A Dynamic Model Identification Package for the da Vinci Research Kit

Aabha Thamanakar, Ankush Singh Bhardwaj, Chinmayee Prabhakar, Nikunj Reddy Polasani, Prasanna Natu

Abstract:

A dynamic model is essential to perform dynamic simulations and model based control. Our work is based on the research paper "A Dynamic Model Identification Package for the da Vinci Research Kit" [1]. The package is capable of modelling the parallelograms , springs, counterweight, and tendon couplings used in dVRK. A dVRK model was also simulated on Unity 3D to perform trajectory planning. The package developed can also ve applied on robots with similar characteristics as dVRK through simple configuration.

Introduction to da Vinci Research Kit (dVRK):

A dVRK is a telerobotic surgical system derived form the first generation da Vinci Surgical System provided by intuitive surgical. The robot uses a minimally invasive surgical approach.

The PSM utilized for this research contains three arms, each with seven degrees of freedom (DOF). It is important to note that in previous research, the last degree of freedom, which corresponds to the opening and closing motion of the gripper, was not taken into consideration because the focus was on computing the position and orientation of a frame attached to the center of the gripper with respect to the base frame. However, in this paper, the gripper jaws are designed and actuated as two separate links, which is considered in the model.

The dVRK-ROS package uses the DH convention based on kinematic frames that are located on the joint axes, however practically all the joints are operated off axis using a combination of cams, links, cables, and pulleys. To connect the joint movements, as described in the dVRK-ROS package, with the motor torques, extra coordinate axes must be established. Numerous types of joint coordinates are defined in the dVRK-ROS package to construct the link between the robot joint motion and the torque of each motor.

Methodology:

The robot parameters and DH were defined. The friction considered at motor joints were Coloumb and Viscous. The DH parameters were calculated using the robot parameters of the dVRK present at the AIM lab. Each link's frame is specified by applying the modified Denavit-Hartenberg (DH) standard. Recursively, depending on the transformation from one frame to its preceding frame, the homogeneous transformation for each frame is determined. The DH parameters were calculated, additional coordinate axes were defined to relate the joint motions with motor torques. "qd" was defined in the kinematic modelling given by:

$$q=[(qb)^T(qa)^T]^T$$

We have the geometry function, this creates the kinematic chains and draws the links of the robot. of the dVRK PSM and estimating its DH parameters. These calculations are broken up into many packages that describe their respective functions based on modeling of geometry, kinematics, and dynamics. The inertial component of the links and the different friction parameters (Coloumb, Viscous, etc) are considered while formulating the geometric model. Each link's frame is specified by applying the modified Denavit-Hartenberg (DH) standard. Recursively, depending on the transformation from one frame to its preceding frame, the homogeneous transformation for each frame is determined.

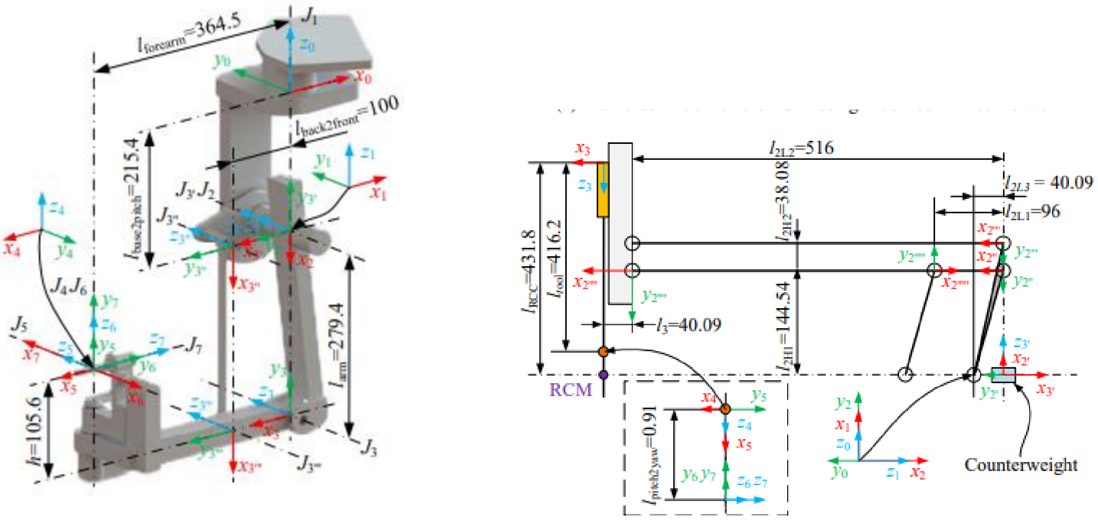

Fig 1 (a) MTM frame definition using modified DH convention, Fig 1 (b) Dimension definition of the PSM

link	joint type	prev	succ	a_{i-1}	α_{i-1}	d_i	θ_i	link inertia	motor inertia	friction	spring
1	R	0	2	0	$\pi/2$	0	$q_1+\pi/2$	✓	✕	✓	✕
2	R	1	$2', 3'$	0	$-\pi/2$	0	$q_2-\pi/2$	✓	✕	✓	✕
$2'$	-	2	$2'', 2'''$	l_{2L3}	0	0	$\pi/2$	✕	✕	✕	✕
$2''$	R	$2'$	$2'''', 2''''', 3$	l_{2H1}	0	0	$-q_2+\pi/2$	✓	✕	✕	✕
$2'''$	R	$2'$	-	$l_{2H1}+l_{2H2}$	0	0	$-q_2+\pi/2$	✓	✕	✕	✕
$2''''$	R	$2''$	3	l_{2L2}	0	0	q_2	✓	✕	✕	✕
$2'''''$	R	$2''$	-	l_{2L1}	0	0	$q_2+\pi$	✓	✕	✕	✕
3	P	$2''''$	4	l_3	$-\pi/2$	$q_3-l_{RCC}+l_{2H1}$	0	✓	✕	✓	✕
$3'$	P	2	-	l_{2L3}	$-\pi/2$	q_3	0	✓	✕	✕	✕
4	R	3	5	0	0	l_{tool}	q_4	✕	✓	✓	✓
5	R	4	6, 7	0	$\pi/2$	0	$q_5+\pi/2$	✕	✓	✓	✕
6	R	5	-	$l_{pitch2yaw}$	$-\pi/2$	0	$q_6+\pi/2$	✕	✕	✓	✕
7	R	5	-	$l_{pitch2yaw}$	$-\pi/2$	0	$q_7+\pi/2$	✕	✕	✓	✕
M_6	R	-	-	0	0	0	q_6^m	✕	✓	✓	✕
M_7	R	-	-	0	0	0	q_7^m	✕	✓	✓	✕
F_{67}	R	-	-	0	0	0	q_6-q_7	✕	✕	✓	✕

link	joint type	prev	succ	a_{i-1}	α_{i-1}	d_i	θ_i	link inertia	motor inertia	friction	spring
1	R	0	$2, 3'$	0	0	$-l_{base2pitch}$	q_1	✓	✕	✓	✕
2	R	1	3	0	$-\pi/2$	0	$q_2+\pi/2$	✓	✕	✓	✕
3	R	2	4	l_{arm}	0	0	$q_3+\pi/2$	✓	✕	✓	✕
$3'$	R	1	$3''$	0	$-\pi/2$	0	$q_2+q_3+\pi$	✓	✕	✓	✕
$3''$	R	$3'$	$3'''$	$l_{back2front}$	0	0	$-q_3-\pi/2$	✓	✕	✓	✕
4	R	3	5	$l_{forearm}$	$-\pi/2$	h	q_4	✓	✕	✓	✕
5	R	4	6	0	$\pi/2$	0	q_5	✓	✕	✓	✓
6	R	5	7	0	$-\pi/2$	0	$q_6+\pi/2$	✓	✕	✓	✕
7	R	6	-	0	$-\pi/2$	0	$q_7+\pi$	✓	✕	✓	✕
M_4	R	-	-	0	0	0	q_4^d	✕	✓	✓	✕

Fig 2: Modelling Description (DH convention) of the PSM and MTM. PSM Links 1 to 7 correspond to the links described in Fig 3. M_6 and M_7 correspond to the friction and inertia modelling of Motor 6 and Motor 7. F_{67} corresponds to the friction modelling between link 6 and link 7.

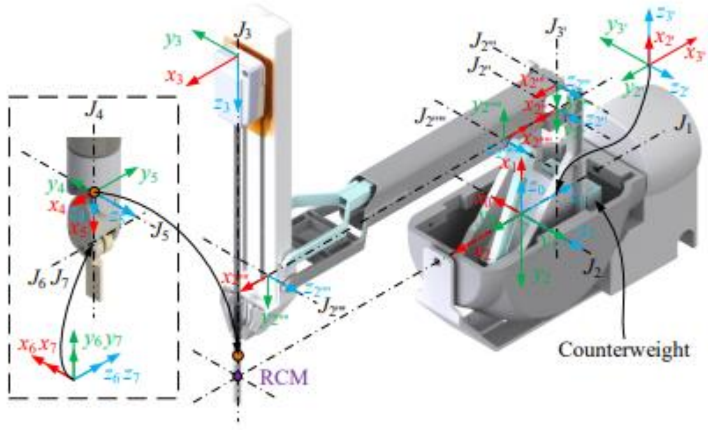

Fig 3: (a) Frame definition of the PSM using modified DH Convention (The dimensions for the geometric modelling are obtained from the user guide of the dVRK.

The Kinematics library defined then creates kinematic chains of the MTM and PSM respectively. The geometry library then plots these kinematic chains defined at the home configuration.

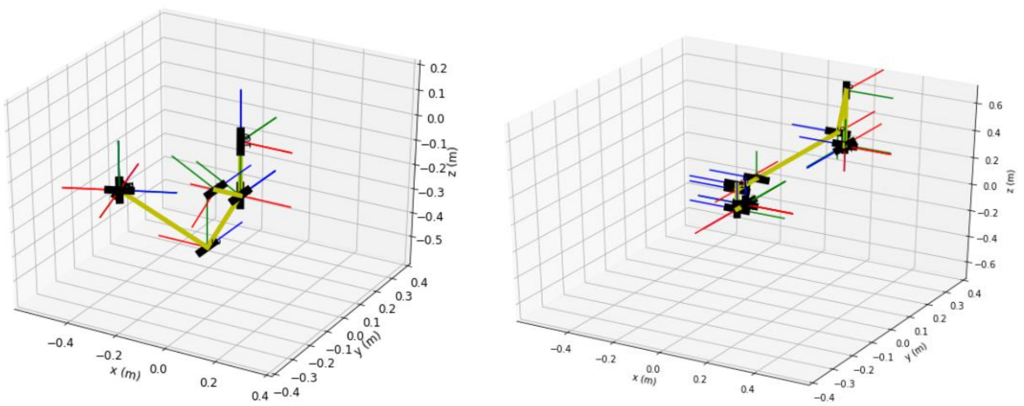

Fig 4 (a) Geometric representation of MTM (b) Geometric representation of PSM with defined DH values.

The equations of motion are expressed in linear form of dynamic parameters by employing the barycentric parameters [1], in which the mass of link is first employed, followed by the first moment of inertia, to describe the equations of motion as a linear form of dynamic parameters.

```
In [7]: robot_def.bary_params
```

Out[7]: $[L1xx, L1xy, L1xz, L1yy, L1yz, L1zz, l1x, l1y, l1z, m_1, Fc_1, Fv_1, Fo_1, L2xx, L2xy, L2xz, L2yy, L2yz, L2zz, l2x, l2y, l2z, m_2, Fc_2, Fv_2, Fo_2, L4xx, L4xy, L4xz, L4yy, L4yz, L4zz, l4x, l4y, l4z, m_4, L5xx, L5xy, L5xz, L5yy, L5yz, L5zz, l5x, l5y, l5z, m_5, L6xx, L6xy, L6xz, L6yy, L6yz, L6zz, l6x, l6y, l6z, m_6, L7xx, L7xy, L7xz, L7yy, L7yz, L7zz, l7x, l7y, l7z, m_7, L8xx, L8xy, L8xz, L8yy, L8yz, L8zz, l8x, l8y, l8z, m_8, Fc_8, Fv_8, Fo_8, L9xx, L9xy, L9xz, L9yy, L9yz, L9zz, l9x, l9y, l9z, m_9, Fc_{10}, Fv_{10}, Fo_{10}, Ia_{10}, K_{10}, Fc_{11}, Fv_{11}, Fo_{11}, Ia_{11}, Fc_{12}, Fv_{12}, Fo_{12}, Fc_{13}, Fv_{13}, Fo_{13}, Fc_{14}, Fv_{14}, Fo_{14}, Ia_{14}, Fc_{15}, Fv_{15}, Fo_{15}, Ia_{15}, Fc_{16}, F$

Fig 5: Barycentric Parameters are defined

The dynamic base parameters [2], also known as lumped parameters, a minimal set of dynamic parameters that may adequately represent the dynamic model of a robot are determined. While base parameters are sufficient to accurately express a robot's dynamics in dynamic equations, standard parameters are necessary for effective recursive Newton-Euler-based dynamic algorithms

The dynamics library written calculates the kinetic energy of each link and there by the torques by adding motor inertia and friction. The dynamics takes into account the springs, counter weight and the parallelograms in the dVRK. The dynamic model is developed using the Euler-Lagrange equation. The dynamics of the dVRK are modeled using the closed-chain robot Euler-Lagrange equation . The Lagrangian defined by this technique is provided by L = K - P, where K is the robot's kinetic energy and P is its potential energy. Springs and motor inertias are modeled independently and are not included in L.

```
In [11]: sympy.Matrix(dyn.base_param)
```

Out[11]:

$$\left[\begin{matrix} 1.0m_9 \\ 1.0m_8 \\ -0.5l2y + 0.5l6y - 0.5l7y + 1.0l8z - 0.07227m_4 - 0.09131m_5 - 0.07227m_6 - 0.07227m_7 \\ 0.5l2x - 0.5l6x + 0.5l7x + 1.0l9x + 0.020045m_4 + 0.020045m_5 + 0.020045m_6 + 0.020045m_7 \\ 0.5l2y - 0.5l6y + 0.5l7y + 1.0l9z + 0.07227m_4 + 0.09131m_5 + 0.07227m_6 + 0.07227m_7 \\ -0.5l2x + 0.5l6x - 0.5l7x + 1.0l8x - 0.020045m_4 - 0.020045m_5 - 0.020045m_6 - 0.020045m_7 \\ -0.5l1y - 0.5l2z - 0.5l4z - 0.5l5z - 0.5l6z - 0.5l7z + 1.0l9y \\ -0.5l1y - 0.5l2z - 0.5l4z - 0.5l5z - 0.5l6z - 0.5l7z + 1.0l8y \\ 1.0l5y \\ 1.0L2xy + 1.0L6xy + 1.0L7xy + 1.0L8xz + 1.0L9xz + 0.2159l2x \end{matrix}\right]$$

Fig 6 : Dynamic model of the PSM

Challenges:

Though the project was based on research paper, the code provided couldn't be run directly as many libraries and functions have depreciated over time (paper coded in python 2.7, we have made our code work with python 3). Some of the packages had to be re written like the geometry, kinematics and dynamic.

The formulation of Lagrange was quite complicated. The dynamics package consumed most of our time to just understand and make it work, especially with differentiation of the lagrange.

Also, the motors of the PSM are positioned at the base of the robot and the joints are driven through cables bringing elasticity, backlash and non-linear friction into the equation complicating the model. Additionally, the existence of a 1-DOF double parallelogram and a counterweight in the PSM complicated the Kinematic modeling.

We did attempt trajectory optimization but couldn't complete it on time. The dVRK ros stack had lot of issues and threw errors constantly. We then decided to make a simulation on Unity 3D. Later when we couldn't move the robot in unity 3D we reverted back to ros and debugged the code which is explained in detail below.

Challenges faced for simulation in gazebo/unity

The simulations of the dynamic modelling were particularly challenging, the dvrk stack in the github repo of the paper was continually failing, and we were not able to carry out required simulations since the excitation trajectory would not be replicable even with zero errors while compiling.

We were not able to identify the root cause of this scenario, which is why we decide to try to simulate the robot in two instances of Unity -3D platform. One of the unity instances was a clone of the model included in the research paper and the other instance was mixed reality where we faced ROS parsing issues due to incompatibility with ubuntu versions. We were not able to completely resolve this issue either, since published values from ros would not get read by the unity software.

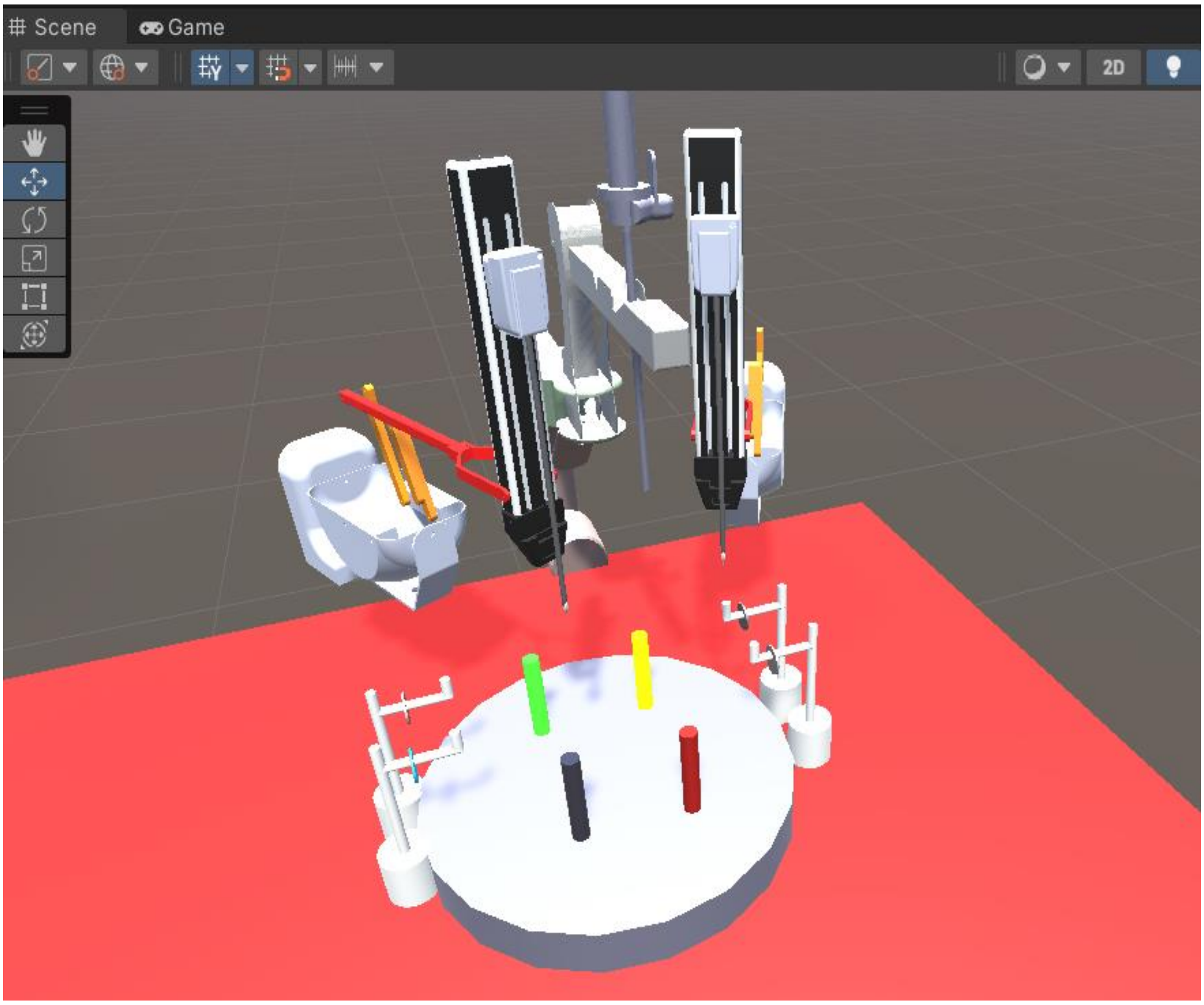

Figure 7. dVRK model in unity [3]

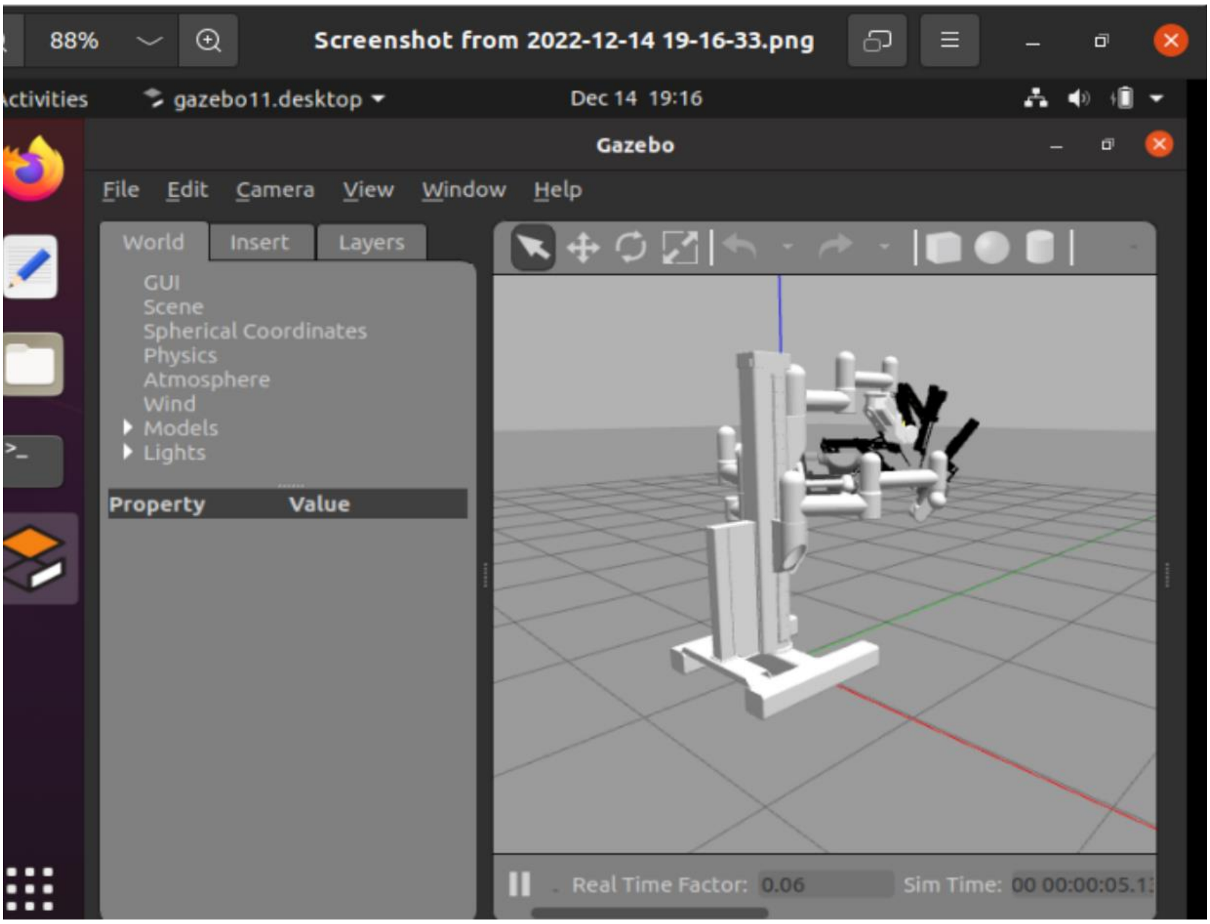

Figure 8. the dVRK model in gazebo [4].

Therefore, we decided to continue the rest of our project using a URDF file, ROS 1 noetic and gazebo. Where we had to work with an older ros stack which met the criteria of our research paper. For this we had to familiarize ourselves with ROS 1 and rectify several instances in the urdf and xacro files to make the robot work. After which successful simulation were ran for all arms of the dVRK for every iteration of the robot.

Results

We successfully created a dynamic robot model for the dVRK using the Lagrange formulation. The dynamic model considers counter weights, parallelogram, springs.

A forward Kinematic simulation was created for the dVRK, PSM. The model was run on ROS and the video clip recording of the resulting motion is attached.

Future Scope:

The robot dynamic model could be used to model robots which have similar design. The trajectory optimization and experimental results can verify the results of this model. Other formulations like Recursive Newton Euler can also be used to create a dynamic model.

References:

[1] Wang, Y., Gondokaryono, R., Munawar, A., & Fischer, G.S. (2019). A Dynamic Model Identification Package for the da Vinci Research Kit. *ArXiv, abs/1902.10875*.

[1] Wang, Y., Gondokaryono, R., Munawar, A. and Fischer, G.S., 2019. A convex optimization-based dynamic model identification package for the da Vinci Research Kit. *IEEE Robotics and Automation Letters*, *4*(4), pp.3657-3664.

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

[3] inproceedings qian2019dvrkxr," dVRK-XR: Mixed Reality Extension for da Vinci Research Kit", Qian, Long and Deguet, Anton and Kazanzides, Peter, Hamlyn Symposium on Medical Robotics, 2019.

[4.] Gondokaryono RA, Agrawal A, Munawar A, Nycz CJ, Fischer GS, An Approach to Modeling Closed-Loop Kinematic Chain Mechanisms, applied to Simulations of the da Vinci Surgical System, Special Issue on Platforms for Robotics Research - Acta Polytechnica Hungarica, Vol 16, No 8, pp 29-48, Nov 2019

Reference:

[1] Y. Wang, R. Gondokaryono, A. Munawar and G. S. Fischer, "A Convex Optimization-Based Dynamic Model Identification Package for the da Vinci Research Kit," in *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3657-3664, Oct. 2019, doi: 10.1109/LRA.2019.2927947.