University of Florida College of Engineering

Chemical Engineering Department

Bifurcation and Non-Linear Instability Laboratory

User Manual

"Taylor-Couette Cells"

Project In-charge: Dr. Ranga Narayanan

Designer: Ronald Wilson

Email: ronaldwilsonk@hotmail.com

Contents

System Assembly

This section elaborates on the parts used to build the model of the experiment.

Figure 1 The DC power supply system

The DC power supply is given in Fig 1. It converts 200VAC to any set DC voltage and current. The power supply system has various configuration parameters that have already been set. Do not change the jumper configuration behind the power supply unit for the experiment. For additional information on the configuration, please refer to the user manual for the Sorensen DCS 150-200.

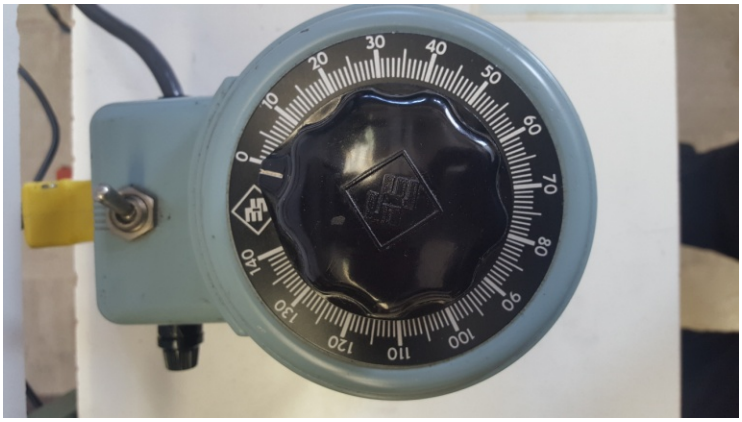


Figure 2 Motor driver for Motor 2

The Motor Driver is shown in Fig 2. It has two terminals. One of them is directly connected to a 110VAC supply and the other to the motor power input. The knob controls the voltage output to the motor which in turn controls the speed of the motor. The driver also has a toggle switch. The toggle switch turns the motor on and off. The motor driver for motor 1 has a 3-way toggle switch. The middle position for the switch is “off”. The other two positions control the max output range from 120 to 140V AC. The positions are marked on the toggle switch. Always use the drivers at 120VAC.

Fig 3 shows the Motor Driver for Motor 1 with the stepper motor mounted on top. The stepper motor driver shaft is coupled mechanically with the knob of the motor driver. The speed control is achieved by actuating the stepper motor in both clockwise and anti-clockwise direction. The resolution of the stepper motor is limited to 1 degree per step. The support mechanism (shown in green) is mounted onto the motor driver. It can be removed if maintenance is required. Refer Section 4 for more infromation. The primary purpose of the support mechanism is to hold as interface between the stepper motor and the motor driver. It also prevents the stepper motor from vibrating and turning about its own axis.

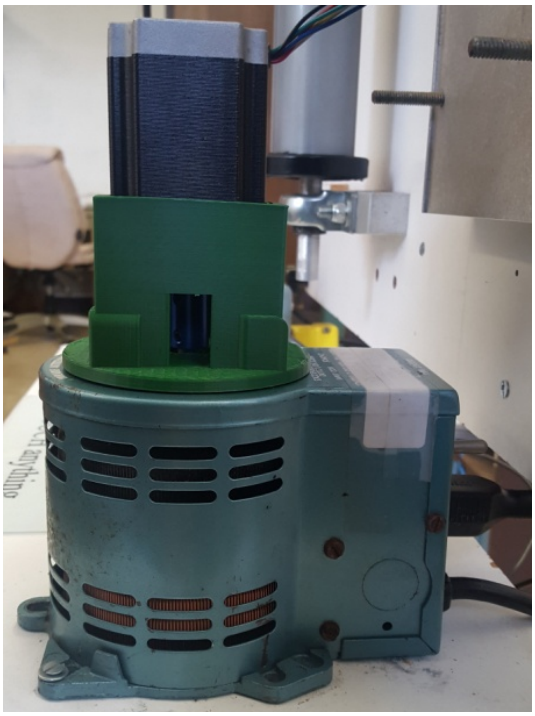


Figure 3 Motor driver for motor 2 with stepper motor mounted on top

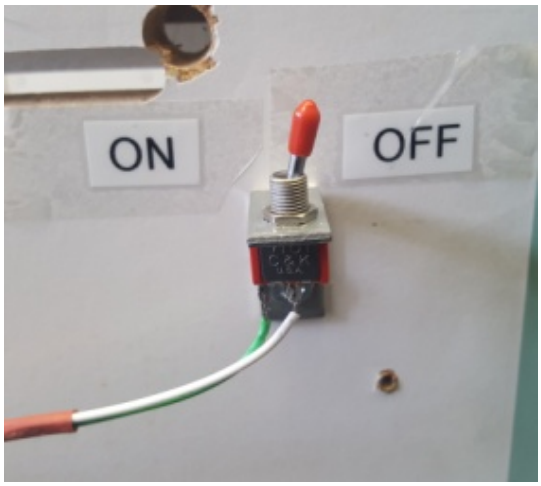


Figure 4 The ON/OFF toggle switch

The turn on/off switch is depicted in Fig 4. The switch is located behind the motors on the experiment structure. The purpose of the switch is to initialize the experimental setup. The turn on phase starts up the motor driver on motor 1 to a preset position. The position is setup empirically to facilitate minimal revolutions on the motor. The minimum revolutions is set in range of 150~200 RPM. This set point can be changed. Refer section 4 for more information.

Figure 5 NI DAQ

The Data Acquisition Unit or DAQ is shown in Fig 5. The device facilitates the transfer of data from and to the interfacing computer and the experiment. For the current experimental setup (Build 1.1), the functionality for the DAQ is limited to setting the input RPM for Motor 1 control system. The input RPM is set through a GUI on the computer connected to the DAQ using a USB.

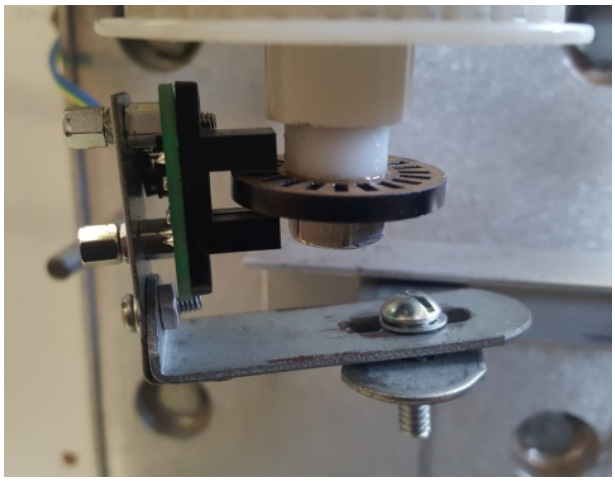


Figure 6 Encoder

The encoder is shown in Fig 6. The device measures the speed at which the motor is turning. The encoder is laser-based. The encoder returns a square pulse train as the laser is pulsed through the encoder disk to the receiver on the other side. The encoder disk has 20 openings through which the light can pass through to the sensor. The number of pulses with respect to a particular time frame can give the speed of the motor. In this experiment, the numbers of pulses are counted for a time period of one second and the speed is calculated in RPM.

Figure 7 AC Motor

The motor is show in Fig 7. It runs at 110VAC and has a max RPM of 10000. However, for this experiment, the max RPM will be electronically limited to 5000. The Voltage-RPM characteristic for this motor is non-linear in nature. Hence, the PI controller was used maintain a constant speed at all times.

The stepper motor driver is shown in Fig 8. The driver requires a 20VDC supply to start working. The configurations are preset and do not have to be changed. To change configuration, please refer the datasheet for the CW250/2/8. The stepper motor is connected in full coil bipolar configuration.

Figure 8 Stepper Motor Driver

The main processor for the experiment is the Arduino Uno shown in Fig 9. It is interfaced with the computer using a USB. The code and wire mapping for the Arduino is available on Github repository.

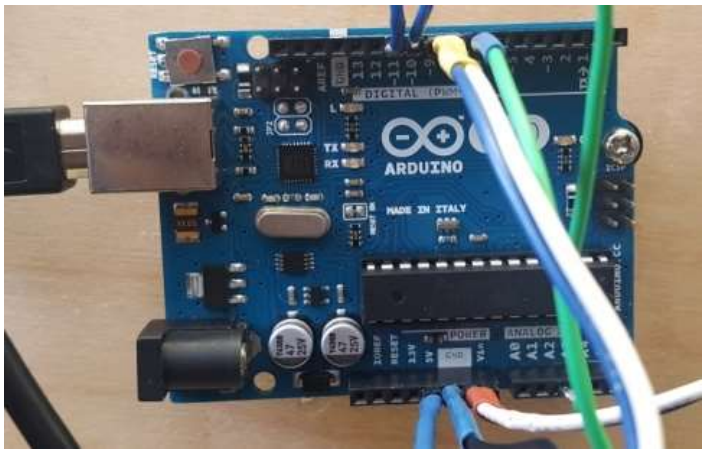


Figure 9 Arduino Uno (Primary Processor)

There is a voltage regulator installed in the setup (Fig 10). The regulator steps down the input from 20VDC to 5VDC to support the main processor and the encoders. The input limitation for the regulator is 36V. However, to enforce safe limits and a higher life cycle for the components, an operating limit of 20-25VDC should be enforced.

Figure 10 Voltage Regulator

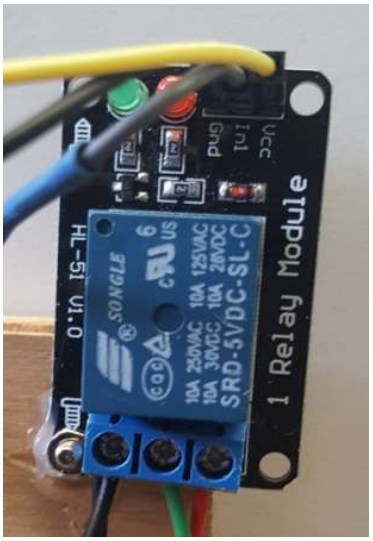
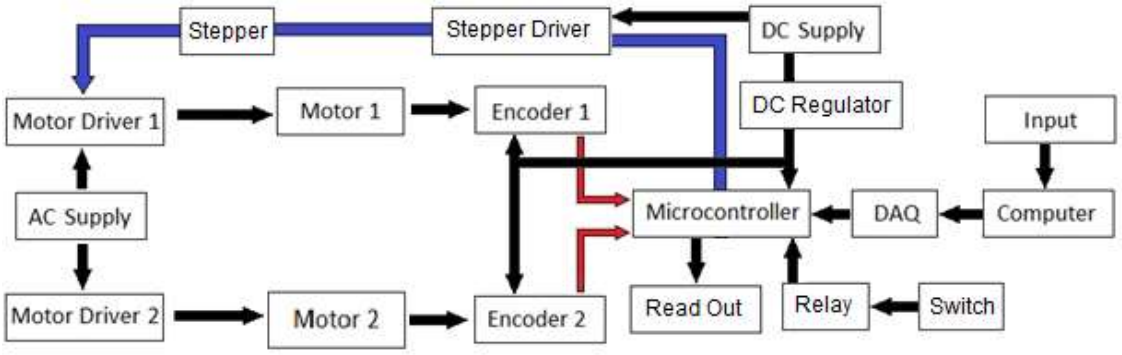


Figure 11 Relay Module

The relay shown in Fig 11 is to supplement the switch. It supports the switching process by providing 0 and 5VDC for the Arduino to process. The switch cannot be used an instant kill switch. The shutdown process takes around 10-15 seconds to complete.

System Model



The system is designed around a central feedback loop. The loop contains motor 1 and its associated components. Once the system is initialized using the on/off toggle switch, the system brings the motor to the initial set point saved in the processor. Then the input value is read from the DAQ. The DAQ outputs a voltage value that is proportional to the input RPM. The max RPM being limited to 5000, the input RPM from the DAQ is divided by 1000 and outputted as voltage (0-5V). This voltage is read by the Arduino using the onboard Analog-Digital Converter. Motor Driver 2 has to be tuned manually by-hand until the desired Taylor-Couette ring pattern is achieved.

The controller implemented in the processor is based on an improvised Proportional-Integral method. The system actuates the stepper proportional to the error between the set RPM and the current RPM obtained from the encoder. However, as the error approaches zero, the controller goes procedural. The steps are limited to one step at a time. A delay of 200 ms is introduced in the system to get rid of transient response. The transient RPM for motor 1 is readable through serial read from the processor. Once the system stabilizes, a “Motor 1 is stabilized” comment is displayed along with the RPM for motor 2. The system stabilizes as long as the RPM of the motor and the input RPM comes within 30 RPM of each other. This separation is considered necessary to rule out effects of noise generated from the system. The noise is primarily attributed to the gradual change in resistance of the coils in the motor due to continuous operation.

Startup Procedures

1. Connect the power supply terminals of the motors to a 110VAC supply. Switch on the toggle switches for each motor driver to ON (Always at 120V, if manual setting is required)

2. Connect the DC power supply system to 220VAC supply. Make sure that no terminals behind the power supply system are short circuited.

3. Turn on the switch as shown in Fig 12. '1' represents ON and '0' represents OFF. Both the displays on the system should now read "0.0" with a red LED turned on at Current knob.

Figure 10 Turn on/off switch and displays

4. Turn the Current knob clockwise. Now turn the Voltage knob clockwise slowly. The display would change as the voltage knob is turned. Turn in till the value reaches 20.0V in the voltage display. If the Current LED turns red again, turn the current knob clockwise till it turns off and the green led on the Voltage knob turns back on. Check for this indicator during operation. Increase current if required !!!

Figure 11 Control knobs

5. Turn on the computer and plug in the two USB cables. One from the DAQ and the other from the Arduino.

6. Turn on the LabVIEW application and set an input RPM in the text box. The LabVIEW interface is shown in Fig 14. Enter the RPM in text box marked as '2'. Then press the button shown in '1' to start transmitting the RPM to the processor.

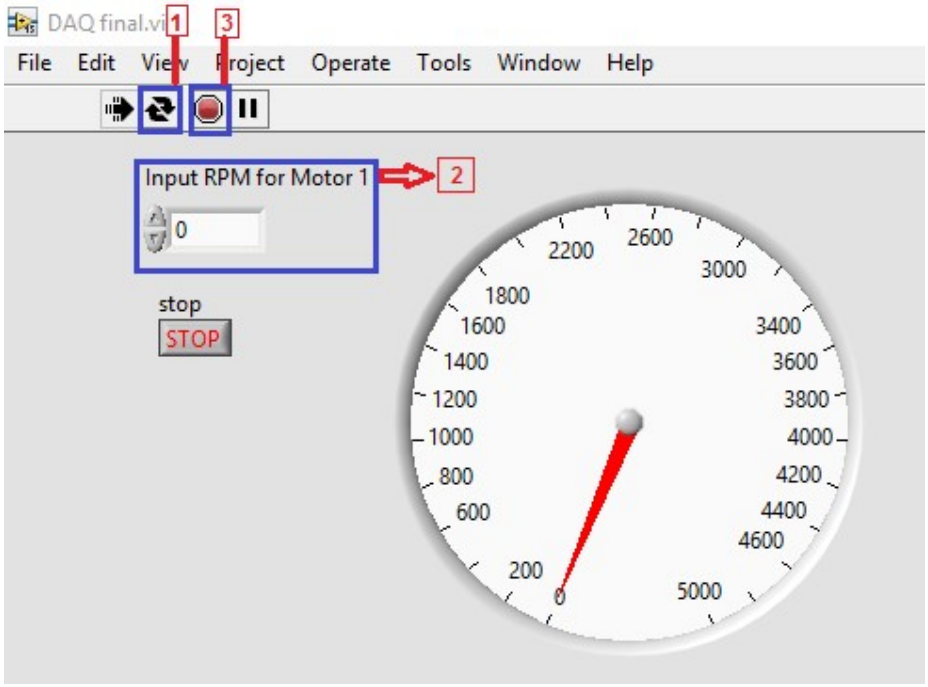


Figure 12 LabVIEW interface

7. Open the Ardunio application. The working version is Build 1.1. After opening the application, go to the menu shown in Fig 15 and select the active port. The active port will be available for selection and will be named differently for different computers. After selecting the active port, the menu will change to the screen shown in Fig 16.

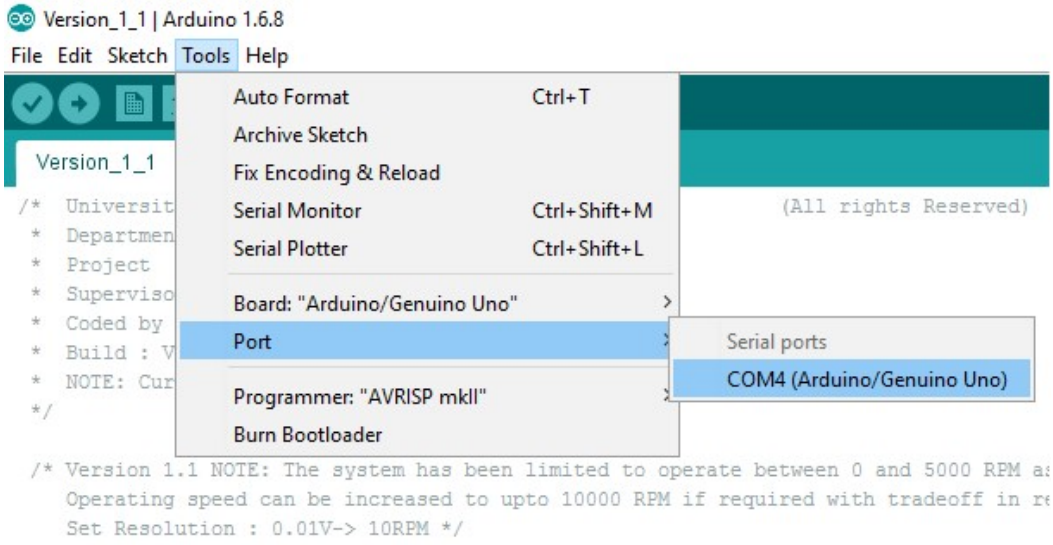


Figure 13 Port not selected

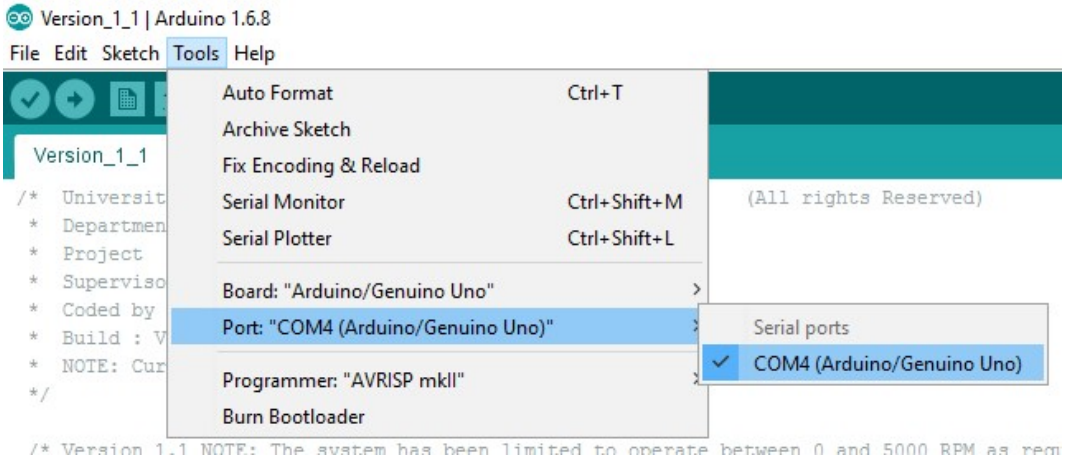


Figure 14 Port selected

8. Navigate to Tools->Serial Monitor on the taskbar. This window keeps the user updated on the status of the experiment. The navigation procedure is shown in Fig 17. **Keep the window open till the experiment is complete !!**

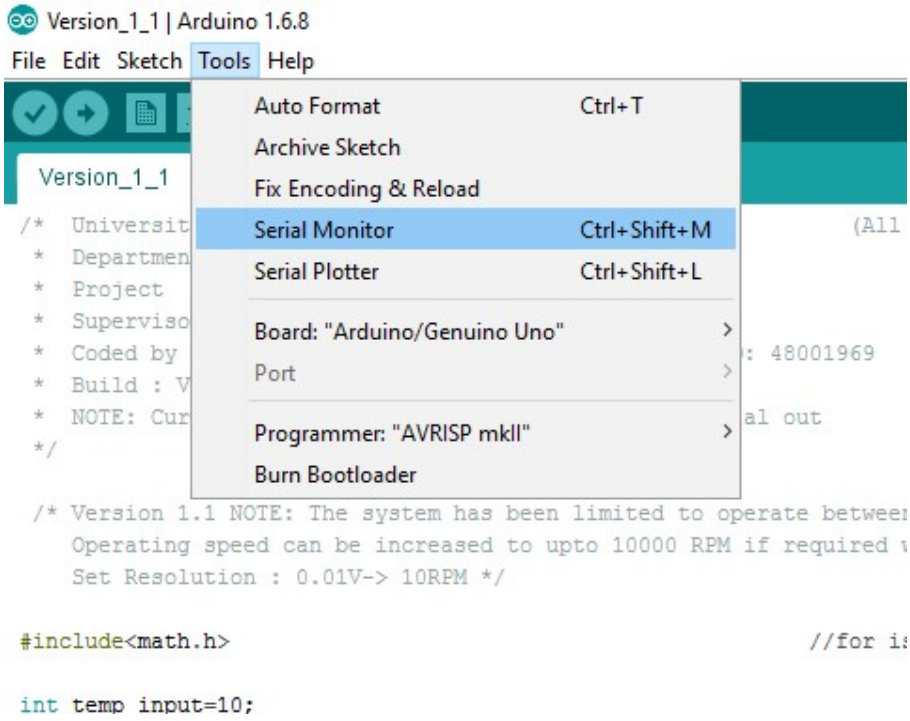


Figure 15 Serial Monitor

6. Turn on the toggle switch to the ON position. The stepper motor would now be initialized to the start point. Once set, the controller would be switched on and bring the motor to the set RPM. Wait till the stabilization prompt is displayed on the Arduino Serial Monitor Window.

7. Continue the experiment as required by turning the motor driver knob for motor 2 and changing the input RPM parameter on the LabVIEW application.

8. Once the experiment is concluded, turn off the experiment by switching the on/off toggle switch to the off position. Wait till the prompt “Shutting down” is displayed on the serial monitor and the transition process is complete. Turn off the motor driver toggle switches. From Fig 14, press button ‘3’ on the LabVIEW application. This turns off the input RPM transmission from the DAQ.

9. Turn off the DC power supply system by turning the Voltage knob counter-clockwise till the display shows “0.0”. Now, turn the Current knob counter-clockwise till the knob reaches position 0. A red LED on the current knob will be turned on. Turn off the switch on the DC power supply system. Make sure all power cords are disconnected.

WARNING: Do not touch the metal surfaces of any power supply system while in operation. Keep the experimental space free of moving debris and tools.

Troubleshooting

Problem 1:

The primary processor has a volatile memory. If the device is not shut down properly, the current position of the stepper motor will not be stored or reset to 0 i.e. Motor 1 will start turning before the system is initialized. This can also be caused if the DC power supply was not turned on/off properly.

Solution:

If this situation occurs:

- Turn off the DC Power supply system directly using the main switch as shown in Fig 12.
- Turn off the power switch on both the motor drivers.
- Reset the knobs to their original positions
- Check if the main on/off toggle switch is in the off position

After following the emergency steps given above, take the tool shown in Fig 18. It is a hex key (or Allen key). The key can be used to manually reset the stepper motor to the zero position.

Figure 16 The Allen/Hex key

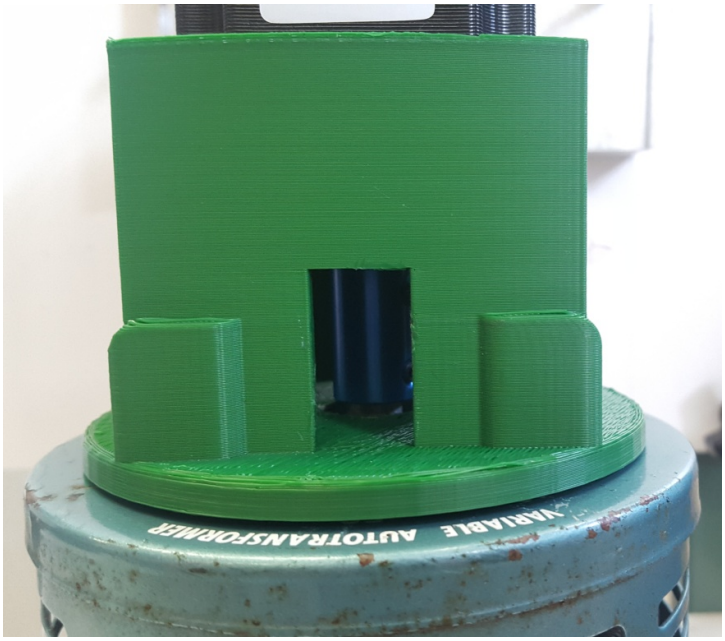


Figure 17 Coupler with lock screw

Insert the key into the coupler and loosen the lock screw on both sides. Once loosened, pull out the stepper motor along with the coupler (blue) and the support structure (green). Make sure to pull it out

parallel to the drive shaft. For reference, the mating mechanism is shown in Fig 20. Reset the motor driver by turning the control shaft counter-clockwise and taking it back to zero volts. Replace the removed part back in and tighten the lock screws on the coupler.

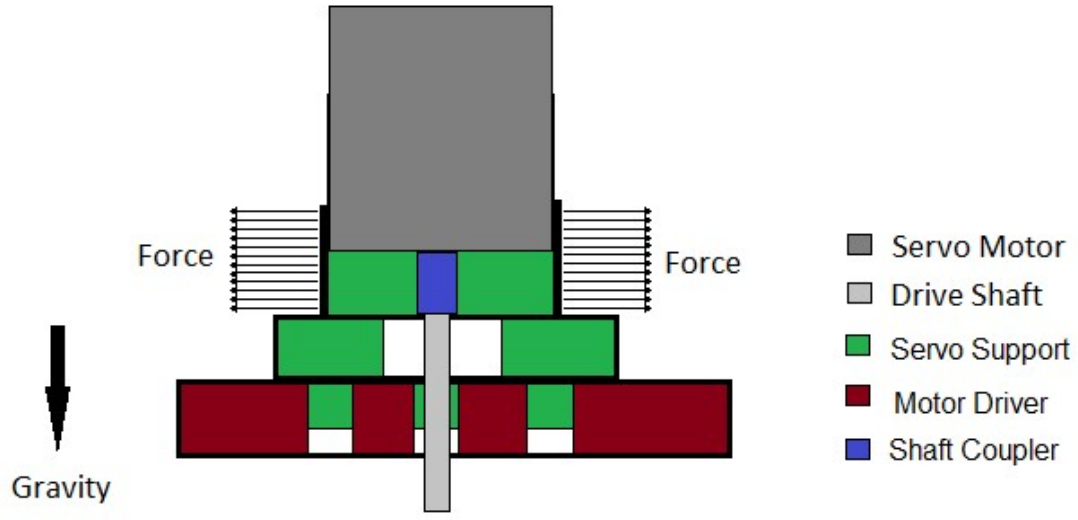


Figure 18 Stepper Motor attachment

Problem 2: Changing operating parameters for the system

Solution: Every parameter used to fine tune the system are declared and explained along with their wire map in the code. Please refer to the code for the processor. The code can be obtained on the online repository https://github.com/ronaldwilsonk/Taylor-Cells-Controller. Make sure to clone and use the repository. The master code for build 1.1 is meant to be a backup and should not be changed !