Guide by Touch: Enabling Drag-and-Drop Control in Robotic Arms

Yifan Wang
SDIM, SUSTech
Shenzhen, China
wangyf2021@mail.sustech.edu.cn

Zixuan Zhong
SDIM, SUSTech
Shenzhen, China
zhongzx2021@mail.sustech.edu.cn

***Abstract*—This paper report details of the successful development and implementation of an adaptive compliance control system for robotic manipulators, enhancing their ability to interact safely and effectively with humans and dynamic environments. The system integrates active force control with passive impedance control, allowing for real-time adjustments to external forces. The project's outcomes include three distinct functionalities: free movement under manual guidance, trajectory tracking with dynamic interaction, and autonomous return to initial positions post-disturbance. The system's performance was validated through rigorous testing, demonstrating improved precision, adaptability, and safety in human-robot interaction.**

***Index Terms*—Adaptive Compliance Control, Robot Arm, Industry Safety**

I. Introduction

The integration of robotics into various sectors such as manufacturing, healthcare, and service industries has been marked by a continuous evolution towards more sophisticated and versatile machines. Robotic manipulators, in particular, have become indispensable for their precision, repeatability, and ability to operate in environments that may be hazardous or inhospitable to human workers. However, the challenge of ensuring these machines can work safely and effectively alongside humans or in unpredictable environments has led to the development of compliance control techniques.

Compliance in robotics refers to the ability of a manipulator to exhibit flexibility and adaptability in its movements, allowing it to respond to external forces and maintain safe interactions with its surroundings. Traditional robotic systems often operate under rigid, pre-programmed motions, which can be limiting when it comes to dynamic tasks that require a degree of give-and-take. The need for robotic manipulators to perform in a manner that is both safe and adaptable has driven the research and development of adaptive compliance control systems.

Adaptive compliance control systems are designed to dynamically adjust the behavior of a robotic manipulator in response to external forces or changes in its operational environment. These systems aim to enhance the performance and safety of the manipulator by incorporating real-time adjustments to its control parameters, thereby allowing for a more nuanced and responsive interaction with the world.

II. Review

The field of robotics has witnessed significant advancements in compliance control techniques, pivotal for ensuring safety and effectiveness in human-robot interaction (HRI). This section reviews the pertinent literature that has informed the development of adaptive compliance control system.

Active compliance control is an integral technique for Tri-Co robots, which are designed to coexist, cooperate, and exhibit cognitive behavior alongside humans [1]. Wang et al. propose a closed-loop compliant control scheme for the dynamic identification of a cascade elbow joint, which is crucial for robots to perform tasks such as handshakes with humans. Besides, accurate dynamic modeling of robotic systems is essential for whole-arm compliance control, which allows for simplification of robot complexity and independence from force sensors [2]. Li et al. address the challenges of formulating accurate dynamic models and propose a recursive parameter identification method that incorporateing prior dynamics knowledge.

And in the context of robot-assisted rehabilitation, subject-specific compliance control is vital for personalized training programs [3]. Miao et al. introduce a new compliance control strategy for upper-limb rehabilitation systems, where real-time adjustments to training trajectories are made based on human-robot interaction forces and user position within a subject-specific workspace. The experimental results from healthy subjects indicate that the proposed compliance adaptation strategy is feasible and has the potential to enhance training safety and efficacy.

Active compliance control is crucial for robotic hands to successfully grasp objects without causing damage [4]. Sadun et al. provide an overview of active compliance control techniques, which are increasingly employed in various robotic applications. The paper discusses force control and impedance control as the two main methods for establishing active compliance control, highlighting several control strategies for safe grasping.

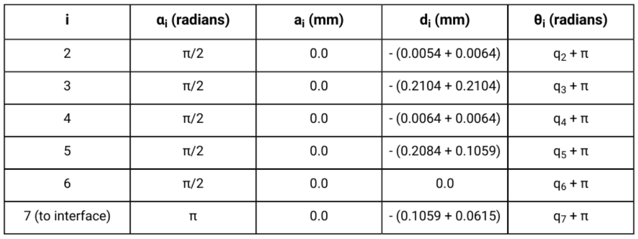
Also, Doria et al. present a novel kinematic method based on the concept of the ”Mozzi axis” to analyze the compliance properties of an industrial robot [5]. The study identifies joint compliances using a modal method that focuses on specific modes of vibration dominated by the compliance of individual joints. The Mozzi axis approach offers an intuitive and geometric representation of the robot's compliance effects on performance, which is particularly useful for applications such as robotic assembly and machining where compliance can compensate for end-effector positioning errors or lead to chatter vibrations.

From active compliance control for handshakes to subject-specific compliance in rehabilitation and the novel Mozzi axis approach for compliance analysis, these studies collectively contribute to the advancement of adaptive compliance control systems. The integration of these techniques into the project aims to address the need for improved safety, adaptability, and efficiency in human-robot collaborative tasks.

III. Modeling

The dynamic function of the Kortex Gen3 is built manually in this project, using Lagrange method. The step by step modeling processes are presented in the following.

A. Derive the T



i	α_i (radians)	a_i (mm)	d_i (mm)	θ_i (radians)
2	π/2	0.0	- (0.0054 + 0.0064)	$q_2 + \pi$
3	π/2	0.0	- (0.2104 + 0.2104)	$q_3 + \pi$
4	π/2	0.0	- (0.0064 + 0.0064)	$q_4 + \pi$
5	π/2	0.0	- (0.2084 + 0.1059)	$q_5 + \pi$
6	π/2	0.0	0.0	$q_6 + \pi$
7 (to interface)	π	0.0	- (0.1059 + 0.0615)	$q_7 + \pi$

Figure 2: DH table from Kortex

Based on the frams given in Figure (1), and the corresponding DH table (Figure 2), the transformation matrix T can be derived repectively.

$$^{i-1}T_i = \begin{bmatrix} cos(\theta_i) & -cos(\alpha_i)sin(\theta_i) & sin(\alpha_i)sin(\theta_i) & a_icos(\theta_i) \\ sin(\theta_i) & cos(\alpha_i)cos(\theta_i) & -sin(\alpha_i)cos(\theta_i) & a_isin(\theta_i) \\ 0 & sin(\alpha_i) & cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 3: Transformation from Kortex

B. Derive Lagrangian

To begin with the Lagrange method, the Lagrangian is needed. The lagrangian **L** is defined as:

$$\mathbf{L} = \mathbf{K} - \mathbf{U} \tag{1}$$

where the **K** is the kinematic energy and **U** represents the potential energy.

The **K** of Gen3 should contain the kinematic energy of all seven links. For i-th link(the number is counted from bottom 1 to the top 7), the kinematic energy $\mathbf{K}_i$ is:

$$\mathbf{K}_i = \frac{1}{2} m_i {v_i}^T v_i + \frac{1}{2} w_i^T I_i w_i \tag{2}$$

where the m_i is the mass of i-th link, and the I_i is the Inertia Tensor of i-th link, $v_i, w_i \in R^{3\times1}$ represent the linear and angular velocities, respectively.

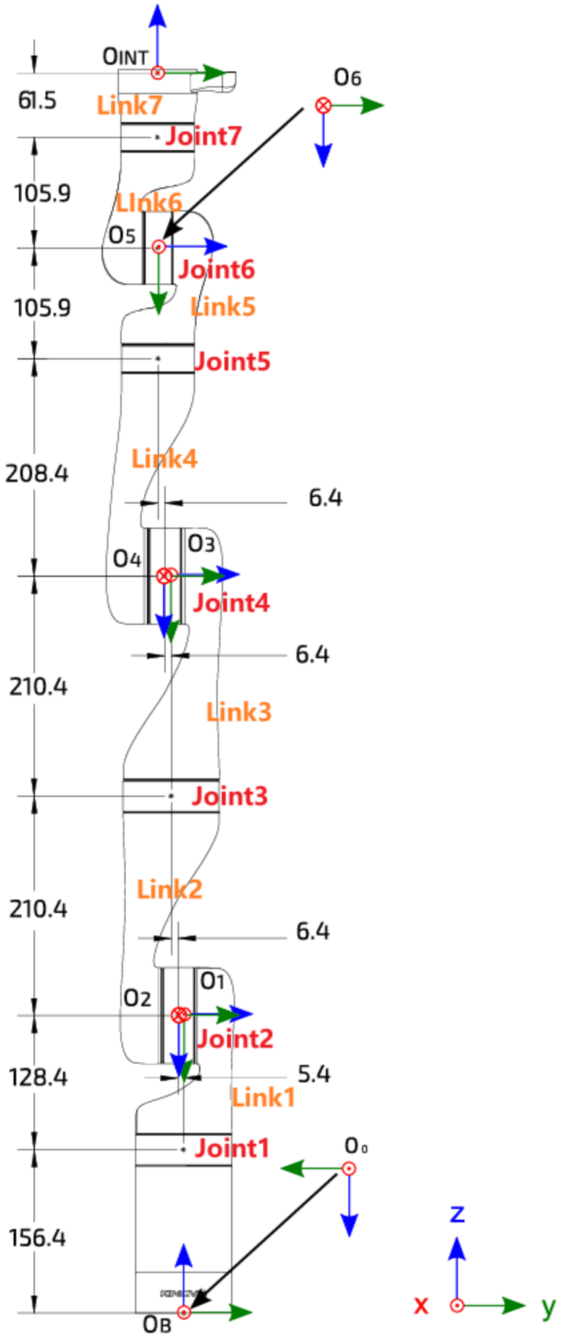


Figure 1: model frame from Kortex

However, the velocity in Cartesian Space is hard to calculate and the velocity in Joint space is easy to control. Hence the jacobian matrix is needed. When working in the Joint space, the joints are marked from the bottom 1 to the top 7 in order, with the notation:

$$\begin{aligned} \Theta &= [q_1, q_2, q_3, q_4, q_5, q_6, q_7]^T \\ v_i &= J_{iv} \cdot \dot{\Theta} \\ w_i &= J_{iw} \cdot \dot{\Theta} \end{aligned}$$

The jacobian matrix is divided to J_v, J_w, with respect to v, w. For i-th link, the jacobian is in form of:

$$J_{ik,k\in v,w} = [j_{ik1}, j_{ik2}, \ldots, j_{ikn}], \tag{3}$$

where $n = 1, 2, \ldots, 7$, $j_{ikn} \in R^{3\times 1}$ In Eq.(3), for example, notation J_{iv} is the linear velocity jacobian matrix of the i-th link, and in J_{iv}, the column j_{ivn} is the coefficient with respect to joint q_n.

In general, the j_{ivn} and j_{iwn} are given like:

$$j_{ivn} = \begin{cases} [0,0,0]^T, n > i \\ {}^0z_n \times (p_i - p_n), n \leq i \end{cases}$$

where ${}^0z_i$ is the i-th link's frame's z direction unit vector in the base frame, p_i and p_n are the position of the mass center of the i-th link and position of the origin of the n-th frame (The origins of frames built by Kortex are all in the center of each joint).

$$j_{iwn} = \begin{cases} [0,0,0]^T, n > i \\ {}^0_i r_3, n \leq i \end{cases}$$

Given jacobian matrix J_{iv}, J_{iw}, the Eq. (2) can be written as

$$\mathbf{K}_i = \frac{m_i}{2}\dot{\Theta}^T J_{iv}^T J_{iv}\dot{\Theta} + \frac{1}{2}\dot{\Theta}^T J_{iw}^T I_i J_{iw}\dot{\Theta}$$

and the potential energy of i-th link can be derived as

$$\mathbf{U}_i = m_i g h_i$$

where the h_i is the height of i-th link's mass center in base frame.

Therefore, the Lagrangian $\mathbf{L}$ can be obtained by

$$\begin{aligned} \mathbf{L} &= \mathbf{K} - \mathbf{U} \\ &= \sum_{i=1}^{7} \mathbf{K}_i - \sum_{i=1}^{7} \mathbf{U}_i \end{aligned} \qquad (4)$$

C. Derive the Dynamic Function

The Euler-Lagrange equation in matrix form is

$$\frac{d}{dt}\left(\frac{\partial \mathbf{L}}{\partial \dot{\Theta}}\right) - \frac{\partial \mathbf{L}}{\partial \Theta} = \tau.$$

First, the Eq.(4) is rewritten as

$$\mathbf{K} = \sum_{i=1}^{7} \mathbf{K}_i = \frac{1}{2}\dot{\Theta}^T \mathbf{M}(\Theta)\dot{\Theta} \qquad (5)$$

where $\mathbf{M}(\Theta) = \sum_{i=1}^{7}(m_i J_{iv}^T J_{iv} + J_{iw}^T I_i J_{iw})$.

Substitute the Lagrangian into Eq.(5), then the following formulas can be obtained.

$$\frac{d}{dt}\left(\frac{\partial \mathbf{K}(\Theta,\dot{\Theta})}{\partial \dot{\Theta}} - \frac{\partial \mathbf{U}(\Theta)}{\partial \dot{\Theta}}\right) - \frac{\partial \mathbf{K}(\Theta,\dot{\Theta})}{\partial \Theta} + \frac{\partial \mathbf{U}(\Theta)}{\partial \Theta} = \tau$$

where the $\tau \in R^{7\times 1}$, represents the torques applied to each joint.

$$\begin{aligned} \frac{d}{dt}\left(\frac{\mathbf{M}^T(\Theta) + \mathbf{M}(\Theta)}{2}\dot{\Theta}\right) - \frac{\partial \mathbf{K}(\Theta,\dot{\Theta})}{\partial \Theta} + \frac{\partial \mathbf{U}(\Theta)}{\partial \Theta} &= \tau \\ \frac{d}{dt}\left(\mathbf{M}(\Theta)\dot{\Theta}\right) - \frac{\partial \mathbf{K}(\Theta,\dot{\Theta})}{\partial \Theta} + \frac{\partial \mathbf{U}(\Theta)}{\partial \Theta} &= \tau \\ \mathbf{M}(\Theta)\ddot{\Theta} + \frac{d\mathbf{M}(\Theta)}{dt}\dot{\Theta} - \frac{\partial \mathbf{K}(\Theta,\dot{\Theta})}{\partial \Theta} + \frac{\partial \mathbf{U}(\Theta)}{\partial \Theta} &= \tau \end{aligned}$$

The equation can be transformed to form

$$\mathbf{M}(\Theta)\ddot{\Theta} + C(\Theta,\dot{\Theta}) + G(\Theta) = \tau$$

where

$$\mathbf{M}(\Theta) = \sum_{i=1}^{7}(m_i J_{iv}^T J_{iv} + J_{iw}^T I_i J_{iw}) \qquad (6)$$

$$\mathbf{C}(\Theta,\dot{\Theta}) = \left[\sum_{i=1}^{7}\frac{d\mathbf{M}(\Theta)}{dq_i}\dot{q}_i\right]\dot{\Theta} - \frac{\partial \mathbf{K}(\Theta,\dot{\Theta})}{\partial \Theta} \qquad (7)$$

$$\mathbf{G}(\Theta) = \frac{\partial \mathbf{U}(\Theta)}{\partial \Theta} \qquad (8)$$

D. Simulation

Once the

$$\begin{aligned} \ddot{\Theta} &= \mathbf{M}(\Theta)^{-1}[\tau - \mathbf{G}(\Theta) - \mathbf{C}(\Theta,\dot{\Theta})] \\ \dot{\Theta}_d &= \dot{\Theta} + \ddot{\Theta}\cdot\Delta T \\ \Theta_d &= \Theta + \dot{\Theta}_d\cdot\Delta T + \frac{1}{2}\ddot{\Theta}\Delta T^2 \end{aligned}$$

$$\begin{aligned} E &= \Theta - \Theta_d \\ \dot{\Theta} &= -k\cdot E \end{aligned}$$

IV. Implementation

A. Derive $\mathbf{M}(\Theta)$

To get the matrix $\mathbf{M}(\Theta)$, jacobian matrix is needed first. For each link, both linear and angular jacobian matrices are need. For all seven links, 14 jacobian matrices are calculated first. Then $\mathbf{M}(\Theta)$ can be derived by Eq.(6).

B. Derive $\mathbf{C}(\Theta,\dot{\Theta})$

It is hard to get the analysis solution, but the numerical solution is easy to calculate. For example,

$$\begin{aligned} \Theta_{max} &= \Theta + \Delta_i \\ \Theta_{min} &= \Theta - \Delta_i \end{aligned}$$

where vector Δ_i means the i-th element is small constant δ. The component of $\mathbf{C}(\Theta,\dot{\Theta})$ can be derived like:

$$\frac{d\mathbf{M}(\Theta)}{dq_i}\dot{q}_i = \frac{\mathbf{M}(\Theta_{max}) - \mathbf{M}(\Theta_{min})}{2\delta}\dot{q}_i.$$

The $q_i, \dot{q}_i \in R$, $\sum_{i=1}^{7}\frac{d\mathbf{M}(\Theta)}{dq_i}\dot{q}_i \in R^{7\times 7}$ and $\left[\sum_{i=1}^{7}\frac{d\mathbf{M}(\Theta)}{dq_i}\dot{q}_i\right]\dot{\Theta} \in R^{7\times 1}$.

However, slight difference occurs in the part $\frac{\partial \mathbf{K}(\Theta,\dot{\Theta})}{\partial \Theta}$. For i-th element of the $\frac{\partial \mathbf{K}(\Theta,\dot{\Theta})}{\partial \Theta}$,

$$[\frac{\partial \mathbf{K}(\Theta,\dot{\Theta})}{\partial \Theta}]_i = \frac{\mathbf{K}(\Theta_{max},\dot{\Theta}) - \mathbf{K}(\Theta_{min},\dot{\Theta})}{2\delta}$$

C. Derive $\mathbf{G}(\Theta)$

It is straightforward to get $\mathbf{G}(\Theta)$ by applying the same trick

$$[\mathbf{G}(\Theta)]_i = \frac{\mathbf{U}(\Theta_{max}) - \mathbf{U}(\Theta_{max})}{2\delta}$$

Thus, we have presented all the necessary calculations for the dynamic equations. Algorithm 1 provides the basic code for dragging and moving in Gen3, while Algorithm 2 illustrates the method for handling interruptions during task execution in Gen3.

Algorithm 1 Drag and Move

```
Establish the connection
while True do
    $\dot{\Theta}, \Theta, \tau, \leftarrow$ Read from Gen3
    $\tau \leftarrow$ Threshold filter($\tau$)
    $\mathbf{M}(\Theta) \leftarrow$ Eq.(6)
    $\mathbf{C}(\Theta, \dot{\Theta}) \leftarrow$ Eq.(7)
    $\mathbf{G}(\Theta) \leftarrow$ Eq.(8)
    $\ddot{\Theta} \leftarrow \mathbf{M}(\Theta)^{-1}[\tau - \mathbf{G}(\Theta) - \mathbf{C}(\Theta, \dot{\Theta})]$
    $\dot{\Theta}_d \leftarrow \dot{\Theta} + \ddot{\Theta} \cdot \Delta T$
    $\Theta_d \leftarrow \Theta + \dot{\Theta}_d \cdot \Delta T + \frac{1}{2}\ddot{\Theta}\Delta T^2$
    Control input $\dot{\Theta}_u \leftarrow -k(\Theta - \Theta_d)$
    Send $\dot{\Theta}_u$ to Gen3
end while
```

Algorithm 2 Interrupt Gen3

```
Establish the connection
while True do
    Gen3 do original Task
    Read data and process $\mathbf{M}, \mathbf{C}, \mathbf{G}$
    $\ddot{\Theta} \leftarrow \mathbf{M}(\Theta)^{-1}[\tau - \mathbf{G}(\Theta) - \mathbf{C}(\Theta, \dot{\Theta})]$
    if $\ddot{\Theta} >$ threshold then
        Get $\dot{\Theta}_d, \Theta_d$ and $\dot{\Theta}_u$
        Send $\dot{\Theta}_u$ to Gen3
    else
        Send control of original task
    end if
end while
```

V. Result

The demonstration videos can be found in the zip package, which includes demonstrations of normal 'Drag and Drop', 'Interrupting the Gen3', and 'Guide and Repeat'.

VI. Discussion

Although the project achieved certain results in the demonstration videos, there is still much room for improvement:

- The control currently uses only proportional control (P control). When the control input is too large, significant oscillations occur. Other control methods could be considered.
- The computation is too intensive, resulting in a low control input frequency. Due to the lengthy calculation process, one cycle takes about 0.05 to 0.1 seconds, leading to a control frequency of 10Hz to 20Hz. Optimizing some calculations to reduce computational load or using C++ for implementation could improve the control frequency.
- For convenience, this project uses velocity control. However, force control could achieve better results. By controlling the torque, not only position but also velocity can be tracked, achieving better tracking of the dynamic reference.

VII. Conclusion

In this project, based on the official Kortex Gen3 manual, we have established the dynamic equations of the Gen3 robotic arm from scratch and implemented the simulation through code, including 'Drag and Drop', 'Interrupting the Gen3', 'Guide and Repeat', and achieved good results.

References

[1] B. Wang, J. Huang, G. Shen, and D. Chen, "Design of admittance controller with sliding mode based on disturbance observer for elbow joint actuated by pneumatic muscles," *Industrial Robot: the international journal of robotics research and application*, vol. ahead-of-print, 05 2020.

[2] L. Yang, "Dynamic model identification for whole-arm compliance control," *Journal of Mechanical Engineering*, vol. 58, no. 3, pp. 45–54, 2022.

[3] Q. Miao, Y. Peng, L. Liu, A. J. McDaid, and M. Zhang, "Subject-specific compliance control of an upper-limb bilateral robotic system," *Robotics Auton. Syst.*, vol. 126, p. 103478, 2020.

[4] A. Sadun, J. Jalani, and J. Sukor, "An overview of active compliance control for a robotic hand," vol. 11, pp. 11872–11876, 01 2016.

[5] A. Doria, S. Cocuzza, N. Comand, M. Bottin, and A. Rossi, "Analysis of the compliance properties of an industrial robot with the mozzi axis approach," *Robotics*, vol. 8, p. 80, 2019.

Contribution

Two people contribute equally to the project and finish the project together.