

Fall 2019 ME751 Final Project Report
University of Wisconsin-Madison

Simulation of Robotiq Linkage-Based Robotic Gripper Interacting with a Cylindrical Handle Using Chrono

Mike Hagenow

December 14, 2019

Abstract

When humans interact with constrained objects in the environment, it is common to allow slip in orientation to avoid awkward configurations of the arm. One question in robotics is how a gripper that is designed to perform only rigid grasps can be configured or controlled to allow a similar type of slip. This project looks at modeling a common linkage-based robotic gripper in Project Chrono and running simulations of grasping a cylinder, which is supposed to provide an analog for grabbing a cylindrical handle. One main goal of the project was to collect grip force data from simulation to determine whether grip force contains signal artifacts that can inform slip control. The other main goal was to explore the Project Chrono framework and learn both about use and capabilities. The project is successful in modeling and simulating the gripper in constrained interaction and learning about the capabilities of Project Chrono. The simulation result force transients are consistent with expected physics, but do not lead to an obvious observer-based design for controls. However, the results show promise for two future directions to explore Project Chrono as a tool to look a gripper environment interactions.

All code and build instructions are provided in the following repository:

<https://github.com/mhagenow01/751finalproject>

Contents

1. General information	4
2. Problem statement.....	4
3. Solution description	4
4. Overview of results. Demonstration of your project	5
4.1. Chrono Tutorials	5
4.2. Gripper Kinematic Simulation	5
4.3. Friction Testing	6
4.4. Gripper Cylinder Complementarity and Penalty Methods.....	7
4.5. Analysis and Potential Controls Strategy	8
5. Deliverables:	10
5.1. Files included in repository	10
5.2. Building the files in the repository	11
5.3. Running the Repository.....	11
6. Conclusions and Future Work	12
References	12

1. General information

1. Home Department: Mechanical Engineering
2. Current status: PhD student
3. Mike Hagenow (Team Leader)
4. I release the ME751 Final Project code as open source and under a BSD3 license for unfettered use of it by any interested party. Note: Some of the meshes are from another GitHub repository, https://github.com/a-price/robotiq_arg85_description, which is also under a BSD license.

2. Problem statement

We have developed a method for recognizing directions of slip when a user performs a demonstration using instrumented tongs. Our intent is then for the robot to perform a similar action. A major challenge is that slip is hard to control as it involves modulating forces to allow prescribed levels of motion. Grippers also commonly have only unilateral force sensing and control capabilities. Given the lack of available sensing, I think this is a great opportunity to use simulation to better understand the interaction of the gripper with the object to be able to think about control strategy without having to explicitly instrument the gripper further.

This implementation uses Project Chrono in order to develop a friction model and simulations involving a common linkage-driven robot gripper with a cylindrical handle (e.g. toaster oven, fridge). Running simulations has the potential to give us insight as to how we should dictate our gripper force control loop to interact with the object.

3. Solution description

Implementing this project required building skills using Chrono, doing kinematic modeling for systems with several bodies and geometric constraints, as well as implementing various friction models and exporting data. In order to achieve these goals, the project was broken up into four main milestones:

- Milestone 1: Complete a subset of the Chrono tutorials to get familiar with the interface
- Milestone 2: Model the kinematics of the gripper and perform a kinematic analysis that demonstrates the functionality of the linkage-based design of the gripper
- Milestone 3 – Set of a friction/contact problem against a rotating cylinder utilizing both the penalty and complementarity methods.
- Milestone 4 – Extract simulation data and compare with some experimental results to determine whether a grip-force based controls strategy for slip is feasible.

All of the code is structured to be run independently. Many of the files have redefinitions of the gripper geometry, but with various tweaks for the desired task or friction model.

4. Overview of results. Demonstration of your project

4.1. Chrono Tutorials

To gain competency with the Chrono platform, I first had to go through a few of the tutorials to understand the basic structure of how the simulations are constructed. Below is a list of the tutorials and demo files that I worked with and the main takeaways as far as code structure and functionality:

Installation guides (Core Chrono Module + Linking to Chrono) - these instructions and tutorials were well designed. Chrono makes good use of CMake and is configured to make linking libraries a very straightforward process.

demo_CH_buildsystem – This tutorial gives a good overview for creating rigid bodies, adding markers, and constraining them together with links. The slider crank also provides an example of a constant angular velocity motor, which is what I used for early testing.

demo_CH_functions – This tutorial was helpful for creating custom functions, which I used for the driving constraints, both for the kinematics of the gripper earlier on and to define a function to turn the cylinder. There are also some good prebuilt function classes for things like sine waves, but I actually thought just creating the ChFunction instantiation was easy enough!

demo_CH_stream – This was interesting to look over as far as filestreaming with Chrono. In my project, I actually ended up just using the std library filestreaming, but the Chrono functionality seems much more advanced and would be useful for more involved projects.

demo_IRR_crank/my_example – These were used to get some basic idea of how to construct the irrlicht visualization. These included setting colors, moving the camera, creating a floor for reference, etc.

Reference manual - In addition to sample code, I found it very useful to look over the reference manual. A few examples of sections I reviewed include ChFunctions, Collision Shapes, Motors, and Simulation System.

4.2. Gripper Kinematic Simulation

The first main milestone of the project was to model the kinematics of the gripper in Chrono. This turned out to be a bit more challenging than originally anticipated. Since the gripper is often used with commercially available robots, there is a ROS package that had some of the meshes and dimensions for the bodies and joints. In ROS, these are expressed in a URDF (unified robot description format) file, so I needed to transcribe the geometry to the Chrono format. The major challenge was that not all of the geometry was defined. In ROS, only serial chain robots are supported. Since this gripper is a linkage-based parallel mechanism, the way it is displayed in ROS is a bit of a visual hack. As a result, there were no dimensions for the redundant joint that is attached from the outer finger to the inner finger (gripper end).

For proper linkage kinematics, proper dimensioning is crucial as small errors can greatly affect the range of motion of the mechanism. The process was mostly trial and error to try and best determine geometry that lead to similar kinematics as are observed on the actual gripper.

When developing the kinematic demonstration, which opens and closes the gripper using a custom function imposed on the most proximal joint to the cuff, small errors in the geometry cause the redundant joints to build up error and cause stability issues for the system. I also ran into problems where I needed to make sure I had to play with my integration step size to get proper simulations, likely because of this same error.

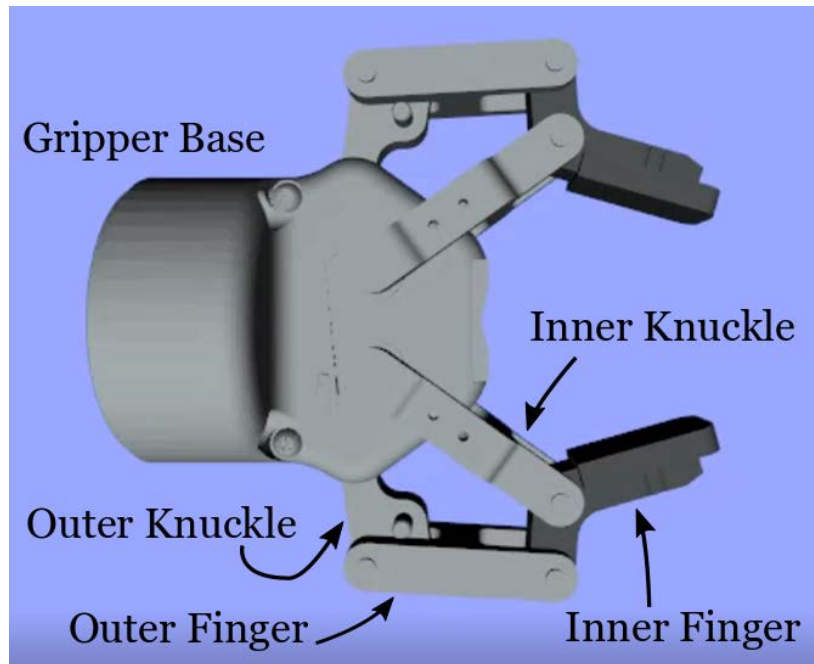


Figure 1: Gripper Kinematic Model in Chrono

Overall, the project was successful in the goal to replicate the gripper kinematics and provide a demonstration where the gripper opens and closes in Chrono.

4.3. Friction Testing

Before doing the cylinder grasp analysis, I created a friction testing playground for the complementarity method. The goal of this playground was to work on the collision geometry and to experiment with the contact properties.

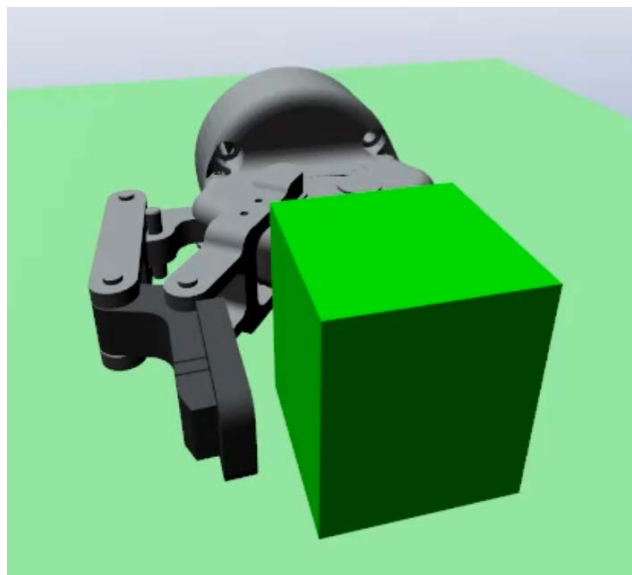


Figure 2: Gripper Pushing Block in Chrono

Since the visualization was created using stl meshes for the pieces of the gripper, collision geometry was still required in order for the gripper to interact with other objects in the environment. I experimented with adding basic primitive collision shapes (in this case, all boxes) to various links of the gripper geometry. In the simulation, I didn't add any collision properties to the interior link as it was difficult to get the orientation correct and it was leading to other unintended collision issues. As a result, you can see during the simulation that the block briefly penetrates the inner finger. In the future, I will try and get the collision models refined so that the gripper can undergo more complex interactions with the environment. In this case, only the collision with the gripper pads was needed for the cylinder grasping. This playground was also used for testing the contact and friction properties. Overall, it provided a good troubleshooting platform to find issues with the gripper model and friction.

4.4. *Gripper Cylinder Complementarity and Penalty Methods*

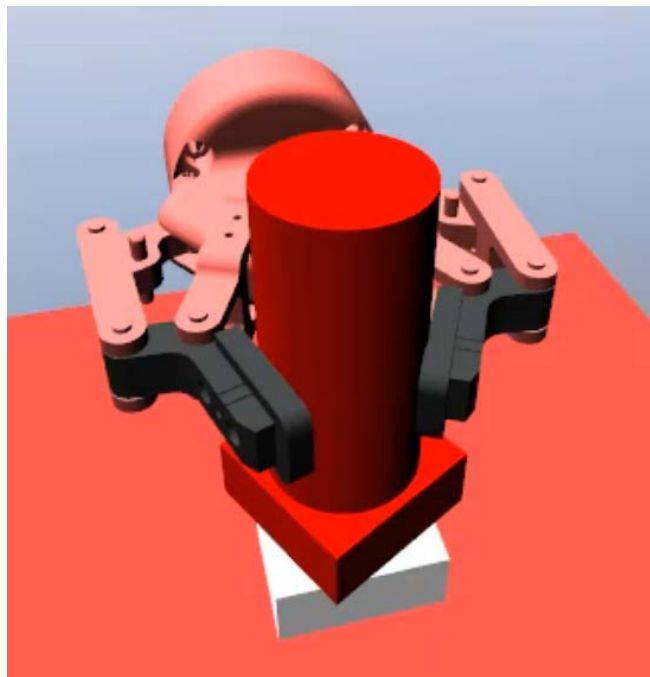


Figure 3: Cylinder Grasp in Chrono (Complementarity)

Once friction had been explored, two different simulations were developed: one that uses the complementarity method and one that uses a penalty-based method. As far as implementation in Chrono, switching between the methods was relatively trivial. I ran into some issues getting the contact to function properly and the cause ended up being that I was using old functions that did not appear to have the intended effect in the newest version of Chrono.

In order to create the gripping effect, a cylinder and stand were created using the ChEasy rigid body classes in Chrono. The largest challenge in building the cylinder grasp was determining the thickness and position of the cylinder. Compared to a purely parallel gripper, the linkage-based gripper has two options for grasping: friction and form-fit grasping [1]. If the object is located near the end of the gripper fingertips, the design will naturally lead to a friction-type grasp. For this project, I chose to do more of a

form-fit grasp, which is more similar to how a person would naturally grasp a handle. The gripping motion is produced by a constant-torque motor on the outer-knuckle link. The cylinder is driven by a custom function that starts stationary and then turns at a fixed rate (to simulate grabbing and then slipping). In the penalty method, there is intentionally a fairly large amount of penetration in order to highlight the effect of this method. Overall, this section produced the intended outcome: a set of simulations employing the two friction models to simulate interaction between the gripper and a handle.

4.5. Analysis and Potential Controls Strategy

For the final milestone and analysis, one challenge was to export and plot data in an easy way. To accomplish this, I created bash scripts that run the Chrono files, which log data to a file and then run python scripts that pick up the data and plot using matplotlib. This proves to be a valuable way to get the simulation performance out of the C++ Chrono library while getting the easy prototyping and visualization flexibility of python.

The goal of exporting data was to look at the grip force on one of the gripper fingers to see if the transients and simulations contained any interesting artifacts that might be able to inform control design when the grip force is the only instrumented and thus, observable data (common for robotic grippers). The simulations from the two contact methods yielded similar and expected results. There is some transient in the contact forces from when the gripper makes contact. Otherwise, the contact force for gripping maintains a constant value from the applied grip torque leading to a constant kinetic friction imposed via the friction coefficient. These effects can be seen in the below plots.

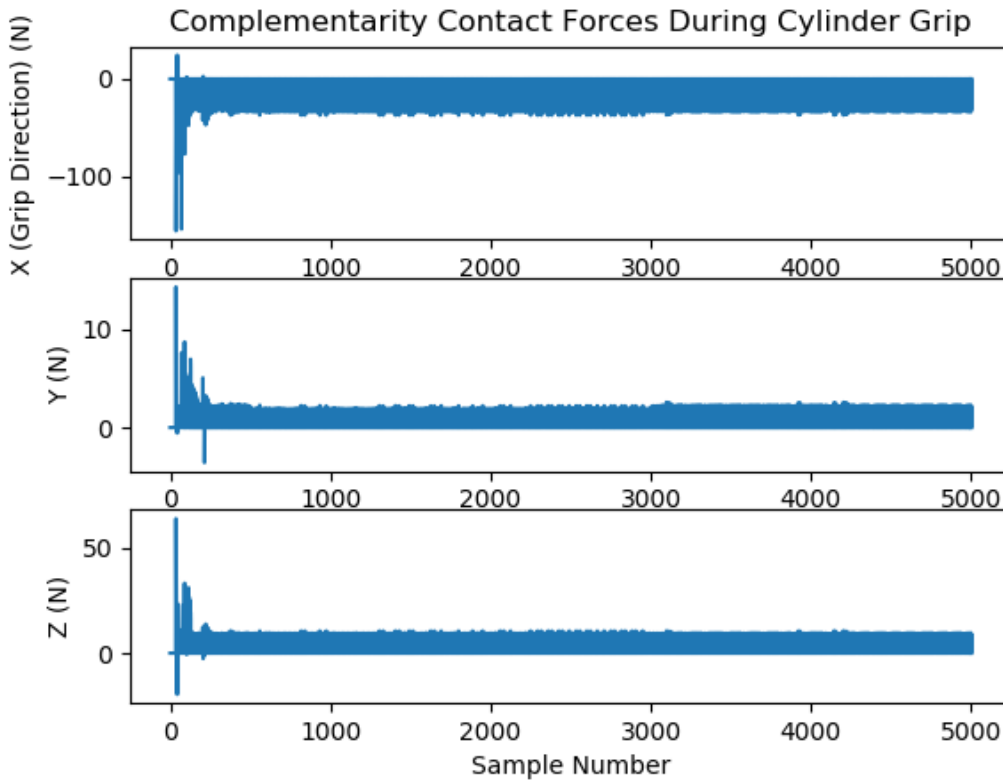


Figure 4: Gripper contact forces during cylinder grasp (penalty method had similar results)

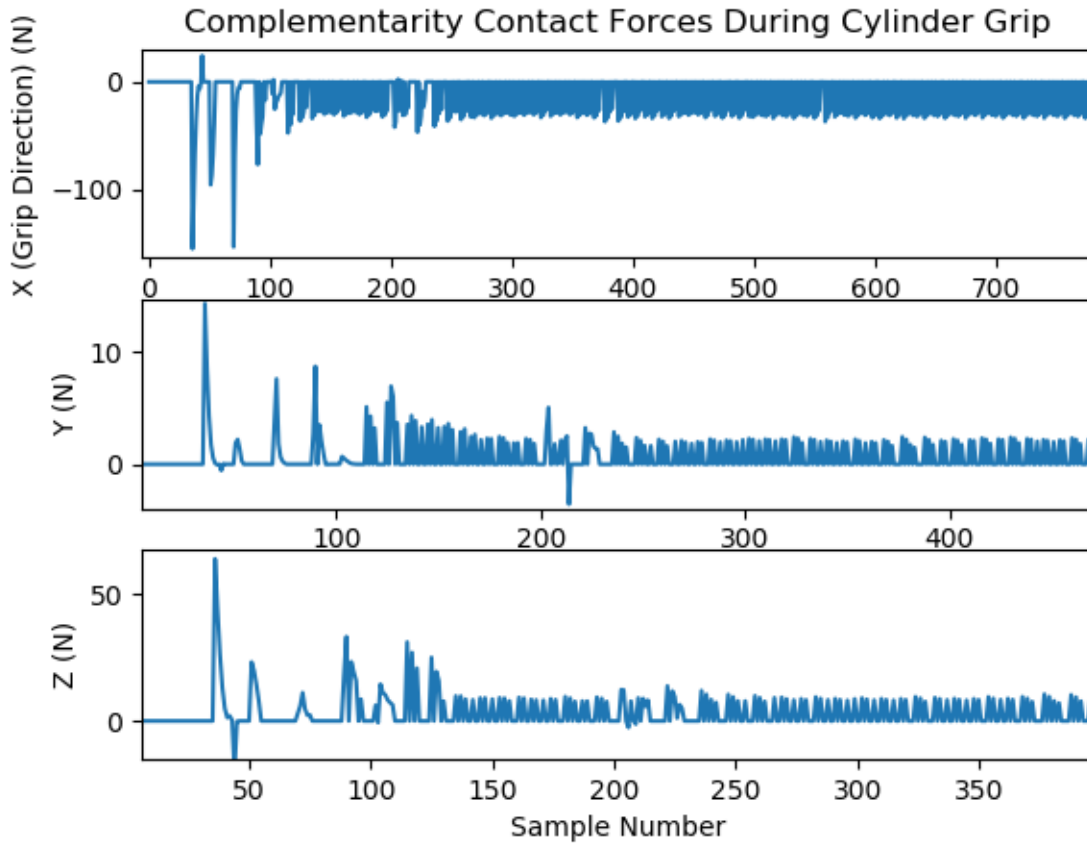


Figure 5: Gripper contact forces during transient (zoomed view)

While this corroborates the expected behavior for the friction model, it turns out to be a very different profile from what is observed in practice during demonstrations of slip. We have access to a pair of instrumented tongs that have 6-axis ATI force torque sensors in each of the handles. I performed similar demonstrations of slipping on a cylindrical handle and extracted an estimated grip force. As seen below in Figure 7, the profile is not constant. The grip force has much greater variability partially due to the lack of proper force control. While a human is much less likely than a robot to regulate force properly, these artifacts would likely also be present in the gripper force measurements due to a low control bandwidth. As a result, it is difficult to draw any conclusions for a control strategy based purely on the gripper simulations (which properly characterize an ideal interaction with the handle), however, it inspires many ideas for future work where simulation could help inform the state of the gripper and slip without having to rely purely on this one measurement. These are discussed further in the conclusions and future work section. Overall, while this simulation did not directly inspire any controls strategy, it was a successful exercise in learning to run and extract data from Chrono for visualization and analysis in python. The simulations were also successful in creating the friction models and visualizing the transient between gripping and eventual slip on the cylinder.

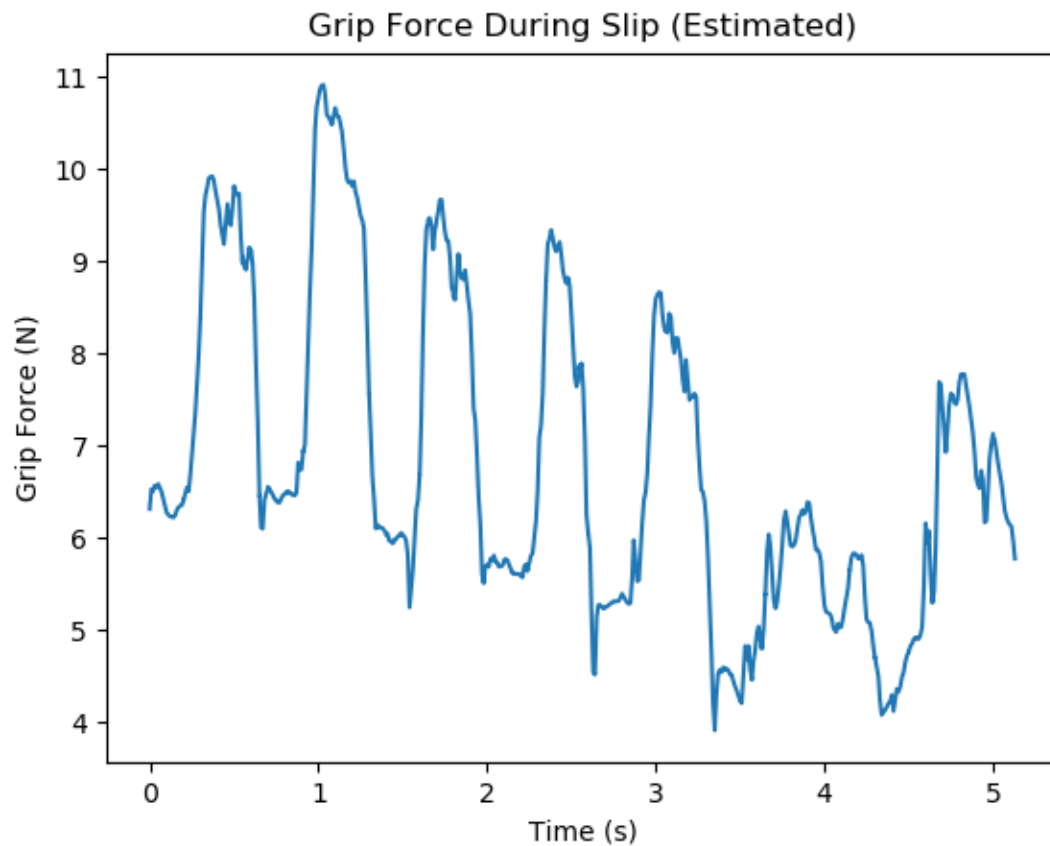


Figure 6: Grip forces during human demonstration of cylinder slip

5. Deliverables:

5.1. *Files included in repository*

Below is a list of code that was created to support this project:

- /gripper_slip_simulations/gripper_kinematics.cpp
- /gripper_slip_simulations/gripper_friction_test_comp.cpp
- /gripper_slip_simulations/gripper_complementarity_friction.cpp
- /gripper_slip_simulations/gripper_penalty_friction.cpp
- /gripper_slip_simulations/gripper_complementarity_friction_plot_generator.cpp
- /gripper_slip_simulations/gripper_penalty_friction_plot_generator.cpp
- /scripts/plot_results.py
- /scripts/run_complementarity.bash
- /scripts/run_penalty.bash

5.2. Building the files in the repository

The project builds against the Chrono C++ API. To build the gripper simulation files, you will need to follow the steps outlined here: http://api.projectchrono.org/development/tutorial_install_project.html

Additionally, since the project uses some relative paths, please follow the below steps to build successfully:

- Chrono should be placed in the root directory of the GitHub repository (i.e., at the same level as the gripper_slip_simulations directory)
- Chrono is configured to build all files into the 'build' directory in the root
- gripper_slip_simulations contains a CMakeLists that can also be used through CMake to build the gripper simulation files. I did all testing with Visual Studio 2017. It was tested on two separate machines. **The executables should be built to the build/Release/ directory** (requires that Visual Studio is set to 'Release'). The Bash scripts for plot visualization look for the exes in this directory (relative paths).

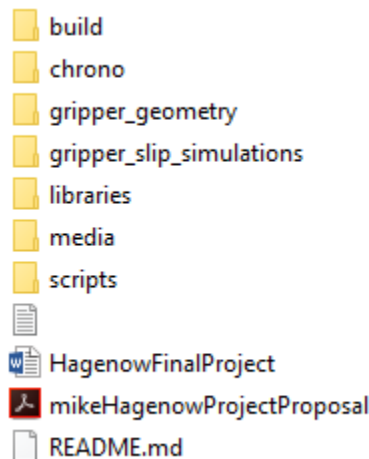


Figure 7: Example of intended file structure

5.3. Running the Repository

Once the files have been build, this repository contains four exe files that can be run alone that include irrlicht visualization (*gripper_kinematics*, *gripper_friction_test_comp*, *gripper_complementarity_friction*, *gripper_penalty_friction*).

Note: If you do not want to build the files, I have also included a video in the media folder (*gripper_simulation_videos.mp4*) that shows all four of the simulations.

In order to create the plots in the report, there are bash scripts in the /scripts/ directory that allow you to run the penalty and complementarity methods and plot the contact forces from one of the gripper fingers: *run_complementarity.bash*, *run_penalty.bash*

These scripts will run the Chrono simulation, save the data to a file, and then plot the data in python. **It is important these be run from the scripts directory as they rely on relative paths.**

They also require the following python packages: numpy, matplotlib. I tested using Python 3.7.0 on windows and using the Git Bash terminal (Note: I had to run *alias python='winpty python.exe'* for the system to recognize the python command).

6. Conclusions and Future Work

This project gave opportunity to utilize ME751 material in kinematic modeling, constraints, and using friction/contact models. The project was successful in achieving the four major milestones: chrono tutorials, kinematic modeling and simulation of a linkage-based gripper, setup of complementarity and penalty-based contact with a cylindrical handle, and exporting contact data for analysis/comparison to actual grip/slip data from a pair of instrumented tongs. I believe that Chrono has demonstrated its value as a tool that could be useful for future simulation in my work. Here are some specific use cases that are future work based on this project's initial investigation:

- The results of this project demonstrated the utility of a gripper simulation, however, the ideal conditions lead to data that does not corroborate real demonstrations. In future work, I will run demonstrations where the gripping is not done in a purely friction-based manner (leveraging the form-fit grasp). This will require extensive refinement of the collision models to properly model the contact forces. With a refined model, I believe we could inform what locations would be most primed for additional instrumentation (e.g. pressure sensors) based on locations where large forces are often applied during form-fit grasping as determined by running a set of demonstrations with differing initial conditions.
- We have recently learned that our gripper only provides static grip force capabilities, but does not allow for closed-loop control of forces. Using Chrono's closed loop (man-in-the-loop) functionality, we can demonstrate the utility of such a capability without having to prototype a physical gripper.

References

[1] Robotiq 2F-85 Gripper Manual. https://assets.robotiq.com/website-assets/support_documents/document/2F-85_2F-140_Instruction_Manual_e-Series_PDF_20190206.pdf