

Development and Control of Modular 5 DoF Robotic ARM using ROS

^{1st} Saksham Sangwan

Department of Electrical
Engineering Delhi Technological
University New Delhi, India

^{2nd} Vishal Sharma

Department of Electrical
Engineering Delhi Technological
University New Delhi, India

^{3rd} Karan Singh Bora

Department of Electrical
Engineering Delhi Technological
University New Delhi, India

Abstract—This paper presents the design and development of a robust 5-DoF robotic arm. Articulation of the robotic arm is achieved by linear actuators and DC brushed motors. The onboard processing is achieved by Nvidia Jetson Nano along with off-the-shelf motor drivers and microcontrollers for feedback and control. The safety of electronic components and battery management is also kept in mind while designing the sub-system. The software stack is based on Robot Operating System (ROS) to facilitate distributed computing.

Index Terms—Control system, inverse kinematics, PID, Robot Operating System

I. INTRODUCTION

In this paper, a 5-DOF robotic arm is designed and developed which is capable of lifting objects and performing intricate tasks with high precision in sensitive work environments. The arm uses a combination of linear actuators and DC motors to operate its various links and joints. The end-effector uses DC motors for proper gripping motion using a 2-claw modular and adaptive gripper. All joints and movements are made precise by the use of closed-loop control, through implementation of potentiometer and rotary encoders. At the heart of the system, we have an Nvidia Jetson Nano along with several microcontrollers (Arduino Mega) sending signals to motor drivers for motor actuation. A robust and flexible software architecture has been developed upon the existing framework of Robot Operating System. Two digital cameras have also been deployed for providing necessary visual feedback.

OpenCV in conjunction with ROS has been used to compress and collect visual data from the cameras ensuring optimal FPS. OpenCV can also be utilised for object detection to provide autonomous capabilities to the arm. A combination of forward and inverse kinematics has been implemented to ensure ease of control along with minimal computation power. The power distribution system is straightforward and efficient. A kill switch has also been implemented for safety purposes. Through proper commands and necessary modifications as per the task at hand, the arm can be used for a variety of real-world applications including assembly line in factories, astronaut assistance onboard a rover, and bomb defusal.

Our objective through this project is to develop a general-purpose robotic arm which can be utilized in various industrial as well as educational scenarios. It can also be seamlessly



Fig. 1. Final Developed 5 DOF Robotic Arm

integrated with several other sub-systems. The design has been kept modular for easy manufacturing and maintenance. The off-the-shelf products used are easily available, have proper documentation and support to speed-up the development process of the arm.

II. MECHANICAL SUB-SYSTEM

The robotic arm is designed to provide stability and keep the weight as low as possible. Carbon fibre tubes and aluminium links along with 3D-printed end-effector jaws are used. Low weight nylon gears are also used in place of mild steel, and a worm gear is attached at the base for 360 degree rotation of the arm.

TABLE I
SELECTION OF MATERIALS

Part(s)	Material	Reason for Selection
Links	Carbon Fiber	Excellent strength to weight ratio
Gripper, Joints	Aluminium	Good strength ratio and cost efficient
Worm	Mild Steel	High strength and good machinability

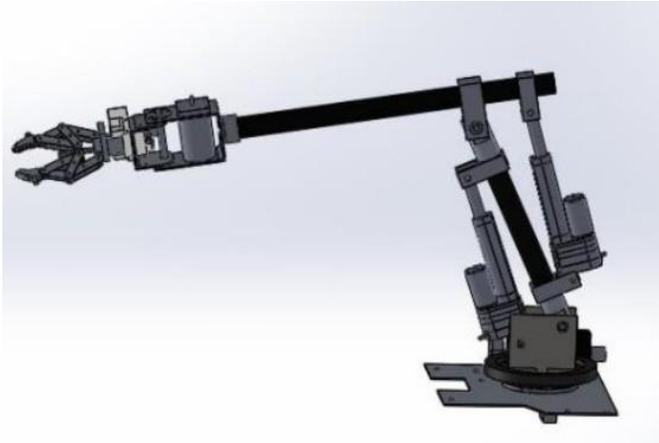


Fig. 2. Mechanical Design - Full Assembly

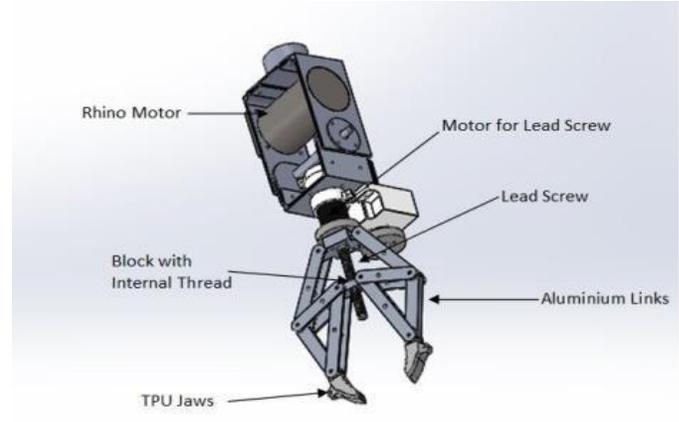


Fig. 4. Gripper Assembly

A. Link Selection

After extensive research and analysis, single tube type links and orientation capability of 5-DOF has been finalized. Bending and shear tests were the primary criteria for selecting the design and material. Carbon fiber links along with aluminum connectors have been incorporated in the design to maintain ideal strength to weight ratio. The thickness of the carbon fiber tubes was selected after several iterations using Ansys 18.

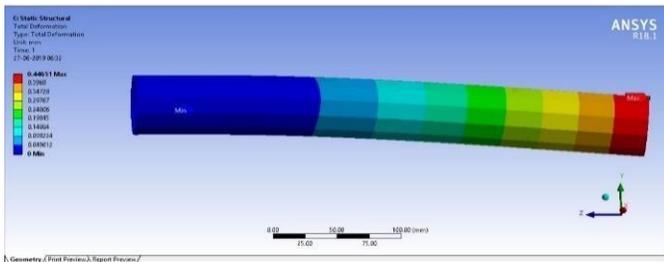


Fig. 3. ACP Analysis of Carbon Fiber Links

B. Gripper Assembly

The mechanism of our gripper is based on lead screw mechanism to increase its gripping capabilities. The jaws of the gripper are made by 3D printed TPU along with spring capabilities which will further improve its grip. Nylon gears are used in the gripper to get required motion as it provides light weight and optimum strength for low torque operations.

III. ELECTRICAL SUB-SYSTEM

Our robotic arm's electrical and electronic sub-system consists of linear actuators and DC brushed motors for actuation, microcontrollers, potentiometers for feedback, power electronics converters (DC-DC buck-boost converter and motor drivers), battery, battery management circuits and safety circuits.

A. Motor and Actuators

We have used two heavy duty linear actuators with dynamic thrust of 800 N and static load of 2,225 N with a 4-inch stroke length. These actuators have a built-in 10K potentiometer that provides us absolute position feedback. The mounting points of linear actuators have been decided so that the shoulder joint has 90 degrees range of motion and elbow has 150 degrees range. We have a worm wheel on the swivel base, to provide the 360-degree rotation of the robotic arm.

Linear actuators have higher efficiency and precision when compared to DC Brushed motors. Thus the joints that experience the most load i.e the elbow and shoulder, for them we used linear actuators. This drastically reduced the jitters and backlashes which were difficult to avoid if we used DC brushed motors. The built-in potentiometer was used to provide feedback of actuator length. DC brushed motors were used for base motor and all the end effector motion. The end effector can perform roll, pitch motion and gripping action. DC Brushed motors are not only easy to control but their controllers are low cost, require less size and easily available.

B. Control System

We have designed a custom control system to control position of linear actuators and DC brushed motors. The control system that we have designed is very simple and can be divided into following parts:

- 1) Sensors
- 2) Microcontroller Unit
- 3) Actuator/Motor Controlling Unit

We are using Independent Joint Control Mechanism that further utilizes Inverse Kinematics to move all the joints in 3D space simultaneously. Independent joint control (IJC) is a conventional and simple control mechanism for motion control of a joint. In our case we use each link of robotic arm independently, i.e., both the linear actuators and the base motor have their own control system loop.



Fig. 5. Assembled Robotic Arm

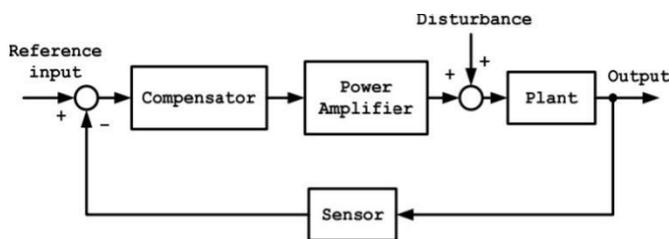


Fig. 6. IJC Mechanism Flow [1]

Variations and events like vibration in sensor mount, gear backlash and load changes cause unpredictable effects on control mechanism that ultimately can cause degradation in accuracy and performance of PID control [2].

C. Sensors

We used potentiometers to sense the position of the motor shaft and length of actuator stroke in the robotic arm. Actuators have inbuilt 10K potentiometer which requires 5V and ground input and gives analog linear output (0-5V) that is converted to digital signal when interfaced with MCU. The resolution of the potentiometer can be increased by using a MCU with higher order ADC. For example, in our case we used Arduino Mega which has 10 bit ADC, i.e., an input analog signal ranging from 0-5 V can be converted to digital signal ranging from 0-1023.

D. Microcontroller Unit

In this paper, we used Arduino Mega which has 8-bit ADC and plays a major role in processing the signal received from the potentiometer. For implementing PID algorithms efficiently, we made our own libraries that made our code efficient and fast. The PID coefficients were set by trial and error method. With its three-term functionality covering treatment to both transient and steady-state responses, proportional-integral-derivative (PID) control offers the simplest and yet most efficient solution to many real-world control problems.

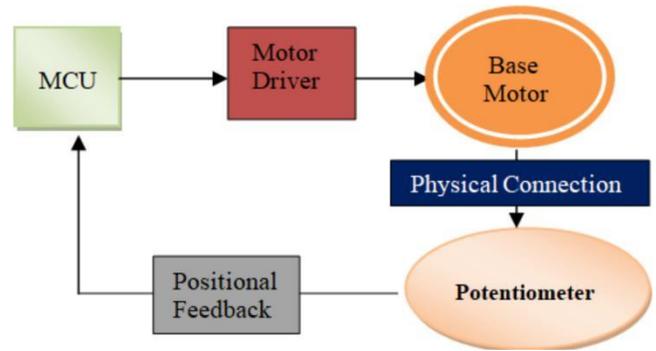


Fig. 7. Base Motor Control Loop

E. Motor Controllers

The main controlling unit of motors are motor drivers. Based on the set reference values and the signal received from the sensors, a PWM signal and direction signal is generated by the MCU, which is interfaced with I/O pins of the motor controller. Motor controllers generate a response accordingly and control the motors connected to output terminals. The microcontroller used has the ability to provide PWM signal with resolution up to 255 bits, which when interfaced with the motor driver is able to precisely control the motor's speed and position. The motor drivers are selected on the basis of current and voltage rating of motors, battery used and MCU that is interfaced. We have used Cytron MD10A in our robotic arm considering parameters mentioned above.

F. Inverse Kinematics

We have implemented a combination of forward and inverse kinematics in order to operate the Robotic Arm in the surroundings. Thus, it is easy for user to operate the arm and it does not load the CPU with complex matrix calculations as well. If we use a IK algorithm for higher DoF robotic arms then the matrix calculations get quite complex and require high computational power. Hence, we use Inverse kinematics in 3 joints i.e., base, shoulder and elbow joints. In rest of the joints, we use forward kinematics.

G. Power Distribution System

Our power distribution model is simple but efficient. We have used buck-boost converters in order to distribute power from a single 24 V battery to different components which have different voltage requirements. We have used PDBs and XT60s that increase the reliability of the system with ease in changing connections or removing faulty modules over time. For accommodating the power requirements of other systems, UBEC's (universal battery eliminator circuit) and buck-boost converters are used to convert the fed voltage to required voltage levels i.e.,

DC-DC Conversion to 12V rails and 5V rails. The 12V rails are used to isolate the 12V motors from any of the 12V electronics as they may experience malfunctions due to electromagnetic interference from the motor. UBEC's are used for their high efficiency, easy handling

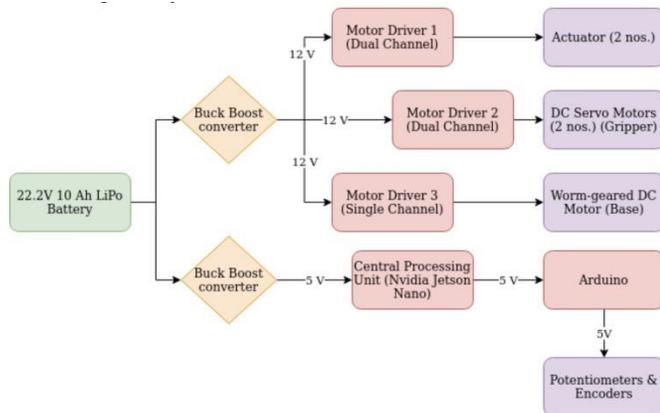


Fig. 8. Power Distribution Flowchart

and low rise in temperature while in use for a prolonged time.

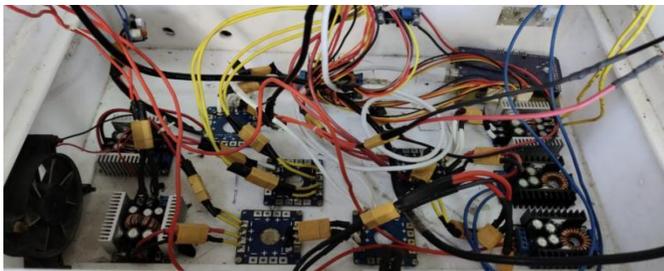


Fig. 9. Power Distribution System in Final Developed Phase

H. Safety Methods

Following safety measures have been employed to ensure safety of our electrical/ electronic system:

1) Limiters:

We have designed a special safety circuit that cuts off current from motor drivers whenever a mechanical link collides with another link with the help of a limiter. However, if the user moves the arm away from the link then the current is allowed to flow in the opposite direction. For this we have used a 10 A diode.

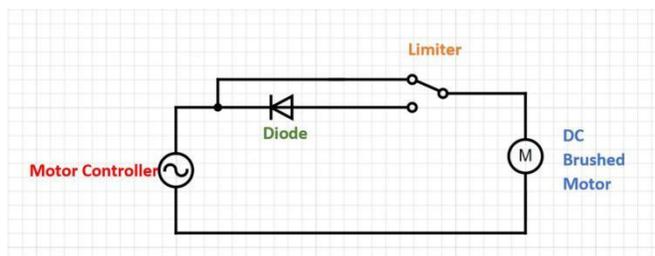


Fig. 10. Safety Circuit Schematics

2) Kill Switch:

The battery is connected to the rest of the power system through an emergency disconnect switch acting as a kill switch; easily visible and reachable in case of an emergency or power failure. Battery is also connected to low voltage buzzer alarms for notifying low levels.

IV. SOFTWARE SUB-SYSTEM

The software architecture is modular and easy to understand. The arm is remotely controlled via joystick commands that send data over a network to the Jetson Nano. First, the joystick, which is connected to a base station laptop via USB, is mapped between threshold values via a python node integrated using ROS. The next python node converts the mapped value to

```

---
header:
  seq: 9414
  stamp:
    secs: 1325530130
    nsecs: 146351623
  frame_id: ''
axes: [-0.0038758506998419762, -0.0038453321903944016, -0.0, -0.999969482421875, 0.0, 0.0]
buttons: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
---
header:
  seq: 9415
  stamp:
    secs: 1325530130
    nsecs: 146351623
  frame_id: ''
axes: [-0.0038758506998419762, -0.0038453321903944016, -0.0, -0.999969482421875, 0.0, 0.0]
buttons: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
---
header:
  seq: 9416
  stamp:
    secs: 1325530130
    nsecs: 146351623
  frame_id: ''
axes: [-0.0038758506998419762, -0.0038453321903944016, -0.0, -0.999969482421875, 0.0, 0.0]
buttons: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
---

```

Fig. 11. Joy Node Terminal Logs

an integer between the threshold PWM values that can be provided to the motor (-255/255) and sends the command to the Jetson Nano for further communication via a dedicated and customized ROS message. This message is also used to send the exact location coordinates- if the arm is several meters away from the control station. The PWM signal is then sent to an Arduino-ROS node that further communicates the signal to the motor driver accurately for the movement of the particular joint. This flow of channel is within the ROS environment which makes it easier for us to integrate multiple hardware such as Nano and Arduinos along with scripts performing different actions in Python. The commands are encoded into a ROS message and sent over a dedicated ROS topic for latency- free, fast paced communication between the base station laptop and the Nano on the arm. There are multiple advantages linked to integrating the ROS [3] environment. This saves the computation power required for each signal and makes a strong connection following TCP/IP protocol. Also, the architecture is easily scalable to multiple robots moving in distinct environments. The environment uses a CLI that is user-friendly.

A. Camera and openCV

The robotic arm consists of two webcams for visual feedback (Logitech C525 and Logitech C270). The feed is encoded

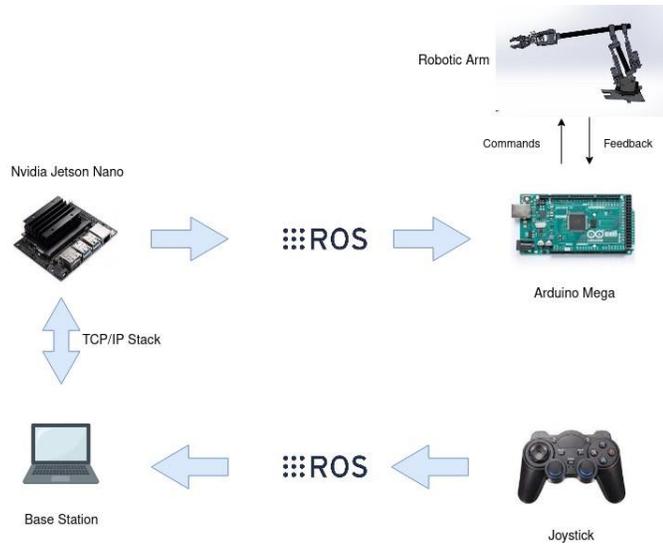


Fig. 12. Complete Control Flow of the Arm

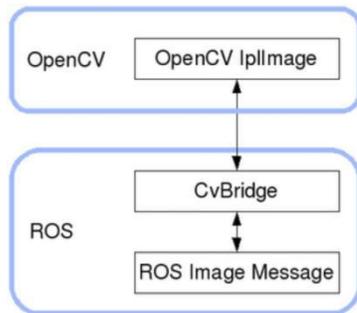


Fig. 13. Interfacing Camera and OpenCV using ROS

and then transmitted via ROS to ensure optimal FPS. The cameras are connected to the Nano via USB for ease of use. OpenCV is an open-source stable library for 2D image processing, used in a wide variety of applications. Much of ROS makes use of 3D sensors and point-cloud data, but there are still many applications that use traditional 2D cameras and image processing for which OpenCV is ideal. The encoding and decoding of the video is achieved through OpenCV.

V. CONCLUSION

In this paper, we presented the process to develop the end-to-end system for a 5-DoF robotic manipulator. The arm is made of carbon fibre and aluminium links. DC brushed motors and linear actuators are controlled by motor drivers and Arduino microcontrollers. The main onboard processor is Nvidia Jetson Nano. Robot Operating System environment is implemented for better control and stability. Webcams are implemented on two locations on the arm for visual feedback. Closed loop feedback system along with PID control is implemented and the arm is operated with a mix of inverse and forward kinematics. The robotic arm can be operated wirelessly and is optimum for mobile deployment. It has

a wide range of applications in the industry such as bomb defusal, mars/moon rover and industrial automation.



Fig. 14. Arm Performing Precision Task

ACKNOWLEDGMENT

We would like to express our gratitude towards the Department of Electrical Engineering, Delhi Technological University for their guidance and support.

REFERENCES

- [1] W. Chung, L.-C. Fu, and S.-H. Hsu, *Motion Control*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 133–159. [Online]. Available: https://doi.org/10.1007/978-3-540-30301-5_7
- [2] N. Shimkin, *Nonlinear Control Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 2886–2889. [Online]. Available: http://doi.org.https.nlic.proxy.nbdlib.cn/10.1007/978-3-540-29678-2_0_21
- [3] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “Ros: an open-source robot operating system,” in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.