Implementation and Testing of Inverse Kinematics on Robotic Arm

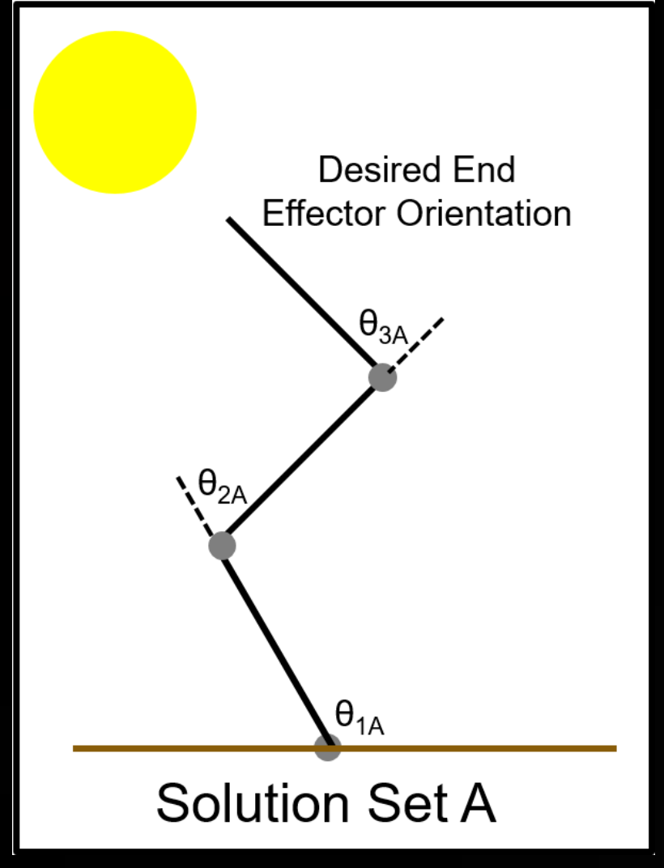
Carter Franz – Saint Louis University
Mentors: Levin Guillermo (EV41) and John Rakoczy (EV41)

Abstract:

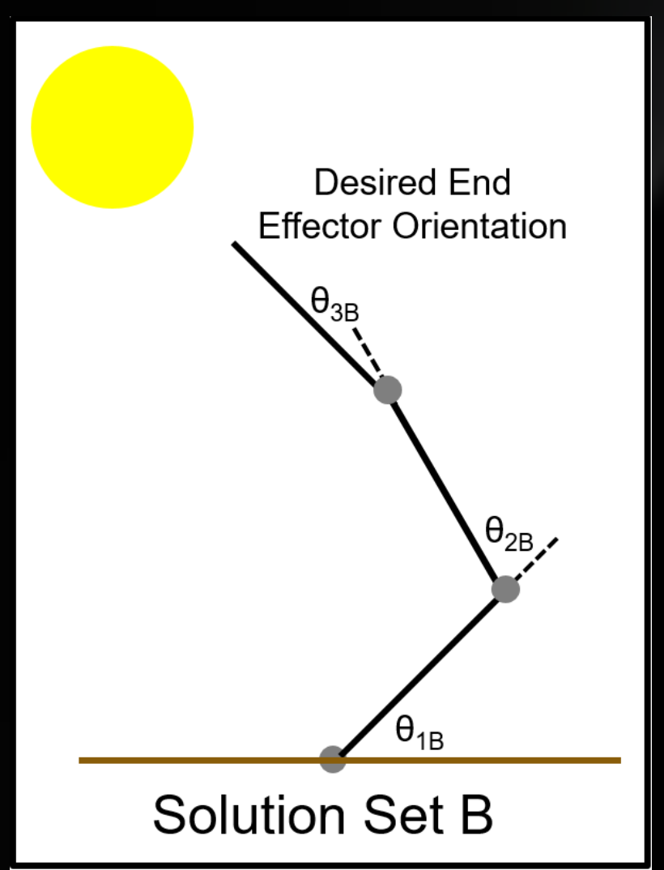
COSIE (Coronal Spectrographic Imager in the Extreme Ultraviolet) is a proposed solar tracking ISS imaging payload that will help bridge the theoretical gap between the physics of the low corona and the heliosphere. This scientific instrument requires high pointing accuracy, on the order of arcseconds. The instrument is mounted onto a three revolute joint robotic arm in order to track the roll, pitch and yaw motion of the Sun. The goal of this project is to construct a prototype model of the robotic arm and implement the proposed analytical inverse kinematics algorithm. In robotics, the inverse kinematics problem is solving for the set of joint angles that achieve the desired end effector location and/or orientation. In this case, orientation is the focus. Depending on the configuration, multiple sets of joint angle solutions may exist. Due to the complexity of robotics, typically iterative methods are used to solve for the joint angle solution sets. However, in this case, an analytical solution exists. A small robotic arm representative of the full size hardware was constructed. The inverse kinematics algorithm, originally in MATLAB/Simulink, was converted into C in order to interface with the motors. This C software was implemented on a Windows PC and microcontroller, and serial communication between the two was established, allowing the motors to be directly controlled by the inverse kinematics algorithm. Testing the inverse kinematics on a physical system will allow the validity and accuracy of the analytic solution to be verified.

Background:

- Inverse kinematics in robotics is the solving for the set of joint angles to achieve a specific end effector orientation and/or location. For this application, only orientation was considered.
- Inverse kinematics is traditionally solved iteratively, but for this configuration an analytic solution exists. This prototype was constructed to test the validity of the analytic solution.
- When only considering orientation, two valid solutions exist.

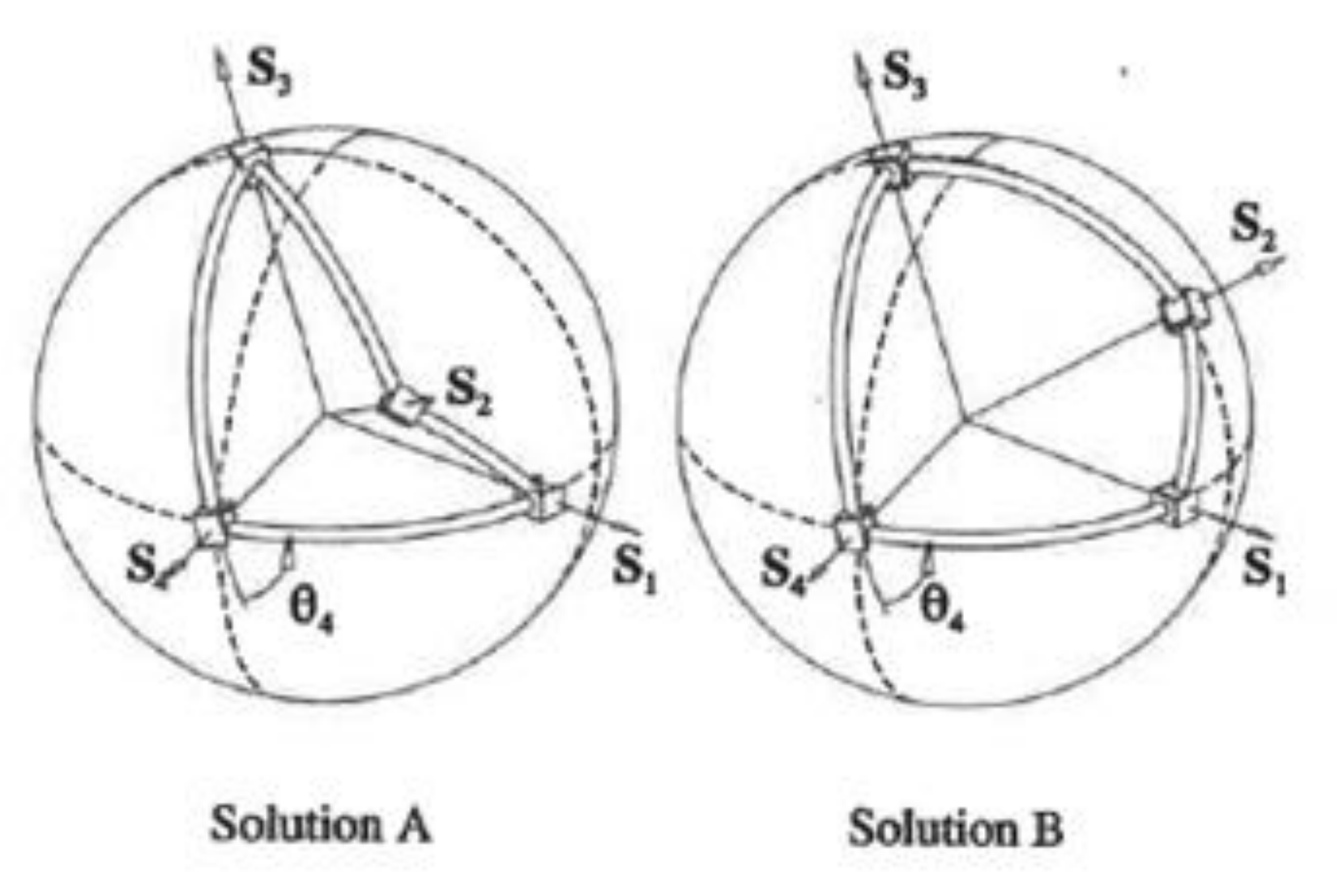


Solution Set A



Solution Set B

The robot coordinate systems are labeled in the manner described in the reference. Given a desired end effector location and orientation, a fictional fourth link can be constructed from the end effector to the robot base. This "closes the loop" and the links form a polygon, which can be used to construct an equivalent spherical mechanism. The joint angle solution sets can be determined using the relationships derived in the reference for the corresponding spherical mechanism.



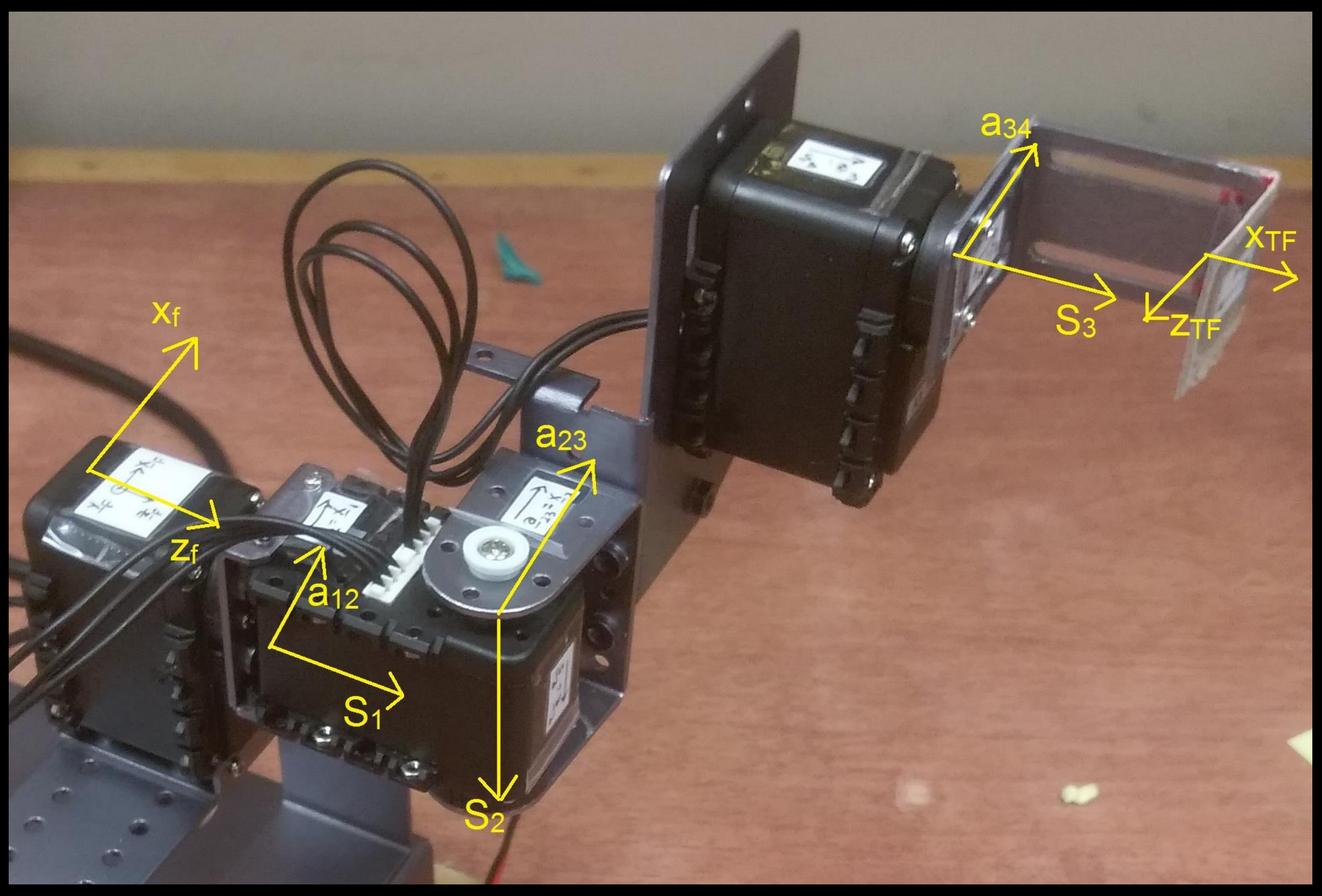
Solution A

Solution B

Ref: Carl D. Crane, III , Joseph Duffy, Kinematic Analysis of Robot Manipulators, Cambridge University Press, New York, NY, 1998

Hardware:

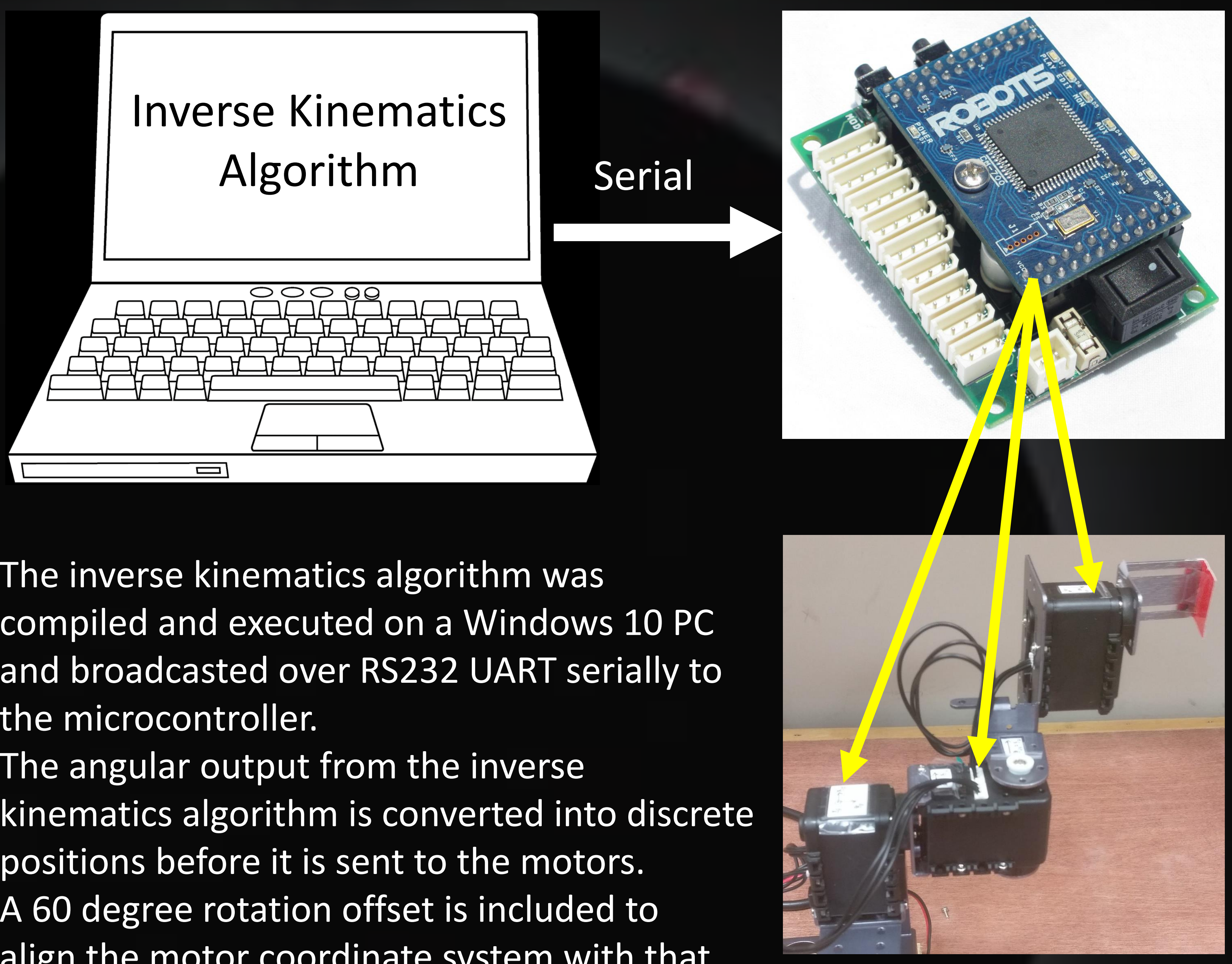
- A prototype representative of the original system was constructed as a three revolute (RRR) robotic arm. The arm was built using three Dynamixel AX-18f servo motors as actuators, with Dynamixel brackets to join the motors together. The motors give the arm three degrees of freedom (roll, pitch, yaw), with the third rotational axis offset a short distance. This offset makes the assumption of a ball joint no longer valid.
- The robotic arm is controlled by the Robotis CM-700 board, which houses an ATMega 2561 microcontroller. The microcontroller communicates with the motors using a daisy-chained serial connection.
- The various coordinate systems for the robot were established and labeled.



Acknowledgements:

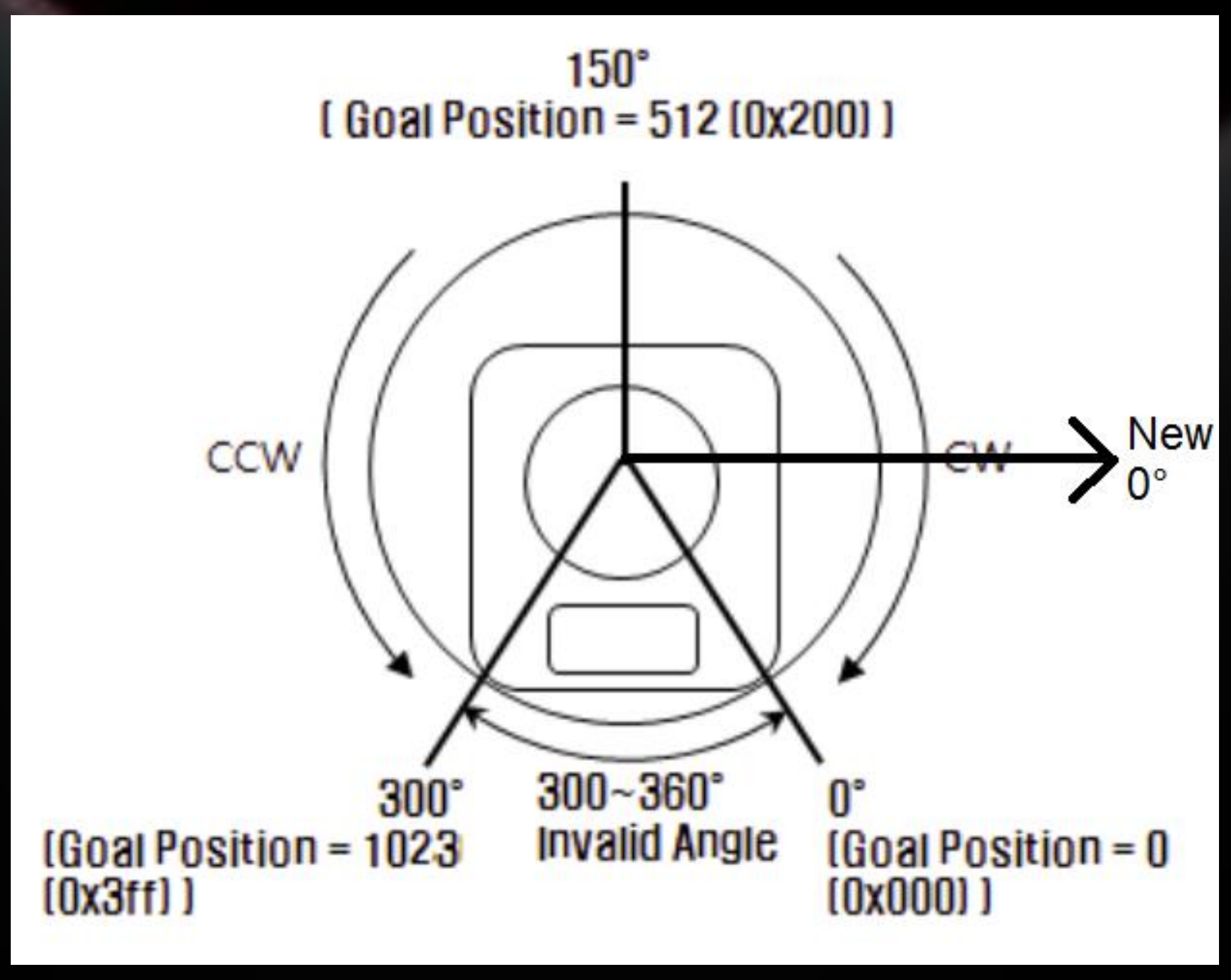
Big thanks to Levin Guillermo (EV41) and John Rakoczy (EV41) for your mentorship and teaching over the course of this project. I would also like to thank NASA and the Missouri Space Grant for giving me this opportunity.

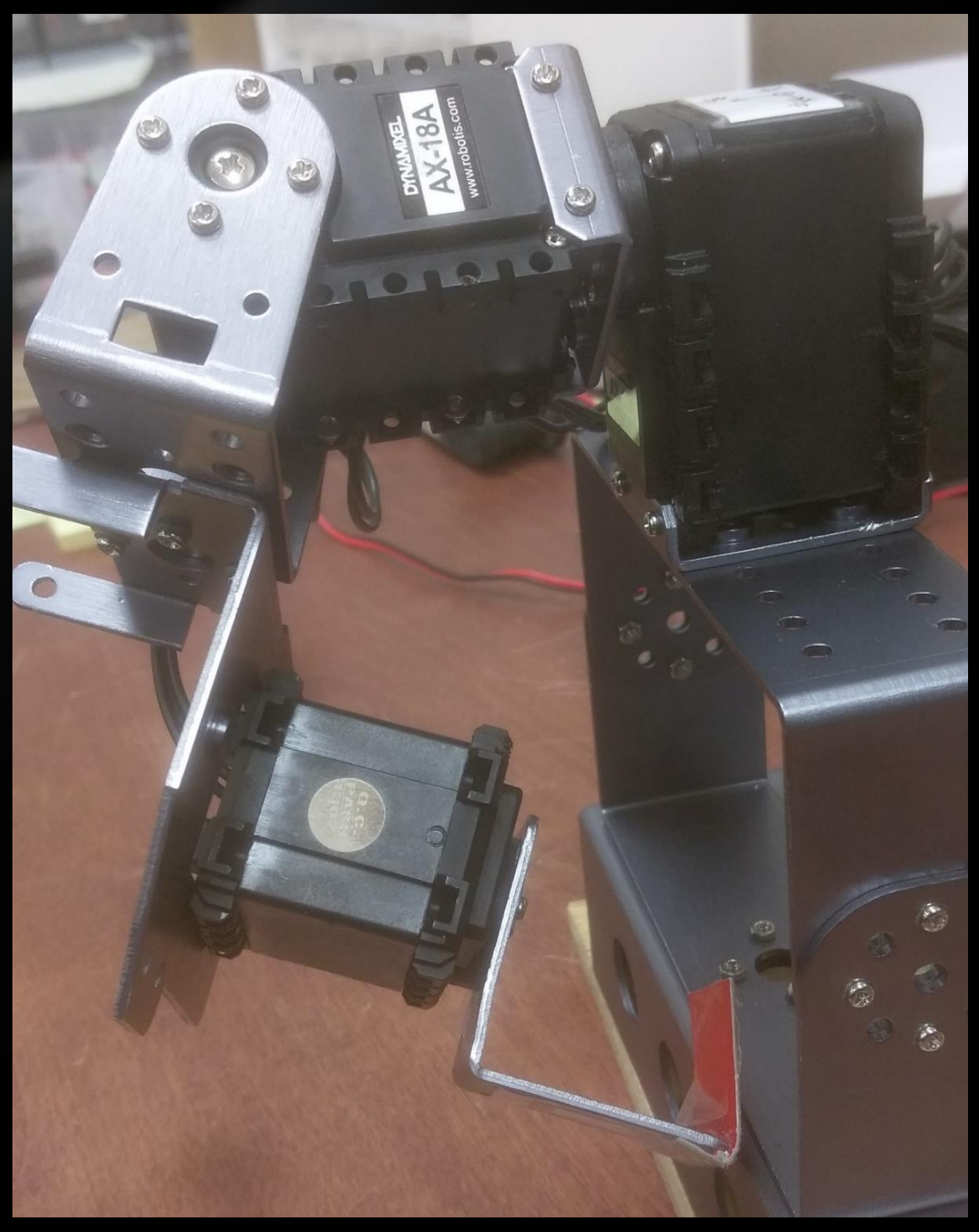
Software:



- The inverse kinematics algorithm was compiled and executed on a Windows 10 PC and broadcasted over RS232 UART serially to the microcontroller.
- The angular output from the inverse kinematics algorithm is converted into discrete positions before it is sent to the motors.
- A 60 degree rotation offset is included to align the motor coordinate system with that of the robotic joint, with additional corrections for edge cases.
- The software defaults to angle set A, but uses angle set B if A is not realizable. If neither is reachable, it returns to the default position.

Limitations:





Hardware:

- There is a 60° region that the motors cannot achieve. Despite the logic in the software, there are desired orientations that are not possible because at least one motor in each solution is required to be in this invalid region.
- The motor brackets, wiring, and overall mount also physically limit the robot workspace due to collisions and limited ranges of motion.

Software:

- Path planning and collision avoidance was not considered in this implementation. The robot will sometimes impact other motors while traveling to the desired position. This prevents the robotic arm from achieving the desired orientation.
- The microcontroller did not have the processing power available to perform the inverse kinematics itself, requiring it to be tethered to a more powerful system. This increases response time and limits mobility of the arm assembly.

Conclusions:

A prototype of the COSIE robotic arm was built, implementing the proposed inverse kinematics algorithm. The algorithm was tested and verified.

Future Work:

The next step is integration of a sun sensor. A sun sensor provides continuous feedback for tracking motion of the Sun. This will mimic COSIE flight operational conditions.