Robotic Arm Mini-project
Part 1 – Servo motors

What is it about?

We now start the Robotic Arm mini-project. It will last until Christmas. Over the coming 6 labs, you will design your own robotic arm (with at least 2 degrees of freedom, but you can add more), program it with ROS, build a 3D visualisation for it, and, if time permit, use a 3D motion planner to control it automatically!

During this first part of the Robotic Arm project, you will learn how to control servo motors with the Arduino Uno.

Aims

At the end of this lab, you should:

- Know how to wire a servo-motor onto an Arduino board
- Know how to control a servo-motor with PWM signals
- Understand the torque characteristics of the servo-motor

Specific Challenges

- This is the first part of a six-weeks long mini-project. Plan ahead!

Note

As usual, **document in your lab journal your findings**. Add **code snippets**, **screenshots**, **pictures** and link to **videos** as needed.

And do not forget: **write your lab journal as a text file using the Markdown syntax** and **push your journal and the pictures on GitHub**.

Part I

The Robotic Arm mini-project

Over the coming 6 labs, you will **design your own robotic arm** (with at least 2 degrees of freedom, but you can add more), **program it with ROS**, build a **3D visualisation** for it, and, if time permit, use a **3D motion planner** to control it automatically!

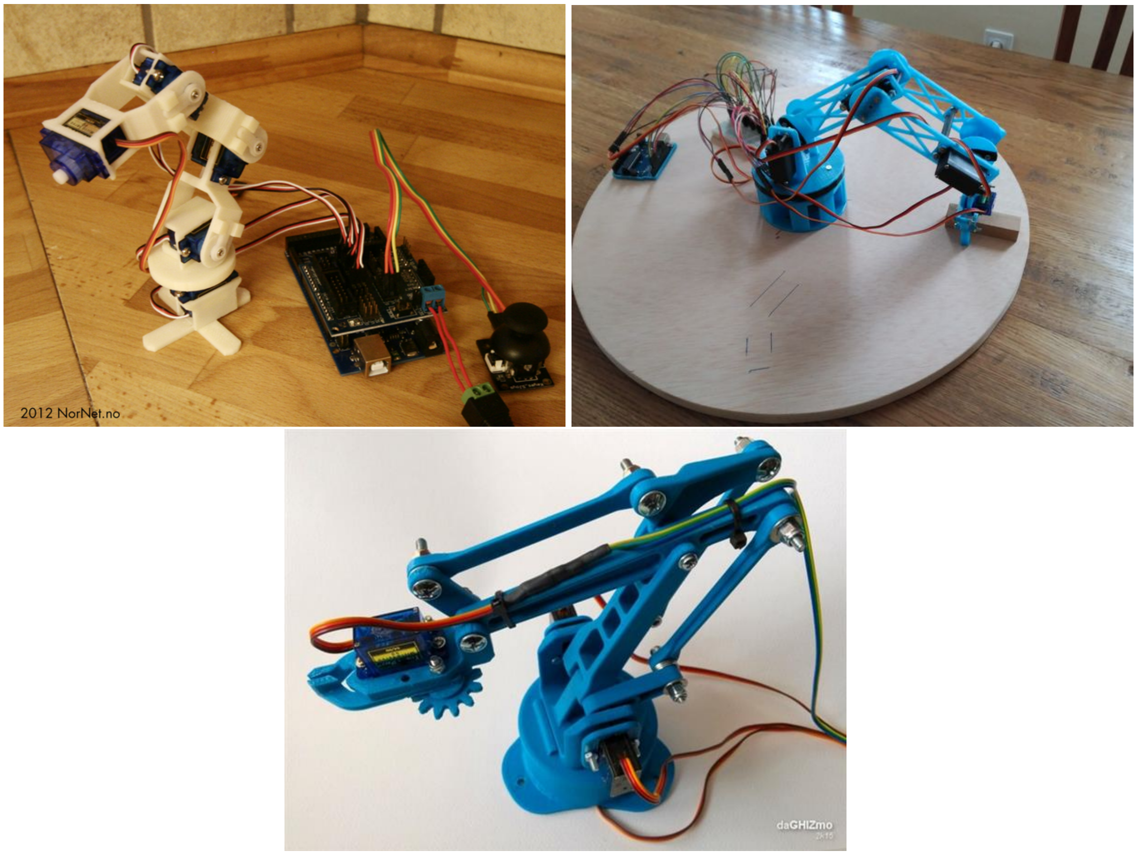


Figure 1: Three examples of 3D printed arms. The first one has 5 degrees of freedom, the second one 4 DoFs, the last one 3 DoFs.

Note

As for previous lab, you are expected to work in pairs. We only have enough servo-motors kits for 25 groups. **If you are not yet in a group, please form a pair today**.

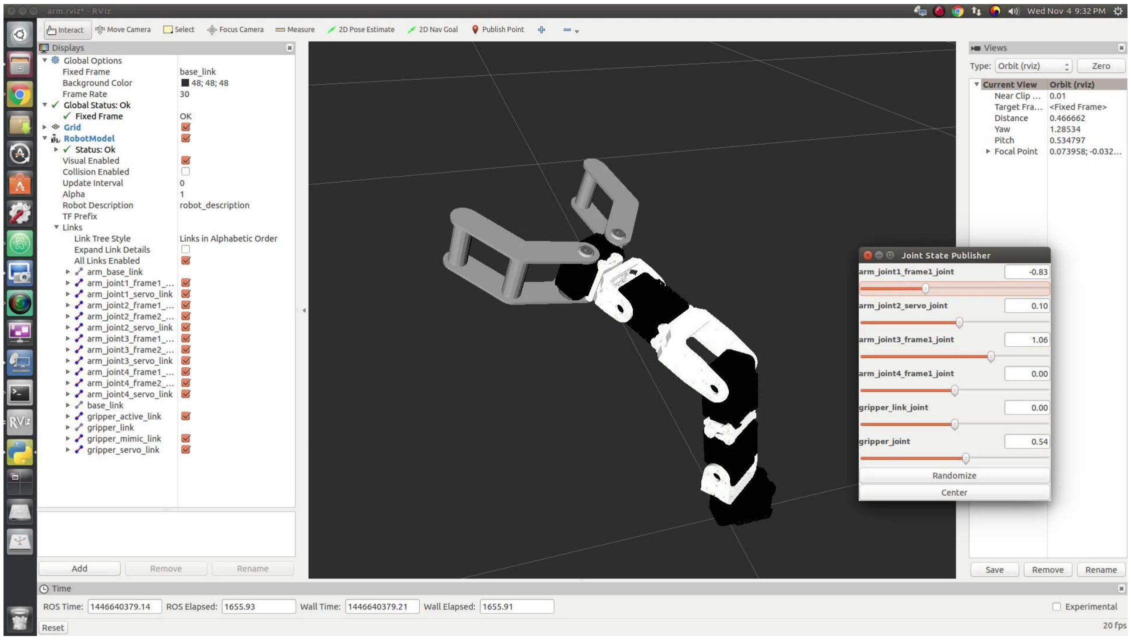
Each group will be given a servo-motor kit with 2 RC mini-servos. This allows everyone to build a robot arm with 2 degrees of freedom.

Taking it further

If you wish, you can create a design with additional degrees of freedom. **However, you will have to source any additional actuator by yourselves**. Small servo-motors or stepper motors can be found online for less than 5 pounds.

Working plan

- **This week**: control of two servo-motors with the Arduino; use of potentiometers to control them
- **Week 2**: Tutorial led by Jake on 3D design for effective 3D printing
- **Week 3**: Design of the arm continued; arm assembly
- **Week 4**: ROS on Arduino; control of the servo using ROS
- **Week 5**: 3D model of the arm, visualisation of the arm in RViz
- **Week 6**: Finalisation of arms; if time permit, 3D motion planning



Global Options	
Fixed Frame	base_link
Background Color	■ 48; 48; 48
Frame Rate	30
✓ Global Status: Ok	
✓ Fixed Frame	OK
Grid	☑
RobotModel	☑
✓ Status: Ok	
Visual Enabled	☑
Collision Enabled	□
Update Interval	0
Alpha	1
Robot Description	robot_description
TF Prefix	
Links	
Link Tree Style	Links in Alphabetic Order
Expand Link Details	□
All Links Enabled	☑
arm_base_link	
arm_joint1_frame1_...	☑
arm_joint1_servo_link	☑
arm_joint2_frame1_...	☑
arm_joint2_frame2_...	☑
arm_joint2_servo_link	☑
arm_joint3_frame1_...	☑
arm_joint3_frame2_...	☑
arm_joint3_servo_link	☑
arm_joint4_frame1_...	☑
arm_joint4_frame2_...	☑
arm_joint4_servo_link	☑
base_link	
gripper_active_link	☑
gripper_link	
gripper_mimic_link	☑
gripper_servo_link	☑

Type: Orbit (rviz)

Current View	Orbit (rviz)
Near Clip ...	0.01
Target Fra...	<Fixed Frame>
Distance	0.466662
Yaw	1.28534
Pitch	0.534797
Focal Point	0.073958; -0.032...

arm_joint1_frame1_joint	-0.83
arm_joint2_servo_joint	0.10
arm_joint3_frame1_joint	1.06
arm_joint4_frame1_joint	0.00
gripper_link_joint	0.00
gripper_joint	0.54

ROS Time: 1446640379.14 ROS Elapsed: 1655.93 Wall Time: 1446640379.21 Wall Elapsed: 1655.91 □ Experimental

Figure 2: Later during the project, you will create a 3D interface to control your arm, using the standard ROS tools like RViz, pictured here.

Part II

Arduino RC servo control

The RC servomotor (Fig. 3) is a unit consisting of a small electric motor driving a gear train. It often uses a potentiometer to measure angular position measurement. Often such a unit is small and cheap although more expensive designs using metal gears and digital control circuitry are also available. You will use such digital servo-motors in ROCO224, next semester.

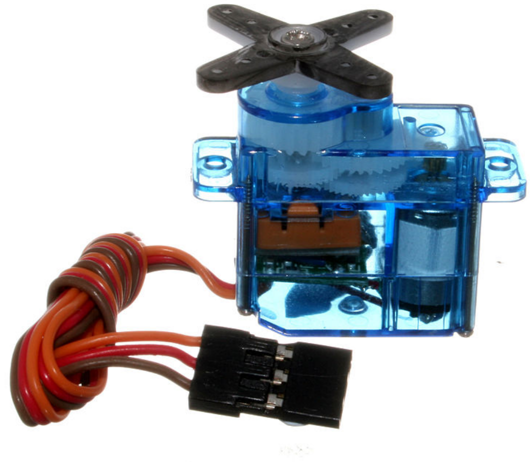


Figure 3: Micro-servo motor

The RC servo generally incorporates circuitry that implements position feedback control (Fig. 4). Output position compared to the commanded target position. This gives rise to error signal. The error drives the electric motor in appropriate direction. When it becomes zero the servo stops moving and reaches equilibrium. A RC position servomotor usually sets output target angle as specified by control pulse width as show in Fig. 5.

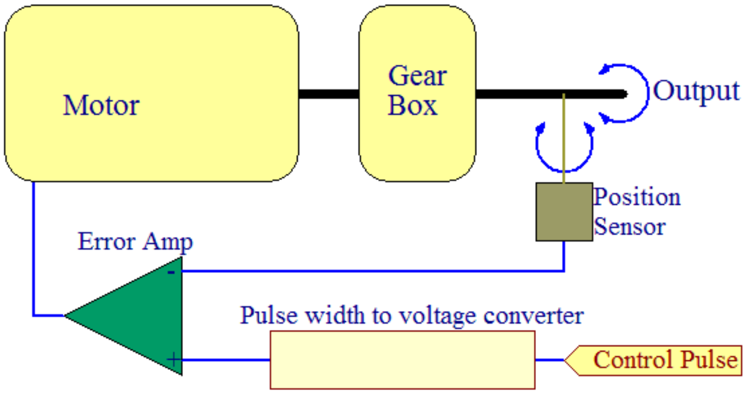


Figure 4: RC servo control showing feedback pathways

A RC servomotor typically has three connections:

- Normally the black wire is ground, the red wire is connected to V_{ref} (5V) and the white (or yellow) wire is the control pulse input.

- If this is not the case on your servo, check its datasheet
- You will need pins to connect the female plug to the female Arduino outputs

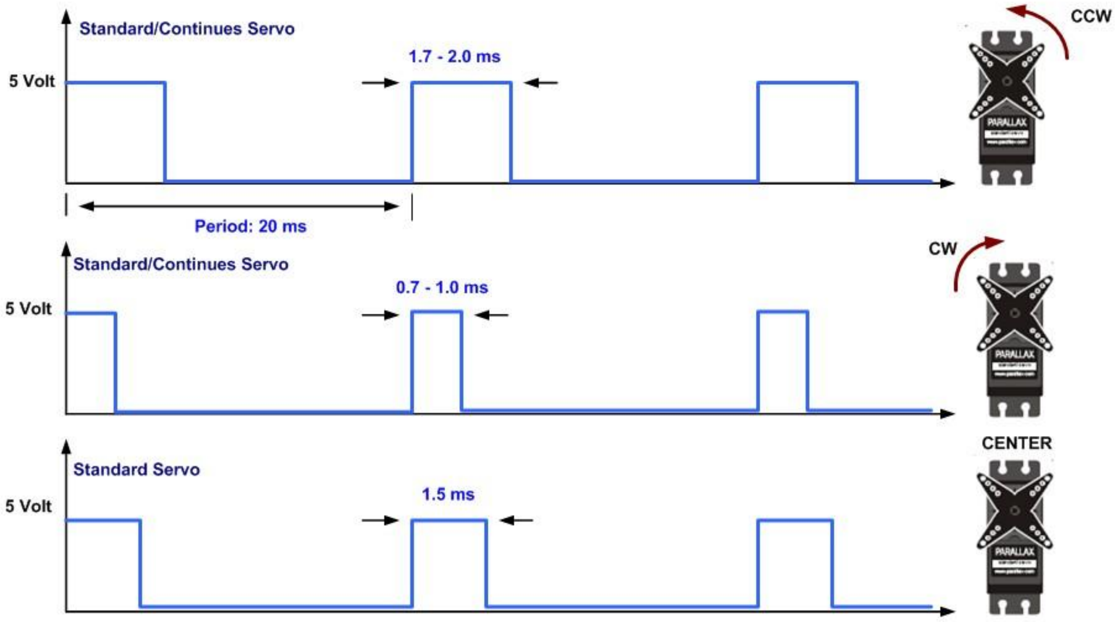


Figure 5: RC servo operation by pulse width control

Step 1 – Control an RC servo

1. Attach the RC servo power connections to the Arduino power output pins
2. Attach the control input to a PWM output pin on the Arduino (e.g. port 9)
3. Write a simple program that runs the servo to generate an output movement back and forward over its entire range so its movement follows a low frequency sine wave of frequency around 0.2Hz

Step 2 – Control the servo with a potentiometer

1. Wire the provided potentiometer to one of the Arduino analog port. Write a program that reads the value of the potentiometer and writes it onto the serial port.
2. Use the potentiometer to rotate the servo-motor.

Step 3 – A robot arm mock-up

1. Using pieces of cardboard, create an arm and attach it to the motor shaft (you can draw inspiration from the picture below, or come up with your own ideas)

2. Estimate the torque of the servo-motor by hanging a fixed mass to the arm, at increasing distance of the motor shaft. Document the process, and check your estimate using the servo-motor datasheet (you can find it easily online)

3. Modify your arm to insert the second servo motor

4. Write a program that control both servos with the two provided potentiometers.

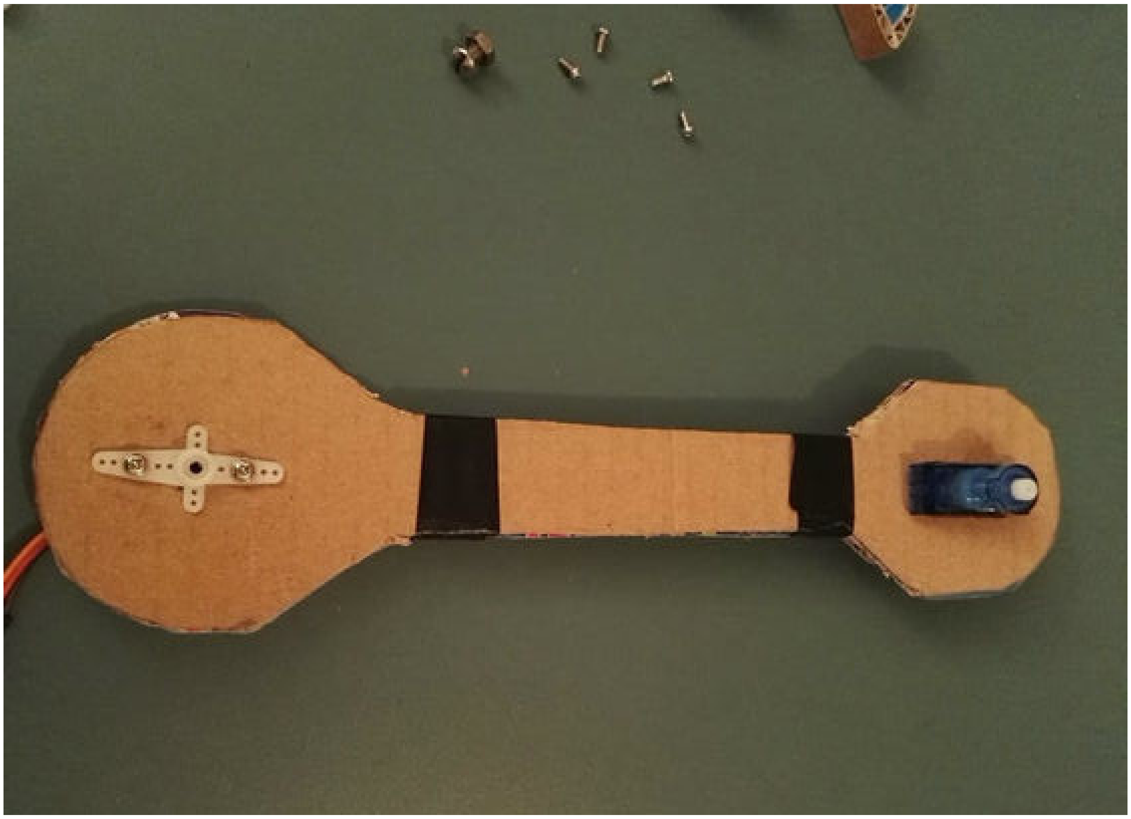


Figure 6: A possible assembly of 2 servos onto a cardboard arm. Taken from `http://www.instructables.com/id/CARDBIRD-the-Cardboard-Robotic-Arm/`

Note

Document your findings with photographs and a short video of RC servo operation.