

Platform Requirements

Purpose

This document defines the functional and quality requirements of the RP10 platform control software. Bulleted items are succinct representations of the requirements. Paragraphs and tables describe or justify the requirement statements.

Elicitation

RIT KGCoe professors Wayne Walter of the Department of Mechanical Engineering and Kenneth Shu of Department of Computer Engineering were interviewed and revealed important aspects of the platform:

1. Precision-oriented commands are not possible without motion sensors to verify if motion has been completed. Thus, the team should develop the most fundamental commands for robot control.
2. Robot applications will be built on top of the platform. For example, the RP10 could be integrated with a sensor network or a robotic arm.

Scope

See scope as described in the project plan.

Terminology

API	Application programming interface, a package of software that can be reused by many applications to perform the same functions
MCU	Microcontroller unit, a small board on the robot that includes a processor to run software to directly control the robot hardware and receive and send data to a PC
Drive Motor	When power is set, rotates a wheel to drive the platform forward or backward. Rotation axis is parallel to the ground.
Steer Motor	When power is set, rotates a wheel joint to allow turning. Rotation axis is perpendicular to the ground.
Motor Module	A pair of one drive motor and one steer motor.
Idler Module	A wheel that can spin but is never powered and is installed simply as a support fixture.
Motor Bay	A grouping of motor modules such that some parameters set on one motor module in the group will apply to the others. The wiring of electronics dictate the platform's bays.

Overall Description

Our product provides user-driven or programmatic control of the robotic platform. The associated hardware was created by combining the work of two previous senior projects. The P08201 team built the RP10 frame and electronics while the P08205 team built the two available motor modules, originally for an RP1 robotic platform. Our system will communicate with the robotic platform and enable the user to give basic commands to the robotic platform.

Software will be created in multiple parts:

- Graphical or textual control software on the PC
- An API, the foundation for robot control through PC software
- MCU software, an intermediary between the API and the robot hardware

Features

Motor Control

Multiple facets of motor control are described below. The platform includes 2 RP1 motor modules and 2 idler modules in the current configuration.

Motor Variability

The platform hardware can be installed with a varying quantity of motor modules. For example, one could install 4 motor modules and no idler modules for 4-wheel drive in the future.

- The RP10 platform software will support control of 1 to 4 motor modules.

The platform software does not have the capability to discover the number of motor modules at runtime.

- The user will specify the quantity of motor modules (1 to 4) installed in the control application.

Communication

- The platform software will send to and receive commands from the MCU via a serial cable connection.

Bay Configurations

Linear Movement

Linear motor control is based on bays, which are determined by the physical locations of the motors on the platform. Two sides share linear movement power and direction, so while the two sides can act independently, the individual motors cannot. The user will be able to give commands to each bay 'forward' and 'backward', but the platform direction will be determined by the angle of the wheels at the time. Because the wheels are able to rotate 360 degrees, 'forward' and 'backward' do not reflect the direction the platform will travel, but the two different linear rotations that the wheels are capable of.

When giving commands, the robot will rotate the wheels until told to stop. No distance measuring will take place for the basic commands. It will be like turning knob, more power makes the wheel turn faster, less makes it turn slower. The motor will continue to turn until told to stop turning.

More complex commands will execute a set of the basic commands in order to perform a task. Rotating the wheels a given amount, for example, will involve tracking how fast the wheel is turning, how far it has already turned, and giving the command to stop at the appropriate time.

Rotation

The rotation commands are capable of non-simultaneous individual motor control. The direction of the rotation is set on a 'bay' basis, while the power to each motor is set individually. So if two motors in the same bay want to turn different directions they are not able to do so at the same time. However, because the power is based on the individual motors, we can turn the first motor in the bay one direction, change the direction setting, and then turn the second motor in the other direction.

Rotation commands will work the same as linear commands. The basics will simply provide the rotation motor with a given power setting until told to do something else. More complex commands will utilize a set of the basic commands.

- The RP10 control software will control the platform in accordance to the electronic bay configurations.

Power Setting

Each motor is associated with a proportional power level between 1 and 100 percent. The power level determines the relative speed of the wheel or joint connected to the drive or steer motor.

- The proportional power of each installed drive motor will be set to an integer between 1 and 100 percent to sustain motion until stopped.
- The proportional power of each installed steer motor will be set to an integer between 1 and 100 percent to sustain motion until stopped.

Direction setting

Each motor spins a component in a specific direction independent of the power level. As we can't know how the wheels are oriented when they start up, directions have to be vague. Positive direction is just the opposite spin direction (could be clockwise or counter-clockwise depending on perspective) of the negative direction. The only requirement is that both directions are supported. A user will have to play with the robot to determine how the directions relate to the robot's physical configuration.

- Each installed drive motor will be set to spin either in positive or negative direction relative to the wheel rotation axis.
- Each installed steer motor will be set to spin either in positive or negative direction relative to the wheel joint rotation axis.

Motor stop

Once a motor power is set, it will sustain motion until it is explicitly stopped.

- Each installed drive motor can be individually stopped by setting power to zero.
- Each installed steer motor can be individually stopped by setting power to zero.
- The platform can be emergency-stopped, reducing the power of all motors to zero.

Diagnostics

Battery voltage reading

The platform includes 2 sets of 6 light-emitting diode (LED) for a person to estimate the relative charge in the batteries.

- The MCU software will read the battery voltage.
- The API will support retrieval of the current battery level on the platform in volts.
- The API will provide the maximum battery level on the platform in volts.
- The control application will display the current level of battery power (to be determined exactly what units).

Encoder reading

Encoders are hardware components that translate rotational motion into electrical signals. A relative encoder indicates how far the wheel has turned. An absolute encoder indicates the current angular position of the wheel. Each motor module includes an encoder. When the document was written, the installed relative encoders were not functional. However, fully operational encoders could be installed in the future.

- The MCU software will read the value of each encoder.
- The API will support retrieval of each encoder's current value.

Reliability

The team has to implement some reliability measure as successful communication between the PC and the MCU is not guaranteed.

Heartbeat Ping

If the RP10 was moving forward and the connection to the PC was cut, the RP10 would continue moving until someone turned off the switch on the platform. In order to prevent this scenario, the PC will ping the RP10 with message at periodic intervals, and the RP10 will acknowledge with a message to the PC.

- If the RP10 MCU does not receive a heartbeat message from the PC in the expected time frame of x seconds, the MCU will terminate any ongoing operations and turn off.

Extensibility

The MCU includes ports for external input and output. One could connect any hardware to these ports to communicate with the MCU. These ports allow extension of the platform for robot applications.

- The MCU will read external input ports.
- The API will allow retrieval of the values of external input ports.
- The MCU will write to the external output ports.
- The API will allow one to set the external ports' output values.

Interfaces

User interfaces will be provided for both text and graphical formats. All commands will be available in both, but the GUI will provide more monitoring of each motor's state.

Graphical

The GUI will encompass a variety of user inputs. Commands will be organized by how they affect each bay or motor. Buttons, input fields, and keyboard controls will be provided. It will cover the entirety of the API.

Text

A few simple commands will be available for fast command-line control.

Operating Environment

API

The control application and any other API-dependent software will require Windows XP or higher with .NET Framework 3.0 or higher. For development, Visual Studio 2008 is recommended.

Platform

The platform will be connected to the PC running the control software by serial cable. It is also assumed that the platform is on a surface that allows it to move freely. Placing the robot on a very uneven, or low friction surface will result in little or erratic movement. The platform cannot detect these surfaces will at runtime.

Associated Documentation

- A user manual will describe how to use the platform control software and the API.

Assumptions

Wheel Orientation

The software cannot yet determine the initial direction that the wheels are pointing. The user must orient the wheels to comply with their intended usage of the robot.

External Interface Requirements

Microsoft Robotics Development Studio

Using MRDS, an engineer could create an application to control an RP10 simulation. When satisfied with the simulation's results, they could switch the application to target the real platform.

- The API must be integrable with the MRDS simulation environment.