

Colin Rubow

Machine Learning Project Progress Report

To review, the goal of the project is to create a machine learned model of the forward kinematics of the UWRL Utah neck brace. This brace has three motors on the base which attach to the shoulders, and are joined through three kinematic chains of two revolute joints followed by a spherical joint. The kinematics are too difficult to derive an analytical solution and so machine learning may be a good substitution.

Many others have used machine learning to model forward kinematics. In [2], a neural network combined with a genetic algorithm was used to model the forward kinematics of a Delta parallel robot system. [4] uses support vector machines to model the forward kinematics of a Stewart platform. [1] uses a neural network to model the forward kinematics of the Stewart platform. An interesting aspect of [1] is that they used the inverse kinematics to create a training dataset that was used to train the neural network. [3] used kernel extreme learning machine to learn the forward kinematics of a '6-DOF parallel robot'. All of these sources use machine learning to learn the forward kinematics of parallel robots and are thus directly applicable to our parallel robot.

To increase the accuracy and chances of a machine learned model of accurately describing the forward kinematics, the forward kinematics have been derived up to a certain point where the problem has been reduced somewhat to the following problem: given three circles in three dimensional space, to find a point on each of the circles which form a triangle congruent to given triangle. The reason for this new problem is that the endpoints of each of the three chains form a triangle. This triangle is fixed through the design process and the robot effectively moves this triangle around in three dimensional space. A forward kinematics solution must therefore preserve this triangle. Therefore, given the three motor angles, the first link of each chain is fixed, and the remaining link has an unknown angle, which means that all candidate points form a circle in three dimensional space. There are three such circles and there must be a point on each of these circles whose triangle is congruent to the triangle fixed by design. It is assumed that only one solution will exist with a given set of motor angles, but it is possible that there are more than one solution, or even none.

The system of equations that are to be solved are thus:

$$\begin{aligned} r_{O_3A_1}^{O_3} &= r_{O_3O_1}^{O_3} + r_{O_1M_1}^{O_3} + l_{A_1} \left(\cos \tau_1 R_{M_{1x}}^{O_3} + \sin \tau_1 R_{M_{1y}}^{O_3} \right) \\ r_{O_3A_2}^{O_3} &= r_{O_3O_2}^{O_3} + r_{O_2M_2}^{O_3} + l_{A_2} \left(\cos \tau_2 R_{M_{2x}}^{O_3} + \sin \tau_2 R_{M_{2y}}^{O_3} \right) \\ r_{O_3A_3}^{O_3} &= r_{O_3M_3}^{O_3} + l_{A_3} \left(\cos \tau_3 R_{M_{3x}}^{O_3} + \sin \tau_3 R_{M_{3y}}^{O_3} \right) \end{aligned}$$

Where τ_1, τ_2, τ_3 are the unknowns to solve for and there are a couple of other constraints due to symmetry and such things. The meaning of each variable is not important in this paper. Suffice to say that each expression is a three dimensional vector.

A physical realization of the robot has been constructed. The three motors at the base joints are DYNAMIXEL XM430-W350-R servo motors which are equipped with angle measurement sensors. This allows the measurement of the angles which are the inputs to the forward kinematics. To create a machine learned model, the output of the forward kinematics needs to be known as well. There were three options considered to accomplish this.

The first was with the use of a Vicon motion tracking system. Three motion tracking tags could be placed on the robot end effector. The first tag would track the position of the end effector, the other two tags could be used to track the orientation of the end effector by taking cross products to create a frame, and subtracting a 'home-position' frame from it to get the orientation of the end effector at each time sample. This strategy ran into a difficulty in that there does not seem to be a viable way to sync the motor angle data and motion capture data, meaning, there is no guarantee that with this strategy the motor angles actually correspond to the captured motion data. This is because the motor angles are read over a specific digital protocol and the VICON system needs either an analog signal or some work to take data from a Raspberry Pi or something, which it is not designed to do.

The second strategy, one of the ones implemented, is to use inertial measurement units (IMUs) to measure the orientation of the end effector. One IMU is placed on the fixed base of the robot and the other is placed on the end effector. By subtracting the base IMU data from the end effector IMU data we get the orientation of the end effector relative to some home position.

The third strategy, used to generate a training dataset, is inspired from [1] by using the inverse kinematics. The inverse kinematics of our robot is solved analytically so this method is viable. It is noted that it shouldn't be used to generate a testing dataset because such acquisition doesn't match the situation we are creating the model for.

The angles of the motors were measured by attaching a motor driver to the motors and communicating to this driver via USB. The IMU data was measured with a MyRio acquisition computer that was attached via USB to the computer. A Labview program was modified to pull the desired data while I perturbed the robot in its various configurations to get data for the whole workspace. The specific data pull was the timestamp, the three motor angles, the six XYZ Euler representation angles of the two IMU sensors, and whether an orientation was attainable or not.

The data was post processed by only keeping data when the end effector orientation was attainable, (despite being able to move the robot into an orientation the workspace of the robot is smaller than the perturbable space; for a good reason). Furthermore, some of the IMU data returned intermittent zeros. The data would process fine, but every once in a while a small train of zeros would return for one of the angle measures and then revert back to actual measurement values. To deal with this data, it is assumed that a linear interpolation is adequate for describing the motion in this unknown region. This assumption is more valid for shorter periods of zeros.

With the goal of having a machine learned model of the forward kinematics, the next step of the process is to decide on the type of model to use. Preferably, the model type will be able to incorporate the derived equations above into it. This can be done by creating a new dataset that has the positions of the center of these circles as inputs and the end effector orientation as outputs. Another way is by training to acquire the desired parameters of the derived equations. Other ways are sure to be available and need to be researched.

After the model type is decided on, the architecture of the model will need to be designed. For example, with learning trees, the depth of the tree, how numerics are handled, and other decisions that need to be made. For a neural network the number and size of layers needs to be decided on. Of course, different model types and architectures can be experimented on and the best performing model can be

selected. In such an experiment, more training sets should be made to ensure the models are trained on datasets that more closely resemble the population set.

A few test datasets can be made to test the models in the same way. The final test would be to implement a sort of control process on the robot that requires forward kinematics. If the robot is able to be controlled, the model is good.

The reason for having multiple datasets is to deal with the random factors of the data collection process.

It is also possible that a way to use the Vicon motion capture can be developed. Using the Vicon system instead of the IMU's would give much more accurate and reliable data. I am not optimistic about this plan though. I will email one of the Vicon representatives to see what they think of the situation.

Sources

- [1] D. K. S. Chauhan and P. R. Vundavilli, "Forward Kinematics of the Stewart Parallel Manipulator Using Machine Learning," *Int. J. Comput. Methods*, vol. 19, no. 08, p. 2142009, Oct. 2022, doi: 10.1142/S0219876221420093.
- [2] C. Liu, G. Cao, and Y. Qu, "Safety analysis via forward kinematics of delta parallel robot using machine learning," *Safety Science*, vol. 117, pp. 243–249, Aug. 2019, doi: 10.1016/j.ssci.2019.04.013.
- [3] J. Ma, X. Duan, and D. Zhang, "Kernel extreme learning machine-based general solution to forward kinematics of parallel robots," *CAAI Transactions on Intelligence Technology*, vol. 8, no. 3, pp. 1002–1013, 2023, doi: 10.1049/cit2.12156.
- [4] A. Morell, M. Tarokh, and L. Acosta, "Solving the forward kinematics problem in parallel robots using Support Vector Regression," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 7, pp. 1698–1706, Aug. 2013, doi: 10.1016/j.engappai.2013.03.011.