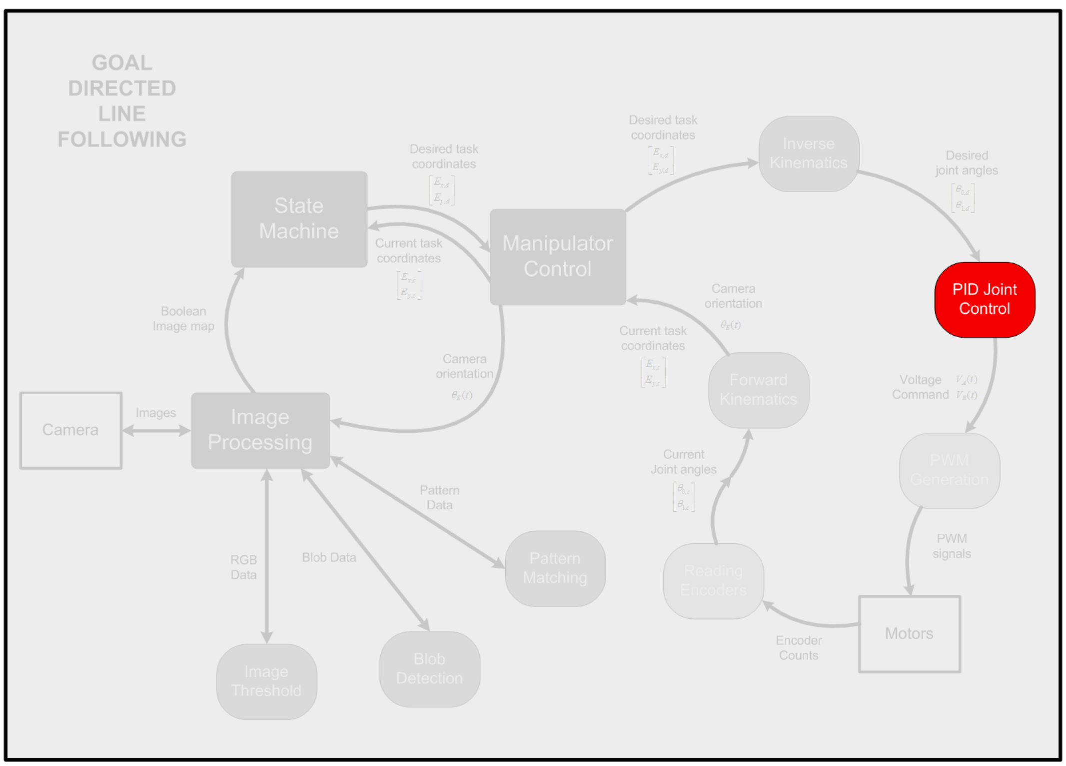
PID Position Control

Topics Covered

- QNET Mechatronic Systems position control
- Proportional-integral-derivative (PID) compensator
- Qualitative tuning according to specifications

Prerequisites

- The QNET Mechatronic Systems is set up according to the Quick Start Guide.
- Inverse Kinematics laboratory experiment.
- Forward Kinematics laboratory experiment.



1 Background

In most systems, there are two types of control methodologies - open loop control and closed loop control. In the former, the control effort is proportional to the output variable. An example of this system is turning the air conditioning system in a car ON/OFF manually. However, to automate this system, a feedback from the system can be used to modulate the control effort and bring the output variable to a setpoint or the desired value. An example is automated air conditioning, in which, a desired temperature value is provided. The system applies high cooling initially, but reduces the cooling effort as the temperature approaches the desired value.

A variety of closed loop control systems can be used, such as PD control, PI control, PID control, lead-lag compensation etc., which depend on the nature and stability of the system to be controlled (referred to as the *plant*) as well as disturbances in the environment. In case of the air-conditioning system example in a car, the plant refers to the car's air-conditioning system, and the disturbances can include the opening/closing of the car windows/doors, external weather, the number of people in the car etc.

In case of the QNET Mechatronic Systems, a feedback controller is used to drive the manipulator to a desired position, while using feedback based on the actual position/speed of the manipulator.

1.1 Response Characteristics

A typical response to a desired step signal of R_0 at time t_0 is shown in Figure 1.1. The maximum value of the response is denoted by the variable y_{max} and it occurs at a time t_{max}. The percent overshoot is found using

$$PO = \frac{100\,(y_{max} - R_0)}{R_0}. \tag{1.1}$$

From the initial step time t_0, the time it takes for the response to reach its maximum value is

$$t_p = t_{max} - t_0. \tag{1.2}$$

This is called the *peak time* of the system.

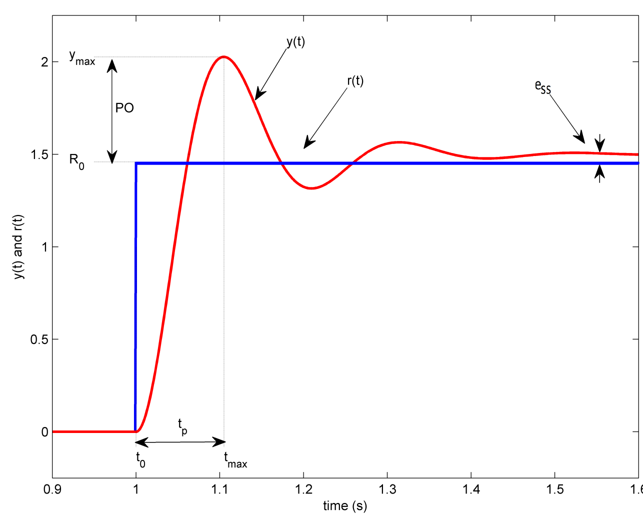


Figure 1.1: Typical response to a step input

Lastly, the difference between the desired value R_0 and the actual value once the response settles, is called the steady state error e_{ss}.

1.2 Proportional Compensation

A proportional compensator drives the plant based on the difference between the current position of the system and the desired position. This contribution of this difference to the control response is tuned by the *proportional gain* k_p which can be found either experimentally or calculated based on system requirements such as rise time. Greater proportional gain will result in a system with a shorter *rise time*, that is, the time needed for the system to reach the desired position. However, since the system is constantly accelerating toward the set point, large proportional gains will generally lead to a system with large overshoot and a slow settling time, and oscillations about the desired set point for a long time.

1.3 Derivative Compensation

To deal with the overshoot and oscillation caused by a proportional compensator, many systems implement a derivative compensator in parallel. This compensator drives the plant based on the rate of change of the position (or velocity) of the system. As with proportional control, this derivative is magnified by the *derivative gain* k_d. The derivative compensator effectively acts as added damping in underdamped systems. To improve the stability of systems with derivative compensation, low-pass filtering is often added to prevent spikes in the derivative component of the compensation due to signal noise.

1.4 Integral Compensation

In many cases, the combination of proportional and derivative gains will result in a system that does not settle sufficiently close to the setpoint. In this case, and integral compensator may be added. This compensator drives the system based on the integral of the error over time magnified by the *integral gain* k_i. This component of the controller increases the longer the system remains far from the setpoint. As the integral gain responds to accumulated errors from the past, it can cause the current value to overshoot the setpoint value.

1.5 PID Control

The proportional, integral, and derivative control can be expressed mathematically as follows

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{de(t)}{dt}. \tag{1.3}$$

The corresponding block diagram is given in Figure 1.2. The control action is a sum of three terms referred to as proportional (P), integral (I) and derivative (D) control gain. The controller Equation 1.3 can also be described by the transfer function

$$C(s) = k_p + \frac{k_i}{s} + k_d s. \tag{1.4}$$

The functionality of the PID controller can be summarized as follows. The proportional term is based on the present error, the integral term depends on past errors, and the derivative term is a prediction of future errors.

Attempts to implement such a PID controller may not lead to a good system response for real-world systems, because measured signals always include measurement noise. As described in 1.3, a low pass filter is used to suppress measurement noise. The combination of a first order low-pass filter and the derivative term results in a high-pass filter $H(s)$, which is used instead of the direct derivative.

A standard methodology in tuning the system gains manually consists of the following steps:

1. Set the derivative and integral gains to 0, and gradually increase the proportional gain till sustained oscillations are observed.
2. Increase the derivative gain gradually till the oscillations disappear and the system is critically damped. Any further damping should increase the rise time.

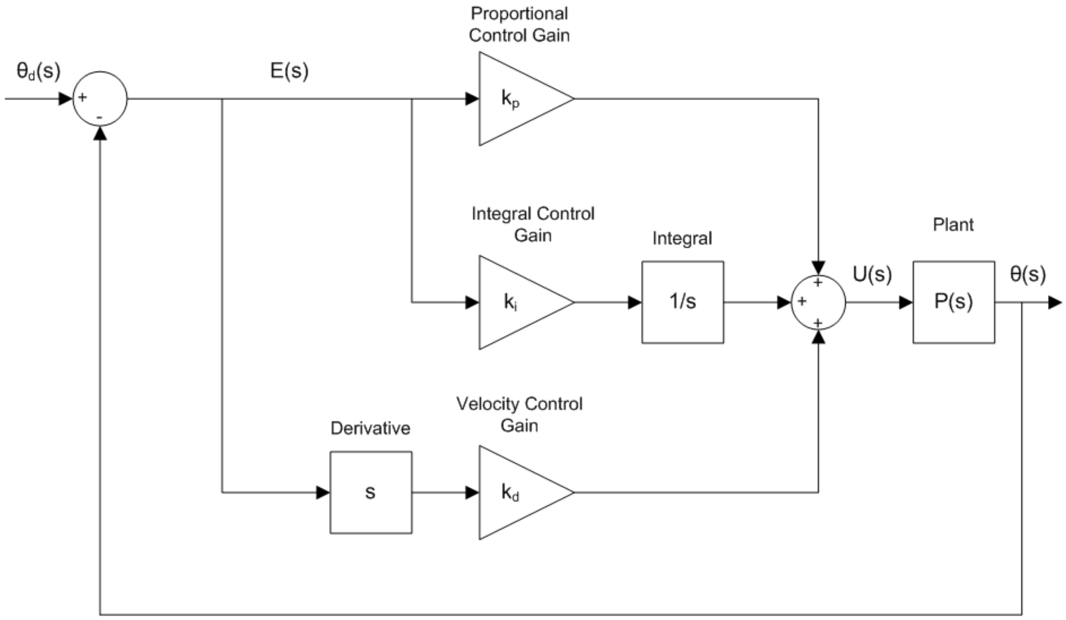


Figure 1.2: Block diagram of PID control

3. Increase the integral gain until the steady state error falls within the desired error threshold.

This process is often iterative, especially when an integral term is used. The proportional and derivative gains may have to be tuned to compensate for the addition of the integral term. Retuning may also be required if the sampling rate changes, or the nature of disturbance in the system changes (for example, the damping in the joints).

1.6 Voltage Saturation

In proper control system design, the voltage commanded by the controller should never exceed the maximum voltage that can be handled by the system, in this case, being the QNET Mechatronic Systems motors. Note that these motors have a nominal voltage of $\pm$18 V. Ensuring this condition will provide complete control authority over the system's dynamics. The QNET Mechatronic Systems implements a dynamic voltage saturation algorithm, that does not allow the commanded voltage to be greater than 12 V for more than 0.5 s at a time. The commanded voltage is otherwise saturated at $\pm$18 V.

2 In-Lab Exercises

2.1 PID Controller

Open `Mechatronic Systems.lvproj`, and under `Quanser ELVIS RIO | Subsystems`, open `PID Joint Position Control.vi`. Within the loop labelled PID Joint Position Control Loop, open the sub-VI `PID.vi`, which should be similar to Figure 2.1. Compare it to Figure 1.2. Note that this VI multiplies the error by the proportional gain, the integral of the error by the integral gain, and the derivative of the error by a derivative gain. Also note that the Integrator block must be added after the integral gain. If the integrator is added before, it will continue integrating the errors even if the integral gain is 0, which will cause large control commands when the integral gain is raised to a non-zero value for the first time.

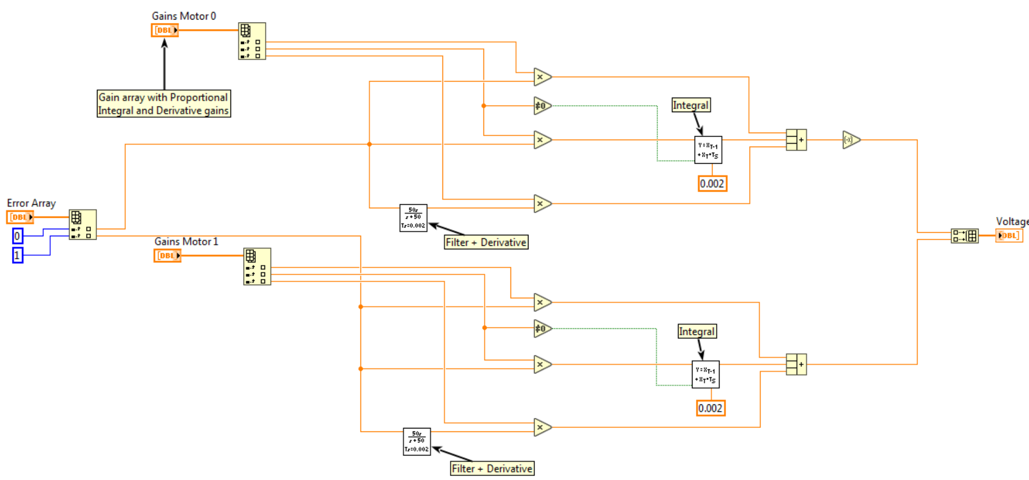


Figure 2.1: Applies a step voltage and displays measured and simulated QNET Mechatronic Systems speed.

2.2 Proportional Control

1. How does a proportional gain k_p affect the response of a system, and how do you expect the response to change when increasing it?

2. Making sure that all the gain dials are set to 0, run `PID Joint Position Control.vi`. Once the Calibration bar is full, hold the arm connected to motor 1 against the hard-stop between the arms, preventing any motion. While holding the arm so, increase the k_p gain gradually to 1.5. Increase this gain further to 2.5 and then 5, and comment on the response.

   Note: If this is the first time any of the VIs is being run, calibration might take up to 10s.

3. Given the specification that the controlled system has a peak time of less than 0.5s, find a proportional gain value that gets a response as required. Is it practical to use purely proportional control? Why or why not?

2.3 Derivative Control

1. How does the derivative gain affect a system's response, and how do you expect the response to change when increasing it?

2. While still holding the arm connected to motor 1 as in the previous step, and with the k_p gain set to 5, increase k_d to 0.25, 0.65 and then 1.2, and comment on the response.

2.4 Integral Control

1. What is the steady state error in your graph from the previous question with k_p and k_d gains 10 and 1.5 respectively? How can this be eliminated and describe how/why it works?

2. While still holding the arm connected to motor 1 as in the previous step, and with the k_p and k_d gains set to 5 and 0.65 respectively, increase k_i to 0.5, 1 and then 1.5, and comment on the response.

2.5 Tuning to specifications

1. Fine tune the gains further if needed and verify that the specifications are all met (less than 5% overshoot, 0.5s peak time and $\pm$ 2% steady-state error).

2. Keep the controller running with your tuned gains. What is the peak voltage in the Voltage Command (V) graph? Keeping subsection 1.6 in mind, is this problematic?

Although the voltage is saturated at a value much lower than the commanded voltage from the controller, this is acceptable in such a situation. In this lab, the desired and actual set-point vary by $\pm$1 rad. The QNET Mechatronic Systems manipulator is at rest once it approaches steady-state. When the desired set-point changes, an abrupt speed requirement causes the derivative compensator to output a large value, which leads to the voltage spike. This can be avoided by supplying a trajectory of points instead, which in turn remove the large step difference between the desired and actual set-points initially. Hence, path planning algorithms 'soften' the step signal, in turn removing the voltage spike. This is the case with the path planning algorithms used in the rest of the labs.

Quanser Inc.
119 Spy Court
Markham, Ontario
L3R 5H6
Canada
info@quanser.com
Phone: 1-905-940-3575
Fax: 1-905-940-3576

Printed in Markham, Ontario.

For more information on the solutions Quanser Inc. offers, please visit the web site at:
http://www.quanser.com