Study of forward and inverse kinematics and test of ROS implementation of PSM arm in dVRK

Robotics Project
Master's degree in Computer Science and Engineering
University of Verona
Filippo Mantovan VR439218

***Abstract*—This document concerns the study of the forward and inverse kinematics of the PSM arm present in the da Vinci Research Kit (dVRK). Moreover, it deals with the test of the ROS implementation of the PSM created by John Hopkins University [1] contained in the dVRK and an implementation of the inverse kinematic in closed form. The test was done by installing the dVRK tool and using RViz to see how the robot moves. Furthermore, a Python script has been developed, which uses python functions (binded to ROS topics) made available by the dVRK, which allow to retrieve information on the position of the end effector and on the value of the joint variables. In this script a closed form solution of the manipulator inverse kinematics is implemented.**

I. Robot structure

The psm arm has 7 degrees of freedom and moves a surgical instrument around a fixed fulcrum point, that acts as an invariant respect to the PSM joint configuration. This point is called Remote Center of Motion (RCM). The last degree of freedom - the seventh -, corresponding to the opening and closing motion of the gripper, is not considered because the focus is in computing the position and orientation of the center of the end effector, without considering the opening functionality of the gripper. This decision was also taken by those who developed the API of the dVRK, in fact, as you will see in the Python script, only 6 joints are requested and provided. These six degrees of freedom are combined in a RRPRRR sequence, where R correspond to Revolute joint and P correspond to Prismatic joint.

Referring to the figure 1, these six degrees of freedom are:

- all the arm rotate about axis J_1 of an angle ϑ_1
- the two parallelograms (in black in the figure) constitute a mechanism that allows rotation of the surgical instrument about axis J_2 of an angle ϑ_2
- the variable length d_3 allows the surgical instrument to translate along the J_3 axis
- the surgical instrument can also rotate of an angle ϑ_4 about axis J_4
- the surgical instrument rotate about axis J_5 of an angle ϑ_5
- the surgical instrument rotate about axis J_6 of an angle ϑ_6

The axes J_1, J_2, J_3, J_4 collide at one point: RCM. The position of the RCM does not depend on the variables of the joints.

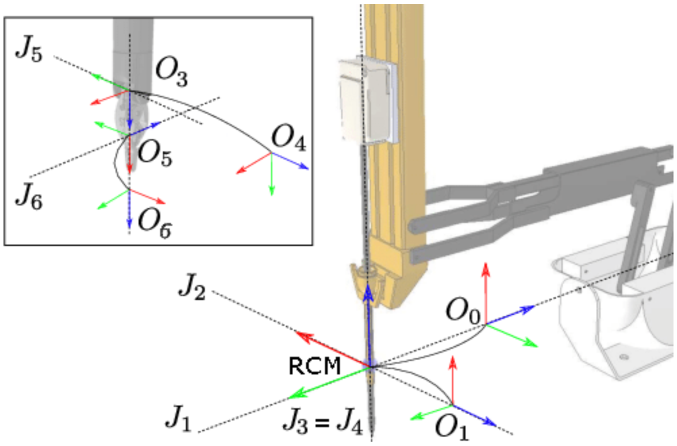

Figure 1: Patient Side Manipulator (PSM) kinematic description (partially taken from [2])

Donald L. Pieper [3] demonstrated that a sufficient condition for the kinematic structure of a six-degree-of-freedom manipulator to be resolvable is that the axes of three consecutive rotational joints intersect at a single point for all manipulator configurations. When the axes of the last three joints intersect at a single point, the last three joints are said to identify a spherical wrist. In our case, the joints on the axes J_4, J_5, J_6 are orthogonal, but they constitute a non-spherical wrist, because they don't intersect at the same point, as can be seen from the figure.

Given a certain pose of the End Effector, in general, not all positions of the joints are possible; in fact there could be configurations that require one or more joints to assume values that are outside the joint limits. So, this is an analysis of the possible admissible values of the joints (angles are in degree, except the third joint which is written in meters):

Joint 1:	-70° / +70°
Joint 2:	-50° / +50°
Joint 3:	0m / +0,235m
Joint 4:	-175° / +175°
Joint 5:	-175° / +175°
Joint 6:	-175° / +175°

II. Forward kinematic

The forward kinematic has as its target to define the end organ pose in function of the values assumed by the joint variables.

Below, the DH table:

i	a	α	d	ϑ
0 - 1	0	$+\frac{\pi}{2}$	0	ϑ_1 * $+\frac{\pi}{2}$
1 - 2	0	$-\frac{\pi}{2}$	0	ϑ_2 * $-\frac{\pi}{2}$
2 - 3	0	$+\frac{\pi}{2}$	d_3*	0
3 - 4	0	0	0.4162	ϑ_4 *
4 - 5	0	$-\frac{\pi}{2}$	0	ϑ_5 * $-\frac{\pi}{2}$
5 - 6	0.0091	$-\frac{\pi}{2}$	0	ϑ_6 * $-\frac{\pi}{2}$

Table 1: DH parameters

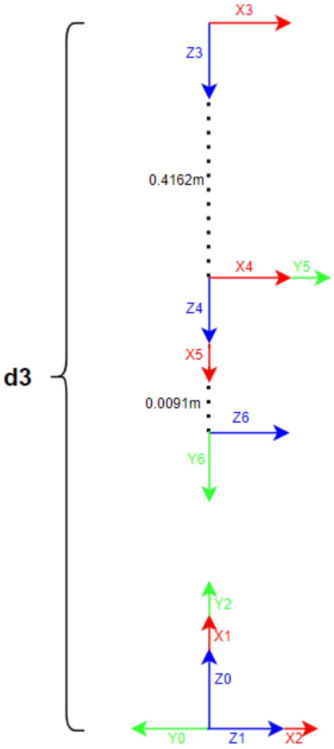

Figure 2: DH frames of PSM

Figure 2 shows the frames used to define the parameters of the DH table. The notation used for the DH table provides that the joint variable is Z, that is the rotations and translations occur along the Z axis. Counterclockwise rotations are positive, while clockwise rotations are negative. Moreover, in order to pass from one reference system to the other, translation occurs first, and then occour rotation. Therefore, denoting with letter d the translation along the Z axis, ϑ the rotation along the Z axis, letter a the translation along the X axis, α the rotation along the X axis, the order used is: $a \rightarrow \alpha \rightarrow d \rightarrow \vartheta$

It is important not to forget the order as it is easy to go wrong because usually a different order is used $d \rightarrow \vartheta \rightarrow a \rightarrow \alpha$

In the case of rotational joints, the joint variable is the ϑ, while in the case of prismatic joints, the joint variable is d. Be careful that for better readability, these letters have been indicated in the DH table followed by the asterisk symbol, just to indicate that they are free variables. Although there are free variables, I have preferred to indicate in any case also the rotations along the axes.

A very important thing to note is that the RCM point (remote center of motion), the point around which the manipulator moves, is placed exactly on the base frame (X0, Y0, Z0) in the figure 2.
Below is a list of the names attributed to the joints in the dvrk

- Joint 1R: outer yaw
- Joint 2R: outer pitch
- Joint 3P: outer insertion
- Joint 4R: outer wrist roll
- Joint 5R: outer wrist pitch
- Joint 6R: outer wrist yaw

A. Explanation of DH parameters

With reference to figure 2 and table 1, below is an exhaustive explanation of each parameter, collected for each step. Note that the reference systems have the number that identifies them one less than the number of the joint in which they are located. Therefore, the reference system number 6 corresponds to the end effector.

[0 - 1] The Z axis corresponds to the rotational joint 1, this means that the joint can rotate around the Z axis with an angle ϑ_1. To pass from reference system 0 to reference system 1 it is necessary to follow the order of actions defined previously. So you have to rotate along the X axis of $+\frac{\pi}{2}$ and then rotate along the Z axis of $+\frac{\pi}{2}$. The two reference systems, as can be seen in the figure, coincide as no translations have been made along the X or Z axis.

[1 - 2] The Z axis corresponds to the rotational joint 2, this means that the joint can rotate around the Z axis with an angle ϑ_2. To pass from reference system 1 to reference system 2 you have to rotate along the X axis of $-\frac{\pi}{2}$ and then rotate along the Z axis of $-\frac{\pi}{2}$.

[2 - 3] The prismatic joint J_3 can move along the Z axis by a free variable d_3. To pass from reference system 2 to reference system 3 you have to rotate along the X axis of $+\frac{\pi}{2}$ and then you have to translate by d_3 along the z axis.

[3 - 4] The rotational joint 4 can rotate around the Z axis by an angle ϑ_4. To move from frame 3 to frame 4 you have to move by a constant 0.4162m along the Z axis.

[4 - 5] The joint 5 can rotate along the Z axis with its free variable ϑ_5. To go from reference system 4 to 5 you

have to rotate along the X axis of - $\frac{\pi}{2}$ and then rotate along the Z axis of - $\frac{\pi}{2}$.

[5 - 6] The joint 6 can move with an angle ϑ_6 around Z. To move from frame 5 to frame 6 you have to move by a constant 0.0091m along the X axis, rotate along the X axis of - $\frac{\pi}{2}$ and then rotate along the Z axis of - $\frac{\pi}{2}$.

III. Inverse kinematic

The inverse kinematic problem concerns the determination of the joint variables, once the position of the end effector has been assigned. The closed-form determination of inverse kinematics requires geometric intuitions for the identification of those significant points on the structure with respect to which to express positions and / or orientations in function to a small number of joint variables, or requires algebraic intuitions for the identification of the significant equations containing the unknowns to be manipulated.

We have to remember a sufficient condition for a manipulator with 6 degrees of freedom which, if satisfied, allows us to guarantee the existence of a closed form solution of the manipulator; this condition is satisfied if:

- Or 3 consecutive rotational joints have axes that intersect at one point (eg. Spherical wrist)
- Or 3 consecutive rotational joints all have parallel axes

This condition is not satisfied in the PSM manipulator because in the last 3 rotational joints it happens that:

- The first two have coincident triads
- The last triad is shifted by an offset equal to 0.0091m from the other two

Having said that, since the sufficient condition is not satisfied, we can assert that either the solution in closed form does not exist or it is difficult to derive. The solution currently implemented in dvkr is iterative and it is based on Newton's algorithm.

Now I will introduce you step by step my solution in closed form for the calculation of the values of the six joints starting from the pose of the end-effector. First of all, to facilitate the calculation, the RCM point was used, characterized by the fact that it never changes in any robot configuration and, as previously mentioned, it coincides with the base triad. Furthermore, the coordinates of the EE are expressed in reference to the base triad.

The first thing to note is that the position and orientation of the final organ that is returned by the python functions (ros topic) has been altered by a matrix (tool-tip offset) which is used to orient the triad of the frame 6 in a triad that respects the convention of the end-effector (with the outgoing z) since as you can see in the figure 2 the reference system 6 does not respect the convention for the EE triad. By multiplying the matrix that defines the altered pose of the EE and the inverse matrix of the tool-tip offset, it is possible to obtain the matrix without alteration. This operation allows us to obtain the values of the normal Z of joint 6 which will be used later to calculate the values of the various joints.

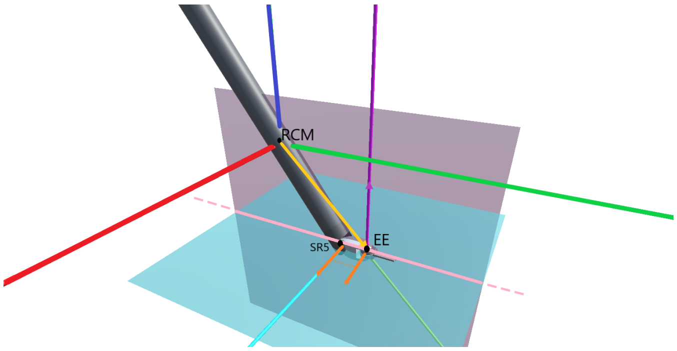

Figure 3: End-effector to SR5

Now that various premises have been introduced, let's proceed with the calculation of the various joints. First, it is necessary to calculate the x, y, z position of the reference system 4-5 before being able to calculate the value of the first 3 joints (take a look at the figure 3 as you read the procedure to better understand), to do this the idea is as follows:

1) Vectorial product between the yellow vector which is the distance vector between RCM and EE and the purple vector (normal z of the pose of the EE after applying the inverse matrix of the tool-tip offset as explained above). What I get is the orange vector indicating the normal outgoing from the gray plane.
2) The equation of the two planes as for each we have a point belonging to the plane and the normal outgoing:
   - Blue plane = normal vector in purple and the EE point
   - Gray plane = normal vector in orange and RCM point
3) The intersection between the two planes which as a result gives me the equation of the line in pink. At the implementation level I used a function that instead of the equation of the line returns me two points belonging to the line obtained through intersection. Thanks to these two points it is possible to construct a director vector that allows me to move by 0.0091m or -0.0091m along the line to be able to find the coordinates of the frame 5 (since it is not possible to distinguish which is the correct direction in which to move to arrive at frame 5 starting from EE, the idea is to consider both possibilities and then obtain two different solutions of the inverse kinematics and appropriately checking if both respect the joint limits). Seen in a practical way, given two points A and B:
   - Build the vector $\overrightarrow{BA}$ = (XB-XA,YB-YA,ZB-ZA)
   - Normalize it by dividing $\overrightarrow{BA}$ by its length
   - You can now move from A to B by a certain offset in this way: A + offset * $\overrightarrow{BA}$-normlized
   - In my case offset = ± 0.0091m

Now that we have the point corresponding to the reference system 5 we can calculate the value of the first three joints as follows (see images number 4, 5, 6). Note that with X, Y, Z letters is indicated the coordinate of the reference system 5 and not the end effector point.

$$\theta_1 = \arctan(\frac{X}{Z})$$

$$\theta_2 = \arcsin(\frac{Y}{d_3})$$

$$d_3 = \sqrt{X^2 + Y^2 + Z^2}$$

Be aware that d_3 is not really the value obtained from this formula, but to get it is necessary adding a constant of 0.0156m as d_3 also extends to the part where Z-RCM is positive and therefore you have to consider it to get the calculations right.

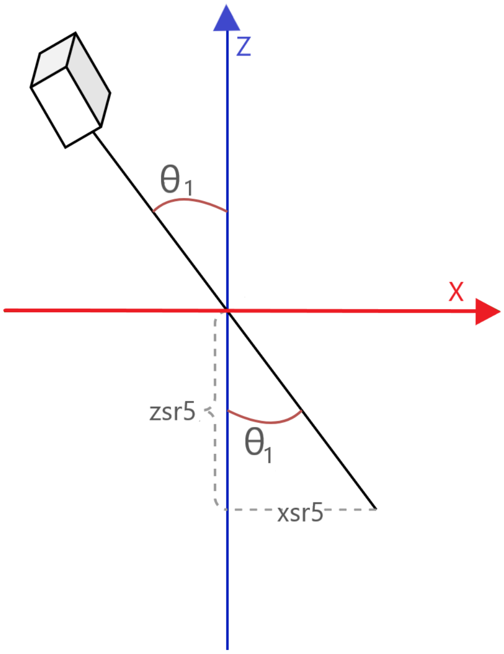

Figure 4: Diagram of the angle ϑ_1

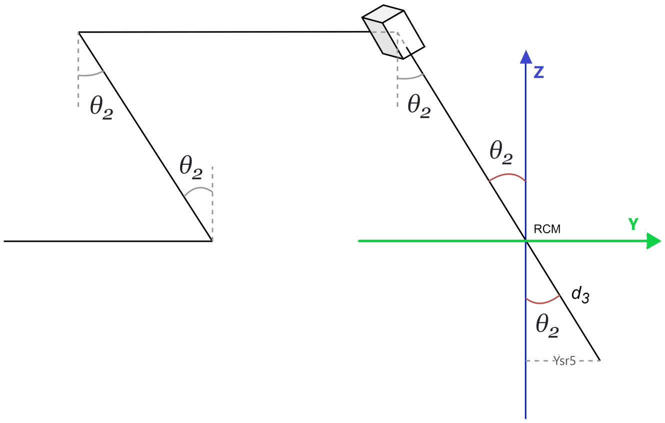

Figure 5: Diagram of the angle ϑ_2

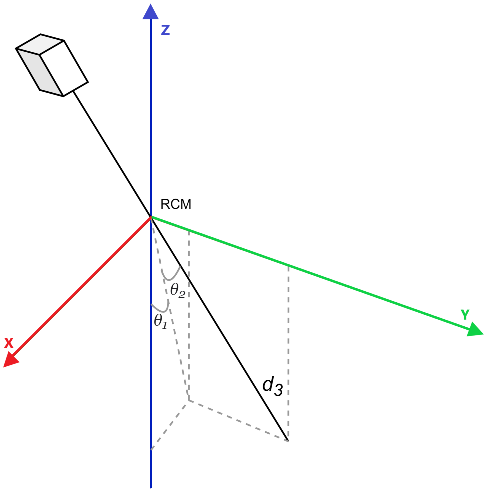

Figure 6: Diagram of the length d_3 (prismatic joint)

The last three joint values are a little more complex to calculate, in order:

- Joint 4:
  - *ny* = normal vector corresponding to the y axis
  - *base_to_pitch* = vector starting from base frame and directed to frame 5
  - Normalize the vector *base_to_pitch*
  - *normal_pitch* = vectorial product between the yellow vector which is the distance vector between RCM and EE and the purple vector mentioned above (Figure 3)
  - Normalize the vector *normal_pitch*
  - Vectorial product between *ny* and *base_to_pitch*
  - Calculate the angle between the npitch vector and the vector obtained in the previous point and this angle corresponds to the value of ϑ_4
- Joint 5 (Figure 7):
  - Vector starting from frame 5 directed to EE (purple)
  - Vector starting from frame 5 directed to base frame (yellow)
  - Calculate the angle between these two vectors (orange) and the complementary of the angle just obtained corresponds to the value of ϑ_5
- Joint 6:
  In order to calculate the value of this joint it is necessary to indicate in the kinematic field in the file ”console-PSM1_KIN_SIMULATED.json” the file ”psm-large-needle-driver-tip.json” (by default a different one is used) . All this because in this file a tooltip-offset matrix is used which, in addition to orienting the EE triad correctly, translates it by an offset equal to 0.01m. So thanks to this it is possible to define a vector that gives us the direction from EE (frame 6) to TIP (EE translated by 0.01) and therefore it is possible to calculate the value of the sixth joint. More in detail:
  - Vector starting from EE directed to frame 5
  - Vector starting from EE directed to TIP

- And similarly for the calculation of ϑ_5 now you can calculate the angle between these two vectors and the complementary of the angle just obtained corresponds to the value of ϑ_6

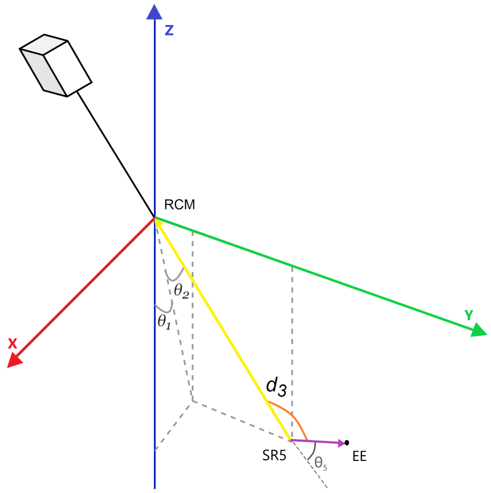

Figure 7: Diagram of the angle ϑ_5

With regard to the correctness of the sign of the calculated angles, with respect to ϑ_1 and ϑ_2 the sign is considered opposite to the calculated one. For the angles ϑ_4, ϑ_5, ϑ_6 on the other hand, the technique of the scalar product was used which, based on the result of this product (negative or positive), it is possible to determine the sign of the angle.

IV. Simulation

In the da VinciResearch Kit (dVRK), created by John Hopkins University, there is the ROS implementation of the PSM [1]. This implementation publishes robot status on various ROS topics. Using RViz to view the robot and the official API that are made available by the dVRK. I wrote a script, available here [4], that uses the direct kinematics already implemented with random joint values (respecting the joint limits) and starting from the pose of the EE obtained it calculates the value of the six joints as shown above and calculates the error. The script generates 1 or 2 possible solutions for the inverse kinematics for each pose of the EE.

In my github repository, in addition to the script, you can find two pdf files of notes where I reported all the reasoning, considerations and the way I approached the realization of the project, in more detail the following things were reported:

- There is all the path made for the installation of the dvrk and ros environment up to the start of the psm manipulator and the related problems encountered
- I have described in detail all the dvrk files that I have analyzed and all the considerations made
- The binding between python functions and related ros topics and what the various functions consist of
- Various hand-made schematics on direct kinematics and some ideas for inverse kinematics are shown
- There are some useful theoretical hints
- There are various links to the sites I used for the various calculations made (plane equation, vector product, etc...)

There are three things to specify:

- I have considered only EE positions that have a z coordinate value less than 0
- In cases where the EE tends to be very close to the origin there may be an inconsistency with the value of the calculated joints
- It may happen that the ϑ_5 angle turns out to have the opposite sign to what it should be

As an experimental result I ran the script by randomly generating 450 sequences of joint values and starting from the pose of the EE obtained I calculated the inverse kinematics in closed form. In 34 cases out of 450 the ϑ_5 angle was found to be of the opposite sign and below in figure 8 there is the set of EE points considered.

If you want to see a simulation of this script you can take a look at the video that is in [5] where the algorithm is tested on ten random positions of the EE and the relative values of the joints are obtained with the inverse kinematics in closed form.

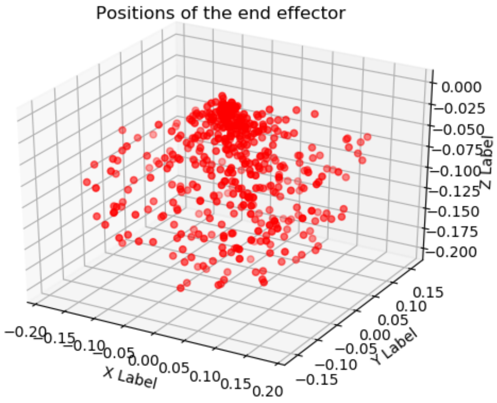

Figure 8: Workspace explored.

References

[1] J. H. University. sawintuitiveresearchkit. [Online]. Available: https://github.com/jhu-dvrk/sawIntuitiveResearchKit/wiki

[2] M. Selvaggio. Psm kinematic description. [Online]. Available: https://www.researchgate.net/figure/PSM-kinematic-description_fig3_328241499

[3] D. L. Pieper, "The kinematics of manipulators under computer control," Ph.D. dissertation, Stanford University, 1968.

[4] F. Mantovan. Ik solution in closed form about the psm arm of the dvrk. [Online]. Available: https://github.com/Filippo39/IKsolution_dvrk

[5] ——. Closed-form inverse kinematics of the psm manipulator. [Online]. Available: https://www.youtube.com/watch?v=D5um$_3PUvrg$