Stacking Blocks with a UR-5

ENPM 662 – Robot Modeling

David A Smart
12/18/17

Introduction:

The UR-5 by Universal Robots, shown in Figure 1, is a six degree of freedom fully-revolute manipulator arm.

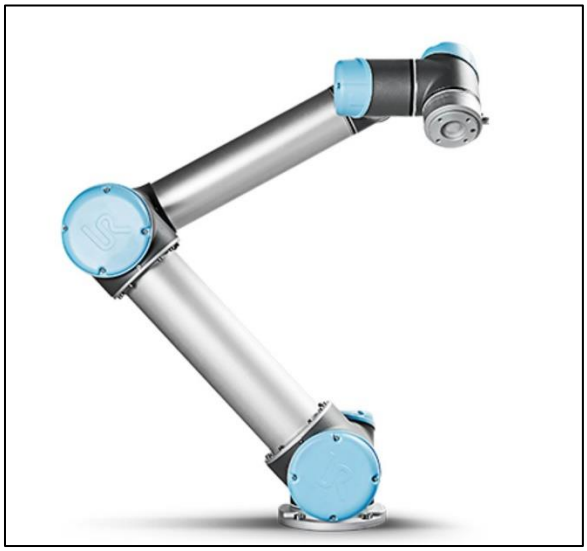
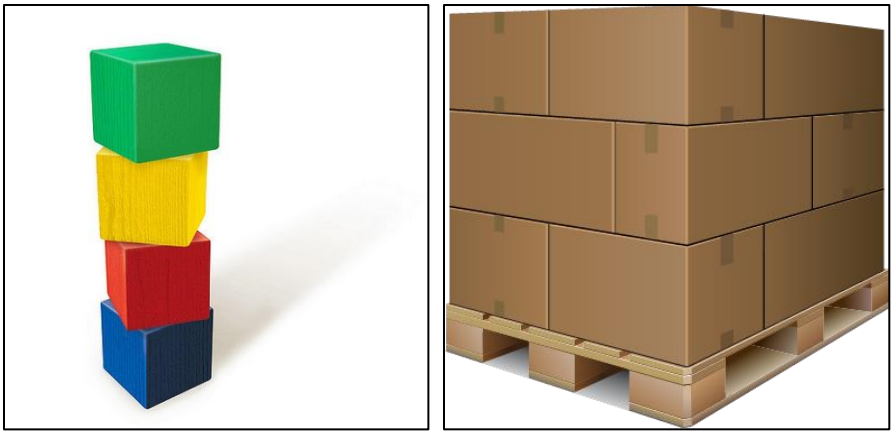


Figure 1: The UR-5 Arm [6].

Stacking single small cubes into slender towers, like that shown in Figure 2a, is not a typical robot application, because there is no current use for it, however, it is a surprisingly difficult and therefore interesting task. Arranging larger rectangular prisms in a brick like overlapping pattern, known as palletizing, as shown in Figure 2b, is commonplace in packaging. The UR-10, simply a larger version of the UR-5, has already been applied to stacking towers of irregularly shaped rocks. This task involved center of mass calculations, contact point calculations, and path planning for each rock in the stack [1]. Although the general motion is similar for all three tasks, palletized stacks are inherently more stable and therefore small positioning errors do not affect the performance, whereas a slight miscalculation or misplacement could be disastrous when stacking rocks. The effect of errors in stacking cubes falls somewhere in the middle.



(a) (b)

Figure 2: (a) Stack of Four Wooden Cubes [7], (b) Palletized Boxes [4].

For this project, it was proposed to simulate the UR-5 moving a stack of blocks from one location to another location within the UR-5's workspace (850mm radius). This was to be accomplished by moving one block at a time, such that the order of the blocks will be reversed. A simple two finger gripper will be used to grasp the square blocks.

The first goal was to accomplish the stacking task using motion control under idealized conditions. For this case, the arm's links and the blocks are treated as weightless, all the links are perfectly ridged, the joint encoders are perfect, and all the positions of all the blocks are known to the robot. The second goal is to accomplish the stacking task using motion control under non-ideal or realistic conditions. In this case, the arm links and blocks will have weights causing there to be gear backlash, the links will have a flexibility, the joint encoders will have random error, and a simulated force sensor on the gripper finger tips will be used to sense when the gripper has reached the next block in the stack. Because the positioning error will be captured to some degree with the encoders, although not error free themselves, it would be possible to implement a feed-back control loop, however, this will be reserved until the end in the case of spare time.

The workspace is known to be a sphere of radius 850mm with a cut-out cylinder of radius 148mm oriented along the z_0-axis to avoid singularities [5]. For both the first and the second goal, all the UR-5 joints can rotate a full ± 360°, and can rotate at a speed of ± 180°/sec [5].

For the second goal, the UR-5 is known to weigh 15-kg total [5] and is assumed to be uniform in density, the flexibility of the links will be assumed to be that of Aluminum. The gripper will be assumed to be made of Aluminum as well and the weight will be calculated from its volume. The cubes will be made of wood and weigh less than what is remaining of the 5-kg payload capacity of the UR-5 [5]. Furthermore, for the second goal, the maximum gear backlash is ± 0.008° [3] and the max encoder error is ± 0.001525° [2], based upon the hi-quality components used in the UR-5. The encoder error will be assumed to take a gaussian distribution. The backlash should always work to rotate the links toward vertical because of gravity, however, this would be difficult to implement in the model, therefore a constant negative error will be used.

Methods:

Table 1 shows the proposed timeline. This timeline was not followed whatsoever because other assignments were considered of greater importance. This is turn was because the other assignments were due at an earlier date than this report, and this project was considered an easier task than the other term projects. The project was not started until December 13, a full week after the late date on the schedule. Consequentially, there was an exceptional amount of stress on the author who was already sick from a lack of rest due to other projects. The actual timeline can be seen in Table 2. The explanation of each activity listed in Table 2 folows.

Table 1: Proposed Timeline

Milestone	**Date**
Idealized Models Finished	November 8th
First Version of MATLAB Control Script Written	November 15th
Realistic Models Created	November 22nd
Second Version of MATLAB Control Script Written	November 29th
Final Report Completed	December 6th

Table 2: Actual Timeline

Activities	**Date**
Models Imported, Failed Error Resolution	December 13th
Model Remade in Simscape, Single Block Placed in World	December 14th
"Tree-Structure Robots" Investigated	December 15th
MATLAB Control Script Written, Single Block to Stack of Blocks	December 16th
Final Report Written	December 17th
Contact Forces Attempted	December 18th

The author has used CAD modeling software for modeling extensively, therefore, Autodesk, was preferred over Simscape Multibody for creating the robot model because of familiarity. Prior to the project-proposal a simple model of the UR-5 was created in Autodesk and was imported successfully into Simscape such that Signal Builder Simulink controls were used to reposition the arm model. After the project proposal was submitted, a STEP-model for the UR-5 from the Universal Robots website was found and used to create a better model. This model was imported into Simscape, but had numerous simulation errors which could not be resolved. Despite

the reduction in accuracy, it was decided to return to the original model which had been proven to work. Oddly, this too had simulation errors, despite no changes being made. These errors were resolved, however, when attempting to add a gripper, different errors occurred that could not be resolved. At this point the author gave up importing models, because the errors were prohibitive. A Simscape Multibody model was painstakingly created, and all attempts at accurate modeling of the shape were abandoned. All the models spoken of above can be seen in Figure 3, below.

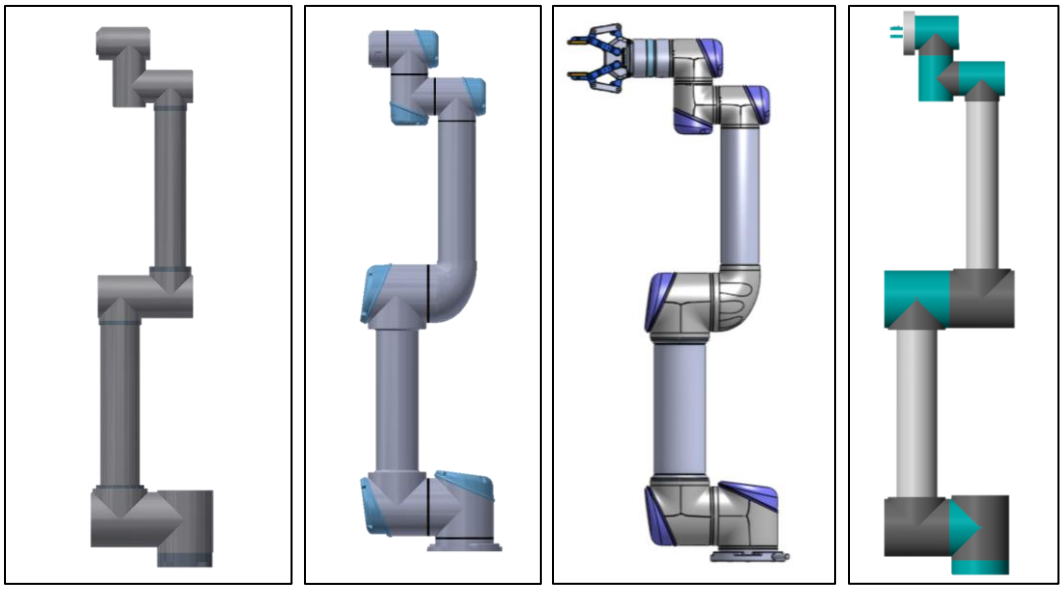


Figure 3: Original Model of UR-5, STEP-model of UR-5, Second Model of UR-5 with Gripper, Simscape Multibody Model of UR-5 with Gripper (left to right)

To add blocks which are initially located within the Simscape Multibody world, one must attach the blocks to the world frame using a rigid transform. To allow the blocks to be picked up by the robot, a 6-DOF joint without actuation must be added. To avoid the blocks falling endlessly under the influence of gravity the gravity vector must be reset to zero.

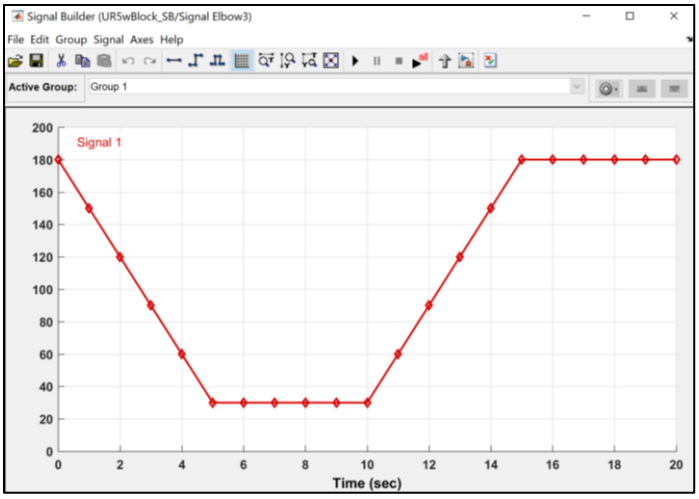


Figure 4: Signal Builder Simulink GUI

Originally the Signal Builder Simulink block, as shown in Figure 4, was used to control the joint angles of the Multibody model, but this was limited in that each signal must be manually setup. It was considered preferable to write a MATLAB function to create the signals for the joint angles depending on the block positions. Outputting variables to the MATLAB workspace proved futile because the variables are not output to the workspace until the simulation was completed. Outputting variables to an internal MATLAB function proved futile also, because the variables were output as zeros until the simulation was completed. Therefore, an external script was given the same information used for setting the block positions and used that to create signals for the joints. Unfortunately this did require the script to be changed if the block positions were changed, however, once set up it was significantly faster than changing each Signal Builder one at a time and avoided guess-and-check work. Just as the simulation could not output any variables, the external script could not be input the control signals in real time. It was discovered that the simulation could read in premade MAT-files, so the script was used to store the signals in MAT-files and these were read into the simulation.

There is an alternative robot modeling framework in MATLAB using “tree structures” to represent the robot which can be visualized using regular plots, as shown in Figure 5. These can “tree structures” can be made from Universal Robot Description Format (URDF) files. The UR-5’s URDF file is available online, however, the model created in Simscape did not match perfectly. These “tree structure” robots can also be constructed manually using the Denavit–Hartenberg (DH) parameters to specify the transformations from one joint frame to the next. Figure 6 and Table 3 show the DH parameters for the UR-5. MATLAB has built in functions for these “tree-structure robots” that can quickly calculate the needed joint angles for a given end effector position. However, placing constraints on the inverse kinematics solver to avoid useless configurations was not intuitive and was never fully understood by the author.

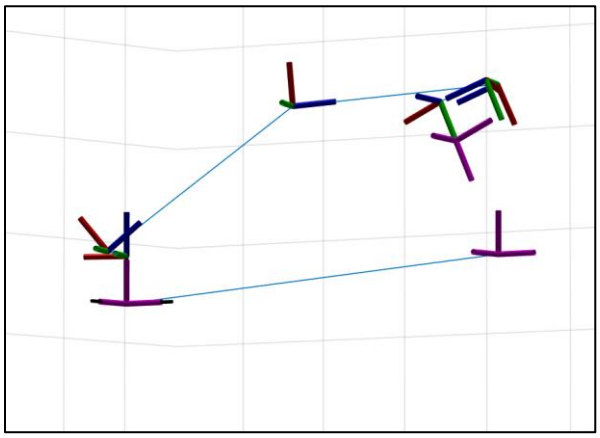


Figure 5: UR-5 URDF “Tree-Structure” Robot

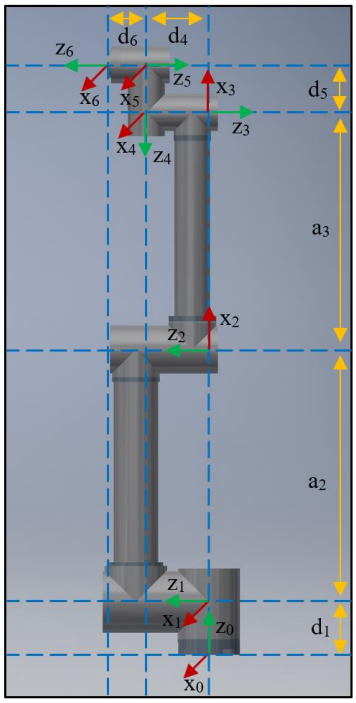


Figure 6: Model of UR-5 with DH Axes Overlaid

Table 3: UR-5 Denavit–Hartenberg Parameter Table

Link-i	Transition	θ_i	d_i	a_i	α_i
1	0→1	θ_1	d_1	0	90°
2	1→2	θ_2+90°	0	a_2	0
3	2→3	θ_3	0	a_3	180°
4	3→4	θ_4+90°	-d_4	0	-90°
5	4→5	θ_5	-d_5	0	90°
6	5→6	θ_6	-d_6	0	180°

Because the “Tree Structure” Robot approach did not pan out, the inverse kinematics were derived, but first the problem was simplified. The UR-5 does not have a classical “spherical wrist”, where the final three revolute joints rotate about a central point, therefore kinematic decoupling is

not possible. All six degrees of freedom are required for positioning and orientation of the end-effector. However, this was simplified by placing the Blocks such that the UR-5's base angle (θ_1) was simply the same angle as the first rigid transformation from the world frame to the block frame. This was accomplished by adding a second transformation from the world frame to the block frame to align the blocks with the gripper. The angles for Wrist 2 and 3 (θ_5 and θ_6) were set such that the gripper would always be grabbing the "left and right" (as opposed to near and far) sides of the block. This can be more clearly understood pictorially, as shown in Figure 7. The equations for the joint angles are given by equations 1-6. Figure 8a and 8b also helps to explain the rest of the derivation for the inverse kinematics.

$$\theta_1 = \theta_{Block} \tag{1}$$

$$\theta_5 = 90° \tag{2}$$

$$\theta_6 = 0° \tag{3}$$

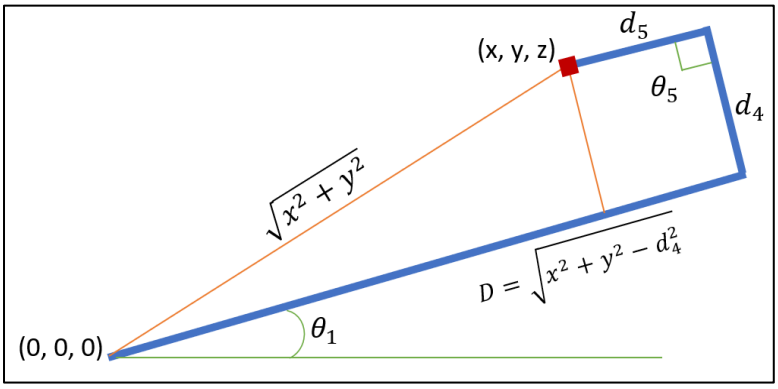


Figure 7: Deriving the Inverse Kinematics - Top View

$$\theta_3 = 360 - cos^{-1}\left(\frac{a_2^2 + a_3^2 - E^2}{2a_2a_3}\right) \tag{4}$$

where

$$E = \sqrt{D^2 + (d_1 - h)^2} \ , \ D = \sqrt{x^2 + y^2 - d_4^2} + d_5 \ , \ h = d_6 + z$$

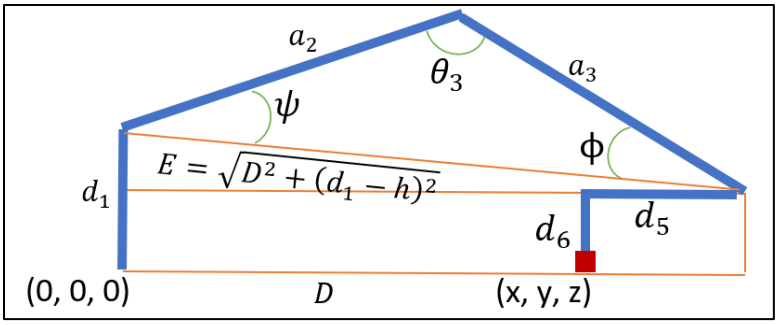


Figure 8a: Deriving the Inverse Kinematics - Side View

if $d_1 > h$

$$\theta_2 = 360 - \left(\psi + 90 - atan2(d_1 - h, D)\right) \tag{5a}$$

$$\theta_4 = 360 - \left(\phi + atan2(d_1 - h, D)\right) \tag{6a}$$

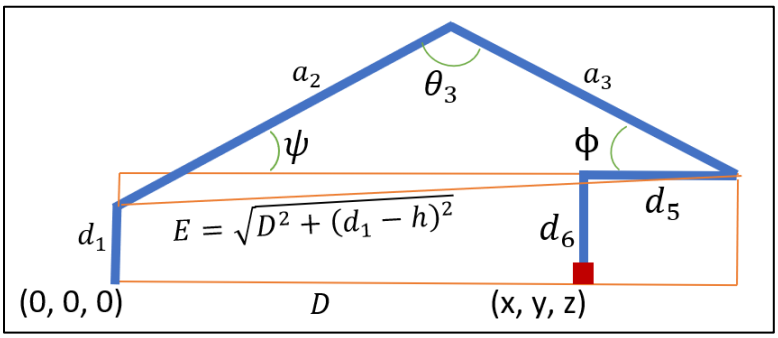


Figure 8b: Deriving the Inverse Kinematics - Side View

if $d_1 < h$

$$\theta_2 = 360 - \left(\psi + 90 + atan2(h - d_1, D)\right) \tag{5b}$$

$$\theta_4 = 360 - \left(\phi - atan2(h - d_1, D)\right) \tag{6b}$$

where $\psi = cos^{-1}\left(\frac{a_2^2 - a_3^2 + E^2}{2a_2E}\right)$, $\phi = cos^{-1}\left(\frac{-a_2^2 + a_3^2 + E^2}{2Ea_3}\right)$

The “360-”s and the “+90”s were simply inlcuded to account for diffrence in the raw angle and the required joint angle in the model.

An example sequence of motions is shown in Figure 9.

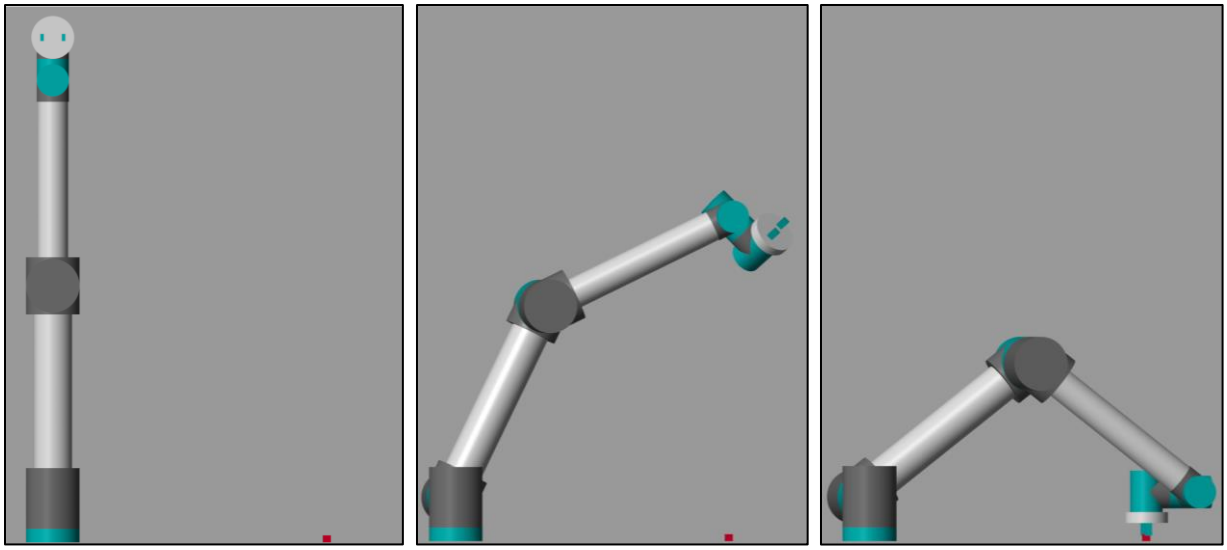


Figure 9: UR-5 Model Reaching for Block

Once the model was proven to successfully position itself to pick up a single block for a range of positions, more blocks were added to form a 6-block stack, as seen in figure 10. This transition was simple, all that was required was making the script into a loop to creating the joint angle signal for each block in sequence, and reversing their drop positions (which as 90 degrees from their initial position). Adding in the backlash and encoder errors did virtually nothing to the accuracy of the arm positioning.

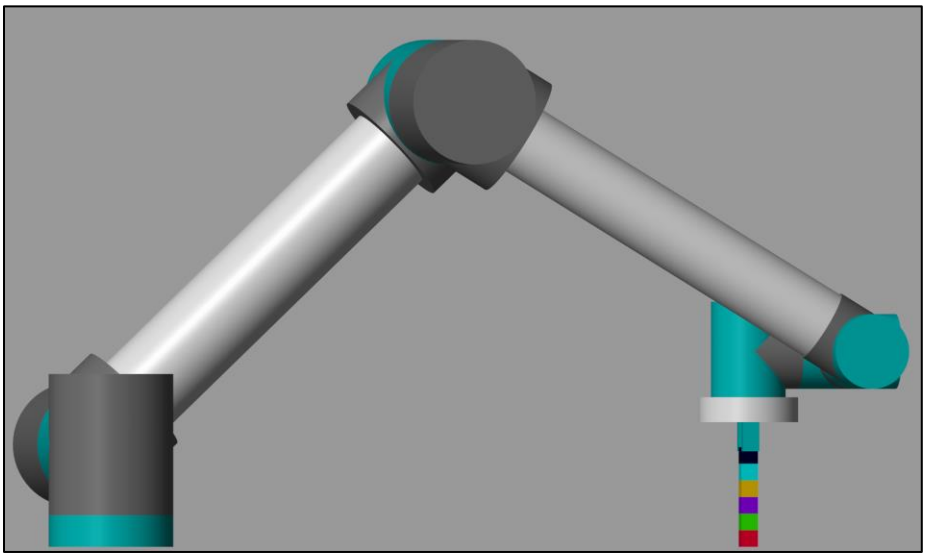


Figure 10: UR-5 Model Reaching for Top Block in Stack

At this point the model robot was able to position itself to pick up the blocks, as well as place them, but to pick up the blocks, contact forces between the blocks and the robot grippers were required. The “Contact Forces Library” for simscape multibody was downloaded from Mathworks file exchange, but proved to be more difficult than the online demo video made it out to be. The grippers still passed through the block… or it threw the block … or the simulation got stuck when the grippers clamped down on the block. An example which came with the downloaded library showed a simple robot moving a block from one conveyor belt to another, but upon inspection the reason for the issues with the simulation was never resolved.

Because the model was never able to pick up the blocks, the model was never made to accurately model the weight of the UR-5 and have the joints controlled with torques. Despite the failure to achieve the goals set forth in the proposal, this report was required to be completed, so the work completed was documented. The author created a github page to share his code files: https://github.com/DavidASmart/ENPM-662-Fall-2017-Semester-Project-/tree/master

Works Cited:

[1] Furrer, Fadri, et al. "Autonomous Robotic Stone Stacking with Online next Best Object Target Pose Planning." (2017).

[2] “Harmonic Drive® strain wave gear.” *Harmonicdrive.net*, Harmonic Drive, LLC., 2017, http://www.harmonicdrive.net/technology.

[3] “RESOLUTE™ with RESA.” *Renishaw.com*, Renishaw, PLC, 2015, www.renishaw.com/en/resolute-absolute-encoder-system-with-resa-rotary-angle-ring--10939.

[4] Multi-Box and Pallet Shipping. June 27 2013. *Shipvine*, http://www.shipvine.com/content/show/b2b-warehousing-fulfillment.

[5] Universal Robots. “UR5/CB3 User Manual.” Universal Robots, 2014.

[6] UR-5. 2014. *Universal Robots*, https://www.universal-robots.com/products/ur5-robot/. Accessed Oct. 24 2017.

[7] Vanallen, Peter. Photoshop made stack-o-blocks. May 22 2009. *Flicker*, https://c2.staticflickr.com/ 4/3346/3553159729_99a7523ac5_z.jpg?zz=1. Accessed Oct. 24 2017.