ABSTRACT

In this paper, we propose a linear parameter varying (LPV) control design approach for trajectory tracking in a robotic system, intended to be involved in an image-guided teleoperated cardiac surgery. The robot is eventually aimed to guide a 3 degree-of-freedom medical tool (a catheter) inside the left ventricle (LV) and achieve the implantation of a prosthetic aortic valve. The successful delivery of the valve from the apical entrance to the aortic annulus strongly depends on the precise navigation of the catheter such that its probable collision with the LV’s changing environment is avoided. The LPV control strategy is utilized here due to its ability to capture the nonlinearities of the designed robot manipulator and adapt in real-time based on the varying end effector’s angle. The simulation studies demonstrate promising results achieved for a guaranteed safe navigation through LV.

INTRODUCTION

In the last decade, employment of biomedical robots in surgical applications has attracted much attention of many researchers in this field. A broad range of medical procedures from highly precise brain surgeries to minimally invasive cardiac interventions and MRI guided teleoperated operations fall into aforementioned applications [1]. In minimally invasive robot-assisted medical interventions, where robot’s miniature end effectors make it possible for surgeons to reach the target through minute incision points, significant achievements including shortened recovery time justify deployment of robots [2]. On the other hand, in teleoperated surgeries when due to the existence of biomedical apparatuses like imaging systems, the operator can not touch the body, or when surgeons have to perform on a patient from a distance, surgical robots play an important role in medical procedures [3]. Compensation of motion artifact caused by the respiration or heart beating is another reason behind letting teleoperated surgical systems be involved in beating heart operations [4]. Being equipped with such system, operators are enabled to work in a virtual environment adroitly to navigate the catheter to reach the target in a safe and accurate way.

Master and slave robots form the very basic elements of a typical teleoperated surgery [5, 6]. The slave robot is directly in contact with patient and manipulates the catheter. Meanwhile, the surgeon actuates the master robot using visual feedbacks from the imaging system and provides commands to the slave unit. The master robot can further contribute in the process by providing the surgeon with tactile sensing while feeding the forces exerted on the end effector, back to the operator’s hand [7]. This latter capability changes the master robot to a haptic device which prevents the operator from applying excessive forces that might be brutal for the tissues. There is a rich literature on various designs for haptic devices suitable for different applications.

For a robot to satisfy the performance requirements, the design of an advanced controller is as important as the robot mechanical design or the selection of actuators and sensors. The controller should be able to provide robot with proper control effort such that the expected stability criterion and desirable performance requirements are met. Based on the dynamics of the robot and to accomplish the task it is designed for, many different control objectives along with a variety of control methods could be considered.
In this paper, we propose a control strategy for trajectory tracking of a slave robot intended to navigate a medical catheter inside the heart’s left ventricle. The designed robot’s highly nonlinear dynamics along with the challenging mission of guiding catheter in heart’s dynamic environment to ensure that it never collides with the boundary, necessitates the implementation of a sophisticated control scheme. The slave robot should be capable of maintaining translational, rotational and bending displacements close to the surgeon’s needs. Due to the environment uncertainties and external disturbances, a robust control strategy is essential for robot to quickly and effectively follow the reference trajectory provided by the master robot. In this paper, we study the use of a linear parameter varying (LPV) control approach [8] that fits within our application of interest due to the nonlinear dynamics of the designed robot manipulator acting as the slave unit. Simulation results shows acceptable specifications regarding both stability criteria and performance requirements.

MECHANISM

In this paper, we are interested in the design and control of a robot manipulator that acts as the slave unit in a teleoperated medical intervention and in particular guiding a medical catheter inside the left ventricle. The design configuration is shown in Fig. 1. This manipulator is a coherent component of a robotic assisted system for implanting a prosthetic aortic valve in beating-heart cardiac surgery. The system consists of several sections. A brief description of the system’s various components is presented next. The interested reader is referred to [9] for more details. The surgical robot is divided into two proximal and distal manipulators. While former part is compensating for the motion of body due to respiration and heart beating, the slave robot (the distal part) placed at top of the compensatory manipulator navigates the catheter inside the left ventricle. Reference trajectories are provided by the master unit that also feeds the haptic sensing back to the surgeon’s hands. To make the loop closed, the imaging system provides operator with the visual feedback.

The slave robotic unit is aimed to insert the catheter inside the left ventricle through the apical entrance and guide it along a non-straight trajectory all the way towards aortic valve. This non-straight path is depicted in Fig. 2. The considered three degrees of freedom including translation, rotation and bending in slave robot ensure the safe navigation of catheter inside the left ventricle.

Fig. 3 illustrates the main parts of the slave robot in detail. The catheter in Fig. 2(b) is placed in the tool holder shown in Fig. 3. The translational DOF is actuated by two DC motors 1 and 2 which slide the medical tool back and forth. DC motor 3 is actuated to rotate the catheter inside an identified safe region that the catheter is allowed to be. Using DC motor 4, valve component in Fig. 2(b) is bent and oriented toward aorta. A cable from the point $X_T$, in Fig. 2(b) passes through the tool holder and wraps around the shaft of the DC motor 4. The bending part of the system, i.e., valve component in Fig. 2, bends when DC motor is active. In the present paper, we will not consider the actuation of the the bending part.

DYNAMIC MODEL OF THE ROBOT

Due to the structure of the designed robot, the dynamic equation of each part is decoupled from others. Obtaining the dynamic equations corresponding to the rotational and bending degrees of freedom is straightforward. In the remainder of this section, we will only discuss the development of dynamic equation for the translational part and corresponding control algorithm.

Using the Euler-Lagrange equation [10], one can derive the dynamic model of the translational motion of the mechanism. Since the translational part consists of two symmetric components, for the sake of simplicity, we only show the derivation of the dynamic equation corresponding to one side in Fig. 4. Half of the handle weight is modeled as the concentrating mass $m_2$. In the following equations, $m_1$ and $L$ are mass and length of arms,
respectively, and \( I_i \) is the moment of inertia with respect to the mass center.

\[
\begin{bmatrix}
\frac{1}{2} m_1 L_1^2 + 2(m_1 + 2m_2)L_2^2 \sin^2 \theta + 2I \dot{\theta} \\
(m_1 + 2m_2)L_2^2 \sin(2\theta) \dot{\theta}^2 + B \dot{\theta}
\end{bmatrix}
\]

Therefore, the dynamic equation of the system is determined to be:

\[
\tau_m = M - B \dot{\theta}
\]  

(3)

Defining the state variables as \( x_1 = \theta \) and \( x_2 = \dot{\theta} \), the following state-space representation for the system dynamics is achieved:

\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= \frac{\tau_m - (m_1 + 2m_2)L_2^2 \sin(2x_1)x_2 - Bx_2}{2m_1L_2^2 + 2(m_1 + 2m_2)L_2^2 \sin^2 x_1 + 2I} + w \\
y &= x_1
\end{align*}
\]  

(4)

In the above state-space representation, \( w \) is the disturbance acting on the system, that reflects the effect of the external forces to the end effector when it touches the boundaries. It can also capture any undesired exogenous inputs interfering with the normal performance of the system. For example in the current design, friction exists between the end effector and the hole in the front wall as depicted in Fig. 3. This friction force that changes its direction can be considered as an external disturbance. The system control input \( \tau_m \), i.e., the DC motor torque, has the ability to be transformed to a corresponding electrical current using the DC motor basic equation. The torque constant links the mechanical torque output to the armature current as \( M = Kl \), where the constant \( K \) is provided by the the motor data sheet. The system output is the end effector angle \( \theta \) in Fig. 4, that is obtained by making use of a high resolution encoder. In practice, the operator provides the translational part with a reference input regarding to the desired position of the end effector. Using the kinematic equation \( l = 2L \cos(\theta) \), the end effector angle and position relate to each other. Therefore, the controller can be designed to achieve the angle trajectory tracking.

**LPV model of the designed robot**

In this section, we show how the system dynamics is reformulated in an LPV form. First, we present a general description of LPV systems. An LPV system can be described in state-space by:

\[
\begin{bmatrix}
x(t) \\
y(t)
\end{bmatrix} = 
\begin{bmatrix}
A(p) & B_1(p) & B_2(p) \\
C_1(p) & D_{11}(p) & D_{12}(p) \\
C_2(p) & D_{21}(p) & 0
\end{bmatrix}
\begin{bmatrix}
x(t) \\
y(t)
\end{bmatrix} + 
\begin{bmatrix}
w(t) \\
u(t)
\end{bmatrix}
\]  

(8)
where \(x, w, u, z\) and \(y\) are the state vector, external disturbance input vector, control input vector, controlled output vector and sensor measurement vector, respectively. Dynamics of the system above is characterized in terms of the LPV scheduling parameter vector \(p(t) \in \mathcal{F}_x^y\), with the parameter space \(\mathcal{F}_x^y\) defined as the set of allowable parameter trajectories

\[
\mathcal{F}_x^y \triangleq \{ p(t) \in C(\mathbb{R}, \mathbb{R}^s) : p(t) \in \mathcal{P}, | \dot{p}_i(t) | \leq v_i, i = 1, 2, \ldots, s, \forall t \in \mathbb{R}_+ \}
\]

where \(\mathcal{P}\) is a compact subset of \(\mathbb{R}^s\), and \(\{v_i\}_{i=1}^s\) are nonnegative numbers. These parameters are assumed to be unknown a priori, but they can be measured or estimated in real-time.

To obtain an LPV description of the nonlinear model (4), we define appropriate scheduling parameters such that the nonlinearities in the model are hidden. This method is referred to as the quasi-LPV modeling. For the state-space representation (5) we introduce scheduling parameter vector \(p = [\rho_1, \rho_2]^T\) as

\[
\rho_1 = \frac{-m_1 + m_2}{2m_1L^2 + 2(m_1 + m_2)L^2 \sin^2 \theta + 2I} \theta - B \\
\rho_2 = \frac{1}{2m_1L^2 + 2(m_1 + m_2)L^2 \sin^2 \theta + 2I}
\]

Then, the quasi-LPV model can be expressed as:

\[
\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & \rho_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \rho_2 \end{bmatrix} \tau_m + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w \\
y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.
\]

(11)

The scheduling parameter vector \(p\) is a vector of two elements that depend on the angle and the angular velocity. Assuming that the encoder reading, i.e., the angle, is the only available measurement, angular velocity needs to be calculated indirectly using other information. In the following, we describe a method to calculate \(\dot{\theta}\) indirectly from the DC motor speed-torque line rather than taking the derivative of \(\theta\) over time.

**DC motor key information**

In a DC motor, the balance equation links input electrical power to the output mechanical power and power losses [11]. Treating frictional losses as an external torque, the electrical losses in motor winding result in the balance equation as:

\[
P_{\text{electrical}} = P_{\text{mechanical}} + P_{\text{loss}}.
\]

Replacing each part with the corresponding equations leads to

\[
UI = \omega M + Rf^2
\]

(13)

where \(U\) is nominal input voltage, \(R\) is the winding resistance, \(M\) is the motor torque, and \(R\) and \(I\) are the rotor winding resistance and electric current, respectively. Replacing \(T\) from (5) into (13), we determine following speed-torque linear equation

\[
\omega = \frac{U}{k} - \frac{RM}{k^2}.
\]

(14)

The angular velocity and the updated scheduling parameter are calculated by using this linear relationship. This equation is used to calculate the angular velocity and update scheduling parameter \(p\) in (9) to avoid the calculation of the angular derivative in real-time and the corresponding issues of dealing with the measurement noise.

It is noted that to provide a DC motor with voltage or current control signal, one can use a driver hardware system. Many of the drivers output some signals for monitoring purposes including the rotor current and angular velocity of the rotor shaft. The latter one can be used to calculate the LPV parameter \(\rho_1\).

**Control design overview**

The control design objective is to provide required torque to the DC motors with feeding electrical currents such that the position of the end effector follows a reference trajectory. Using inverse kinematic, one can translate the end effector position to a control problem for angular tracking. This holds true due to the low weight of the robot components. For this part of the designed robot, we seek for a controller that tracks a reference for the angular reference \(\theta\) in Fig. 4. The maximum torque input the DC motors can bear is indicated in the motor catalog provided by the manufacturer. Also, each motor has a mechanical time constant to avoid the fast changes in the control signal. Therefore, the control design procedure should take into consideration limits on both the control signal and its rate of variation. Another important aspect of the design is its ability to attenuate the effects of external disturbance inputs.

**LPV GAIN SCHEDULING CONTROL DESIGN PROCEDURE**

For the LPV system represented by (8), the induced -\(\ell_2\)-gain from the input \(w\) to the output \(z\) (or the so called \(\mathcal{H}_\infty\) norm) is defined as:

\[
\|T_{zw}\|_{\ell_2} = \sup_{\rho \in \mathcal{F}_x^y} \sup_{\|w\| \neq 0} \frac{\|z\|_2}{\|w\|_2}
\]

where \(T_{zw}\) denotes an input-output operator, i.e., \(z(t) = T_{zw}(w(t))\). If the signals \(w\) and \(z\) are chosen to reflect the system performance specification, a positive bound on \(\|T_{zw}\|_{\ell_2}\) implies that the system has satisfied the desired performance specification. The Bounded Real Lemma (BRL) is used to solve this analysis problem by solving an eigenvalue problem (EVP) [12].
The design problem is now to synthesize an LPV controller of the form
\[
\begin{bmatrix}
\dot{x}_c(t) \\
u(t)
\end{bmatrix} =
\begin{bmatrix}
A_c(\rho) & B_c(\rho) \\
C_c(\rho) & D_c(\rho)
\end{bmatrix}
\begin{bmatrix}
x_c(t) \\
y(t)
\end{bmatrix} +
\begin{bmatrix}
\hat{A}_K \\
\hat{B}_K
\end{bmatrix} + \begin{bmatrix}
\hat{C}_K \\
\hat{D}_K
\end{bmatrix} w(t) + \begin{bmatrix}
\hat{A}_{K2} \\
\hat{B}_{K2}
\end{bmatrix} c_0 + \begin{bmatrix}
\hat{C}_{K2} \\
\hat{D}_{K2}
\end{bmatrix} c_1 + \begin{bmatrix}
\hat{A}_{K3} \\
\hat{B}_{K3}
\end{bmatrix} c_2 + \begin{bmatrix}
\hat{C}_{K3} \\
\hat{D}_{K3}
\end{bmatrix} c_3 + \begin{bmatrix}
\hat{A}_{K4} \\
\hat{B}_{K4}
\end{bmatrix} c_4 + \begin{bmatrix}
\hat{C}_{K4} \\
\hat{D}_{K4}
\end{bmatrix} c_5 + \begin{bmatrix}
\hat{A}_{K5} \\
\hat{B}_{K5}
\end{bmatrix} c_6 + \begin{bmatrix}
\hat{C}_{K5} \\
\hat{D}_{K5}
\end{bmatrix} c_7 + \begin{bmatrix}
\hat{A}_{K6} \\
\hat{B}_{K6}
\end{bmatrix} c_8 + \begin{bmatrix}
\hat{C}_{K6} \\
\hat{D}_{K6}
\end{bmatrix} c_9,
\]

such that the closed-loop system formed by the interconnection of the open-loop system (8) and the LPV controller (15) is asymptotically stable and its induced $\mathcal{L}_2$-gain is less than $\gamma$. The so-called basic characterization theorem provided below gives the procedure to find such a controller.

**Lemma 1. Basic Characterization Lemma [13]:** Given the open-loop LPV system represented by (8) and the positive scalar $\gamma$, there exist an LPV controller of the form (15) if one can find parameter-dependent symmetric matrices $X(\rho)$ and $Y(\rho)$ and parameter-dependent matrices $\hat{A}_K, \hat{B}_K, \hat{C}_K$ and $\hat{D}_K$ such that for all $\rho \in \tau_D$, the following LMI problem has a feasible solution.

\[
\begin{bmatrix}
X + XA + \hat{B}_K C_2 + (\star) & * & * & * \\
\hat{A}_K^T + A + B_2 \hat{D}_K C_2 - \hat{Y} + AY + B_2 \hat{C}_K + (\star) & D_1^T & * & * \\
(\hat{X} B_1 + \hat{B}_K D_2)^T & (B_1 + B_2 \hat{D}_K D_2)^T & -\gamma I & * \\
C_1 + D_12 \hat{D}_K C_2 & C_2 Y + D_12 \hat{C}_K & D_{11} + D_12 \hat{D}_K D_{21} & -\gamma I \\
\end{bmatrix} < 0
\]

Then, the controller’s state-space matrices are determined by following the steps below:

1. Solve for $M$ and $N$ from the factorization problem $NM^T = I - XY$.
2. Compute the controller matrices from

\[
\begin{align*}
D_c &= \hat{D}_K \\
C_c &= (\hat{C}_K - D_c C_2) M^{-T} \\
B_c &= N^{-1}(\hat{B}_K - B_2 \hat{C}_K) \\
A_c &= N^{-1}(XY + NM^T + \hat{A}_K - X(A - B_2 D_c C_2) Y) \\
&\quad - \hat{B}_K C_2 Y - XB_2 \hat{C}_K) M^{-T}.
\end{align*}
\]

**LPV control design procedure for the robot**

Shown in Fig. 5 is the closed-loop configuration used in the control design process with $r, u$ and $w$ being the reference input, control input, and exogenous input, respectively. The control input is the torque applied to the DC motor, and $w$ is the disturbance acting on the catheter. Also, $y$ represents the measured output, and $z$ is the controlled output. In the control design configuration, the output vector is considered to be $z = \begin{bmatrix} We \\ Wu \end{bmatrix}$ where $e$ is the tracking error. The reason for this selection is to achieve: (i) a small tracking error over the low frequency range, and (ii) a low control effort avoiding the presence of high frequency content.

**The design of the dynamic weights:** We solved an $S - KS$ mixed sensitivity problem for the design of an LPV controller. To this purpose, bounds on these two transfer functions need to be defined [14] with $W_e$ to be a high pass filter and $W_u$ to be a low-pass filter with appropriate crossover frequencies.

There are a number of points that need to be taken into consideration in the selection of dynamic weights, which are discussed in the following.

**Remark 1.** The bandwidth of the input weight $W_u(s)$ can be determined from the mechanical time constant of the DC motor, and its gain is tuned such that the amplitude of the control effort does not exceed the maximum continues torque that the motor can endure. These values can be found in the DC motor data sheet.

**Remark 2.** In the design problem under study, the weight gains were optimized using extensive simulations for a variety of reference inputs.

**Further discussion**

This section briefly addresses some of the issues encountered in solving the LMI synthesis conditions of Lemma 1, as well as computing the control state-space matrices in (15). A standard approach to solve the parameterized LMIs of Lemma 1 is to initially pick some basis functions to represent the dependency of the matrix variables on the LPV parameters and then grid the parameter space. Hence, finite-dimensional LMIs are solved at the grid points and are then checked on a finer grid. For a description of the approach the interested reader is referred to [13]. Once the LMIs are solved, the controller matrices are then computed from (16). It should be noted that the controller state-space matrices are also parameter-dependent and updated in real-time based on the real-time calculation of the LPV parameter vector $\rho(i)$ using the available measurements.

In the control design procedure, the Lyapunov matrices $X$ and $Y$ are considered to be constant. This leads to an LMI problem in Lemma 1 and the controller matrices in (16) to be independent of derivative of the LPV parameters. As mentioned...
earlier, for any $\rho \in \mathcal{F}_P$, the infinite-dimensional LMIs in Lemma 1 should hold true. To determine a finite-dimensional LMI problem, a gridding over the parameter space needs to be done. From the robot mechanism $\theta$ can vary in $[0, 2\pi]$. The angular velocity $\dot{\theta}$ depends on the application; however, it is noted that the whole mechanism moves slowly and does not sweep more than 45 degrees per second. Therefore, it is assumed that $\dot{\theta} \in [-\pi/4, \pi/4]$. Using these bounds on $\theta$ and $\dot{\theta}$, one can compute the range of variations for the LPV parameters $p_1$ and $p_2$. We will show the LPV parameters variation in the next section.

SIMULATION RESULTS

In this section we present simulation results demonstrating the viability of the developed LPV controller for the designed robot. For validation of the designed controller, we implemented the robot dynamic model in Simscape toolbox in MATLAB and the model of the DC motor in Simpower toolbox. The characteristics of the DC motors were selected from the Maxon motor data sheet [11]. The model EC-pole 22 brushless is used for DC motors 1 and 2 and the ECmax 22 brushless for the third and forth motors. Although we designed the controller for angle tracking in the rotational part, as mentioned before, in practice, the reference inputs provide end effector position in 3-D. Using kinematic equations, the position is transformed to the corresponding translational, rotational and bending values for each part of the robot. The range of the parameter variations is a key point in LPV controller design procedure. It was found from extensive open-loop and closed-loop simulations that $p_1 \in [-3.81, 3.81]$ and $p_2 \in [51, 667]$. The dynamic weights are designed to shape the closed-loop transfer functions $S$ and $KS$ as illustrated in Fig. 6. Figure 7 shows the response of the translational part of the robot to a series of ramp reference inputs. These reference inputs are used to show a typical translation profile for the end effector. Here, the designed LPV controller provides control action only for this part so that the end effector moves back and forth to follow the trajectory. It is noted that for simulation purposes, we assumed measurements mixed with the band-limited white noise and also a friction between rod and the supporting wall in Fig. 3.

In terms of the control design, the main focus of the paper has been on the development of an LPV controller for the nonlinear system dynamics modeled in the LPV form (11). However, the rotation and bending parts need to be controlled as well. The dynamics of these parts are linear and could be controlled using a linear controller designed by the standard linear control design methods. We used a model reference adaptive controller (MRAC) with output feedback [15] for tracking of the reference input. Using an adaptive controller while the designed LPV controller takes care of the translational motion ensures a precise tracking for varies reference trajectories. Figure 8 shows the closed-loop system response to a typical trajectory on the surface of a cylinder involving both translational and rotational parts. Shown in Fig. 9 is the corresponding tracking error of the rotational part demonstrating less than 1% error.

FUTURE WORK

The future work will be to build a prototype of the designed robot and implementing the LPV controller in dSPACE real-time controller. The prototyped system along with the control logic will be eventually integrated with the MRI images intraoperatively.

ACKNOWLEDGMENT

This work was supported by the National Science Foundation under Grant CNS-0932272. All opinions, findings, conclusions or recommendations expressed in this work are those of the authors and do not necessarily reflect the views of our sponsor. The authors would like to thank F. Shirazi for his valuable comments and E. Yeniaras for providing Fig. 2.

REFERENCES

Figure 7. RESPONSE TO A VARYING-SLOPE RAMP REFERENCE INPUT

Figure 8. A TYPICAL REFERENCE TRAJECTORY FOR THE END EFFECTOR


