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

This is the published version of a paper published in .

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

Zhang, S., Dai, S. (2019)

Real-time kinematical optimal trajectory planning for haptic feedback manipulators

Simulation, 95(7): 621-635

<https://doi.org/10.1177/0037549718815755>

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-77208>

Real-time kinematical optimal trajectory planning for haptic feedback manipulators

Shiyu Zhang and Shuling Dai

Simulation: Transactions of the Society for Modeling and Simulation International
2019, Vol. 95(7) 621–635
© The Author(s) 2018
DOI: 10.1177/0037549718815755
journals.sagepub.com/home/sim

Abstract

To obtain real-time haptic interactions in virtual cockpit systems (VCSs), a real-time trajectory planning method based on kinematical optimization for haptic feedback manipulators (HFMs) is presented in this paper. Firstly, the control panel area is extracted in the workspace of the HFM, in which the interacting point is located. Then a feasible interacting configuration is calculated as the objective configuration of the trajectory encoded by a parametric representation. The trajectory planning problem is formulated as a non-linear optimization problem based on kinematics, which is solved in real-time by finding a good initial solution with machine learning methods. Simulations show that trajectories with a compromise between safety and rapidity can be calculated in real-time by this method, which provides a basis for haptic interaction in VCSs.

Keywords

Real-time trajectory planning, haptic feedback, virtual cockpit system, kinematical optimization, machine learning

1. Introduction

The virtual cockpit system (VCS) is a flight simulator with human–computer interaction based on virtual reality technology, in which the physical cockpit and control mechanisms are replaced by virtual display and motion-tracking devices. Compared with the physical cockpit system, the advantages of higher flexibility in the structure and functionality, as well as the lower cost, make the VCS meaningful in flight simulation.¹

Haptic feedback is of great significance for better immersion and interaction in VCSs. However, efforts in this direction are still very limited.² In order to achieve haptic interactions in the VCS without sacrificing the advantages of low cost and high flexibility, a haptic feedback system based on the servo serial manipulator is presented. On the basis of the traditional VCS, a mechanical system including a serial manipulator (called a haptic feedback manipulator, HFM) and a concise control panel equipped with different kinds of button linked with the end-effector is placed in front of the user. The position and posture of the user's hand are measured by motion-tracking sensors as the hand moves. Once the interactive operation is detected, the user is able to see the virtual hand operating the control panel in the virtual environment through the virtual display. Meanwhile, the HFM brings

the end-effector with the control panel to the objective position the user intends to operate with to provide haptic feedback.

To achieve interactions with human hands, trajectory planning for the HFM according to hand motion is required, which brings the corresponding button on the end-effector to the predicted interacting point at the interacting time. Two key problems need to be solved.

- (1) Interacting configuration determination, that is, determination of the objective configuration of trajectory planning. The objective of trajectory planning is to make the manipulator move from the current configuration to the interacting configuration corresponding to the interacting point, which is located in the simulated control panel area. Therefore, we need to analyze the

State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, China

Corresponding author:

Shuling Dai, State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, XueYuan Road No.37, HaiDian District, Beijing, China.

Email: sldai@buaa.edu.cn

workspace of the HFM, extract the control panel area, and consequently determine the interacting point and its corresponding configuration.

- (2) Real-time trajectory planning. There are three key points. (a) Real-time performance. The trajectory needs to be re-calculated according to the variation of the predicted interacting configuration. One trajectory planning needs to be done in one predicting period. (b) Rapidity of motion. The motion time of the HFM is required to be as short as possible to make sure that the end-effector can reach the objective point before the human hand to provide haptic feedback. (c) Safety of motion. The positions, velocities, accelerations, and torques of the joints of the HFM need to be restricted to avoid injuries to users.

To satisfy the above requirements, firstly, we analyze the workspace of the HFM and extract the optimal control panel area, which is used to determine the interacting point and the corresponding interacting configuration. Then, a non-linear optimization model is established on the rapidity and safety criteria, in which the trajectory is encoded by a parametric representation. Finally, the global minimum is solved by finding a good initial solution through the regression model, which is obtained by machine learning methods with the database generated offline.

1.1. Related work

1.1.1. Haptic feedback in the VCS. To achieve haptic feedback in the VCS, Semi-Virtual Reality Cockpit solutions have been proposed^{3,4} based on the principle that “Seeing is virtual and touching is believing.” The scenes in and out of the cockpit are created by the virtual reality technology, while all touchable parts keep 1:1 proportions to provide haptic feedback. However, due to the large size and poor flexibility, these solutions sacrifice the advantages of the VCS.

Another VCS scheme was developed by STRICOM with haptic feedback achieved by the TOPIT (Touched Objects Positioned in Time) technology.^{5,6} A mechanical system equipped with various kinds of buttons is placed in front of the user. Each button is used to simulate all the buttons of the same type in the real cockpit. When the 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 button can only move in a two-dimensional plane area so that interactions in three-dimensional space are barely achieved.

We improve the TOPIT scheme by employing a serial manipulator as the servo mechanical system, which can

achieve three-dimensional movement with smaller size, larger workspace, and better flexibility.

1.1.2. Trajectory planning for the serial manipulator. The traditional point-to-point trajectory planning is started from interpolation-based methods, such as polynomial interpolation^{7,8} and B-spline interpolation.^{9,10} In general, pure interpolation-based methods are able to accomplish the required tasks, but fail to achieve optimal performance in specific aspects. To overcome this shortcoming, some trajectory planning methods are presented based on non-linear optimization.^{11,12} The non-linear optimization problem is developed with an optimal objective based on time, energy, and power consumption. Constraints such as mechanical structure, time, and obstacle avoidance¹³ are considered as well. Von Stryk and Schlemmer¹⁴ investigated 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. Chettibi et al.¹⁵ presented the optimal planning problem and tried to find a trade-off between time, energy, and power consumption and solved it using the Sequential Quadratic Programming (SQP) method. However, none of the aforementioned optimization-based methods are real-time due to the complex computation of non-linear optimization.

Bäumel et al.¹⁶ presented three pure kinematical objective functions and implemented parallel computation with multiple initial solutions to obtain the global minimum in real-time. However, their method has a stringent demand for hardware due to the parallel computation with 32 CPU cores. Lampariello et al.¹⁷ and Werner et al.¹⁸ take the dynamics into consideration and consider the criterion of minimum power consumption. Several machine learning methods are adopted to find the global minimum in real-time. However, this method is only suitable for the case of a fixed initial robot configuration and a fixed starting point of the target trajectory.

Recently, a different approach to generate trajectories originated from trajectory learning and generalization, learning-by-demonstration (LbD),^{19,20} has been presented. The motion of a manipulator is modeled as a dynamical system^{21,22} whose parameters are learned from demonstration by a human. With this method, natural human-like movement can be achieved by robots. Furthermore, it is suitable for a dynamic environment with obstacles and disturbances, since it is able to model the non-linear and uncertain factors in a mechanical system. However, compared with the optimization-based method, it is unable to obtain the optimal solution according to some criteria.

To achieve the safety and real-time performance of the haptic feedback system, we combine the trajectory planning method based on non-linear optimization with the

Figure 1. Components of the haptic feedback system based on the servo serial manipulator. HFM: haptic feedback manipulator.

real-time solving method based on machine learning. The non-linear optimization problem is established with the criterion of safety and real-time performance to find the trade-off between them. Machine learning methods are implemented for choosing the initial guess of the solver to overcome the high computational complexity of non-linear optimization.

2. Haptic feedback system based on the servo serial manipulator

2.1. Haptic feedback system

A haptic feedback system for the VCS based on the servo serial manipulator is introduced in this paper. As shown in Figure 1, it consists of motion-tracking sensors, computers, and a serial manipulator, called the HFM, linked with a concise control panel. Motion-tracking sensors are used to track hand motion and detect interactions. The concise control panel linked with the end-effector of the HFM is equipped with various kinds of buttons, such as a pressing button, rotating button, and sliding button, which provide haptic feedback to users. Computers work to perform hand trajectory prediction, trajectory planning, and control for the HFM.

The workflow is shown in Figure 2. When the user's hand moves, hand motion data are measured by tracking sensors, and then processed by computers to calculate the position and pose, predict the trajectory of the user's hand, and determine the interacting time t_i and the interacting point p_i . Trajectory planning is performed and the HFM is actuated to bring the corresponding button on the control panel to p_i before t_i . The trajectory of the HFM needs 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 moment of interaction, the user is able to see the virtual hand operating the

control panel in the virtual environment through the virtual display while the corresponding button on the control panel reaches the spiral position of the object the user intends to operate with to provide haptic feedback.

This haptic feedback scheme needs merely a concise control panel with several buttons instead of the full-scale control panel in a real cockpit, for which the size and cost are greatly reduced. Compared with TOPIT, the spatial movement in three dimensions can be achieved with a smaller size and larger workspace by employing a serial manipulator. In addition, different types of aircraft can be simulated only by changing the specific software, making it more flexible in functionality.

2.2. Haptic feedback manipulator

A six-degree-of-freedom (6-DOF) serial manipulator is adopted as the HFM in this paper (Figure 3). The six joints of the serial manipulator are the waist, shoulder, elbow, pitching wrist, yawing wrist, and rotating wrist joint. The position of the end-effector is determined by the waist, shoulder, and elbow joints, while the orientation is determined by the three wrist joints. The pitching wrist and yawing wrist joints are used to adjust the orientation of the concise control panel. In addition, the rotating wrist joint is used to switch the 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 Denavit–Hartenberg (DH) convention²³ is adopted to describe the kinematic chain of a manipulator, as shown in Figure 4. $O_0 - X_0 - Y_0 - Z_0$ 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. 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

Figure 2. Workflow of the haptic feedback system. HFM: haptic feedback manipulator.

Figure 3. The structure of the haptic feedback manipulator. 1: waist; 2: shoulder; 3: elbow; 4: pitching wrist joint; 5: yawing wrist joint; 6: rotating wrist joint.

Figure 4. Denavit–Hartenberg coordinate system of the haptic feedback manipulator.

Table 1. 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

relative rotation between two adjacent links, and d_i is the offset distance along the joint axis, as illustrated in Table 1.

3. Workspace analyses and interacting configuration calculation

3.1. Workspace generation and control panel area extraction

According to the workflow of the haptic feedback system shown in Section 2.1, we first need to predict an interacting point, which serves as the objective of trajectory planning for the HFM. To achieve the consistency of the virtual environment and real word, we need to extract an area in the reachable workspace of the HFM to simulate the control panel, in which the interacting points are located. In addition, as users and the HFM work in a shared workspace and perform physical interaction, the motion range of the HFM is influenced by hand motion to guarantee the reality and safety of human–robot interaction. Therefore, it is necessary to perform workspace analysis of the HFM.

In real cockpits, multiple control panels are distributed almost in all the directions around the pilot. This paper aims at theoretical research on trajectory planning of the haptic feedback servo system and does not consider any specific airplane model type. Top panels and lateral panels are thus neglected. We take the control panel located right forward, for example, and simplify the control panel as a flat rectangular panel on which button distribution is designed artificially.

Firstly, the reachable workspace of the HFM is generated by the Monte Carlo method, which is denoted as $S \subset \mathbb{R}^3$. As described in Section 2.2, the position of the end-effector is determined by the waist, shoulder, and elbow joints. Thus, only three individual variables

Figure 5. Interaction of the end-effector and the user's hand. (Color online only.)

Figure 6. Control panel plane and control panel area.

Figure 7. End-effector trajectory of the haptic feedback manipulator.

$\{\theta_1, \theta_2, \theta_3\}$ are considered to generate the workspace. Multiple sets of $\{\theta_1, \theta_2, \theta_3\}$ are randomly chosen in the allowable motion ranges. Then, the corresponding positions of the end-effector relative to $O_0 - X_0 - Y_0 - Z_0$, denoted as $p \in \mathbb{R}^3$, are calculated by the forward kinematics and plotted as a point cloud. The reachable workspace can be finally approximated by the point cloud by generating enough random points.

Then, the reachable workspace is divided into parts A and B by plane C (called the control panel plane), as shown in Figure 5. The control panel area, in which all the contacts of the user's hand and the end-effector of the HFM performed (denoted as $S_C \subset \mathbb{R}^3$), is located in plane C, as shown in Figure 6. In case of disturbing hand movement and moreover to avoid danger, the motion range of the end-effector is restricted in side A, that is, the area before the control panel plane in reachable workspace. This area is defined as the effect workspace of the HFM (denoted as $S_E \subset \mathbb{R}^3$), which is encircled by a red plane and red dot line in Figure 5. Note that in the base coordinate frame the direction along the x -axis is assigned as the back, while the opposite direction is the front. The hand moves at side B. At the interacting time, the user's hand and the end-effector of the HFM interact at the interacting point located in S_C .

As shown in Figure 8, the reachable workspace is symmetrical with respect to the plane xOz . The z -axis is perpendicular to the ground. The user's hand gets close to the manipulator along the opposite direction of the x -axis. Therefore, we select a plane perpendicular to the x -axis as the control panel plane, which is symmetrical with respect to the plane xOz and is located right forward of the user. The workspace is divided into a series of layers perpendicular to the x -axis, in which a layer is selected as the control panel plane. As the layered result shows (Figure 9), the point distribution in the workspace is non-uniform and a cavity exists inside the workspace. The density of the points close to the center of the workspace tends to be larger, which indicates that the manipulator is more flexible. Likewise, the area of the cross-section becomes larger. However, there is a cavity inside the workspace, which is unreachable for the end-effector and thus the control panel area should be kept away from it. Consequently, to obtain a control panel plane with larger area and better joint flexibility, while keeping it away from the cavity, the first layer back to the cavity is selected as the control panel plane. Finally, a rectangular area with the largest area in this plane is extracted as the control panel area.

3.2. Interacting configuration determination

When the user's hand moves, the trajectory of hand motion is predicted by the haptic feedback system. The interacting point $p_i (p_i \in S_C)$ is obtained by intercepting the trajectory with the control panel plane. Accordingly, the interacting

configuration $q_i = \{q_1, q_2, \dots, q_6\}$ can be determined, which serves as the objective configuration of trajectory planning in Section 4.

According to Section 2.2, the position of the end-effector is only determined by the first three joints. Thus, $\{q_1, q_2, q_3\}$ can be calculated by the inverse kinematics. Then q_4, q_5 , and q_6 are calculated according to the direction of the control panel.

To guarantee the control panel is perpendicular to the x -axis, we have the following:

$$q_4 = -(q_2 + q_3), \quad (1)$$

$$q_5 = -q_1, \quad (2)$$

and according to the required direction of the particular button, q_6 is selected:

$$q_6 = \{0, \pi\}. \quad (3)$$

In addition, if more than one feasible solution of q_i exists, we select the one that has the least deviation from the current configuration as the objective configuration.

4. Real-time trajectory planning based on kinematic optimization

4.1. Problem statement

When the haptic feedback system works, the trajectories of the hand movement and the interacting point are predicted according to the current hand motion periodically with the period T_p . A sequence of predicted interacting points $(p_i^0, p_i^1, \dots, p_i^n) \subset S_E$ is obtained. As the hand moves, the actual workspace converges gradually and the end-effector finally reaches the real interacting point. 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 Figure 7, in which the black lines represent the trajectories generated in every period and the red line represents the actual trajectory of the whole process.

As for the trajectory planning in each period, it is necessary to generate a trajectory from the initial configuration q_0 to the objective configuration q_i , with the initial velocity ω_0 . 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 the parameters. Trajectory planning is formulated as a non-linear optimization problem. The objective function $F(C)$ is defined considering the criteria of safety and motion time. Mechanical and time constraints are applied. We need to find a trajectory subjected to all the constraints to minimize $F(C)$. The optimization problem can be treated as a mapping from (q_0, q_i, ω_0) to C :

$$(q_0, q_i, \omega_0) \mapsto C^* = \min_C F(q_0, q_i, \omega_0, C). \quad (4)$$

The trajectory planning problem is generally costly due to its strong non-linearity and the requirement of frequent re-planning. To achieve the real-time performance, some assumptions and approximations are made to simplify the problem.

- (1) The trajectory planning is implemented in *joint space*. Three joints of the HFM are required to reach the objective configuration at the same time, which are planned independently. Compared with Cartesian space, joint space planning has low computational complexity without large amount of inverse kinematics calculations, since the controller actuates joint motion in joint space. In addition, it is convenient to implement mechanical constraints, for example, constraints of position, velocity, and acceleration of joints. Furthermore, mechanism singularity can be avoided.
- (2) A pure *kinematical optimization* is chosen, in which the objective function and constraints depend only on positions, velocities, and accelerations of joints, not on the torques. Dynamics is significant in the motion of the manipulator, but complicated in computation and costly in time, which are unsuitable for real-time applications. As for haptic feedback application, the joint velocity constraints dominate the manipulator motion, since the motion time is short and the acceleration and deceleration phases take up a slight proportion. Therefore, the details of acceleration and deceleration are less important for overall performance and we just need to conservatively choose the maximally allowed acceleration to ensure that the torque never exceed its limit.
- (3) The movements of the HFM are encoded by *trapezoidal velocity ramps*. As argued above, the precise characteristics of the acceleration and deceleration phases are not essential for the overall performance. Trapezoidal ramps can be easily analytically expressed with a few independent parameters, and significantly simplify the computation since only acceleration and maximum velocity need to be limited.

4.2. Trajectory parameterization

Trapezoidal velocity ramps are employed to encode the trajectories, which consist of three phases as follows: uniform acceleration phases $(0 - t_1)$; maximum velocity phase $(t_1 - t_2)$; and uniform deceleration phase $(t_2 - t_f)$. The acceleration and deceleration phases have the same acceleration a . The initial and maximum velocity are ω_0 and ω_m , respectively.

We define the following:

$$\hat{r}(t, t_1, t_2) = r(t_1 - t) - r(t - t_2), \quad (5)$$

where $r(t)$ is the step function:

$$r(t) = \begin{cases} 1 & t \geq 0 \\ 0 & t < 0 \end{cases} \quad (6)$$

The trajectory of each joint can be represented as follows:

$$\begin{aligned} \ddot{q}(t) &= a\hat{r}(t, t_1, t_2), \\ \dot{q}(t) &= \dot{q}_0 + a\hat{r}(t, t_1, t_2)t, \\ q(t) &= q_0 + \dot{q}_0 t + \frac{1}{2}a\hat{r}(t, t_1, t_2)t^2. \end{aligned} \quad (7)$$

For each joint, the trapezoidal ramp can be defined by two independent parameters: a and t_f . Considering N_J -DOF robots, we need an N_J -dimensional acceleration parameter, $a \in \mathbb{R}^{N_J}$. In addition, to ensure the joint reaches the objective configuration simultaneously, only one parameter for motion time is needed, $t_f \in \mathbb{R}$. Therefore, $N_C = N_J + 1$ parameters are required for this optimization problem in total, $C = (a, t_f)$.

The parametrized trajectory is represented as follows:

$$q(a, t_f, t) = q_0 + \frac{1}{2}a\hat{r}(t, t_1, t_2)t^2. \quad (8)$$

4.3. Kinematical non-linear optimization

4.3.1. Optimization problem. The non-linear optimization problem with non-linear constraints can be described as follows:

$$\begin{aligned} (a^*, t_f^*) &= \min F(a, t_f) \\ \text{subject to} \\ a &\in \mathbb{R}^N, t \in \mathbb{R} \\ h_i(a, t_f) &= 0, (i = 1, 2, \dots, N_h) \\ g_i(a, t_f) &\geq 0, (i = 1, 2, \dots, N_g), \end{aligned} \quad (9)$$

where F is the objective function and h_i and g_i are the equality and inequality constraints, respectively. The dimension of optimization space is $N_J + 1$.

4.3.2. Objective function. Three optimization modes are presented in this section.

(1) Acceleration minimization

Safety is one of the most important factors in human–manipulator interaction applications. To ensure safety, the HFM is required to move softly with accelerations that are as small as possible:

$$F(a, t_f) = \sqrt{\frac{1}{N} \sum_{i=1}^{N_J} \left(\frac{a_i}{a_{i, \max}} \right)^2}, \quad (10)$$

where a_i is the acceleration of the i th joints and $a_{i, \max}$ is the maximally allowed acceleration of the i th joint.

(2) Motion time minimization

To achieve rapidity of motion, the HFM is required to reach the objective configuration quickly with motion time as small as possible:

$$F(a, t_f) = t_f. \quad (11)$$

(3) Compromise of acceleration and time

Simulation results show that the acceleration minimization mode is usually mechanically harmful if working in the acceleration phase all the time (Figure 11(a)) or working with maximum velocity (Figure 11(b)). In addition, working with small acceleration requires large motion time, which may reduce the rapidity performance. Similarly, the motion time minimization mode is less safe due to working with large acceleration (Figure 12).

To compromise the safety and rapidity of the motion, we consider a balance between acceleration and motion time as follows:

$$F(a, t_f) = \mu \frac{t_f}{t_{\max}} + (1 - \mu) \sqrt{\frac{1}{N_J} \sum_{i=1}^{N_J} \left(\frac{a_i}{a_{i, \max}} \right)^2}, \quad (12)$$

which can be converted to the following:

$$F(a, t_f) = \mu \left(\frac{t_f}{t_{\max}} \right)^2 + \frac{1 - \mu}{N_J} \sum_{i=1}^{N_J} \left(\frac{a_i}{a_{i, \max}} \right)^2, \quad (13)$$

where $t_{\max}$ is the maximally allowed motion time. μ is the weighting coefficient, representing the weighting of acceleration and motion time, and $0 \leq \mu \leq 1$. The cases $\mu = 1$ and $\mu = 0$ correspond to the motion time minimization mode and acceleration minimization mode, respectively.

4.3.3. Constraints. The joint offset from the current configuration to the objective configuration, that is, the interacting configuration, is as follows:

$$q_f = q_i - q_0. \quad (14)$$

The trapezoidal velocity ramps are satisfied to the following:

$$t_1 = \frac{\omega_m - \omega_0}{a}, \quad (15)$$

$$t_2 = t_f - \frac{\omega_m}{a}, \quad (16)$$

$$\omega_m = \omega_0 + at_1, \quad (17)$$

$$q_f = at_1(t_f - t_1) - \frac{1}{2}\omega_0(t_f - t_2 - t_1). \quad (18)$$

The positions, velocities, and accelerations of the joints are required not to exceed the mechanical limitations during the motion:

$$\begin{aligned} |q_i(t)| &< q_{i, \max}, i = 1, \dots, N_J, \\ |\dot{q}_i(t)| &< \dot{q}_{i, \max}, i = 1, \dots, N_J, \\ |\ddot{q}_i(t)| &< \ddot{q}_{i, \max}, i = 1, \dots, N_J. \end{aligned} \quad (19)$$

The HFM is required to reach the interacting configuration before the time user's hand reaches the interacting point:

$$t_f \leq t_i. \quad (20)$$

4.4. Real-time solver based on machine learning

The SQP method is adopted to solve the non-linear optimization problem to minimize the objective function described in Section 4.3.2 satisfying the constraints in Section 4.3.3. The solver easily becomes stuck in the local minimum due to the non-linearity of manipulator's kinematics. It would be difficult to obtain the global minimum by randomly choosing an initial solution. To overcome this problem, different initial guesses are chosen to generate a set of local minima, in which the best is taken as the global minimum. This method is rarely used in real-time applications since it takes a prohibitively long time to solve a large number of non-linear optimization problems. For example, it takes about 1 minute and 8 seconds to handle 625 and 81 groups, respectively, of initial guesses for three-degree-of-freedom (3-DOF) robots.

Therefore, a machine learning-based method is presented to choose a good initial guess of $N_J + 1$ parameters and find the global optimization in real-time. Firstly, multiple groups of (q_f, ω_0) are chosen in the work range. Then the corresponding global optima of (a, t_f) are calculated offline, which are used to fulfill the database of a set of (joint state, parameters). The mapping from joint state to parameters is consequently established by regression. During the online procedure, for a given (q_f, ω_0) , a group of parameters is initially obtained through the regression model, which serves as the initial solution $(\tilde{a}, \tilde{t}_f)$ of the optimization problem in Section 4.3. The non-linear optimization problem is finally solved by the SQP method to obtain the global optimum (a^*, t_f^*) .

The detailed procedure is as follows, in which Steps (1) and (2) are offline and Steps (3) and (4) are online.

- (1) Generate database.

N_d groups of the joint state (q_f, ω_0) are chosen in the work range. For the i th group of joint state (q_f^i, ω_0^i) , multiple initial guesses are chosen to calculate the local optimal parameters by SQP, in which the best is taken as the global optimum (a^i, t_f^i) . The N_d groups of joint variations and the corresponding global optimal parameters are collected to generate a database D :

$$D = (q_f^i, \omega_0^i, a^i, t_f^i)_{i=1}^{N_d}. \quad (21)$$

- (2) Establish the mapping from joint variations to parameters.

The motion time is the primary parameter we are concerned about, since the joints are required to reach the objective positions at the same time. Therefore, motion time t_f is first determined by establishing mapping from (q_f, ω_0) to t_f :

$$(q_f, \omega_0) \mapsto t_f = f_1(q_f, \omega_0). \quad (22)$$

Then other parameters are determined according to t_f and (q_f, ω_0) :

$$(q_f, \omega_0, t_f) \mapsto a = f_2(q_f, \omega_0, t_f). \quad (23)$$

- (3) Choose a good initial guess.

For an arbitrary (q_f, ω_0) , an initial guess for the non-linear optimization solver $(\tilde{a}, \tilde{t}_f)$ is chosen through the mapping obtained in Step (2):

$$\begin{aligned} \tilde{t}_f &= f_1(q_f, \omega_0), \\ \tilde{a} &= f_2(q_f, \omega_0, \tilde{t}_f). \end{aligned} \quad (24)$$

- (4) Calculate the global optimization.

The optimal parameters (a^*, t_f^*) are obtained by solving the non-linear optimization problem described in Section 4.3 with the SQP solver, in which $(\tilde{a}, \tilde{t}_f)$ obtained in Step (3) is used as the initial guess for iteration.

5. Results

5.1. Workspace analyses

The link parameters and motion limits of the HFM are chosen, as shown in Tables 2 and 3, respectively.

The workspace is simulated by the Monte Carlo method with 100,000 points. Examples of the obtained point clouds are shown in Figure 8.

The point cloud is divided into 40 layers along the x -axis and the projection on plane yOz of each layer is shown in Figure 9. It can be seen that a cavity exists in Layer 34

Table 2. Link parameters of the haptic feedback manipulator.

Link	α/rad	a/mm	θ/rad	d/mm
L1	$-\pi/2$	0	θ_1	150
L2	0	270	θ_2	0
L3	0	320	θ_3	0
L4	$\pi/2$	0	θ_4	0
L5	0	0	θ_5	0
L6	0	0	θ_6	0

Table 3. Motion limits of the haptic feedback manipulator.

Joint	Motion range ($^\circ$)	$\omega_{\max} (^\circ/\text{s})$	$\ddot{q}_{\max} (^\circ/\text{s}^2)$
Joint 1 (waist)	-135 to $+135$	100	860
Joint 2 (shoulder)	-5 to $+85$	100	860
Joint 3 (elbow)	-10 to $+95$	100	860
Joint 4 (pitching wrist)	-90 to $+90$	300	860
Joint 5 (yawing wrist)	-90 to $+90$	300	860
Joint 6 (rotating wrist)	-180 to $+180$	300	860

and vanishes in Layer 35. Therefore, Layer 35 is selected as the control panel plane and the largest rectangle in it is extracted as the control panel area, which is delineated by the red lines in Figure 10.

5.2. Real-time trajectory planning

5.2.1. Three optimization modes. To analyze the performance of non-linear optimization, we first consider $N_J = 3$. Then, we adopt this method to the 6-DOF HFM, which is detailed in Section 5.3.

Firstly, the velocity profiles of the three optimization modes are depicted.

(1) Acceleration minimization

The case $\mu = 0$ corresponds to the acceleration minimization mode.

We choose $q_f = (0.5, 0.4, 0.3)$, $\omega_0 = (0, 0, 0)$ and obtain $a = (0.2222, 0.1778, 0.1333)$, $t_f = 3.0000$. The velocity profile is as shown in Figure 11(a).

Figure 8. The reachable workspace of the haptic feedback manipulator. **Figure 9.** Projection of each layer of the workspace.

Figure 10. Control panel area. (Color online only.)

Figure 12. Velocity profile of the motion time minimization mode.

Figure 11. Velocity profile of the acceleration minimization mode.

We choose $q_f = (4.2, 1.1, 1.3)$, $\omega_0 = (0, 0, 0)$ and obtain $a = (2.9068, 0.4894, 0.5778)$, $t_f = 3.0000$. The velocity profile is as shown in Figure 11(b).

Note that in this mode, the acceleration is as small as possible and therefore the motion time is quite large so that it reaches the upper limit. In the case of Figure 11(b), for a large expected variation of Joint 1, it works with the maximum velocity for a long time to reduce the acceleration.

(2) Motion time minimization

The case $\mu = 1$ corresponds to the motion time minimization mode.

We choose $q_f = (0.5, 0.4, 0.3)$, $\omega_0 = (0, 0, 0)$ and obtain $a = (15.0000, 10.1196, 15.0000)$, $t_f = 0.4024$. The velocity profile is as shown in Figure 12. All three joints have large acceleration. In addition, both Joints 1 and 2 reach the maximum velocity to ensure rapidity.

(3) Compromise of acceleration and time

We choose $q_f = (0.5, 0.4, 0.3)$, $\omega_0 = (0, 0, 0)$ and $q_f = (4.2, 1.1, 1.3)$, $\omega_0 = (0, 0, 0)$, respectively, with the value of μ changing. The results are shown in Tables 4 and 5, respectively. With the increase of μ , the accelerations increase while the motion times decrease, as shown in Figure 13.

Comparing Tables 4 and 5, we can see that the joint variations of Joints 2 and 3 in Group 2 are larger than those of Group 1, but the accelerations are smaller. This is due to the increase of motion time caused by the significant large variation of Joint 1 in Group 2. It shows that the parameters are dominated by the joints with large variation.

5.2.2. Local minimum and initial guess. In the case $q_f = (0.5, 0.4, 0.3)$, $\omega_0 = (0, 0, 0)$, and $\mu = 0.5$, we uniformly choose 81 groups of initial guesses to solve the

Table 4. Results with different μ ($q_f = (0.5, 0.4, 0.3)$, $\omega_0 = (0, 0, 0)$).

μ	a	t_f
0	0.2222, 0.1778, 1.6554	3.0000
0.2	2.7590, 2.2072, 1.6554	0.8514
0.4	3.8328, 3.0663, 2.2997	0.7224
0.6	5.0245, 4.0196, 3.0147	0.6309
0.8	6.9206, 5.5272, 4.1454	0.5380
1	15.0000, 10.1196, 15.0000	0.4024

Table 5. Results with different μ ($q_f = (4.2, 1.1, 1.3)$, $\omega_0 = (0, 0, 0)$).

μ	a	t_f
0	2.9068, 0.4894, 0.5778	3.0000
0.2	5.8038, 0.6034, 0.7131	2.7005
0.4	8.0276, 0.6421, 0.7589	2.6178
0.6	10.4179, 0.6700, 0.7890	2.5672
0.8	14.3839, 0.6920, 0.8178	2.5216
1	15.0000, 5.3259, 5.3852	2.5167

non-linear optimization problem. The obtained local minima are sorted in ascending order, as shown in Figure 14. Choosing different initial guesses leads to different solutions, which indicates that the solution depends on the initial guess.

The minimum of these local solutions is 0.4979 (Group 1) and the maximum is 0.6509 (Group 81), which is 30.72% larger than Group 1. Therefore, it is necessary to

choose multiple groups rather than only one group of initial guesses to find the global minimum.

The first 20 groups of local minima increase slowly, in which the value of Group 20 is only 0.81% larger than that of Group 1. Therefore, the solution of Group 1 can be taken as the global minimum.

The more groups of initial guesses are adopted, the more accurate the obtained global minimum is, but the longer the calculation time will be. The relative errors and calculation times of different numbers of initial guesses are shown in Table 6. It can be seen that from 81 to 10,000 groups of initial guesses chosen, the relative errors are all very slight but the calculation time increases rapidly. Thus, we choose 81 groups of initial guess to reduce the computation cost as well as satisfy the precision.

5.2.3. Machine learning-based solver. Firstly, we randomly select 1200 groups of joint states in the motion range (shown in Table 3). For each group of joint states, 81 groups of initial guesses are used to calculate the global minimum. Then a database is generated with $N_d = 1200$ groups of (q_f, ω_0, a, t_f) . In this paper, the Support Vector Regression (SVR) and Gaussian Process Regression (GPR) methods are employed to establish the regression models $f_1^{SVR}(q_f, \omega_0)$, $f_2^{SVR}(q_f, \omega_0, t_f)$ and $f_1^{GPR}(q_f, \omega_0)$, $f_2^{GPR}(q_f, \omega_0, t_f)$.

By randomly choosing $N_r = 100$ groups of (q_f, ω_0) and calculating the optimal value of objective function F_p and the corresponding parameters (a_p, t_{fp}) with the machine learning-based solver described in Section 4.4, the relative errors are obtained:

Figure 13. Results with different μ .

Figure 14. The distribution of local minima corresponding to different initial guesses.

Table 6. Results of different numbers of groups.

Number of groups	Global minimum	Calculation time (s)	Relative error
81	0.497947	8.25	2.65826E-05
256	0.497936	27.51	4.10395E-06
625	0.497935	62.26	2.63489E-06
1296	0.497938	130.60	8.97901E-06
10,000	0.497934	1616.73	—

Table 7. Relative errors of objective function value and parameters.

Average relative error	SVR	GPR
F	0.47%	0.27%
t_f	0.45%	0.52%
a_1	2.22%	1.41%
a_2	8.87%	3.89%
a_3	3.12%	2.72%

SVR: Support Vector Regression; GPR: Gaussian Process Regression.

$$\begin{aligned}
 e_F &= \left| \frac{F_p - F_a}{F_a} \right|, \\
 e_a &= \left| \frac{a_p - a_a}{a_a} \right|, \\
 e_{t_f} &= \left| \frac{t_{fp} - t_{fa}}{t_{fa}} \right|,
 \end{aligned} \quad (25)$$

where F_a and (a_a, t_{fa}) are the accurate optimal values of the objective function and the corresponding parameters, respectively.

The relative errors of the objective function value and the parameters are shown in Table 7, and the calculation time is shown in Table 8. The following can be concluded.

- (1) The relative errors of the objective function values obtained by SVR and GPR are 0.47% and 0.27%, respectively, which are both very slight. The GPR

Table 8. Calculation time.

Average time	SVR	GPR
Offline regression/s	0.52	20.56
Online calculation/ms	25.63	25.63

SVR: Support Vector Regression; GPR: Gaussian Process Regression.

Figure 15. Predicted positions at each period and position profiles of the entire course.

Figure 16. Velocity profiles of the haptic feedback course.

model is more accurate than the SVR model, but takes much longer time to build offline.

- (2) The relative errors of t_f are smaller than those of a , since the regression and fitting of t_f are first calculated and then the regression models of a are developed depending on t_f , as illustrated in Section 4.4. Thus the precision of t_f is better than that of a .

Figure 17. Velocity curve of each period.

- (3) The relative error of a_1 is the smallest among the three acceleration parameters. According to the analyses in Section 5.2.1, the parameters are dominated by joints with larger variation. Since the motion range of Joint 1 is much larger than that of the other two joints, the situation that Joint 1 has the largest variation takes up a large proportion (about 81% of the samples). Therefore, the acceleration of Joint 1 obtained by the machine learning-based method has the best precision.
- (4) The online calculations for optimal parameters cost 25.63 ms on average, which can be implemented in real-time.

5.3. Trajectory planning for haptic feedback

In haptic feedback applications, the trajectory is re-planned in every prediction period according to the updated interacting point and time. Considering the 6-

DOF HFM described in Section 2.2, we select $q_0 = (0, 0, 0, 0, 0, 0)$ and $\omega_0 = (0, 0, 0, 0, 0, 0)$ as the initial configuration and velocity of the whole process, respectively. The objective configuration of the first period is $q_c = (0.50, 0.40, 0.30, 1.00, 1.30, 3.14)$. The updated period is set as $T_p = 60$ ms and the updated objective configuration in each period is shown with an asterisk in Figure 15, which is used to re-plan the trajectory. Employing the GPR method, we perform trajectory planning continuously and obtain the position and velocity curve in the joint space of the entire process, shown in Figures 15 and 16, respectively. In addition, the end-effector of the HFM reaches the interacting point in time.

The whole process includes 21 trajectory planning periods, which are illustrated by examples in Figure 17. Trajectory planning of each period on average costs 38.78 ms. The calculation time of each period is below T_p , as shown in Figure 18, which means that trajectory planning for the HFM can be performed in real-time.

Figure 18. Calculation time of each period.

6. Conclusion

A haptic feedback scheme for the VCS based on the servo serial manipulator and a trajectory planning method for it are presented in this paper. We analyze the workspace of the HFM and firstly find the interacting configuration as the objective of trajectory planning. Then trajectory planning based on kinematical optimization is performed and solved in real-time with the machine learning method.

- (1) A haptic feedback system based on the servo serial manipulator is presented and the type of the manipulator is determined.
- (2) An interacting configuration determination method is presented. Firstly, the control panel area is extracted by analyzing the workspace of the HFM. Then the interacting point and the corresponding interacting configuration are determined by intercepting the hand trajectory with the control panel.
- (3) A trajectory planning method based on kinematical optimization and a real-time solver based on machine learning are presented. The trajectory planning problem is formulated as a non-linear optimization problem based on kinematics and the global minimum is found in real-time by choosing a good initial solution with the regression model built offline.

However, some limitations should be considered in future work.

- (1) In this paper, we simplify the control panel of real cockpits, only considering the forward panel. In future work, we would like to develop the HFM by adding DOFs to simulate multiple control

panels in different directions and make it close to the real situation.

- (2) The trajectory planning method presented in this paper is based on kinematics and the trapezoidal velocity profile, which is incapable of torque limitation. In future work, we would like to extend this method to dynamics by considering torque and energy in non-linear optimization and using polynomial interpolation for trajectory parameterization.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

1. Dai S, Lei X and Mei J. Virtual cockpit system. *J Syst Simulat* 2002; 14: 488–492.
2. Wang D, Jiao J, Zhang Y, et al. Computer haptics: haptic modeling and rendering in virtual reality environments. *J Comput Aid Des Comput Graphic* 2016; 28: 881–895.
3. Yu T. *Semi-virtual reality cockpit technology research based on data glove*. PhD Thesis, Nanjing University of Aeronautics and Astronautics, 2008.
4. Tang Y. *Research on key technologies for augmented semi-virtual reality cockpit*. PhD Thesis, Nanjing University of Aeronautics and Astronautics, 2012.
5. CGSD. Tactile feedback for a force-reflecting haptic display. Technical report, Computer Graphics Systems Development Corporation, 1998.
6. Stonum R. A survey of immersive technology for maintenance evaluations. Technical report, Battelle Memorial Institute, 1998.
7. Kim KW, Kim HS, Choi YK, et al. Optimization of cubic polynomial joint trajectories and sliding mode controllers for robots using evolution strategy. In: *Proceedings of the IECON'97 23rd International Conference on Industrial Electronics, Control, and Instrumentation, 1997 (IECON 97)*, volume 3, New Orleans, LA, 14 November 1997 pp.1444–1447. Piscataway, NJ: IEEE.
8. Bazaz SA and Tondu B. Online computing of a robotic manipulator joint trajectory with velocity and acceleration constraints. In: *Proceedings of the 1997 IEEE International Symposium on Assembly and Task Planning (ISATP'97)*, Marina del Rey, CA, USA, 7–9 August 1997, pp. 1–6. Piscataway, NJ: IEEE.
9. Lampariello R and Hirzinger G. Generating feasible trajectories for autonomous on-orbit grasping of spinning debris in a useful time. In: *2013 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, Tokyo, Japan, 3–7 November 2013, pp. 5652–5659. Piscataway, NJ: IEEE.
10. Liu X, Meng Z, Ni J, et al. Trajectory planning algorithm for hydraulic servo manipulator of three freedom. *J Zhejiang Univ Eng Sci* 2015; 49: 1776–1782.
11. Miossec S, Yokoi K and Kheddar A. Development of a software for motion optimization of robots-application to the kick motion of the hrp-2 robot. In: *2006 IEEE International*

- Conference on Robotics and Biomimetics*, Kunming, China, 17–20 December 2006, pp. 299–304. Piscataway, NJ: IEEE.
12. Jetchev N and Toussaint M. Fast motion planning from experience: trajectory prediction for speeding up movement generation. *Auton Robot* 2013; 34: 111–127.
 13. Schulman J, Ho J, Lee AX, et al. Finding locally optimal, collision-free trajectories with sequential convex optimization. In: *Proceedings of Robotics: Science and Systems*, Berlin, Germany, 24–28 June 2013. Cambridge, MA: MIT Press.
 14. von Stryk O and Schlemmer M. Optimal control of the industrial robot manutec r3. In: *Computational optimal control*. Berlin: Springer, 1994, pp.367–382.
 15. Chettibi T, Lehtihet H, Haddad M, et al. Minimum cost trajectory planning for industrial robots. *Eur J Mech A Solid* 2004; 23: 703–715.
 16. Bäuml B, Wimböck T and Hirzinger G. Kinematically optimal catching a flying ball with a hand-arm-system. In: *2010 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, Taipei, Taiwan, 18–22 October 2010, pp. 2592–2599. Piscataway, NJ: IEEE.
 17. Lampariello R, Nguyen-Tuong D, Castellini C, et al. *2011 IEEE international conference on robotics and automation (ICRA)*, Shanghai, China, 9–13 May 2011, pp. 3719–3726. Piscataway, NJ: IEEE.
 18. Werner A, Trautmann D, Lee D, et al. Generalization of optimal motion trajectories for bipedal walking. In: *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, Hamburg, Germany, 28 Sept.–2 Oct. 2015, pp.1571–1577. Piscataway, NJ: IEEE.
 19. Gribovskaya E and Billard A. Learning nonlinear multivariate motion dynamics for real-time position and orientation control of robotic manipulators. In: *9th IEEE-RAS international conference on humanoid robots (Humanoids)*, 7–10 Dec. 2009, Paris, France, pp.472–477. Piscataway, NJ: IEEE.
 20. Khansari-Zadeh SM and Billard A. Learning stable nonlinear dynamical systems with Gaussian mixture models. *IEEE Trans Robot* 2011; 27: 943–957.
 21. Kim S, Shukla A and Billard A. Catching objects in flight. *IEEE Trans Robot* 2014; 30: 1049–1065.
 22. Salehian SSM, Khoramshahi M and Billard A. A dynamical system approach for softly catching a flying object: theory and experiment. *IEEE Trans Robot* 2016; 32: 462–471.
 23. Tesar D and Thomas M. Dynamic modeling of serial manipulator arms. *Trans ASME* 1982; 104: 218–228.

Author biographies

Shiyu Zhang received her BSc degree from Hefei University of Technology in 2014. She is now a PhD student at the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University. Her research interests focus on virtual reality and human–robot interaction.

Shuling Dai received his PhD degree from Beihang University in 1997. He is now a Full Professor and Doctoral Advisor at the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University. His research interests focus on virtual reality and distributed simulation.