaining high tracking accuracy between master operator and slave robot is a challenge due to time delays, model uncertainties and external interference. If the tracking accuracy cannot be guaranteed, the operating system will fail to meet the requirements, affecting the performance of tasks that require precise control, such as surgery [6] or precision machining [7]. Inaccurate tracking also leads to the loss of targets and difficulties in position tracking, causing the system to deviate from the expected trajectory, and may even lead to unexpected collisions with dangerous objects. Secondly, dynamic response capabilities are also considered vital since the master-slave robot is teleoperation. The current obstacle to its development is that the control algorithm is not precise enough and cannot be adjusted in real-time to adapt to interference. Poorer dynamic responsiveness means the system takes longer to correct errors, resulting in increased control latency. This conducts a negative impact on the response of the system to external changes or disturbances, resulting it unsuitable for tasks requiring fast response times. Furthermore, poor dynamic response capabilities increase safety risks, especially in the field of industrial robotics [8], where unexpected events may endanger personnel or equipment.

Several studies have been conducted on the tracking accuracy of master-slave robots in teleoperation systems. Although hybrid control algorithms have been proposed for tracking [9]-[10], they do not consider external environmental factors and have the risk of over-adjustment, thus failing to achieve more accurate trajectory tracking. In order to solve these problems, researchers [11]-[13] used new algorithms to obtain higher tracking accuracy. Naqshrao et al. [11] proposed the extended state observer (ESO) predictive tracking control algorithm. Li et al. [12] proposed a control algorithm based on iterative learning, and Kim et al. [13] proposed a nonlinear predictive control algorithm based on deep reinforcement learning. However, these algorithms are complex, require a large amount of data training time, and have high computational complexity, which is not conducive to improving the dynamic response capability of the system.

Researchers also studied the dynamic response capability of teleoperation systems. Zhang et al. [14] proposed an adaptive H∞ control method suitable for teleoperation systems to improve the dynamic response capability. Some researchers improved the response speed of the controller by optimizing the design of fuzzy logic controller [15]. Others proposed a constant-delay non-singular finite-time control method based on terminal sliding modes to speed up the convergence time of synchronization errors in nonlinear remote control operating systems [16]. However, the above researches focus more on improving the dynamic response capability of the system and considers position tracking accuracy as a secondary factor. Therefore, none of research can analyze position tracking accuracy and dynamic response capabilities together to balance system performance.

This article proposes a master-slave teleoperation system control model based on model predictive control (MPC) with a real-time dynamic adaptive parameter adjustment algorithm. We define the trajectory tracking objective function, model it as an optimization problem to minimize the prediction error, and solve the optimization problem at each control moment to determine the optimal control action in the future, improving tracking accuracy and dynamic response capability effectively. In the context of the previous research, our article provides the following contributions:

(i) This article proposes a dynamic optimization model of the master-slave teleoperation robot system based on MPC. The model is utilized by the integral barrier Lyapunov function, including dynamic system modelling, environmental force representation, target trajectory tracking, and prediction error minimization. This model only adds a minimal calculation while reducing tracking errors effectively, thereby enhancing tracking accuracy.

(ii) This article proposes a real-time dynamic adaptive parameter adjustment algorithm, including parameter adjustment based on MPC prediction feedback, neural network dynamic parameter adjustment and robust optimization dynamic parameter adjustment. The dynamic response capability of the system is improved with sufficient tracking accuracy.

DOI: 10.54254/2755-2721/67/2024MA0061

(iii) This article proposes theoretical simulations based on the Simulink platform and validates them on an actual robot operating platform. These findings offer valuable insights applicable to various fields, involving remote surgery and space exploration in analogous scenarios.

The remainder of this article is organized as follows. Section II models the master-slave teleoperation robot system based on MPC and its environmental dynamics. The operation process of the master-slave teleoperation system based on MPC, adaptive neural network and robust term compensation is introduced in detail. Section III conducts theoretical simulation and comparative experiments, and Section IV summarizes this article.

2. Master-slave teleoperation system based on mpc

We conduct a dynamic model of the master-slave teleoperation system and utilize MPC to optimize the control of the model. We feed the predicted values generated by MPC back to correct the actual tracking trajectories in real-time, and employ adaptive neural networks and robust term compensation to adjust parameters dynamically.

2.1. Improved Teleoperation Model Based on MPC

This article takes the status of the robot and environmental information as input and uses the MPC model to predict future trajectory of the robot. Then, the error between the predicted trajectory and the desired trajectory is minimized by optimizing the control input, and the control strategy is updated in each control cycle.

Constructing a teleoperation system model:

$$x(k+1) = f\big(x(k), u(k)\big) \tag{1}$$

$$y(k) = h\big(x(k)\big) \tag{2}$$

We propose an integral barrier Lyapunov function:

$$V(z, \alpha) = \sum_{i=1}^{n} V_i(z_i, \alpha_{i-1}) \tag{3}$$

$$V_i(z_i, \alpha_{i-1}) = \int_0^{z_i} \frac{\sigma k_{c_i}^2}{k_{c_i}^2 - (\sigma + \alpha_{i-1})^2} d\sigma, i = 1, \ldots, n \tag{4}$$

Where, $z_i = x_i - \alpha_{i-1}, \alpha_1, \ldots, \alpha_{n-1}$ are continuously differentiable functions satisfying $| \alpha_i | \leq A_i < k_{ci+1}$ for positive constants A_i, $i = 0,1,..,n-1$.

We minimize the function J, Eq. 5 by MPC:

$$V = min_u \ \ J\big(x(k), u(\cdot)\big) \tag{5}$$

The teleoperation system consists of a pair of master-slave manipulators with 3 degrees of freedom. The dynamics of the manipulators in the Cartesian space are expressed in a specific form [17]:

$$\begin{aligned} M_{xj}(q_j)\dot{x}_j + C_{xj}(q_j, \dot{q}_j)x_j + g_{xj}(q_j) + f_{xj}^t(q_j, q_j) \\ +\delta_{xj}(q_j, \dot{q}_j, t) = u_{xj} + F_{jv} \end{aligned} \tag{6}$$

Where, $j \in \{m, s\}$ denotes the identity of the master and slave robots. $q_j \in R^{3\times1}$,$\dot{q}_j \in R^{3\times1}$,$\ddot{q}_j \in R^{3\times1}$ denote the acceleration, speed and position of the robot joint space respectively. M_{xj} is the inertia matrix, C_{xj} is the centrifugal force and Coriolis force matrix, g_{xj} is the gravity term matrix, f_{xj}^i is the joint friction moment, δ_{xj} is the unknown bounded external disturbance term, u_j represents the control input, $F_{mv} = F_h$ represents the master operation the robot exerts force, and $F_{sv} = F_e$ represents the contact force between the slave robot and the environment.

We consider the environmental force modelling of the slave as follows:

$$F_e = \Phi^T(t)\theta(t) \tag{7}$$

Where, $\Phi(t) = [\dot{x}_s(t), x(t), 1]^T$ and $\theta(t) = [B_e, K_e, C_e]^T$, B_e, K_e and C_e represent the damping coefficient matrix, stiffness coefficient matrix and constant vector of the environment respectively.

DOI: 10.54254/2755-2721/67/2024MA0061

To avoid the torque being transmitted directly in the signal transmission channel, we consider using adaptive neural networks to simulate environmental forces. The neural network is applied to generate optimal weights, and the weights are transmitted from the slave or the master end through the signal transmission channel to avoid errors caused by direct transmission of torque, thereby minimizing deviations, improving accuracy, and generating reference trajectories.

$$F_{ev} = W_{ev}^{T} S_{ev}(X_{ev}) \tag{8}$$

Where $X_{ev} = [\dot{x_s}, x_s]^T$ is the input vector, virtual environment parameter $W_{ev} \in R^{ol\times o}$, number of hidden layer nodes l and number of neural networks $o = n$.

There is an optimal value $\widehat{W_{ev}}$ for the estimated environmental parameter W_{ev}^{*} such that the absolute value of the estimated deviation $|\varepsilon|$ of F_{ev} is minimized, it is satisfied:

$$F_{ev} = W_{ev}^{*} S_{ev}(X_{ev}) + \varepsilon \tag{9}$$

In addition, we propose an effective strategy that combines robust terms with adaptive neural networks to solve the control problems of nonlinear systems caused by disturbances and unknown nonlinear terms [18].

The target impedance system designed for the master reference trajectory is written as follows:

$$M_r \ddot{x}_r + C_r \dot{x}_r + G_r = \xi_1 F_h - \xi_2 \hat{F}_{ev} \tag{10}$$

The slave reference trajectory is written as follows:

$$x_s^r = \frac{1}{\chi^2 s^2 + 2\chi s + 1} x_m\left(t - T_f\right) \tag{11}$$

the tracking error between the reference vector r and the model output y, Eq. 12 by MPC:

$$min_u \ \sum_{i=N_1}^{N_2} r(k+i|k) - y(k+i|k) \tag{12}$$

s.t.

$$u_{lb} \le u(k+j|k) \le u_{ub} \tag{13}$$

$$y_{lb} \le y(k+i \mid k) \le y_{ub} \tag{14}$$

$$\forall\, i \in \{N_1, \cdots, N_2\} \text{ and } j \in \{(0, \cdots, N_u\} \tag{15}$$

We refer to the predicted state $k+i$ at time point k as $x(k+i|k)$. A sequence of states is indicated by $x(\cdot)$:

$$x(k+i) \forall i \in (0, \cdots, N_u) \Rightarrow x(\cdot) \tag{16}$$

$$u(k+i) \forall i \in (0, \cdots, N_u) \Rightarrow u(\cdot) \tag{17}$$

$$y(k+i) \forall i \in (N_1, \cdots, N_u) \Rightarrow y(\cdot) \tag{18}$$

The constraint formulation is abbreviated by indicating that the sequence x $(\cdot)$ being in the feasible set X_f:

$$x_{lb} \le x(\cdot) \le x_{ub} \Rightarrow x \in X_f \tag{19}$$

Considering the system model of this article, we solve the optimization problem by MPC as:

$$\min_U \sum_{k=0}^{N-1} \left(y_{k+1} - y_{ref}\right)^T Q\left(y_{k+1} - y_{ref}\right) + u_k^T R u_k \tag{20a}$$

$$s.t. \quad x_{k+1} = A x_k + B u_k, k = 0, \dots, N-1 \tag{20b}$$

$$y_k = C x_k, k = 1, \dots, N \tag{20c}$$

$$u_k \in U, k = 0, \dots, N-1 \tag{20d}$$

where x_k, u_k, y_k denote the $k-th$ step predictions of system's states, inputs and outputs, respectively. N denotes the prediction horizon, $R > 0$ and $Q \ge 0$ are weighting matrices of appropriate dimensions, y_{ref} is the output reference to be tracked and U is the polyhedral constraint set. By solving (20) with a

DOI: 10.54254/2755-2721/67/2024MA0061

given initial condition $x_0 = x(t)$, the optimization yields the optimal input $sequenceU^\star = \{u_0^*, \dots, u_{N-1}^*\}$ from which only the first element, i.e., u_0^* is implemented to the plant. At the next time instant, (20) is solved again for a new value of state measurements $x(t)$ and the whole procedure is repeated.

The optimal control problem (20) can be cast as a QP of the form after straightforward algebraic manipulations:

$$J^*(x_0) = \min_U U^T HU + x_0^T FU \quad (21a)$$

$$s.t. \quad GU \le w + Ex_0 \quad (21b)$$

where $x_0 = x(t)$ is the initial condition. The QP problem is strictly convex since R is assumed to be positive definite.

2.2. Real-time dynamic adaptive parameter adjustment algorithm

We conduct a real-time dynamic adaptive parameter adjustment algorithm to optimize the model, utilize the predictive feedback function of MPC to obtain the optimal trajectory, and use adaptive neural networks and robust terms to eliminate the influence of external interference and model uncertainty to improve position tracking accuracy and dynamic responsiveness.

On the master side, the motion information of the master operator and the virtual environment parameters of the slave robot are used to realize the reconstruction of environmental force. The reference trajectory of the master is generated based on the behavior of the operator to ensure the operator to possess better force perception ability. The reference trajectory generated by the master, the master dynamics model and the reconstructed environmental force are used as inputs to perform the master model predictive control. Each sampling point also uses rolling optimization, and the results are fed back to the master dynamics model for correction. Then the controller is designed on the master side to track the reference trajectory of the master side.

The position signal of the master is transmitted to the slave, and after passing through the slave reference trajectory generator, the reference trajectory of the slave is generated. The reference trajectory generated from the slave, the slave dynamics model and the environmental force are used as inputs for predictive control. At each sampling point, the trajectory of future sampling points is predicted, rolling optimization is performed, and the results are fed back to the slave dynamics model for correction. Then the controller is designed on the slave to track the reference trajectory of the slave. At the same time, the neural network is used to estimate the environmental parameters of the slave and transfer them to the master.

The real-time dynamic adaptive parameter adjustment algorithm of master-slave teleoperation system is shown in Figure 1.

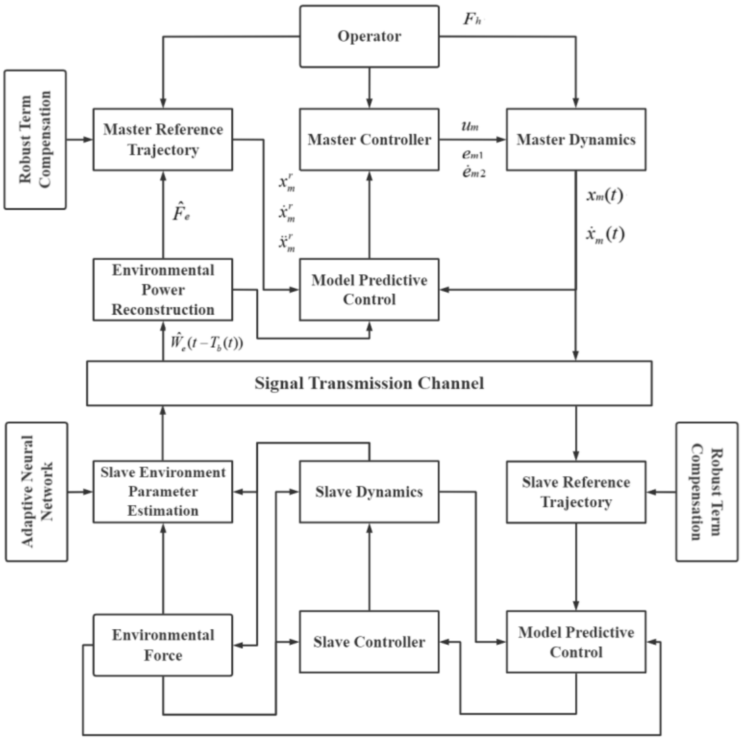

Figure 1. Real-time dynamic adaptive parameter adjustment algorithm of master-slave teleoperation system

The specific operation steps are as follows:

(i) The adaptive neural network is used to approximate the nonlinear part and the external disturbance uncertainty part in the environmental force information of the slave. According to the prediction error index, dynamically adjust the parameters in the neural network through the error back propagation algorithm to generate the slave environmental parameters estimate.

(ii) The slave environment estimation parameters are transmitted to the master through the signal transmission channel. On the master end, the motion information of the master end and the virtual environment parameters of the slave are input to realize the reconstruction of environmental forces.

(iii) Taking the reconstructed environmental forces and operator behavior as inputs, a robust compensation term is designed to offset factors such as model errors, unmodeled dynamics and external disturbances, and the parameters are dynamically adjusted to generate a reference trajectory for the master.

(iv) The master end model predictive control is performed using the reference trajectory generated by the master end, the master end dynamic model and the reconstructed environmental forces as inputs. Each sampling point adopts rolling optimization. The MPC controller will use the current system status and model to predict the system behavior in the future period, generate an optimized control input sequence, and feed the results back to the master end dynamic model for correction.

(v) The actual trajectory of the master end dynamics is used as input, and the slave end reference trajectory is generated through the signal transmission channel. Among them, robust terms are used to offset factors such as model errors and external interference.

(vi) The slave reference trajectory, slave dynamic model information and slave environmental force parameters are input to perform slave model predictive control. The current system status and model of the slave are used to predict the system behavior in the future period, and an optimized control input sequence is generated, which is fed back to the slave dynamics for actual trajectory correction.

DOI: 10.54254/2755-2721/67/2024MA0061

3. Case study

We apply a pair of 3-DOF Phantom Omni robots as the research object, analyze the master-slave robots respectively, discuss from two aspects: position tracking accuracy and dynamic responsiveness (error threshold convergence time and speed tracking accuracy), and verify the effectiveness of the model from the perspective of theoretical simulation. The model based on MPC with a real-time dynamic adaptive parameter adjustment algorithm is called the Improved Model, and the model that is not being used is called the Traditional Model.

In the simulation, the impedance matrices related to the behavior of human are set as $M_d = diag[0.45,0.45,0.45]$, $C_d = diag[1.5,1.5,1.5]$, $G_d = diag[0.08,0.08,0.08]$. The operating force F_h and the environmental force F_e are set as:$F_h = [0.15\sin t + 0.1,\ 0.2\cos t + 0.05,\ 0.07\sin t]^T, F_e = [0.05sint,\ 0.04sint, 0.03sint]^T$.

3.1. Analysis of master robot performance

The initial values of the joint angular velocities of the master ends are 0. The initial value of the joint angle of the master end is $q_m(0) = [-0.6, -0.3, -0.2]^T$. In the Traditional Model, we apply the master controller parameters as:$\omega_{j1} = diag(65,75,65), \omega_{j2} = diag(55,135,60)$. In the Improved Model, we apply the master controller parameters as: $\omega_{j1} = diag(30,134,45), \omega_{j2} = diag(26,140,48)$. The model prediction parameters are: $Q = F = diag(10,10,10), R = diag(0.1,0.1,0.1)$. The adaptive neural network and robust term parameters are: $\alpha_j = 7.15, \beta_j = 0.001$.

The improved model outperforms the traditional model in position tracking errors in the x, y, and z axes and 3D space. The standard deviation of the Traditional Model in the x, y, and z axes and 3D space are 2.86cm, 1.85cm, 2.61cm, and 2.55cm, respectively, while 2.51cm, 1.68cm, 2.33cm, 2.28cm in the Improved Model. The standard deviation of the improved model in the x, y, and z axes and 3D space is reduced by 12%, 9%, 11% and 11%% respectively compared with the traditional model from a percentage point of view. The maximum deviation of the Traditional Model in the x, y, and z axes and 3D space are 22.10cm, 14.54cm, 17.17cm, and 30.47cm, respectively, while 18.28cm, 13.48cm, 14.48cm, 25.74cm in the Improved Model. From a percentage point of view, the maximum errors of the improved model in the x, y, and z axes and 3D space are reduced by 17%, 7%, 16%, and 15% compared with the traditional model, respectively. Additionally, the total absolute deviation of the Traditional Model in the x, y, and z axes and 3D space are 61.10m, 46.33m, 60.64m, 105.71m, respectively, while 59.99m, 36.80m, 57.27m, 94.69m in the Improved Model. The total absolute errors of the improved model in the x, y, and z axes and 3D space are smaller than the traditional model by 2%, 20%, 6%, and 10% respectively from a percentage point of view. Detailed data is shown in Table I.

Table 1. Comparative analysis of master robot solutions

Case	State	Standard Deviation (cm)	Maximum Deviation (cm)	Total Absolute Deviation (m)
Traditional Model	X	2.86	22.10	61.10
	Y	1.85	14.54	46.33
	Z	2.61	17.17	60.64
	3D	2.55	30.47	105.71
Improved Model	X	2.51	18.28	59.99
	Y	1.68	13.48	36.80
	Z	2.33	14.48	57.27
	3D	2.28	25.74	94.69

We install the master robot speed error convergence threshold to 0.1m. The speed error threshold convergence times of the traditional model in the x, y, and z axes and 3D space are 0.31s, 0.23s, 0.28s, and 0.31s respectively. The speed error threshold convergence times of the Improved Model in the x, y,

DOI: 10.54254/2755-2721/67/2024MA0061

and z axes and 3D space are 0.25s, 0.20s, 0.23s, and 0.25s respectively. The speed error threshold convergence time of the improved model in the x, y, and z axes and 3D space is shortened by 19%, 13%, 18%, and 19% respectively from a percentage point of view compared with the traditional model. The speed tracking error range of the improved model is % smaller than that of the Traditional Model. In addition, the Improved Model also has smaller speed tracking fluctuations in the x, y, and z axes, with smaller variance, and higher data stability, as shown in Figures 2 and 3.

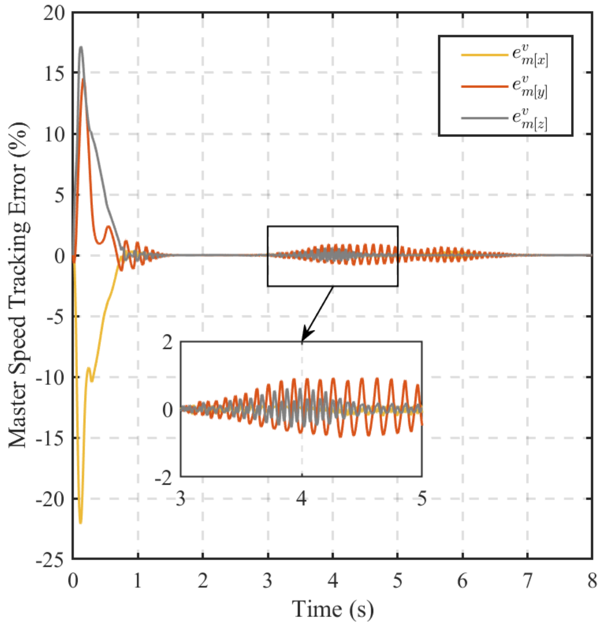

Figure 2. Master Speed Tracking Error in Traditional Model

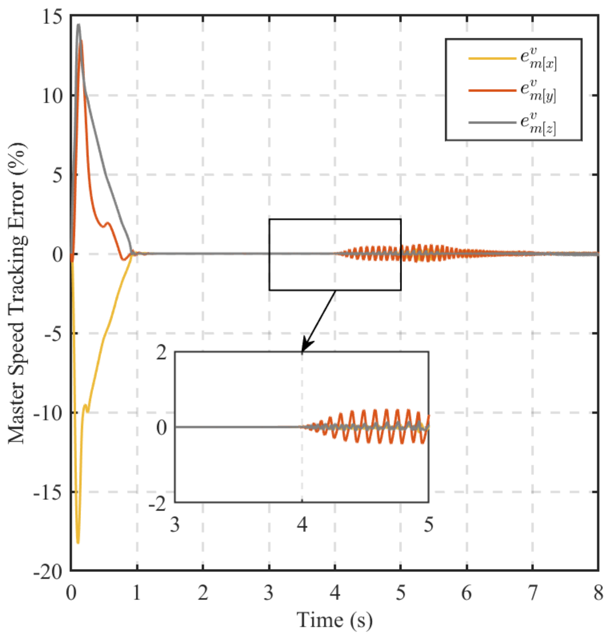

Figure 3. Master Speed Tracking Error in Improved Model

3.2. Analysis of slave robot performance

The initial values of the joint angular velocities of the slaves are 0. The initial value of the joint angle at the slave is $q_s(0) = [0.6, 0.3, 0.2]^T$. In the Traditional Model, we apply the slave controller parameters as: $\omega_{j1} = diag(75,82,65), \omega_{j2} = diag(55,140,60)$. In the Improved Model, we apply the master controller parameters as: $\omega_{j1} = diag(30,117,45), \omega_{j2} = diag(30,106,48)$. The model prediction parameters are: $Q = F = diag(10,10,10), R = diag(0.1,0.1,0.1)$. The adaptive neural network and robust term parameters are: $\alpha_j = 6.8, \beta_j = 0.001$.

The Improved Model outperforms the Traditional Model in terms of tracking slave position trajectories and considering performance in the x, y, and z axes and 3D space. The standard deviation of the Traditional Model in the x, y, and z axes and 3D space are 3.68cm, 8.26cm, 6.15cm, and 6.32cm, respectively, while in the Improved Model they are 2.90cm, 7.74cm, 5.04cm, and 5.60cm. This shows that the standard deviation of the improved model in the x, y, and z axes and 3D space is reduced by 21%, 6%, 18%, and 11%, respectively, compared to the traditional model from a percentage point of view. Moreover, the maximum deviation of the Traditional Model in the x, y, and z axes and 3D space are 21.57cm, 56.77cm, 33.97cm, and 62.12cm, respectively, while in the Improved Model, they are 17.96cm, 52.71cm, 27.41cm, and 57.70cm. This means that the maximum errors of the improved model in the x, y, and z axes and 3D space are reduced by 17%, 7%, 19%, and 7% respectively, compared to the traditional model from a percentage point of view. Furthermore, the total absolute deviation of the Traditional Model in the x, y, and z axes and 3D space are 166.37m, 254.26m, 206.69m, and 402.13m, respectively, while in the Improved Model they are 132.54m, 238.16m, 167.08m, and 355.90m. This indicates that the total absolute errors of the improved model in the x, y, and z axes and 3D space are smaller than the traditional model by 20%, 6%, 19%, and 11% respectively from a percentage point of view. Detailed data is shown in Table II.

Table 2. Comparative analysis of slave robot solutions

Case	State	Standard Deviation (cm)	Maximum Deviation (cm)	Total Absolute Deviation (m)
Traditional Model	X	3.68	21.57	166.37
	Y	8.26	56.77	254.26
	Z	6.15	33.97	206.69
	3D	6.32	62.12	402.13
Improved Model	X	2.90	17.96	132.54
	Y	7.74	52.71	238.16
	Z	5.04	27.41	167.08
	3D	5.60	57.70	355.90

The threshold for the convergence of the speed error of the slave robot was set to 0.1m. The convergence times of the speed error threshold of the Traditional Model in the x, y, and z axes, as well as 3D space, were 0.66s, 0.90s, 0.91s, and 0.91s, respectively. On the other hand, the Improved Model has speed error threshold convergence times in the x, y, and z axes, and 3D space, of 0.52s, 0.83s, 0.86s, and 0.86s, respectively. The speed error threshold convergence time of the Improved Model in the x, y, and z axes, as well as 3D space, is 21%, 8%, 5%, and 5% shorter than that of the Traditional Model. Moreover, the speed tracking error range of the Improved Model is 7% shorter than that of the Traditional Model. The Improved Model also has smaller speed tracking fluctuations in the x, y, and z axes, with smaller variance, and higher data stability, as shown in Figures 4 and 5.

DOI: 10.54254/2755-2721/67/2024MA0061

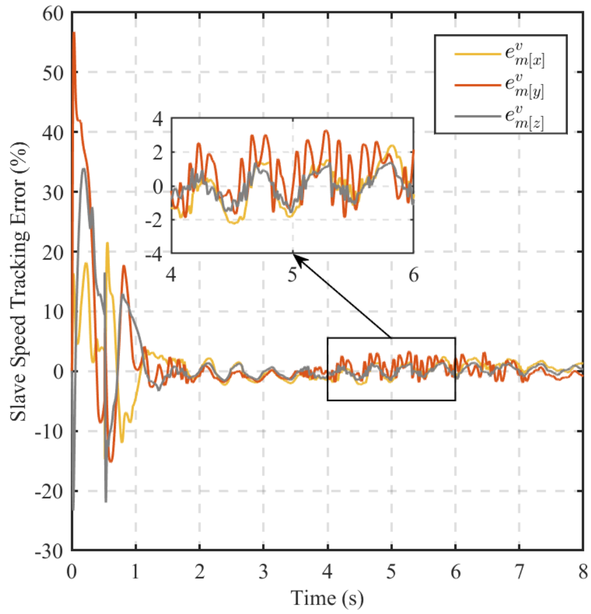

Figure 4. Slave Speed Tracking Error in Traditional Model

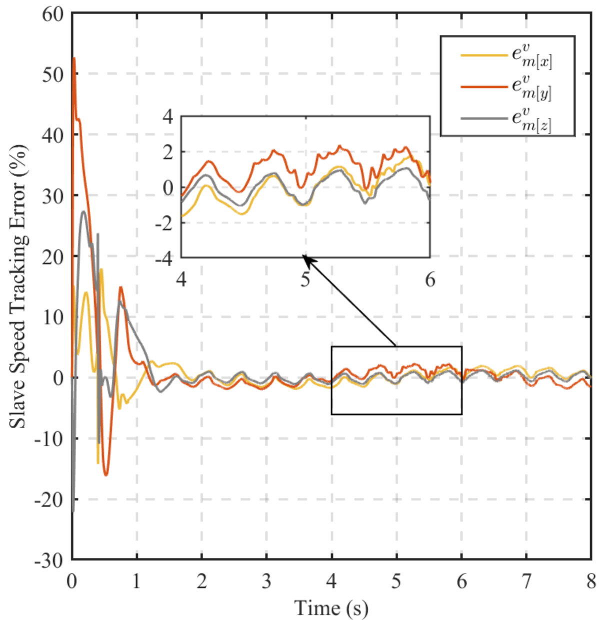

Figure 5. Slave Speed Tracking Error in Improved Model

4. Conclusion

This article proposes a master-slave teleoperation system control model based on model predictive control (MPC) with a real-time dynamic adaptive parameter adjustment algorithm. We develop a teleoperation model based on the integral barrier Lyapunov function and define a trajectory tracking objective function, formulating it as an optimization model to minimize prediction errors. The system corrects the actual tracking trajectories in real-time by using the predicted values generated by MPC. Neural networks and robust optimization are applied to adjust dynamic parameters and eliminate the effects of time delays, external disturbances, and model uncertainties.

DOI: 10.54254/2755-2721/67/2024MA0061

This article achieves reduced position tracking errors for both master and slave robots by using the dynamic optimization model based on MPC. Error convergence times and speed tracking errors are shortened through the real-time dynamic adaptive parameter adjustment algorithm. These improvements indicate enhanced position tracking accuracy and dynamic response capabilities, compared to the Traditional Model. This teleoperation system model with both accuracy and real-time performance presents a practical solution with adequate security and sufficient work efficiency for remote, unstructured, and potentially hazardous applications like telemedicine and space exploration.

Our future work will concentrate on refining and optimizing the proposed model to improve its applicability and robustness in diverse scenarios.

References

[1] Chen, Z., Huang, F., Yang, C., Yao, B.: Adaptive fuzzy backstepping control for stable nonlinear bilateral teleoperation manipulators with enhanced transparency performance. IEEE Trans. Ind. Electron. 67(1), 746–756 (2019).

[2] Mehrjouyan, A., Menhaj, M.B., Khosravi, M.A.: Robust observer-based adaptive synchronization control of uncertain nonlinear bilateral teleoperation systems under time-varying delay. Measurement 182, 109542, (2021).

[3] Y. Maddahi, S. Liao, W. Fung, E. Hossain, and N. Sepehri, “Selection of network parameters in wireless control of bilateral teleoperation manipulators,” IEEE Trans. Ind. Inf., vol. 11, no. 6, pp. 1445–1456, Dec. 2015.

[4] Y. Park, J. Lee, and J. Bae, “Development of a wearable sensing glove for measuring the motion of fingers using linear potentiometers and flexible wires,” IEEE Trans. Ind. Inf., vol. 11, no. 1, pp. 198–206, Feb. 2015.

[5] P. Huang, F. Zhang, J. Ma, Z. Meng, and Z. Liu, “Dynamics and configuration control of the maneuvering-net space robot system,” Advances in Sp. Res., vol. 55, no. 4, pp. 1004–1014, 2015.

[6] Korte C, Nair SS, Nistor V, Low TP, Doarn CR, Schaffner G. Determining the threshold of time-delay for teleoperation accuracy and efficiency in relation to telesurgery. Telemed J E Health. 2014 Dec;20(12):1078-86.

[7] Huang, Z., Zhu, J., Lei, J. Tool wear predicting based on multi-domain feature fusion by deep convolutional neural network in milling operations. J Intell Manuf 31, 953–966 (2020).

[8] T. N. Truong, A. T. Vo and H. -J. Kang, "A Backstepping Global Fast Terminal Sliding Mode Control for Trajectory Tracking Control of Industrial Robotic Manipulators," in IEEE Access, vol. 9, pp. 31921-31931, 2021.

[9] E. Nu˜no, R. Ortega, L. Basa˜nez, An adaptive controller for nonlinear teleoperators, Automatica 46 (1) (2010) 155–159.

[10] Y.-C. Liu, M.-H. Khong, Adaptive control for nonlinear teleoperators with uncertain kinematics and dynamics, IEEE/ASME Transactions on Mechatronics 20 (5) (2015) 2550–2562.

[11] Nageshrao, S., & Sasi, A. (2019). Predictive Tracking Control of Teleoperation Robots using Extended State Observer. Robotics and Autonomous Systems, 113, 150-159.

[12] Li, X., Zhang, Y., & Su, C. Y. (2019). Iterative learning-based control for teleoperation systems with unknown time-varying delay and input nonlinearity. IEEE Transactions on Industrial Electronics, 66(1), 594-602.

[13] Kim, K., Park, Y., & Kim, B. (2022). Nonlinear Predictive Control of Teleoperation Systems using Deep Reinforcement Learning. IEEE Transactions on Industrial Informatics, 18(2), 1131-1139.

[14] Zhang, H., and Liu, Y. (2019). Adaptive H∞ Control for Teleoperation Systems with Time-Varying Delays and Input Saturation. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 49(1), 99-110.

[15] P. Longhini, A. Simeone, "Fuzzy Logic Control for Teleoperation Systems: A Survey," IEEE Transactions on Fuzzy Systems, 2018.

DOI: 10.54254/2755-2721/67/2024MA0061

[16] Y. Yang, C. Hua, and X. Guan, "Finite time control design for bilateral teleoperation system with position synchronization error constrained," IEEE Trans. Cybern., vol. 46, no. 3, pp. 609–619, Mar. 2016.
[17] Makkar G, Hu W, Sawyer W G, et al. Lyapunov-based tracking control in the presence of uncertain nonlinear parameterizable friction[J]. IEEE Transactions on Automatic Control, 2007, 25(10): 1988-1994.
[18] LI Longnan, HUANG Panfeng, MA Zhiqiang. Finite-time Control Method for Robot Teleoperation Based on Time-varying Output Constraints[J]. ROBOT, 2022, 44(1): 19-34, 44.