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**

The text of the abstract comes here. Use font size 12 throughout the document.   
IMPORTANT NOTE: The name of the file for your Final Project report should be me751FirstnameLastname.pdf. Like me751DanNegrut.pdf. Drop this Final Project report in Canvas in the folder “FinalProject751” by Tuesday, December 14, at 2:45 PM. Do not go beyond 10 pages unless you really feel like you have to (page count doesn’t include TOC, abstract, etc.). Feel free to use a LaTeX source as long as you follow this format; in the end, you will be uploading a PDF anyway.

All code and build instructions are provided in the following repository: <https://github.com/mhagenow01/751finalproject>

Contents

[1. Problem statement 4](#_Toc531620937)

[2. Solution description 4](#_Toc531620938)

[3. Overview of results. Demonstration of your project 4](#_Toc531620939)

[4. Deliverables: 4](#_Toc531620940)

[5. Conclusions and Future Work 5](#_Toc531620941)

[References 5](#_Toc531620942)

# 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.

# 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). It is my hope that running simulations will give us insight as to how we should dictate our gripper force control loop to properly interact with the object.

# 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 4 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 able 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.

# Overview of results. Demonstration of your project

## 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 stuff –

demo\_CH\_buildsystem –

demo\_CH\_functions –

demo\_CH\_stream –

something for visualization –

reference manual

## Gripper Kinematic Simulation

Placeholder text

## Gripper Cylinder Complementarity Method

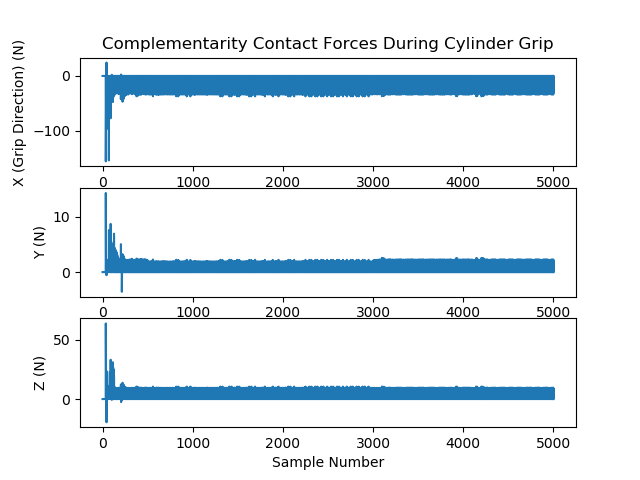
Placeholder text

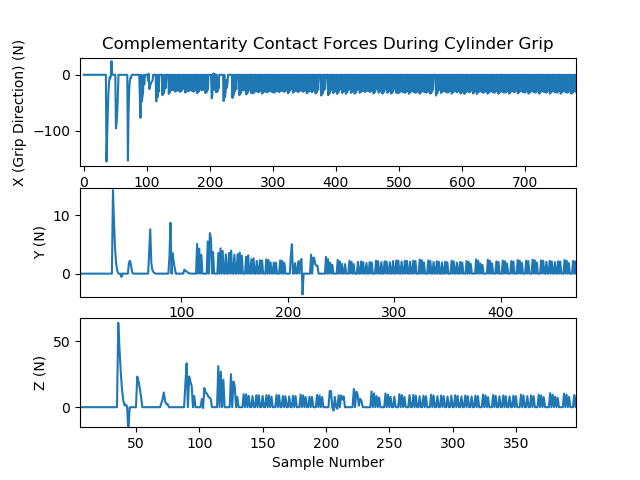
## Gripper Cylinder Penalty Method

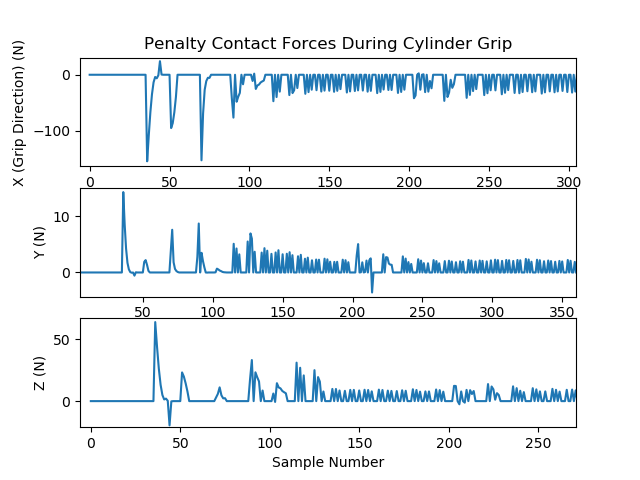
Placeholder text

## Analysis and Potential Controls Strategy

Placeholder text







Explain here what you obtained, explained why the results are good/bad. This is the place where you talk about the outcomes of your Final Project effort. It is not the end of the world if your code doesn’t work as anticipated. Explain here how far you have made it.

Most often, you have a comparison against sequential code, perhaps via a scaling analysis. Make sure you include plots and/or tables to show your results.

# Deliverables:

## Files included in repository

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

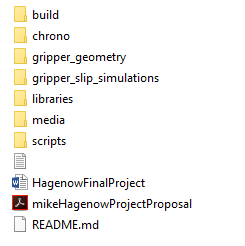
* /gripper\_slip\_simulations/gripper\_kinematics.cpp
* /gripper\_slip\_simulations/gripper\_friction\_test\_comp.cpp
* /gripper\_slip\_simulations/gripper\_complementarity\_friction.cpp
* /gripper\_slip\_simulations/gripper\_penalty\_friction.cpp
* /gripper\_slip\_simulations/gripper\_ complementarity\_friction\_plot\_generator.cpp
* /gripper\_slip\_simulations/gripper\_penalty\_friction\_plot\_generator.cpp
* /scripts/plot\_results.py
* /scripts/run\_complementarity.bash
* /scripts/run\_penalty.bash

## 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 one two separate machines. When I tested, the executables were built to the build/Release/ directory (provided you compile with Release).

[](https://github.com/mhagenow01/751finalproject/blob/master/media/example_file_structure.PNG?raw=true)

## Running the Repository

Once the files have been build, this repository contains 4 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).

# Conclusions and Future Work

Simulations where the slip is in the “whats it called” zone

Using the closed loop to simulate missing gripper functionality

# References

[1] Make sure to give credit where it’s due.