Design of an impedance controller for da Vinci MTM

RBE/ME 501 Final Project Report

Farid Tavakkolmoghaddam Sapan Santosh Agrawal Kartik Patath Simon Hao Yang Tianyu Zhu

***Abstract*— Robots have long been used in the operation theaters primarily for a wide variety of applications ranging from neurosurgery to laparoscopic surgery. This demands a regulated force control between the manipulator and the tissues in which the robot interacts. Although the da Vinci Surgical Robot has risen as a pioneering tool in the field of minimally invasive surgery, the platform lacks a controller that governs the interaction of the robot's end effector with its surrounding environment in a compliant manner. This work presents the dynamics modeling and design of an impedance controller for the Master Tool Manipulator of the da Vinci robot in a simulated environment. *Index Terms*—Impedance control, da Vinci Research Kit, AMBF simulator**

I. Introduction

The concept of a master-slave tele-manipulator was developed in the early 90s [1]. The da Vinci surgical system (by Intuitive Surgical) was the first commercially available robotic system developed for laparoscopic surgery. The robot is comprised of four main components: the Master Tool Manipulator (MTM), the Patient Side Manipulator (PSM), the foot pedal and a stereo viewer . The PSM is comprised of 3 articulated arms and an endoscope arm. Real-time imaging inside patient's body is passed into the stereo viewer which not only allows surgeon to see the operation area but also provides depth of view. Two haptical MTMs are attached to the stereo viewer that gives surgeon the ability to manipulate the wristed motion at the end effector located at the PSM in real-time through hand movement. [2]. The robot provides an ergonomic interface to the surgeon which eliminates some inherent issues commonly associated with traditional laparoscopic surgeries such as the fulcrum effect, lack of depth perception, and unnatural positioning of the surgeon during the procedure [3].

Da Vinci Research Kit (dVRK) is the research version of the clinical system that is made available for the research community [4]. It's an open-source platform as it gives researchers direct access to all sensors and actuators and allows them to freely write/modify all levels of the control software. [5] Similar to the clinical version, the two main components in dVRK are namely the MTM and the PSM. The PSM in the dVRK is comprised of the two articulated arms and one stereo endoscope. The components of the dVRK are shown in Fig. I. The system is controlled using a Proportional Integral Derivative (PID) controller, which lacks the ability to provide a compliant interaction between the end effector and the patient's body when in operation.

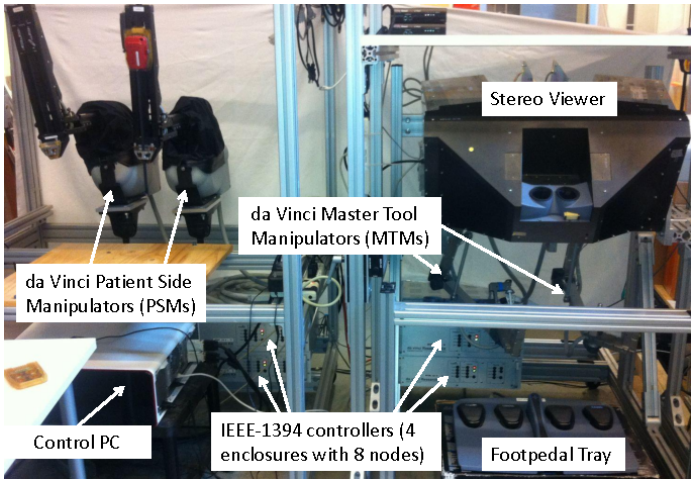

Fig. 1. Main components in a Da Vinci Research Kit [6].

Impedance control has proved to be a useful technique for controlling the interactions of the manipulator with its environment in a safe and compliant manner [7], [8]. An impedance controller can provide the synergy between the position and the force control of the end effector operating along multiple degrees of freedom (DOF) in the Cartesian space [9], [10]. Impedance control is widely used in tasks that require positioning of the end effector along a trajectory, while regulating the force interactions with the environment. Such applications can be found in human-robot cooperative robotics [11], [12] as well as in the surgical robotics [13]. One benefit of using an impedance control on surgical robots is that it enables force control in uncertain and dynamically changing environments such as within the patient's body. Additionally, the sensors at the joints provide compliance in cases of collision with objects. This project aims to design an impedance controller that could be used for the MTM console of the dVRK system which could potentially bring the compliance to the system.

The understanding of MTM's structure is therefore important for the successful implementation of impedance control. As shown in fig 2, The overall structure rotates about the vertical axis of joint 1. Joint 2, 2',2" and 3' forms a parallelogram mechanism which is actuated by joint 2 and 3. Axis of joint 4,

5, 6 and 7 intersects at the same point forming a non-locking gimbal. All joints are actuated by a motor except joint 2' and 2". [14]

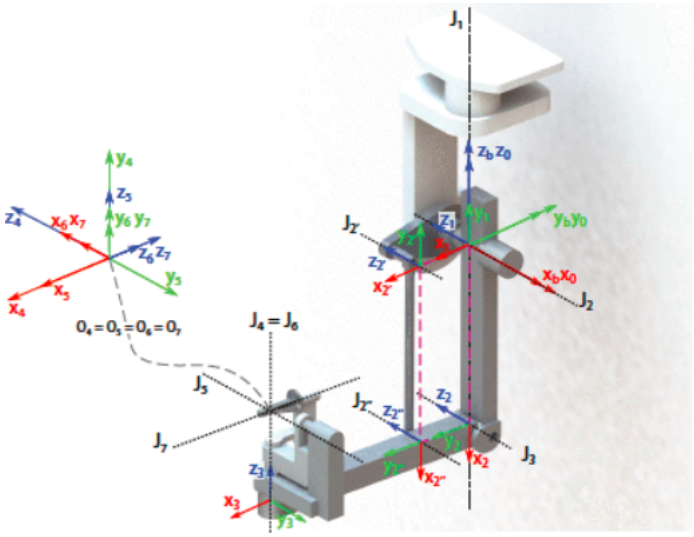

Fig. 2. MTM frame definition using modified DH convention. [14]

Dynamics model of the MTM is first derived using an open-source library called RBDL and an impedance controller was then designed based on the dynamics model. The controller is then tested in a simulated environment to evaluate its performance and the accuracy of the dynamics model. AMBF simulator is used for this work since it supports the simulation of the parallel mechanisms and provides real-time dynamics simulation. The following sections will cover the overview of the methodology used for this project followed by an in-depth overview of the AMBF simulator. Moreover, the dynamics modeling of MTM system is discussed and the results regarding the performance of the impedance controller are demonstrated. Finally in the conclusion section, we give a summary of the project and the future work.

II. Methods

As mentioned previously, the purpose of this project is to design an impedance controller for the da Vinci MTM arm. The requirement for the design of a controller is to have an accurate model of the dynamics of the system. The MTM is consisted of a hybrid parallel and serial links, which makes the dynamics modeling of such manipulator more challenging compared to conventional serial manipulators. Hence, we opted to approach the problem by first obtaining the dynamics of a serial system and then expanding the knowledge to the more complex parallel manipulator which in out case is the MTM arm. The serial manipulator that we selected is an industrial 7-DOF KUKA LBR arm. Our approach is valid for both the robots with slight modifications in the sub-tasks, so that the workflow from the Kuka LBR can be transferred to the MTM. The approach can be divided into three sub-tasks:

1) Dynamic modelling derivation for both manipulators
2) Implementation of the gravity compensation control for both manipulators
3) Design of an impedance controller for both manipulators

The dynamics of both of our systems is derived using an open source library called RBDL. More is coverd on the RBDL library and the procedures used to derive the dynamics for the robots in later sections. We tested our controllers in a simulated environment. AMBF simulator was used since it contains built-in models of both robots and it supports the dynamics simulation as well as the closed loop mechanisms.

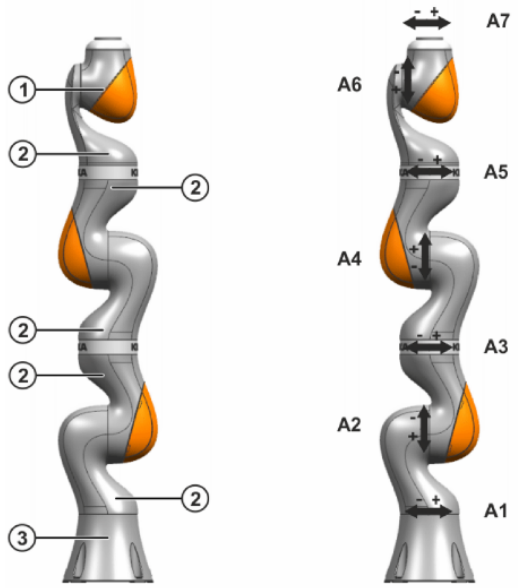

Fig. 3. a) Assembly of the KUKA LBR robot 1. In-line wrist 2. Joint module and 3. Base frame b) shows all 7 axes of rotation in the robot [15].

III. AMBF Simulator

Robot simulators are used to replicate real-world robotic systems while taking into consideration the physical and environmental factors that affect the robot's motion in their operational environment. Simulators are useful tools when developing new controllers since it allows for robotics programs to be conveniently written and debugged off-line without using the physical system. This is beneficial since it prevents any potential damage to the physical system during the development phase and helps to keep the development costs at a minimum. There is a wide variety of options available for robot simulators that are categorized based on the license, application, and their ability to simulation dynamic bodies. In this project, we have opted to use the AMBF simulator for simulation of both of the robotic systems namely the da Vinci MTM and KUKA LBR arm.

AMBF is an open-source simulator that offers a real-time dynamic simulation of multi-bodies including robots, free bodies, and multi-link puzzles. The simulator utilizes CHAI-3D framework to provide real-time haptic interaction via several haptic devices ranging from dVRK Manipulators to commercially available devices such as Razor Hydras and

Geomagic touch [16]. It also provides a python client for training neural networks and reinforcement learning agents on real-time data with the simulation in the loop. Other libraries such as BULLET-Physics, Open-GL, GLFW, yaml-cpp, are also used in the AMBF simulator that make the simulator accommodating to a variety of applications in the field of robotics. The AMBF simulator is capable of simulating parallel mechanisms such as the MTM and has built-in models of both the KUKA LBR and MTM arm in its libraries. The communication between the user and the simulator can be achieved by using either the provided python client or the ROS interface. In this project, the python API has been used for communication with the robot.

A. File description

The YAML format is used for robot description in AMBF as it allows modularity in body description. The individual link descriptions enable us to asynchronously control individual body movements. Because of this modular way of describing a body we can also include parallel mechanisms in our robot description (just like the dVRK MTM which has a parallel mechanism in it). The YAML has two components that recursively describe a robot i) BODY description and ii) JOINT description.

1) JOINT description : The Joint description will describe how two rigid bodies in a robot are connected to each other. It describes each joint with a parent and child, unlike URDF which describes a single parent for the entire robot. Defining a parent for a joint is done by using a Parent axis and a parent pivot which are defined with respect to simulator's world coordinates. Similar description is used for a child. In Fig 4. we show a sample of joint description in YAML.

```
JOINT base-link1:
  name: base-link1
  parent: BODY base
  child: BODY link1
  parent axis: {x: 0.0, y: 0.0, z: 1.0}
  parent pivot: {x: 0.0, y: 0.0, z: 0.103}
  child axis: {x: 0.0, y: 0.0, z: 1.0}
  child pivot: {x: 0.0, y: 0.0, z: 0.0}
  joint limits: {high: 2.094, low: -2.094}
  controller: {D: 2.0, I: 0, P: 1000.0}
  type: revolute
```

Fig. 4. JOINT descriptions of two rigid bodies in YAML

2) BODY description : The body description of a rigid body has the name, mass, spatial orientation, inertia values and their offsets, friction values, damping coefficients along with color information of a rigid body in the robot Fig.5 shows a sample of how the Body description looks like. It also mentions which file contains the rendering information of this specific body (.STL).

```
BODY base:
  name: base
  mesh: base.STL
  mass: 0.0
  collision margin: 0.001
  scale: 1.0
  publish children names: true
  publish joint names: true
  publish joint positions: true
  location:
    orientation: {p: -0.0, r: 0.0, y: 0.0}
    position: {x: 0.0, y: 0.0, z: -1.3}
  inertia: {ix: .001, iy: .001, iz: .001}
  inertial offset:
    orientation: {p: 0, r: 0, y: 0}
    position: {x: 0.001, y: -0.0, z: 0.06}
  publish joint names: true
  publish joint positions: true
  friction: {rolling: 0.01, static: 0.5}
  damping: {angular: 0.95, linear: 0.95}
  restitution: 0
  collision groups: [0]
  color components:
    ambient: {level: 1.0}
    diffuse: {b: 0.0147, g: 0.0147, r: 0.0147}
    specular: {b: 1.0, g: 1.0, r: 1.0}
    transparency: 1.0
```

Fig. 5. BODY descriptions of a rigid body in YAML

IV. Dynamics Modeling

The dynamics of the system describe the motion of the robot considering the forces and moments that are applied on the robot. Understanding and derivation of the dynamics of the robot is important when designing a controller, since the motion of the robot is derived from the dynamics of the system and is used to calculate the error between the desired and actual values of the commanded control signal. Thus, an accurate dynamics model of the robotic system is absolutely neccassary for a successful implementation of a controller. The generic eqation of motion of a rigid body described in the joint space is in the following form:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \tag{1}$$

where M is the inertia matrix, C is the velocity dependant matrix, G is the gravity matrix and τ is the joint torques. $\ddot{q}$, $\dot{q}$, q, are joint acceleration, velocity, and position respectively. The dynamics model for the manipulators can be computed by using different methods namely Euler-Lagrange , Newton Euler [17], parameter identification , and more recently deep learning techniques [18]. We opted to use an open-source library called RBDL to derive the dynamics of both of our robotic systems.

A. Rigid Body Dynamics Library

Rigid Body Dynamics Library (RBDL) [19] is an open-source package which utilizes the Newton-Euler approach to

derive the dynamics of the robot model. The library is capable of deriving the forward and inverse kinematics, as well as the forward and inverse dynamics of the robotic systems. Also, the library has options that provides additional information such as the Jacobian and the inertia matrix to the user. All these capabilities make the RBDL a great tool for calculating the dynamics of a robotic system.

B. YAML to RBDL Parser

Robot models are described in the RBDL using the description files of the robot. Since we are using YAML for robot description we decided to write a universal parser for any robot which has its description written in YAML to create an RBDL model out of it. The tricky part here was in constructing the transformation matrices of each frame of the robot because in YAML, each joint has a parent and child axis associated with it and now we had to calculate the transformation between them using this information. Rest apart, RBDL itself has very less documentation written for modeling a robot, hence we had to figure out how to parse the exact robot model of our system which gives us an accurate representation of our actual system. We finally were able to come up with a YAML to RBDL parser that solves the modeling problem.

C. Dynamic modelling of the manipulators

As stated previously, we first aimed to derive the dynamics model for the KUKA arm using the RBDL. The reason behind that was to get familiar with the RBDL library by having a relatively more generic system to solve for and then extending our work to the MTM arm. We first began by parsing the YAML description file and passing it to the RBDL to obtain the inverse dynamics model of the KUKA arm. By using the inverse dynamics function in the RBDL we were able to calculate the joint torque values provided position, velocity and acceleration input values at each joint.

D. Gravity Compensation for the Manipulators

To check the accuracy of the dynamics model created in the RBDL, we used gravity compensation control solely based on the dynamics of the robot. This is achieved by passing zero values for both the acceleration and velocity terms to the inverse dynamics function. Thus, The generic equation of motion of the system reduces to a simple gravity term which accounts for the gravitational forces that act on each link of the robot and outputs the joint torques that are necessary to hold the robot in any given configuration provided that no initial velocity or acceleration act on the system. To test the gravity compensation the robot is moved to a random position while the gravity compensation is activated. The desired behavior of the controller is to maintain the position of the arm in space with no motion. After testing the dynamics of the KUKA arm, we turned our attention to the MTM arm. The procedure was mostly similar to that of the KUKA arm with the only exception being the use of the loop constraints for defining the closed-loop mechanism of the MTM arm. Similarly we tested the accuracy of the dynamics model of the MTM by the use of a gravity compensation controller. The simulation results for the gravity compensation tests for both robots can be seen in the fig. 6 where robots are placed in random configurations with the gravity compensation enabled. For both robot systems the controller was able to maintain the position of the robots with out using any error compensating terms.

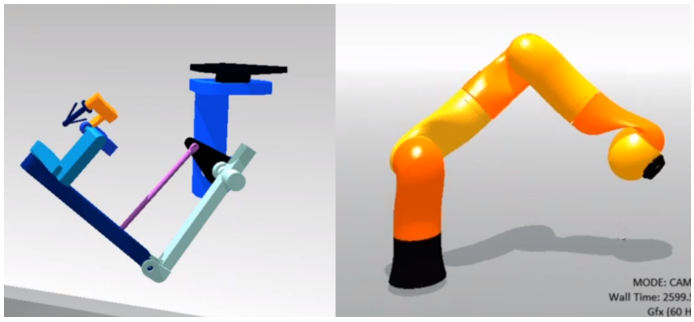

Fig. 6. Testing g of the dynamics model using the gravity compensation. Robots are set at random configurations and the joint values are used as the input to the inverse dynamics function. The function calculates the joint torques required to hold the robot in the specified location.

V. Controller Design

Impedance control has proved to be useful technique for controlling the interactions of the manipulator with its environment in a safe and compliant manner [7], [8]. It provides the synergy between the position and the force control of the end-effector operating along multiple degrees of freedom (DOF) in the Cartesian space [9], [10]. Impedance control is widely used in tasks that require positioning of the end-effector along a trajectory, while regulating the force interactions with the environment. Such applications can be found in human-robot cooperative robotics [11], [12] as well as in the surgical robotics [13]. One benefit of using an impedance control on surgical robots is that it enables force control in uncertain and dynamically changing environments such as within the patient's body. Additionally, the sensors at the joints provide compliance in cases of collision with objects. Currently the da Vinci system lacks this feature and is only controlled using a classical PID controller, which lack the crucial compliance needed for both the MTM and PSM arms. This project aims to design an impedance controller that could be used for the MTM console of the dVRK system.

A. Mathematical Formulation

Let $x_d, \dot{x}_d, \ddot{x}_d$ be the desired end-effector position, velocity and acceleration trajectories in the task space respectively. Therefore, the time domain the impedance controller Force is expressed as:

$$F = \left(\frac{M_x}{M_d} - 1\right) F_{ext} - \frac{M_x}{M_d}(D_d\dot{e} + K_d e) \qquad (2)$$

where M_d, B_d, K_d represent the desired inertia, damping and stiffness matrices determined by designer; F_{ext} is the external contact force; $e, \dot{e}$ are the deviations in the position

and velocities of the end-effector from the desired trajectories in the task space. M_x is the system inertia in task space determined as,

$$M_x = J^{-T} M(q) J^T \tag{3}$$

This Force at the robot end-effector thus can be applied through joint torques along with the Gravity Compensation as,

$$u = J^T F + G(q) \tag{4}$$

Thus, the robot tracks the desired trajectory while the compliance and damping behavior of the robot can be modified as per the requirements of the task.

VI. Results

This section provides results to numerous tests performed with the impedance controller on the KUKA and MTM of dVRK.

1) Trajectory following: In this test, we command to the KUKA robot to follow a straight line along the X axis as shown in fig. 7. The resultant end-effector trajectory in the X-Y plane is depicted in fig [8].

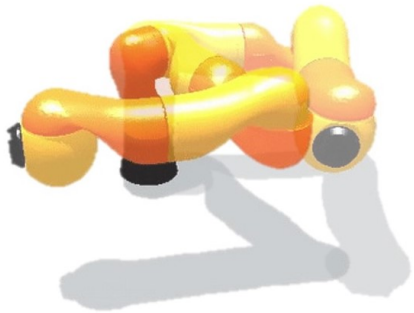

Fig. 7. KUKA-LBR Single Line Trajectory

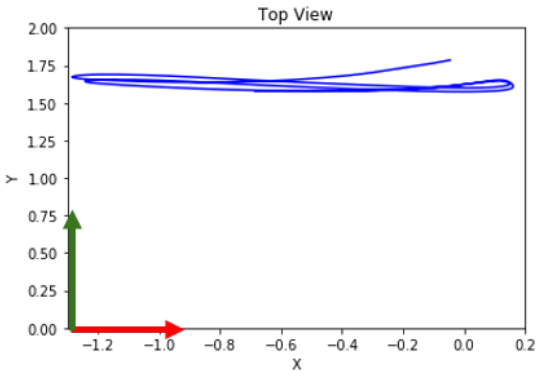

Fig. 8. MTM robot following a Line Trajectory subjected to external disturbance.

Fig. 9. MTM Single Line Trajectory

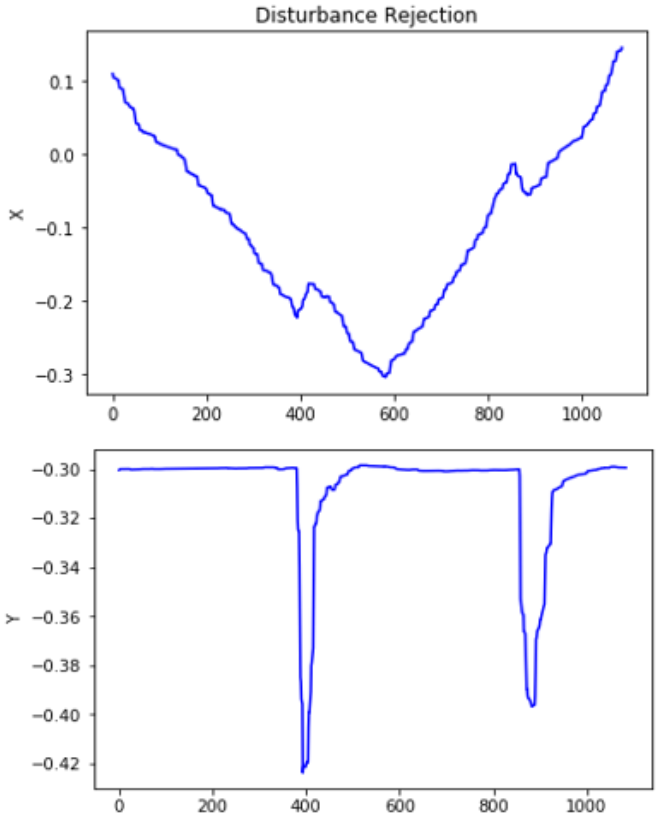

Fig. 10. MTM Robot subjected to external disturbance (a) X position trajectory of end-effector. (b) Y position trajectory of end-effector.

2) Response to external disturbance: In this test, we subject external force to the MTM following a straight line along the X axis, as shown in fig. 9.

From fig [10], we observe that the robot motion is critically damped and it quickly converges to the desired motion. Fig 10(a) and (b) shows the response of the MTM in X and Y direction respectively. Response along Z direction shows behavior similar as that in Y. With the impedance controller we can not only control to motion of the robot but also the interaction of it with the environmental elements. We also

tested the controller response under different stiffness values. As shown in the fig. (11), the robot exerts more resistance to disturbance under high stiffness controller setting.

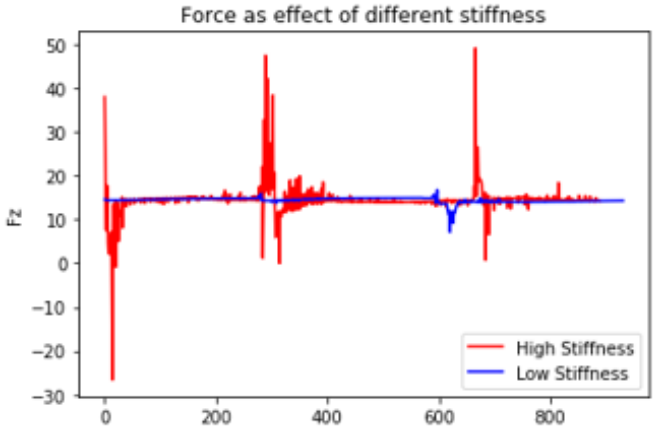

Fig. 11. Force at the end-effector resultant of the disturbance to the MTM under different controller stiffness setting.

3) Motion - Force Selective Control: In this test, we set high stiffness along the X and Y direction while low stiffness along Z direction. When deviated from the desired trajectory, the robot converges to the X_d and Y_d while holding the Z position , as shown in fig. 14. This test demonstrates the controller's ability control and tweak motion and force along different directions. Such abilities are highly useful for tasks that require hybrid force-motion control such as writing on whiteboard.

4) Varying K_p values Influence on trajectory: In this test, we set different K_p values when still command the MTM following a straight line along the X axis, as shown in fig. 9. More specifically, our setting is: $x_{desired} = -0.4m + 0.35m, vx_{desired} = 0.5m/s$, with other axis position locked and velocity to be zero. The result will look like what Figure 13 shows.

The profiles can be divided into $K_p = 0.3/0.5/0.7$, all in three groups. Each group has two figures: the upper one shows how the desired position and real position match with each other. The lower one shows its velocity profile as well.

As K_p increases, time cost for robot finishing three entire loops will decrease, from 5000ms (5s) to 2.5s. Meanwhile, in the velocity profiles the largest noise values will also increase. This phenomenon indicates that K_p will influence the system response rate: since K_p is defined as the feedback parameter of error position, as it increases, the feedback force and real-time velocity will also increase simultaneously. In this case, larger K_p can let the system run more efficiently, but also more easily make it unstable.

5) Controller response to Obstacles in the path: In this experiment, we place a block in the path of the end-effector and observe the robot's response for impedance and PD-position controller. For the PD Controller, the error increases as the robot is expected to follow the time-stamped position trajectory. Eventually the controller throws the box from its

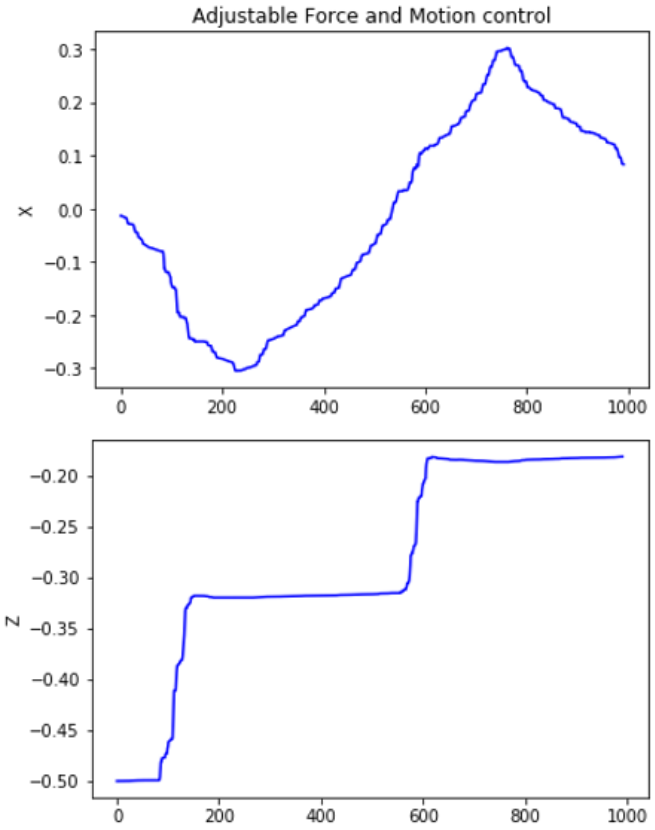

Fig. 12. Controller response for hybrid motion and Force control for MTM robot.

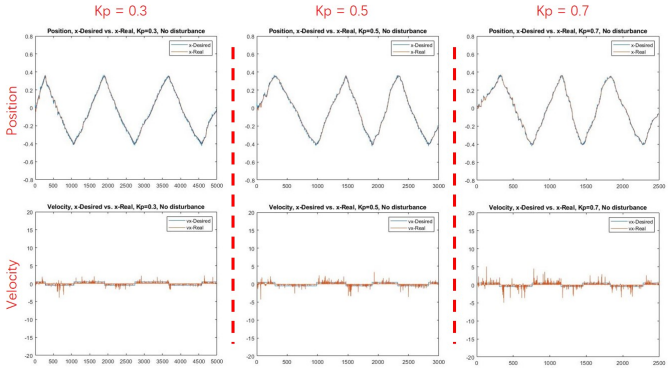

Fig. 13. Kp=0.3/0.5/0.7, Within/out disturbance, tip Position/Velocity profiles of MTM Trajectory

position. However, the impedance controller stops the robot at the position as soon as it touches the box. Thus, impedance controller is highly useful for tasks which involves human intervention in the robot's workspace.

VII. Conclusion

In this work we presented our approach and analysis for the impedance controller implemented on the KUKA LBR and MTM of the dVRK in AMBF simulator. We computed the kinematics and dynamics of the systems using the RBDL.

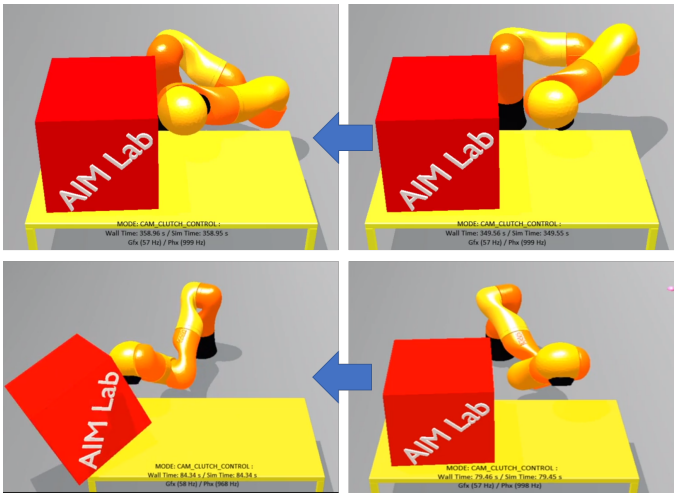

Fig. 14. Comparison between impedance (top), PD (bottom) controller. In both the figures the robot is following a horizontal line trajectory. The impedance controller is set with a low stiffness in the direction of the trajectory making the robot compliant when it makes contact with an external object. The PD controller as expected demonstrates a non-compliant behaviour when making contact with the box and droping it off the table showing the inherent lack of compliance in the classical PD controller.

We also developed a YAML parser which can be used for any open-chain system to automatically generate the robot model in the RBDL. Implementing gravity compensation, Computed Torque controller and task-space impedance controller, we provided a comprehensive analysis of capabilities and shortcomings of these methods. Thus, as a contribution towards the development of AMBF simulator, we have two fully documented working robot model examples with working dynamics and controllers. Also, we added new function in the python API to get inertia of the bodies from AMBF. Overall, this work is a good example for working with ROS, dynamics using the RBDL, AMBF simulator and different controller schemes, providing great learning experience to the students.

Future works would include writing a generalised parser to handle constraints and closed-chain systems. Currently, the AMBF simulator lacks velocity feedback. A plugin to get the joint velocity feedback can allow engineers to implement wide range of controllers. Also, The controllers can be implemented on the real system. A beginning step would be implementing gravity compensation for MTM system.

Finally, as an acknowledgement, we would like to thank Dr. Adnan Munawar for his support and guidance throughout the project.

REFERENCES

[1] J. Ruurda, T. J. Van Vroonhoven, and I. Broeders, "Robot-assisted surgical systems: a new era in laparoscopic surgery." *Annals of the Royal College of Surgeons of England*, vol. 84, no. 4, p. 223, 2002.

[2] I. Surgical, "About da vinci systems," https://www.davincisurgery.com/da-vinci-systems/about-da-vinci-systems, 2019.

[3] U. Leung and Y. Fong, "Robotic liver surgery," *Hepatobiliary surgery and nutrition*, vol. 3, no. 5, p. 288, 2014.

[4] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, "An open-source research kit for the da vinci surgical system," in *IEEE Intl. Conf. on Robotics and Auto. (ICRA)*, Hong Kong, China, 2014, pp. 6434–6439.

[5] Z. Chen, A. Deguet, R. H. Taylor, and P. Kazanzides, "Software architecture of the da vinci research kit," in *2017 First IEEE International Conference on Robotic Computing (IRC)*. IEEE, 2017, pp. 180–187.

[6] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, "An open-source research kit for the da vinci surgical system," *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6434–6439, 2014.

[7] C. Ott, A. Albu-Schaffer, A. Kugi, S. Stamigioli, and G. Hirzinger, "A passivity based cartesian impedance controller for flexible joint robots-part i: Torque feedback and gravity compensation," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 3. IEEE, 2004, pp. 2659–2665.

[8] G. J. Lahr, J. V. Soares, H. B. Garcia, A. A. Siqueira, and G. A. Caurin, "Understanding the implementation of impedance control in industrial robots," in *2016 XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR)*. IEEE, 2016, pp. 269–274.

[9] N. Hogan, "Impedance control: An approach to manipulation: Part itheory," *Journal of dynamic systems, measurement, and control*, vol. 107, no. 1, pp. 1–7, 1985.

[10] T. Yoshikawa, T. Sugie, and M. Tanaka, "Dynamic hybrid position/force control of robot manipulators-controller design and experiment," *IEEE Journal on Robotics and Automation*, vol. 4, no. 6, pp. 699–705, 1988.

[11] R. Ikeura, H. Monden, and H. Inooka, "Cooperative motion control of a robot and a human," in *Proceedings of 1994 3rd IEEE International Workshop on Robot and Human Communication*. IEEE, 1994, pp. 112–117.

[12] R. Ikeura and H. Inooka, "Variable impedance control of a robot for cooperation with a human," in *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, vol. 3. IEEE, 1995, pp. 3097–3102.

[13] T. Haidegger, B. Benyó, L. Kovács, and Z. Benyó, "Force sensing and force control for surgical robots," *IFAC Proceedings Volumes*, vol. 42, no. 12, pp. 401–406, 2009.

[14] G. A. Fontanelli, F. Ficuciello, L. Villani, and B. Siciliano, "Modelling and identification of the da vinci research kit robotic arms," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1464–1469.

[15] KUKArobotics, https://www.kuka.com/en-us/products/robotics-systems/industrial-robots/lbr-iiwa, May 2019.

[16] A. Munawar, "Asynchronous multi-body framework," https://github.com/WPI-AIM/ambf, 2019.

[17] R. Featherstone, *Rigid body dynamics algorithms*. Springer, 2014.

[18] M. Lutter, C. Ritter, and J. Peters, "Deep lagrangian networks: Using physics as model prior for deep learning," *arXiv preprint arXiv:1907.04490*, 2019.

[19] M. L. Felis, "Rbdl: an efficient rigid-body dynamics library using recursive algorithms," *Autonomous Robots*, vol. 41, no. 2, pp. 495–511, 2017.