Individual Component Analysis: Controller

1. Project Description and Individual Component Justification

Anteater Dynamics is working on a low-cost 7 degree-of-freedom robotic arm using proprietary components supplied by the team's industry sponsor ROBOTIS Inc. The motivation behind the project was for hobbyists and robotics enthusiasts to have a middle ground between low-fidelity budget options and cost prohibitive high-fidelity options. One of the main requirements for this project is to keep the total cost under $1000. Below is a grey box diagram of the major functions the robotic system aims to achieve.

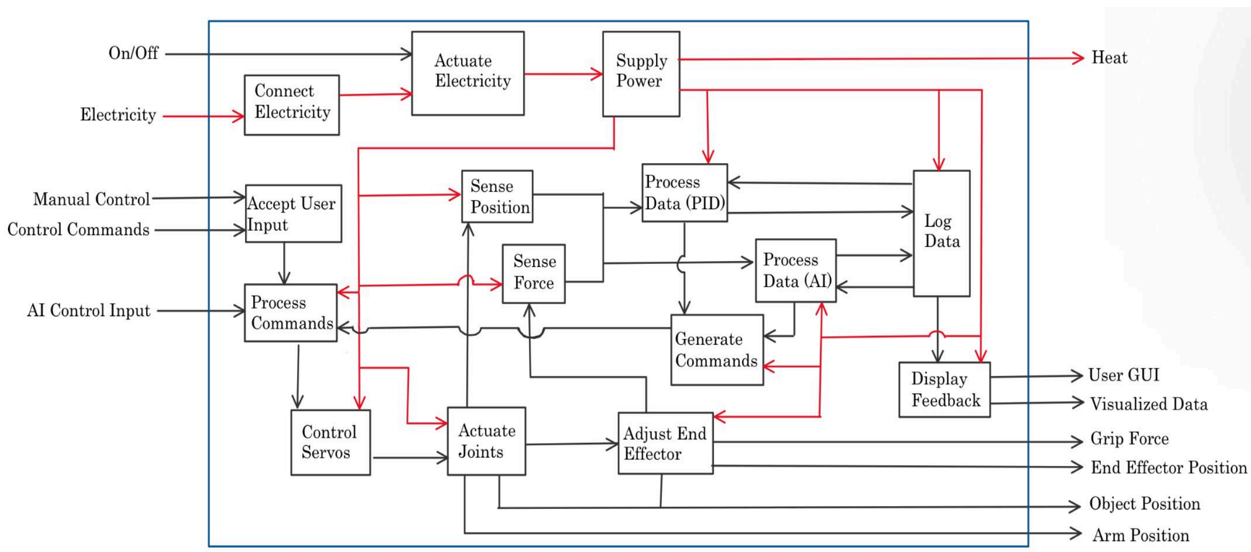

Fig 1. Grey Box Diagram of the Robotic Arm [1]

The controller plays in integral role as this component will receive the manual control and control commands from the user, and process those commands to help achieve the final end effector position through mathematical computation (inverse kinematics), adjust the grip force of the end effector by reading load cell data, as well as displaying data through a GUI. The type of controller is critical as it needs to be able to handle executing all the different functions described in the grey box diagram shown above.

2. Functional Requirements and Other Considerations

The functional requirements of the robotic controller are as follows:

FR-Cr1: The controller shall control six servos at once

FR-Cr2: The controller shall be able to perform mathematical calculations such as forward or inverse kinematics

FR-Cr3: The controller shall process input data from the load cells

While not necessarily requirements there are other considerations and suggestions that should be taken into account. The end-product is aimed toward hobbyists and other robotic enthusiasts as well as researchers. With this broad end-user audience, the controller should be relatively simple to use, or should not have too high of a learning curve. Additionally, cost shall be considered when choosing a controller as the main goal of this project is to keep the final product price to be under $1000.

3. Concepts or Options

From the DYNAMIXEL quick start manual and selection guide, the DYNAMIXEL servos can be controlled three ways, by an Arduino-compatible microcontroller, a single-board computer (SBC), or directly from a PC. [2, 3] For this component the three options will be

1. A generic Windows or macOS laptop with at least 8 GB of RAM.
2. A Raspberry Pi 4 Model B (2 GB RAM) SBC.
3. The OpenRB-150 controller.

For direct PC control, the specifications for the PC will be based on the Lenovo IdeaPad Slim 3i 15.6" available at Costco [4]. This is a newer generation of a laptop currently used by a project member and can operate a variety of different applications including WSL, SolidWorks, and MATLAB. Control of the DYNAMIXEL servos can be achieved via the DYNAMIXEL Software Development Kit (SDK) which allows the DYNAMIXEL to be programmed in several different languages. A diagram of how the PC is connected to the DYNAMIXEL is shown in the figure below.

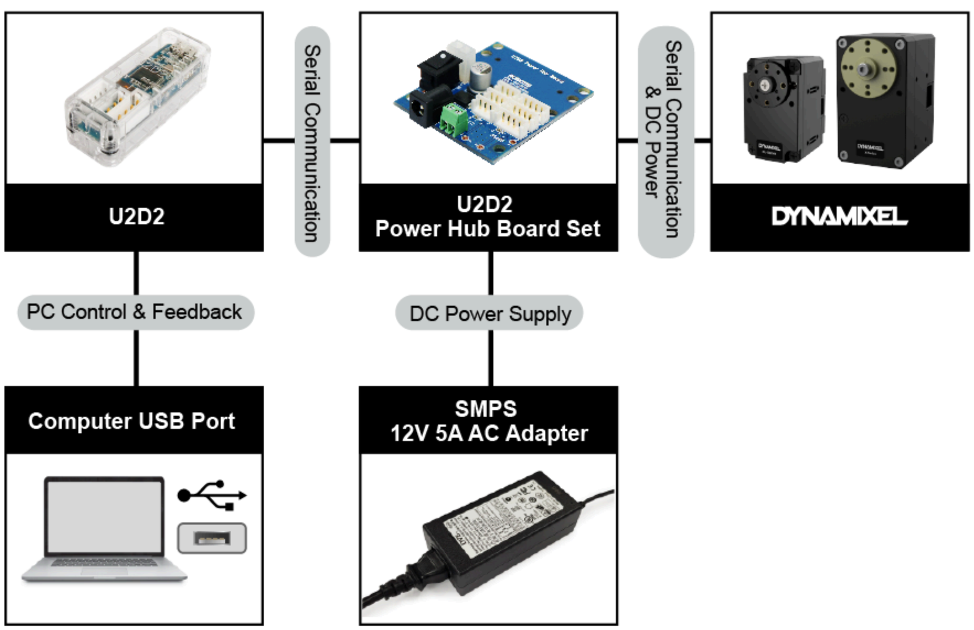

Figure 2. Flowchart highlighting how a PC is connected to the the DYNAMIXEL servos [3]

As seen in the figure, three additional components are necessary for communication between the PC and the servos. The U2D2 converts USB to either TTL and/or RS-485, depending on which model servo is being used. The U2D2 power hub is what connects directly to the DYNAMIXEL and is connected to an external power source.

The Raspberry Pi 4 Model B is an SBC that runs on the Linux operating system. Similar to direct control from a PC, the Raspberry Pi can program the DYNAMIXEL through the Dynamixel SDK, and the flowchart shown in Figure 2. But instead of a Computer USB port, the Raspberry Pi USB port will be used.

Lastly, the OpenRB-150 is an Arduino-compatible microcontroller designed and sold by ROBOTIS. Control of the DYNAMIXEL servos can be achieved through the Arduino IDE and the Dynamixel Arduino library.

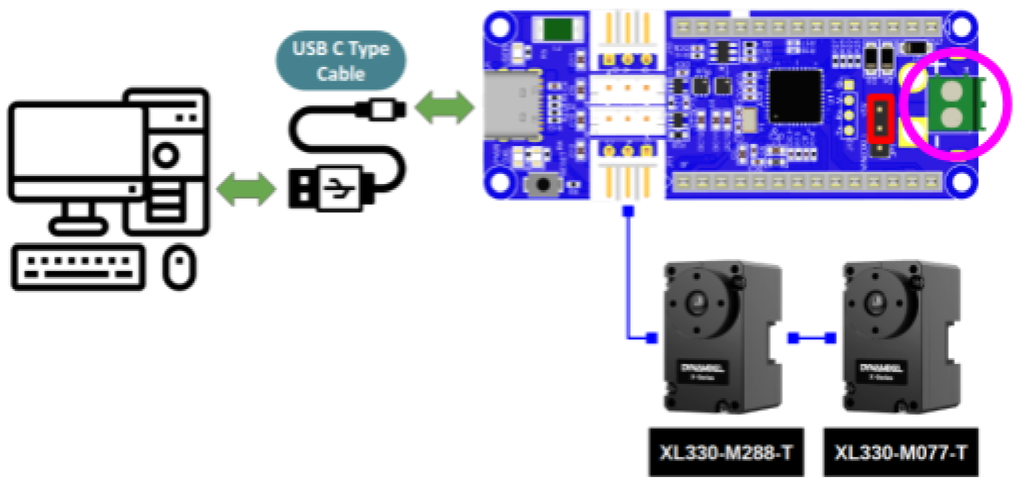

Fig 3. Diagram of OpenRB-150 Connection [3]

As shown in figure above, the OpenRB-150 is the only component needed to control the robotic arm. Note that the OpenRB-150 can be powered through an external source via the pins circled in bright pink. This functionality is needed when using more powerful servos.

4. Pros and Cons Summary

Below is a table representing a summary of the pros and cons of the three different options:

Component	Pros	Cons
Direct PC	Most Versatile	High learning curve
	Highest Computing Capacity	Not Beginner Friendly
Raspberry Pi	Portable	Most Costly Solution
	Strong computing capability	Not beginner friendly
		Limited Versatility compared to PC
OpenRB-150	Arduino-compatible (beginner friendly)	Small memory/Weak computing power
	Cheapest Option	Only 4 TTL Ports
		Only uses TTL ports

Highlighted in green is the chosen component. Further analysis how this decision was made will be discussed in the following section

5. Analysis and Justification

Rating 1-5		Robotic Arm Controller					
		Direct PC		Raspberry Pi 4 B		OpenRB-150	
Objectives	WEIGHT	Rating	Weighted Score	Rating	Weighted Score	Rating	Weighted Score
Cost	40.00%	3.00	24.00%	1.00	8.00%	5.00	40.00%
Computing Power	30.00%	5.00	30.00%	4.00	24.00%	1.00	6.00%
Compatibility/ Versatility	15.00%	5.00	15.00%	3.00	9.00%	1.00	3.00%
Ease of use	15.00%	3.00	9.00%	2.00	6.00%	4.00	12.00%
SCORES	100.00%	3.90	78.00%	2.35	47.00%	3.05	61.00%

The objectives listed in the weighted decision matrix relate to the functional requirements and other considerations discussed in section 2 of this paper. Important to note, an assumption was made that the end-user will already have a PC with similar or better specifications as the Lenovo laptop used in this analysis.

Cost is weighted as the most important factor as the goal of the project is to have a low-cost option for consumers. The OpenRB-150 gets the highest rate in this category, with a single board costing only $24.90. [5] The runner up is the direct PC control–under the assumption the user already has a PC similar in specifications to a Lenovo laptop–as the U2D2 starter pack costs $59.70 [6]. Under this assumption the direct PC controller received the rating of 3, since the cost of this method will only be the U2D2 and its accompanying Power Hub Board. The Raspberry Pi is the costliest option with an additional $44.99 for the SBC itself on top of the $59.70 cost of the U2D2 starter set [7]

Computing power is the second highest-weighted category. All the functional requirements require ample computing power and the direct control of the PC provides sufficient computing power to perform more resource intensive tasks such as calculating the proper thetas for rotation with inverse kinematics, or getting the final position of the end effector with forward kinematics. The Raspberry Pi 4 is the second highest-rated in computing power technology. As an SBC, the Raspberry Pi has a CPU which can process more data than an Arduino [8].

The next two categories carry the same weight, compatibility/versatility and ease of use. Compatibility and versatility is related to which languages and programming environments can be used in the controller. Again the PC had the highest rating since it can use all the languages supported by the Dynamixel SDK (C, C++, C-Sharp, Python, Java, Matlab, Labview, and ROS). [2, 3] While the Raspberry Pi 4 is able to use these languages as well, compatibility is limited for example, since MATLAB cannot be directly installed on and run on a Raspberry Pi. [9] Ease of use relates to how beginner friendly each controller is. While not a functional requirement, it's important to consider given the large end-user audience. The OpenRb-150 has the highest rating here since the DYNAMIXEL can only be used in the ArduinoIDE environment along with the Arduino2Dynamixel library. These limitations help beginners code basic commands to the DYNAMIXEL servos.

Specifications for Chosen Component

From the Decision Matrix, direct PC control is the chosen method for controlling the robot. The specifications for this component will be focused on the U2D2 and U2D2 Power Hub Interface devices.

The U2D2 connects to the PC via a USB Type-A cable. The ports on the U2D2 include a 3 Pin TTL and a 4 PIN RS-485, both for connecting to DYNAMIXEL. The last port is a 4 Pin UART. The maximum baud rate is 6 Mbps. [9]

The U2D2 Power hub has 3 TTL ports and 3 RS-485 to connect to the DYNAMIXEL servos. The U2D2 power hub has an operating voltage of between 3.5V and 24.0 V, with a max current of 10.0A.

References and Notes

[1] Cardona, L., "Problem Definition," *MAE 151A*, PowerPoint presentation, Irvine, CA, Feb. 4, 2025. Slide 11

[2] “DYNAMIXEL Selection Guide,” *ROBOTIS e-Manual*, Mar. 24, 2024. https://emanual.robotis.com/docs/en/reference/dxl-selection-guide/#controller--interface (accessed Feb. 27, 2025).

[2] “DYNAMIXEL Quick Start Guide,” *ROBOTIS e-manual*, 2025. https://emanual.robotis.com/docs/en/dxl/dxl-quick-start-guide/ (accessed Feb. 27, 2025).

[3] “Lenovo IdeaPad Slim 3i 15.6" Touchscreen Laptop - Intel Core 5 120U - 1080p - Windows 11,” *Costco Wholesale*, 2022. https://www.costco.com/lenovo-ideapad-slim-3i-156-touchscreen-laptop---intel-core-5-120u---1080p---windows-11.product.4000260119.html (accessed Feb. 27, 2025).

[4] “OpenRB-150,” *ROBOTIS*, 2025. https://www.robotis.us/openrb-150/ (accessed Feb. 27, 2025).

[5] “DYNAMIXEL Starter Set [US],” *ROBOTIS*, 2025. https://www.robotis.us/dynamixel-starter-set-us/ (accessed Feb. 27, 2025).

[6] “Raspberry Pi 4 Model B - 2GB DDR4,” *Micro Center*. https://www.microcenter.com/product/621439/raspberry-pi-4-model-b---2gb-ddr4?src=raspberrypi

[7] J. Cicolani and Springerlink (Online Service, *Beginning Robotics with Raspberry Pi and Arduino : Using Python and OpenCV*. Berkeley, Ca: Apress, 2018. Available: https://learning.oreilly.com/library/view/beginning-robotics-with/9781484268919/

[8] “Raspberry Pi,” *Mathworks.com*, 2024. https://www.mathworks.com/hardware-support/raspberry-pi.html (accessed Feb. 27, 2025).

[9] Y. Name, “U2D2,” *ROBOTIS e-Manual*. https://emanual.robotis.com/docs/en/parts/interface/u2d2/

[10] Y. Name, “U2D2 Power Hub,” *ROBOTIS e-Manual*. https://emanual.robotis.com/docs/en/parts/interface/u2d2_power_hub/

DYNAMIXEL Selection Guide

Direct PC/SBC:

- Requires U2D2
  - Converts USB to TTL/RS-485 Signal
- Then the U2D2 Power Hub Board Set
  - This also applies to when using the Raspberry Pi
- Power Supply

OpenRB-150:

- Needs an arduino or for our case OpenRB-150
- Power Supply

- Arduino IDE
- 4 TTL connectors

Software Tools:

- Dynamixel Wizard 2.0
  - Works on both u2d2 and OpenRB [Source]
- Dynamixel SDK (software development kit)
  - Supports a lot of different languages
  - Available on all three major OS (Linux, macOS, Windows
- Arduino Library
  - ROBOTIS has library to make coding in arduino ide easier

DYNAMIXEL SDK Guide:

- Can be used for both PCs and SBC
- Can use these languages: C, C++, C#, Python, Java, MATLAB and LabVIEW

Costs (based on the ROBOTIS Website)

Direct PC:

- U2D2 starter set is $59.70
- This will be the only cost for direct PC control
- If you do a la carte the u2d2 will cost 32.10
- The power hub will be 19.00
- 51.10 if bought a la carte

Raspberry Pi 4 Model B:

- U2D2 and then on top of that the actual raspberry pi
- From Micro-center it will cost 44.99

OpenRB-150

- Open RB itself will cost $24.90

Specs:

- 256 KB Flash Memory
- 32 KB SRAM

Can take both power from PC or external power supply

Quick pros and cons
OpenRB-150 is the CHEAPEST Option (Maybe put this in a decision matrix)
OpenRB-150 is also most limiting since it can only be coded in arduino IDE

Raspberry pi:
More powerful than openrb-150 when it comes to computing (has better specs)
Most Costly, assuming user has access to a laptop

PC:
Most versatility, can use a variety of different software programming tools
Least portable of the three. Potentially limiting as well if people don’t have access to a PC (say they only have a chromebook)