<http://www.diva-portal.org>

Postprint

This is the accepted version of a paper published in *Journal of Computing and Information Science in Engineering*. This paper has been peer-reviewed but does not include the final publisher proof-corrections or journal pagination.

Citation for the original published paper (version of record):

Zhang, S., Dai, S. (2018)

Real-Time Trajectory Generation for Haptic Feedback Manipulators in Virtual Cockpit Systems

Journal of Computing and Information Science in Engineering, 18(4):

041015-1-041015-11

<https://doi.org/10.1115/1.4041166>

Access to the published version may require subscription.

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:oru:diva-77207>

Real-Time Trajectory Generation for Haptic Feedback Manipulators in Virtual Cockpit Systems

Shiyu Zhang

State Key Laboratory of Virtual Reality Technology and Systems
Beihang University
XueYuan Road No.37, HaiDian District, Beijing, China, 100191
Email: zhangshiyu@buaa.edu.cn

Shuling Dai*

State Key Laboratory of Virtual Reality Technology and Systems
Beihang University
XueYuan Road No.37, HaiDian District, Beijing, China, 100191
Email: sldai@buaa.edu.cn

To obtain real-time interactions in the virtual cockpit system (VCS), a real-time trajectory generation method based on dynamical non-linear optimization and regression prediction for the haptic feedback manipulator (HFM) is presented in this paper. First, a haptic feedback system based on servo serial manipulator is constructed. Then, the trajectory planning problem for the HFM is formulated as a non-linear optimization problem to balance the motion time and power consumption and ensure the safety of physical human-robot interactions. Multiple optimization problems are solved to generate the optimal database off-line. Finally the classified multivariate regression method is presented to learn the database and generate optimal trajectories with arbitrary initial and objective positions on-line. Results show that trajectories with rapidity, safety and lower power consumption can be generated in real-time by this method, which lay a basis of haptic interactions in the VCS.

1 Introduction

Virtual cockpit system (VCS) is a flight simulator in which human-computer interactions are achieved by the virtual reality technology. Owing to the advantages of smaller size, lower cost and higher flexibility in the structure and functionality compared with the physical one, the VCS is meaningful in the flight simulation [1].

Haptic feedback is of great significance for better immersion and interaction in the VCS. However, efforts in this direction are still very limited [2]. In order to achieve haptic interaction in the VCS without sacrificing the advantages, a haptic feedback system based on the servo serial manipula-

tor is presented. On the basis of the traditional VCS, a servo serial manipulator (called haptic feedback manipulator, HFM) is placed in front of the user. A concise control panel equipped with different kinds of buttons are linked with the end-effector of the HFM. The HFM tracks the hand and brings the corresponding button to the objective position the user operates with to provide haptic feedback.

To achieve the interaction between user's hand and the HFM, trajectory planning and generation for the HFM according to the hand motion is required, which provide the command to actuate the HFM. Three key problems are required to be solved:

Physical human-robot interaction (pHRI) safety. In haptic feedback applications, the HFM and user's hand work in a shared workspace and are required to contact directly. Thus, the parameters of the workspace, kinematics and dynamics of the HFM should be limited to ensure the safety of the user.

Optimal trajectory planning. In the course of haptic feedback, the HFM is required to reach the objective configuration quickly with low power consumption, and ensure the safety of pHRI. Therefore, we would like to take the above factors into consideration to seek for optimal trajectories.

Real-time trajectory generation. When the haptic feedback system works, as the hand moves, the trajectory need to be re-calculated according to the variation of predicted interacting configuration. In one update period, the system predicts the new interacting configuration and accordingly updates the trajectory. Thus, one trajectory generation needs to be done in one update period to ensure the real-time performance.

*Corresponding author.

First, considering the pHRI, we present a safety strategy by establishing motion limitations of the HFM. Secondly, non-linear optimization problem is established and solved to obtain the optimal trajectories, in which the objective function is constructed to balance the motion time and the power consumption, and the constraints are constructed according to the mechanical limits and the safety of pHRI. Finally, database of optimal trajectories is created off-line and new optimal trajectories according to the new inputs are generated on-line by regression and prediction.

2 Related Work

2.1 Haptic feedback in VCSs

To achieve haptic feedback in VCS, Semi-Virtual Reality Cockpit solution [3] is proposed based on the principle that “Seeing is virtual and touching is believing”. Haptic feedback is achieved by employing a 1:1 control panel, but advantages of the VCS are sacrificed due to the large size and poor flexibility. Another VCS scheme was developed by STRICOM with haptic feedback achieved by the TOPIT (Touched Objects Positioned in Time) technology [4, 5]. A mechanical system equipped with various kinds of buttons is placed in front of the user. When user’s hand moves, the servo system brings the desired button to the position the user aims at to provide haptic feedback. The control panel is simplified and different kinds of control panels can be simulated only by changing software. However, the mechanical system is still large. Moreover, the buttons can only move in a 2-Dimension plane area so that interactions in 3-Dimension space are hardly achieved.

We improve the TOPIT scheme by employing a serial manipulator as the HFM, which can achieve the 3-Dimension movement with smaller size, larger workspace and better flexibility.

2.2 Trajectory planning and generation

The traditional point-to-point trajectory planning is started from interpolation-based methods, such as polynomial interpolation [6, 7] and B-spline interpolation [8, 9]. In general, pure interpolation-based methods are able to accomplish required tasks, but unable to achieve the optimal performance in specific aspects. To obtain optimal trajectories, some trajectory planning methods are presented based on non-linear optimization [10, 11]. Non-linear optimization problem is constructed with optimal objectives based on time, energy and power consumption. Constraints such as mechanical structure, time and obstacle avoidance [12] are considered as well. Von Stryk *et al.* investigated the non-linear optimization with three separated criterions of minimum time, minimum energy and minimum power consumption and solved it by a numerical method of combining a direct collocation and an indirect multiple shooting method [13]. Chettibi *et al.* presented the optimal planning problem tried to find the trade-off between time, energy and power consumption and solved it by the Sequential Quadratic Programming (SQP) method [14]. However, all aforemen-

tioned optimization-based methods are not real-time due to the complex computation of the non-linear optimization.

The non-linear optimization is likely to get stuck in local minimum and will be required to be calculated with multiple initial guesses to obtain the global minimum. Due to the large amount of calculations, it is hard to achieve on-line computation. Therefore, quickly computing for the global minimum is still challenging. To generate trajectories in real-time, a promising method is learning with previous data to reduce the on-line computation time. Pan *et al.* proposed a machine learning approach to predict the qualities of initial guesses [15]. Jetchev *et al.* proposed a trajectory prediction method, employing the previous data to establish the map from inputs to outputs by machine learning and predicting new outputs according to new inputs [16]. Hauser considers more general context of nonlinear optimization problems and establishes conditions on how many samples are needed to achieve a given approximation quality [17]. Lampariello *et al.* [18] and Werner *et al.* [19] use the trajectory prediction methods to solve the on-line trajectory planning problem, but limit to the cases with a fixed initial configuration of the target trajectory.

In this paper, we combine the trajectory planning method based on non-linear optimization and the calculation method for global optimization based on machine learning, and then extend them into the haptic feedback application in the VCS. Non-linear optimization problem is constructed according to the principle of power consumption optimization and the safety of pHRI. Furthermore, the classical regression methods are modified by classifying the input variables, which are always complex in the motion planning of the HFM, to achieve trajectory generation with arbitrary initial and final configurations.

3 Haptic feedback system based on servo serial manipulator

3.1 Haptic feedback system

A haptic feedback system for the VCS based on the servo serial manipulator is constructed in this paper. As shown in Fig. 1, it consists of HFM, motion tracking sensors and computers. The HFM is essentially a servo serial manipulator, which is detailed in Section 3.2. A concise control panel equipped with various kinds of buttons, such as pressing button, rotating button and sliding button, is linked with the end-effector of the HFM to provide haptic feedback to users. Each button is used to simulate all the buttons of the same type in the real cockpit to simplify the control panel. Motion tracking sensors are equipped in front of the user and used to collect the motion data such as position and orientation of the hand based on vision. Computers work to perform hand trajectory prediction, trajectory planning and control for the HFM.

The work flow is shown in Fig. 2. When user’s hand moves, hand motion data are measured by tracking sensors, and then processed by computers to predict the hand trajectory and determine the interacting time t_c and the interacting point p_c . Trajectory planning is performed and the HFM

Fig. 1. Components of haptic feedback system based on servo serial manipulator

Fig. 2. Work flow of haptic feedback system

Fig. 3. Interaction of the control panel and user's hand

is actuated to bring the corresponding button on the control panel to p_c at t_c . The trajectory of the HFM is required to be re-planned periodically according to the variation of the predicted hand trajectory and interacting point, while the hand is moving constantly. At the same time, the user is able to see the virtual hand moving in the virtual environment through the virtual display accordingly.

To achieve the consistency of the virtual environment and real world, we extract a fixed area in the reachable workspace of the HFM to simulate the control panel, in which all the interacting points are supposed to be locat-

ed. The reachable workspace and the control panel area are shown as the blue and red area in Fig. 3a, respectively. At the moment of interaction, the user is able to see virtual hand operating the control panel in virtual environment (Fig. 3b), while the corresponding button on the end-effector of the HFM reaches the interacting point which the user intends to operate to provide haptic feedback (Fig. 3a).

3.2 Haptic feedback manipulator

A 6-DOF (degree of freedom) serial manipulator is adopted as HFM in this paper (Fig. 4). The six joints of

the serial manipulator are waist, shoulder, elbow, pitching wrist, yawing wrist and rotating wrist joint, which are all rotating joints. The position of the end-effector is determined by the waist, shoulder and elbow joint, while the orientation is determined by three wrist joints. The pitching wrist and yawing wrist joint are used to adjust the orientation of the concise control panel. And the rotating wrist joint is used to switch states of some buttons to further reduce the number of buttons on the control panel. For example, the state of a sliding button can be translated from on to off, or in reverse, by rotating 180 degrees.

The Denavite Hartenberg (DH) convention is adopted to describe the kinematical chain of the HFM. Four geometric parameters are used to describe the relationship between two adjacent joints: α_i is the twist angle between the axes, a_i is the perpendicular distance between the joint axes, θ_i is the relative rotation between two adjacent links and d_i is the offset distance along the joint axis [20], as illustrated in Table 1. The structure of the HFM is specified by the DH link parameters shown in Table 2. The DH coordinate system is shown in Fig. 5, where $O_0 - X_0 - Y_0 - Z_0$ is the base coordinate frame and $O_i - X_i - Y_i - Z_i (i = 1, 2, \dots, 6)$ are the coordinate frames of the six joints.

Fig. 4. The structure of the HFM (1: waist, 2: shoulder, 3: elbow, 4: pitching wrist joint, 5: yawing wrist joint, 6: rotating wrist joint)

Fig. 5. DH coordinate system of the HFM

3.3 Safety of pHRI

In human-robot interaction applications, robots and humans work in a shared workspace. To avoid injury from

Table 1. DH Link Parameters of the Manipulator and Explanation

Parameter	Meaning
α_i	Angle value from Z_i to Z_{i+1} about X_i
a_i	Perpendicular distance from Z_i to Z_{i+1} along X_i
θ_i	Angle value from X_i to X_{i+1} about Z_i
d_i	Perpendicular distance from X_i to X_{i+1} along Z_i

Table 2. Link Parameters of the HFM

Link	α /rad	a /m	θ /rad	d /m
L1	$-\pi/2$	0	θ_1	l_1
L2	0	l_2	θ_2	0
L3	0	l_3	θ_3	0
L4	$\pi/2$	0	θ_4	0
L5	0	0	θ_5	0
L6	0	0	θ_6	0

robots to humans, measures to ensure safety are required, including active and passive measures [21], in which collision avoidance is a common method [22, 23]. However, in haptic feedback applications, the HFM and user's hand are required to contact directly, performing physical human-robot interactions. Therefore, an active safety measure is proposed in this paper, including motion range limiting (detailed in Section 3.3.1) and power and force limiting (detailed in Section 3.3.2).

3.3.1 Motion range limiting

The reachable workspace is obtained by the Monte Carlo method [24], denoted as $S \subset \mathbb{R}^3$, shown by the blue points in Fig. 3a. A control panel area can be extracted, denoted as S_C , shown by the red points in Fig. 3a, which divides the reachable workspace into the effective workspace S_E and the prohibiting area S_P .

When the haptic feedback system works, the motion ranges of the HFM and user's hand are restricted:

1. The end-effector of the HFM is restricted to move in S_E by trajectory planning and controlling for the HFM. At the same time, user's hand is restricted to move at the side of S_P by monitoring of the sensors and visual feedback through the virtual display.
2. At the interacting moment, user's hand and the corresponding button contact at a point in S_C . All the contacts are performed in S_C .
3. The end-effector of the HFM reaches the interacting point a slight time before user's hand to ensure that the HFM is static when the user manipulating on it.

3.3.2 Power and force limiting

The probability of danger is greatly reduced by limiting the motion range, but unable to be fully eliminated. Unexpected collisions may happen when the HFM or user's hand moves beyond the motion limits due to reasons such as sensor errors. Power and force limiting is presented in collaborative industrial robot safety requirements (ISO/TS 15066 [25]) to reduce the risk for applications in which contact between the robot and human is considered as a normal event.

Power and force limiting can be converted to maximum relative velocity limiting, which can be calculated according to the pain sensitivity threshold of the human body [25]:

$$v_{rel,max} = \frac{p_{max}A}{\sqrt{\mu k}}, \quad (1)$$

where p_{max} is the maximum contact pressure for specific body area, A is the area of contact between robot and body region, k is the effective spring constant for specific body region, μ is the reduced mass of the two-body system:

$$\mu = \left(\frac{1}{m_H} + \frac{1}{m_R} \right)^{-1}, \quad (2)$$

where m_H and m_R are the effective mass of the body region and the robot, respectively.

Therefore, the maximum permissible relative velocity is determined by p_{max} , k , m_H and m_R . According to ISO15066 [25], the contact parameters of the hand are shown in Table 3.

In this paper, we adopt the robot with $m_R = 20\text{kg}$. When $A = 100\text{mm}^2$, we can derive $v_{rel,max} = 1.86\text{m/s}$ according to Equ. 1 and 2.

Table 3. Contact parameters of the hand

$p_{max}(N/mm^2)$	$k(N/mm)$	$m_H(kg)$
3.9	75	0.6

4 Trajectory planning based on non-linear optimization

4.1 Problem statement

When the haptic feedback system works, the trajectory of hand movement and interacting point are predicted according to current hand motion periodically every T milliseconds. A sequence of predicted interacting points $(p_c^0, p_c^1, \dots, p_c^n) \subset S_E$ is obtained. In each period, re-planning of the trajectory is performed, in which the current configuration and the interacting configuration are selected as the initial and objective configurations, respectively. The whole trajectory of the end-effector is shown in Fig. 6.

In each period, it is required to generate a trajectory from the initial configuration q_0 to the objective configuration q_c . The trajectory is encoded by a parametric representation, denoted as $q(C, t)$, where $C \in \mathbb{R}^{N_c}$ is the parameter vector representing the trajectory, and N_c is the dimension of parameters. Trajectory planning is formulated as a non-linear optimization problem. The objective function $F(C)$ is defined according to motion time and power consumption. Constraints are defined according to mechanical limits and the safety of pHRI. We need to find a trajectory which is subject to all the constraints and has minimal $F(C)$. The input state is denoted as $x = (q_0, q_c)$. The optimization problem can be treated as a mapping from x to C :

$$x \mapsto C^* = \min_C F(x, C), \quad (3)$$

denoted as $C^* = O(x)$.

Fig. 6. End-effector trajectory of HFM (Black lines: trajectories generated in every period, Red line: the actual trajectory of the whole process)

4.2 Trajectory parameterization

To ensure the continuity of velocities and accelerations, 5-order polynomial is employed to encode the trajectories. For each joint, the parametric representation of the position is

$$q(t) = c_0 + c_1t + c_2t^2 + c_3t^3 + c_4t^4 + c_5t^5. \quad (4)$$

Then, the velocity and acceleration of each joint are:

$$\dot{q}(t) = c_1 + 2c_2t + 3c_3t^2 + 4c_4t^3 + 5c_5t^4, \quad (5)$$

$$\ddot{q}(t) = 2c_2 + 6c_3t + 12c_4t^2 + 20c_5t^3. \quad (6)$$

The trajectory of each joint has 6 parameters.

4.3 Dynamical non-linear optimization

4.3.1 Optimization Problem

The non-linear optimization problem with non-linear constraints can be described as:

$$\begin{aligned} (\mathbf{c}^*, t_f^*) &= \min F(\mathbf{c}, t_f) \\ s.t. \\ \mathbf{c} &\in \mathbb{R}^{N_{c1}}, t_f \in \mathbb{R}, \\ h_i(\mathbf{c}, t_f) &= 0, (i = 1, 2, \dots, N_h), \\ g_i(\mathbf{c}, t_f) &\geq 0, (i = 1, 2, \dots, N_g), \end{aligned} \quad (7)$$

where F is the objective function, h_i and g_i are the equality and inequality constraints, respectively.

$\mathbf{c}$ is the polynomial parameter vector. In this paper, we only consider the position of the end-effector and the three corresponding joints, $N_J = 3$, thus $N_{c1} = 18$. All the joints are required to reach the objective positions at the same time, and one motion time parameter t_f is added. Therefore, $N_c = 19$.

4.3.2 Objective Function

In the haptic feedback applications, the HFM is required to reach the interacting configuration as quickly as possible with lower power consumption. The objective function is defined as

$$F(\mathbf{c}, t_f) = \beta \cdot \alpha \cdot t_f + (1 - \beta) \cdot P(\mathbf{c}, t_f), \quad (8)$$

$$P(\mathbf{c}, t_f) = \left(\int_0^{t_f} \sum_{i=1}^{N_J} (\dot{q}_i(\mathbf{c}, t) \tau_i(\mathbf{c}, t))^2 dt \right)^{\frac{1}{2}}, \quad (9)$$

where P is the power consumption, τ_i is the torque of the i th joint. α is the proportionality coefficient, making the two items in the same order of magnitudes. β is the weighting coefficient, representing the weighting of the motion time and the power consumption, and $0 \leq \beta \leq 1$. The case $\beta = 1$ and $\beta = 0$ correspond to the motion time minimization mode and the power consumption minimization mode, respectively.

4.3.3 Constraints

The torque $\boldsymbol{\tau}$ at time t is determined by the $\mathbf{q}$, $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ at t

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}), \quad (10)$$

where $\mathbf{M}(\mathbf{q})$, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ and $\mathbf{G}(\mathbf{q})$ are the inertia matrix, the centripetal force and Coriolis force matrix, and the gravity matrix, respectively.

The boundary conditions are

$$\mathbf{q}(0) = \mathbf{q}_0, \quad (11)$$

$$\mathbf{q}(t_f) = \mathbf{q}_c, \quad (12)$$

$$\dot{\mathbf{q}}(0) = \dot{\mathbf{q}}_0, \quad (13)$$

$$\dot{\mathbf{q}}(t_f) = 0. \quad (14)$$

The mechanical limits are

$$|q_i(t)| < q_{i,max}, i = 1, \dots, N_J, \quad (15)$$

$$|\dot{q}_i(t)| < \dot{q}_{i,max}, i = 1, \dots, N_J, \quad (16)$$

$$|\ddot{q}_i(t)| < \ddot{q}_{i,max}, i = 1, \dots, N_J, \quad (17)$$

$$|\tau_i(t)| < \tau_{i,max}, i = 1, \dots, N_J. \quad (18)$$

To ensure the pHRI safety, the total motion time, the final acceleration and the linear velocity of the end-effector are limited.

First, according to Section 3.3.1, the motion time of the HFM should not exceed the interacting time t_c :

$$t_c - t_f \geq T_s. \quad (19)$$

Considering the predicting error, a margin T_s is selected to ensure that the HFM reach the interacting point before the hand.

Additionally, an acceleration threshold for pHRI is defined to make the acceleration as small as possible when the HFM approaches the interacting configuration:

$$|\ddot{\mathbf{q}}(t_f)| < a_s. \quad (20)$$

Also, the linear velocity of the end-effector is restricted according to Section 3.3.2:

$$|\dot{\mathbf{p}}(t)| < V_s, \quad (21)$$

where V_s is the maximum permissible relative velocity obtained in Section 3.3.2, $\dot{\mathbf{p}}(t)$ is the linear velocity in Cartesian space at time t , which can be obtained by the Jacobi equation:

$$\mathbf{V} = [\mathbf{v}, \boldsymbol{\omega}]^T = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}, \quad (22)$$

where $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{6 \times N_J}$ is the Jacobian matrix, $\mathbf{V} \in \mathbb{R}^6$ is the generalized velocity of the end-effector in Cartesian space, composed of the linear velocity $\mathbf{v} \in \mathbb{R}^3$ and the angular velocity $\boldsymbol{\omega} \in \mathbb{R}^3$. And

$$|\dot{\mathbf{p}}(t)| = |\mathbf{v}(t)|. \quad (23)$$

4.3.4 Solution

According to Section 4.2, each joint has 6 polynomial parameters ($c_0, c_1, c_2, c_3, c_4, c_5$). According to the boundary conditions in Section 4.3.3, each joint has 2 independent joint parameters. First, according to Equ. 11 and 13, we get $c_0 = q_0$ and $c_1 = \dot{q}_0$. Then we select (c_2, c_3) as the independent parameters, consequently c_4 and c_5 can be derived by Equ. 12 and 14. Therefore, for three joints, there are 7 independent optimization parameters, denoted as $\mathbf{C} = (c_2^1, c_3^1, c_2^2, c_3^2, c_2^3, c_3^3, t_f)$.

The SQP method [26] is adopted to solve the non-linear optimization problem to minimize the objective function described in Section 4.3.2 satisfying to the constraints in Section 4.3.3, which can be represented as:

$$\mathbf{C}^* = O_{\mathbf{C}_0}(\mathbf{x}, \mathbf{C}). \quad (24)$$

where $\mathbf{C}_0$ is the initial guess for the iterative solver.

To achieve the real-time performance, we elaborately select the initial guess $\mathbf{C}_0$ and modify the solving algorithm, which are detailed in Section 5.

5 Real-time trajectory generation based on regression and prediction

5.1 Overview

Due to the large amount of calculations of dynamics, the SQP solver described in 3.3-D takes long time to converge, which makes it hardly operate on-line. Therefore, a real-time trajectory generation method based on regression and prediction is presented in this paper. First, we select multiple input states $\mathbf{x} = (\mathbf{q}_0, \mathbf{q}_c)$ and solve the optimization problems by the SQP method to obtain the optimal trajectories and generate a database off-line. During the on-line course, for a new input state, the optimal parameters are predicted through the regression model established with the database. The procedure is as below ((1) and (2) are off-line steps and (3) and (4) are on-line steps), shown as Fig. 7:

(1) Workspace analysis

First, N_p groups of joint variables $\mathbf{q} = (\theta_1, \theta_2, \theta_3)$ are selected in the motion range and the corresponding positions in Cartesian space are obtained by the forward kinematics. Then, a 3-Dimension point cloud is depicted and S_E and S_C are determined.

(2) Generate database

N_{p0} groups of $\mathbf{p}_0 \in S_E$ and N_{pc} groups of $\mathbf{p}_c \in S_C$ are selected as the initial points and the objective points, respectively. The corresponding configurations $\mathbf{q}_0$ and $\mathbf{q}_c$ are paired to obtain $N_{p0} * N_{pc}$ groups of input states $\mathbf{x} = (\mathbf{q}_0, \mathbf{q}_c)$. For each $\mathbf{x}$, optimal parameters $\mathbf{C}$ are obtained by solving the optimization problem by the SQP method. The database is created:

$$D = \{\mathbf{x}^{i,j} = (\mathbf{q}_0^i, \mathbf{q}_c^j), \mathbf{C}^{i,j}\} (i = 1, \dots, N_{p0}; j = 1, \dots, N_{pc}). \quad (25)$$

(3) Regression and prediction

For the new input state $\mathbf{x}$, regression analysis is performed with database D and the mapping from current state to the trajectory parameters is established:

$$\mathbf{x} \mapsto \mathbf{C} = f(\mathbf{x}). \quad (26)$$

And the new generated parameters $\tilde{\mathbf{C}}$ are obtained.

(4) Refine parameters

As $\tilde{\mathbf{C}}$ may not satisfy the constraints, we refine $\tilde{\mathbf{C}}$ by solving the optimization problem taking $\tilde{\mathbf{C}}$ as the initial guess. To further reduce the computation time, we simplify the original optimization problem by removing the objective function

$$\mathbf{C}^* = \bar{O}_{\tilde{\mathbf{C}}}(\mathbf{x}), \quad (27)$$

where $\bar{O}$ represents the non-objective optimization.

5.2 Regression prediction method

5.2.1 Classified multivariate regression (CMR)

For a new input state $\mathbf{x} = (\mathbf{q}_0, \mathbf{q}_c)$, predicted parameters $\tilde{\mathbf{C}}$ are obtained by learning from the database. As the non-linearity of the mapping from inputs to outputs, global linear regression is not appropriate. Therefore, two local non-parametric regression methods, k-nearest neighbour (k-NN) and locally weighted regression (LWR), are selected.

Define the similarity of the new input state $\mathbf{x}$ and a state in the database $\mathbf{x}^{i,j}$:

$$\|\mathbf{x}^{i,j} - \mathbf{x}\|^{-2}. \quad (28)$$

Different measurements can be achieved by selecting different forms of norm $\|\cdot\|$.

Then the new parameters are generated by weighting of the parameters in the database. The weighting coefficients are determined by the similarities.

For an input state $\mathbf{x} = (\mathbf{q}_0, \mathbf{q}_c)$, the initial configuration $\mathbf{q}_0$ and the objective configurations $\mathbf{q}_c$ are expected to be considered separately. Therefore, we improve the classical method by classifying the elements of the input variable into several classes and defining similarity for each class. Suppose the input variable is d -Dimension, and divide the elements into m classes. The index set of the i th class is

$$I = \{I_1, \dots, I_{d_i}\}, \quad (29)$$

where d_i is the number of elements in the i th class, and

$$\sum_{i=1}^m d_i = d. \quad (30)$$

Fig. 7. Real-time trajectory generation base on regression and prediction

Define m corresponding selection matrix, and the i th selection matrix $S^l \in R^{d_l \times d}$ is

$$S_l(i, j) = \begin{cases} 1 & i = 1, j = I_1; \dots; i = d_l, j = I_{d_l} \\ 0 & \text{others} \end{cases} \quad (l = 1, \dots, m). \quad (31)$$

Then the similarity of the i th class of the input variable is

$$Sim_l = \|S^l(x^{(i,j)} - x)\|^{-2}. \quad (32)$$

In this paper, we classify the input variable into 2 classes of initial configuration and objective configuration, $m = 2, d = 6, d_1 = d_2 = 3$.

The selection matrix is

$$\begin{aligned} S_1 &= (I_3 \quad O_3), \\ S_2 &= (O_3 \quad I_3), \end{aligned} \quad (33)$$

where I_3 and O_3 are the 3×3 unit matrix and zero matrix, respectively.

Consequently, the similarities of the initial and objective configurations are

$$\begin{aligned} Sim_1(i) &= \|S^1(x^{(i,j)} - x)\|^{-2} \quad (i = 1, \dots, N_{p0}; j = 1), \\ Sim_2(j) &= \|S^2(x^{(i,j)} - x)\|^{-2} \quad (j = 1, \dots, N_{pc}; i = 1). \end{aligned} \quad (34)$$

5.2.2 Classified multivariate k-NN (CM-k-NN)

k-NN is a local linear approximator of a function based on a set of known (state, parameters) pairs. k groups of state

with the largest similarity are selected and the corresponding parameters are weighted to generate new parameters. As to the CM-k-NN, k nearest initial configurations to q_0 and k nearest objective configurations to q_c are selected and paired, then $k \times k$ groups of parameters are weighted to obtain the new parameters:

$$C_{k-NN}(x) = \sum_{\substack{i \in I_1^{NN} \\ j \in I_2^{NN}}} w(i, j) \cdot C^{i,j}, \quad (35)$$

where $I_1^{NN} \in \mathbb{R}^k$ and $I_2^{NN} \in \mathbb{R}^k$ are the sets of the indexes with largest similarities in Sim_1 and Sim_2 , respectively. $w(i, j)$ is the weighting coefficient, which is calculated by the inverse-distance weighting:

$$w(i, j) = \frac{Sim_1(i)}{2k \cdot \sum_{i \in I_1^{NN}} Sim_1(i)} + \frac{Sim_2(j)}{2k \cdot \sum_{j \in I_2^{NN}} Sim_2(j)}. \quad (36)$$

And

$$\sum_{\substack{i \in I_1^{NN} \\ j \in I_2^{NN}}} w(i, j) = 1. \quad (37)$$

5.2.3 Classified multivariate LWR (CM-LWR)

Compared to k-NN which uses only a part of data in the database, LWR is required to use the whole data. The weighting coefficient is reduced sharply as the similarity decreasing:

$$C_{LWR}(x) = \sum_{\substack{i \in I_0 \\ j \in I_c}} w(i, j) \cdot C^{i,j}, \quad (38)$$

where I_1 and I_2 are the sets of the whole indexes of initial configurations and objective configurations, respectively.

Similar to the CM-k-NN, the weighting coefficient is determined by Sim_1 and Sim_2 :

$$w(i, j) = \frac{1}{W} e^{-\frac{Sim_1(i)^{-1} + Sim_2(j)^{-1}}{2\lambda^2}}, \quad (39)$$

$$W = \sum_{\substack{i \in I_1 \\ j \in I_2}} e^{-\frac{Sim_1(i)^{-1} + Sim_2(j)^{-1}}{2\lambda^2}}. \quad (40)$$

And

$$\sum_{\substack{i \in I_1 \\ j \in I_2}} w(i, j) = 1. \quad (41)$$

where λ represents the bandwidth, which determine the descent rate of w as the similarities decrease.

6 Results

We select $l_1 = 0.4$, $l_2 = 0.3$ and $l_3 = 0.4$ in Table 2. And the motion limits of the HFM are selected as shown in Table 4.

6.1 Trajectory planning based on the non-linear optimization

The trajectories obtained by non-linear optimization is shown by the solid lines in Fig. 10. The velocity, acceleration, torque and linear velocity satisfy the constraints. Additionally, Joint 1 and Joint 2 work smoothly. For each joint, a phase with approximate uniform velocity takes up a large proportion.

The average computation time for one optimization problem on the computer with Inter Xeon CPU E5-1603 2.8GHz is 137.51s, which is unable to achieve real-time performance.

We choose $\mathbf{q}_0 = [-0.3, -0.2, -1.7]$, $\mathbf{q}_c = [1.3, 0.5, -0.9]$, and $\alpha = 10$. When $\mu = 0.5$, the motion times (t_f) and power consumptions (P) of the optimal trajectories with different μ s are shown in Table 5. As μ increasing, the motion time increases while the power consumption decreases. When $\mu = 0$, t_f reaches the upper limit. However, when $\mu = 1$, t_f is unable to reach the lower limit. As the velocity, acceleration and linear velocity reach the limits, t_f is unable to further reduce to satisfy all the constraints.

6.2 Database of optimal trajectories

We randomly select 50000 groups of joint values in the motion ranges shown in Table 4. The reachable workspace of

Table 5. t_f and P with different μ

μ	t_f (s)	P (J)
0	2	23.41
0.2	2	23.41
0.4	2	23.41
0.6	1.56	27.82
0.8	1.33	32.80
1.0	1.33	35.30

Fig. 8. The Reachable Workspace and the control panel area of the HFM

the HFM is obtained and divided according to Section 3.3.1 (shown as Fig. 8), in which the control panel area is extracted (depicted with red points in Fig. 8, approximated as a $0.6 \times 0.5m^2$ rectangle).

We randomly select $N_{p0} = 100$ initial points and $N_{pc} = 25$ objective points in S_E and S_C , respectively. As shown in Fig. 9, the initial points and the objective points are depicted with green and red points respectively.

The initial and objective points are paired and the 2500 corresponding configuration pairs are served as the input of the non-linear optimization problem, which is solved to obtain 2500 groups of parameters. The total computation time for database generation is about 108 hours.

6.3 Comparison of two regression prediction method

6.3.1 Motion curve

We choose $\mathbf{q}_0 = [-0.53, -0.13, -1.71]$ and $\mathbf{q}_c = [-0.23, -0.40, -1.12]$ and obtain motion curves by non-linear optimization, CM-k-NN and CM-LWR respectively,

Table 4. Motion Limits of the HFM

Joint	Motion Range ($^{\circ}$)	$\omega_{max} (^{\circ}/s)$	$\ddot{q}_{max} (^{\circ}/s^2)$	$\tau_{max}(Nm)$
Joint 1	-90 $\sim$ 90	100	860	180
Joint 2	+45 $\sim$ 45	100	860	180
Joint 3	-135 $\sim$ 0	100	860	180

Fig. 9. Database of the initial and objective points

shown as Fig. 10. The motion curves obtained by the CMR methods are approximate to the ones obtained by the non-linear optimization and the constraints are satisfied as well.

6.3.2 Precision analysis

We select $N_r = 20$ input states $\mathbf{x}$ randomly and choose 2-norm as the measure of similarity:

$$Sim_l = \|\mathbf{S}^l(\mathbf{x}^{(i,j)} - \mathbf{x})\|_2^{-2} \quad (l = 1, 2). \quad (42)$$

The initial parameters $\tilde{\mathbf{C}}$ are first obtained by CM-k-NN and CM-LWR respectively. Then the optimal objective function value F_p and the corresponding parameters $\mathbf{C}_p$ are obtained by non-objective optimization.

The cost increase is

$$e_F = \left| \frac{F_p - F_a}{F_a} \right|, \quad (43)$$

where F_a is the optimal objective function value obtained by non-linear optimization.

For the k-NN method, the average e_{FS} with different k s are shown by the blue points in Fig. 13. As k increases, the average e_F first decreases and then increases. As only a part of data are used, when k is relatively small, the database is not employed sufficiently, which can lead to large errors. 1-NN is equivalent to a look-up table, which has the largest error. When k is very large, large amount of data are used. However, the coefficient chosen by Equ. 44 reduces slowly as the similarity decreases (shown as Fig. 11a), which means the states with small similarities take a large proportion in

new parameters generation and lead to large error. When $k = 6$, the smallest error is obtained, $e_F = 6.56\%$.

For the LWR method, the average e_{FS} with different λ s are shown by the blue points in Fig. 14. As λ increases, the average e_F first decreases and then increases. Unlike k-NN, LWR employs the whole data in the database. The coefficient chosen by Equ. 39 reduces as the similarity decreases, and the reduction rate decreases as λ increases, as shown in Fig. 11. When λ is relatively small, states with large similarities take quite a large proportion, similar to 1-NN, shown as Fig. 11b. When $\lambda = 0.5$, the smallest error is obtained, $e_F = 5.25\%$.

We amend the k-NN method by choosing different k s for different classes of the input. For a new input state $\mathbf{x} = (\mathbf{q}_0, \mathbf{q}_c)$, k_0 groups of $\mathbf{q}_0$ and k_c groups of $\mathbf{q}_c$ are selected in the database to generate new parameters. The weighting coefficient is

$$w(i, j) = \frac{Sim_1(i)}{2k_c \cdot \sum_{i \in I_1} Sim_1(i)} + \frac{Sim_2(j)}{2k_0 \cdot \sum_{j \in I_2} Sim_2(j)}. \quad (44)$$

When $k_0 = N_{p0}$ and $k_c = N_{pc}$, the k_0 - k_c -NN method employs the whole data and is equivalent to the LWR method with the different weighting coefficient chosen manner. The variation trend of this weighting coefficient is similar to the LWR with larger λ , shown as Fig. 11a.

6.3.3 Time performance

The average regression time (denoted as T_R) of k-NN increases as k increases, as shown in Fig. 12. When $k = 6$, the minimum objective value is obtained, $T_R = 0.58ms$. The maximum T_R is $1.71ms$ ($k = 25$).

The average regression time, non-objective optimization time (denoted as T_O) and the total calculation time (denoted as T_T) of 6-NN and LWR are shown as Table 6. The average non-objective optimization time, and even the total calculation time are far smaller than the average calculation time for the original optimization problem (137.51s). And real-time performance can be achieved by the regression prediction method.

In conclusion, compared with k-NN, LWR has better predicting precision, but costs longer time for regression.

6.4 Influence of database size

According to Section 6.2, it takes a large amount of time to generate the database. When the database size is reduced,

Fig. 10. Motion curve (solid: non-linear optimization, dashed: CM-6-NN, dot-dashed: CM-LWR; red: Joint 1, green: Joint 2, blue: Joint 3)

Fig. 11. Variation trend of the weighting coefficient

the calculation time decreases, but the prediction precision will decrease as well. To balance the precision and time performance, databases with different sizes are employed to analyze the trade-off between them.

Three databases with 100×25 , 60×15 and 20×5 samples are selected. The average cost increases of k-NN and LWR with three databases are shown as Fig. 13 and Fig. 14,

respectively. The precision with smaller database is lower. The minimum average cost increase with three databases are shown in Table 7. When database size decreases, for LWR, the increase of e_F is small. When $\lambda = 0.5$, $e_F = 5.64\%$ (with 20×5 -database), which is smaller than the minimum e_F obtained by k-NN with 100×20 -database. For k-NN, the minimum e_F is 9.13% with 20×5 -database, which is still below

Fig. 12. T_R of k-NN with different k s

Table 6. Calculation time

Time	6-NN	LWR
$T_R(ms)$	0.58	9.87
$T_O(ms)$	12.45	12.81
$T_T(ms)$	13.03	22.68

Fig. 13. e_F with different database sizes (k-NN)

10%. Therefore, the database size can be appropriately reduced when it has no strict requirement of precision, to greatly reduce the calculation time while keep a reasonable error rate.

The regression times with different database sizes are shown in Table 8. When the database size reduces, T_R of k-NN has little change (always very small) while T_R of LWR greatly reduces.

In conclusion, high efficiency can be achieved when select 20*5-database and employ the LWR method with $\lambda = 0.5$. The relative high precision can be obtained with smaller calculation time.

Fig. 14. e_F with different database sizes (LWR)

Table 7. e_F with different database sizes

Database size	k-NN (%)	LWR (%)
100*25	6.56 ($k = 6$)	5.25 ($\lambda = 0.5$)
60*15	7.65 ($k = 10$)	5.37 ($\lambda = 0.5$)
20*5	9.13 ($k = 4$)	5.64 ($\lambda = 0.5$)

Table 8. T_R with different database sizes

Database size	k-NN (ms)	LWR (ms)
100*25	0.58 ($k = 6$)	9.87
60*15	0.60 ($k = 10$)	2.74
20*5	0.49 ($k = 4$)	0.71

7 Conclusion

A real-time trajectory generation method based on non-linear optimization and regression prediction is presented in this paper.

1. A haptic feedback system based on the servo serial manipulator is constructed and the type and parameters of the HFM is selected. Additionally, a pHRI safety strategy is presented. The limitation of motion range, motion time and linear velocity of the end-effector is determined, which is essential to pHRI applications.
2. Trajectory planning based on non-linear optimization is performed. The objective function is constructed to balance the motion time and power consumption, and the constraints are constructed according to the mechanical limits and the safety of pHRI. Feasible and safe trajectories are obtained to actuate the robot move to the objective configuration faster with less power consumption.
3. Optimal trajectories are generated in real-time by generating database of optimal trajectories off-line and re-

gression prediction on-line. CMR method is presented, which is suitable to the situations that elements of input variable belong to different classes and need to be considered separately. Two local non-parameter regression methods, k-NN and LWR, are performed combining with CMR. And the precision and time performance, as well as the influence of the database size are analyzed. LWR has better predicting precision, while k-NN works faster. Better real-time performance is achieved with a reasonable error rate by appropriately reducing the database size.

There are still some limits in this paper.

1. This paper focuses on 3-DOF trajectory planning which actuates the end-effector of the HFM move to the objective position. In the future work, we would like to consider the last three DOFs to control the orientation of the end-effector.
2. The trajectory is encoded by 5-order polynomials, lacking of the uniform motion phase. The velocity increases slowly and the max velocity phase is short. In the future work, we would like to use the piecewise interpolation method to encode trajectories and obtain the better motion efficiency.

References

- [1] Dai, S., Lei, X., and Mei, J., 2002. "Virtual cockpit system". *Journal of System Simulation*, **14**(4), Apr., pp. 488–492.
- [2] Wang, D., Jiao, J., Zhang, Y., and Zhao, X., 2016. "Computer haptics: Haptic modeling and rendering in virtual reality environments". *Journal of Computer-Aided Design & Computer Graphics*, **28**(6), June, p. 881–895.
- [3] Tang, Y., Gu, H., and Li, P., 2012. "Augmented semi-virtual reality cockpit technology". *Journal of Traffic and Transportation Engineering*, **12**(2), Apr., pp. 62–69.
- [4] CGSD, 1998. Tactile feedback for a force-reflecting haptic display. Tech. rep., Computer Graphics Systems Development Corporation.
- [5] Stonum, R., 1998. A survey of immersive technology for maintenance evaluations. Tech. rep., Battelle Memorial Institute.
- [6] Kim, K. W., Kim, H. S., Choi, Y. K., and Park, J. H., 1997. "Optimization of cubic polynomial joint trajectories and sliding mode controllers for robots using evolution strategy". In Proc. IECON, Vol. 3, pp. 1444–1447.
- [7] Bazaz, S. A., and Tondur, B., 1997. "Online computing of a robotic manipulator joint trajectory with velocity and acceleration constraints". In Proc. ISATP, IEEE, pp. 1–6.
- [8] Lampariello, R., and Hirzinger, G., 2013. "Generating feasible trajectories for autonomous on-orbit grasping of spinning debris in a useful time". In Proc. IROS, IEEE, pp. 5652–5659.
- [9] Liu, X., Meng, Z., Ni, J., and Zhu, Z., 2015. "Trajectory planning algorithm for hydraulic servo manipulator of three freedom". *Journal of Zhejiang University (Engineering Science)*, **49**(9), Sept., pp. 1776–1782.
- [10] Miossec, S., Yokoi, K., and Kheddar, A., 2006. "Development of a software for motion optimization of robots-application to the kick motion of the hrp-2 robot". In RProc. ROBIO, IEEE, pp. 299–304.
- [11] Bäuml, B., Wimböck, T., and Hirzinger, G., 2010. "Kinematically optimal catching a flying ball with a hand-arm-system". In Proc. IROS, IEEE, pp. 2592–2599.
- [12] Schulman, J., Ho, J., Lee, A. X., Awwal, I., Bradlow, H., and Abbeel, P., 2013. "Finding locally optimal, collision-free trajectories with sequential convex optimization". In Proc. Robotics: science and systems.
- [13] von Stryk, O., and Schlemmer, M., 1994. "Optimal control of the industrial robot manutec r3". *Computational Optimal Control*, pp. 367–382.
- [14] Chettibi, T., Lehtihet, H., Haddad, M., and Hanchi, S., 2004. "Minimum cost trajectory planning for industrial robots". *European Journal of Mechanics-A/Solids*, **23**(4), July–Aug., pp. 703–715.
- [15] Pan, J., Chen, Z., and Abbeel, P., 2014. "Predicting initialization effectiveness for trajectory optimization". In Proc. ICRA, IEEE, pp. 5183–5190.
- [16] Jetchev, N., and Toussaint, M., 2013. "Fast motion planning from experience: trajectory prediction for speeding up movement generation". *Autonomous Robots*, **34**(1), Jan., pp. 111–127.
- [17] Hauser, K., 2017. "Learning the problem-optimum map: Analysis and application to global optimization in robotics". *IEEE Transactions on Robotics*, **33**(1), Feb., pp. 141–152.
- [18] Lampariello, R., Nguyen-Tuong, D., Castellini, C., Hirzinger, G., and Peters, J., 2011. "Trajectory planning for optimal robot catching in real-time". In Proc. ICRA, IEEE, pp. 3719–3726.
- [19] Werner, A., Trautmann, D., Lee, D., and Lampariello, R., 2015. "Generalization of optimal motion trajectories for bipedal walking". In Proc. IROS, IEEE, pp. 1571–1577.
- [20] Denavit, J., and Hartenberg, R. S., 1955. "A kinematic notation for lower-pair mechanisms". *ASME Journal of Applied Mechanics*, **22**, pp. 215–221.
- [21] De Santis, A., Siciliano, B., De Luca, A., and Bicchi, A., 2008. "An atlas of physical human-robot interaction". *Mechanism and Machine Theory*, **43**(3), Mar., pp. 253–270.
- [22] Corrales, J., Candelas, F., and Torres, F., 2011. "Safe human-robot interaction based on dynamic sphere-swept line bounding volumes". *Robotics and Computer-Integrated Manufacturing*, **27**(1), Feb., pp. 177–185.
- [23] Pereira, A., and Althoff, M., 2015. "Safety control of robots under computed torque control using reachable sets". In Proc. ICRA, IEEE, pp. 331–338.
- [24] Rastegar, J., and Perel, D., 1990. "Generation of manip-

ulator workspace boundary geometry using the monte carlo method and interactive computer graphics”. *Journal of mechanical design*, **112**(3), pp. 452–454.

- [25] International Organization for Standardization, 2010. “Robots and robotic devices-collaborative robots”. *Technical Specification ISO/TS 15066*.
- [26] Hargraves, C. R., and Paris, S. W., 1987. “Direct trajectory optimization using nonlinear programming and collocation”. *Journal of Guidance, Control, and Dynamics*, **10**(4), pp. 338–342.