**Pneumatic Cylinder position control**

**Objectives:**

In this lab, as a group:

* Design and implement a closed-loop control system to regulate the position of a pneumatic cylinder carrying a payload.
* Investigate the effects of tuning the PID controller gains on system response and performance characteristics such as rise time, settling time, and overshoot.
* Compare the controller’s performance when actuating the cylinder from the rod side versus the piston side.

**Introduction**

In this experiment, you will develop and implement a closed-loop position control system for a pneumatic cylinder. The system uses a solenoid-operated binary valve, controlled via Simulink and an Arduino control board.

Through this activity, you’ll gain hands-on experience with pneumatic actuation, real-time control, and PID tuning, while analyzing how system dynamics change under different operating conditions.

**Background**

**Pneumatic Cylinders**

Pneumatic cylinders are mechanical actuators that convert the energy of compressed air into linear motion. They are widely used in industrial automation systems for tasks such as lifting, clamping, pushing, and positioning. A typical pneumatic cylinder consists of a piston that moves within a cylindrical housing. Compressed air applied to either side of the piston generates a force that causes it to move, extending or retracting the rod attached to it.

One important characteristic of pneumatic systems is their rapid response and relatively simple construction. However, the compressibility of air and the non-linear behavior of pneumatic components can make precise position control more challenging compared to hydraulic or electric systems. As a result, closed-loop control systems, such as those using PID (Proportional-Integral-Derivative) controllers, are often implemented to improve positioning accuracy and stability.

**5-Port, 3-Position, Closed-Center Solenoid-Operated Binary Valve**

The pneumatic cylinder in this lab is actuated using a **5-port, 3-position, closed-center solenoid-operated valve**. This type of directional control valve plays a crucial role in directing airflow to and from the cylinder. Here’s a breakdown of its operation:

* **5 Ports**:  
  The valve has five ports — two for connecting to the pneumatic cylinder (one to the rod side and one to the piston side), two exhaust ports, and one inlet port for compressed air.
* **3 Positions**:  
  The valve can switch between three different positions:
  1. **Extend Position**: Connects the compressed air supply to the piston side of the cylinder while exhausting the rod side.
  2. **Retract Position**: Connects the compressed air supply to the rod side while exhausting the piston side.
  3. **Closed-Center Position**: All ports are blocked, isolating both sides of the cylinder and stopping any motion.
* **Closed-Center Configuration**:  
  In the center position, the valve prevents airflow to and from the cylinder, effectively holding the piston in its current position. This is useful for maintaining a steady state or pausing motion without venting pressure.
* **Solenoid-Operated Binary Control:**  
  The valve is actuated by electrically controlled solenoids, each functioning as a binary (on/off) actuator. In this lab, the solenoids receive control signals from an Arduino running a Simulink-based control program. While the solenoid actuation itself is digital, a closed-loop feedback system — using a position sensor and a PID controller — continuously adjusts the cylinder’s position by rapidly switching the valve states at appropriate intervals. This is accomplished through pulse-width modulation (PWM) control, effectively modulating the average airflow to the cylinder and enabling smoother, more precise position regulation.

**Pneumatic System**

Figure 1 illustrates the configuration of a pneumatic cylinder with a payload connected to a 5-port, 3-position, closed-center directional control valve. The valve features five ports: a compressed air supply port (P), two actuator ports (A and B) connected to the piston and rod sides of the cylinder, and two exhaust ports (R and S). In its three positions — extend, retract, and closed-center — the valve directs compressed air to either side of the cylinder or blocks all ports to hold position. Solenoid actuators shift the valve between these positions in response to binary control signals from the Arduino, enabling position control of the cylinder within a closed-loop system.



Figure 1: Pneumatic Cylinder with 5-Port, 3-Position, Closed-Center Solenoid-Operated Valve

**Procedure**

1. **A close up of a machine

   AI-generated content may be incorrect.Power Up the System**
   * Connect the power cord for the Pneumatic Educational Platform and turn on the **Main Switch** located at the back.
   * Turn on the **Compressor Toggle Switch** on the front of the platform to start filling the air tank.
2. **A close-up of a machine

   AI-generated content may be incorrect.Verify Pneumatic Connections**
   * Check that the pneumatic circuit is connected correctly according to the provided schematic diagram (see **Figure 1**).
   * Use the **Filter-Pressure Regulator** to set the system pressure to **75 psi**.
3. **Select the Speed Sensor**
   * A close-up of a circuit board

     AI-generated content may be incorrect.On the **Electronics Bay**, to select the position sensor slide the **latching Switch** so that the **LCD screen** shows (mm &in) on the screen, if not selected already.
   * Latching Switch is labeled as 2 on the figure to the right.
4. A circuit board with wires and a button

   AI-generated content may be incorrect.**Calibrate the Position Sensor**
   * On the **Electronics Bay**, press the **Red Button** to initiate the position sensor calibration.
   * Confirm a successful calibration by checking the message on the **LCD screen**.
5. **Open and Run the Simulink Control Model**
   * Open the provided starting **Simulink model** designed for open-loop control of the pneumatic cylinder.
   * In Simulink, select the **“HARDWARE” tab** from the command ribbon.
   * A screenshot of a computer

     AI-generated content may be incorrect.Click the **“Monitor & Tune”** icon to compile and download the control code to the Arduino.  
     *Note: Simulink will compile the model into C-code, load it onto the Arduino, and initiate communication.*
6. **Verify Model Execution**
   * When prompted by a pop-up window summarizing the model build and deployment process, select **“OK”** to proceed.
   * The system will then begin controlling the pneumatic cylinder in open-loop mode.
7. **Control the Solenoid Valve Manually**
   * Familiarize yourself with controlling the solenoid valve by extending and retracting the cylinder through the Simulink interface.
8. **Develop and Implement Closed-Loop Control**
   * Modify or develop a closed-loop control system that causes the pneumatic cylinder to cycle between **30 mm and 60 mm**, then back to **30 mm** over a **20-second cycle**.
   * Tune your PID gains to meet the following system performance specifications:
     + **Steady-state error ≤ 3 mm**
     + **Percent overshoot ≤ 10%**
     + **±2% settling time ≤ 5 seconds**
   * Use the **minimum control gain values necessary** to satisfy these requirements.
9. **Evaluate System Performance**
   * Use the **cursor measurement tools** in Simulink to analyze system response characteristics (rise time, overshoot, steady-state error, and settling time).
   * Make detailed observations about the dynamic behavior and control performance of the system when actuating:
     + From the **Rod Side**
     + From the **Piston Side**
   * Record and discuss any noticeable differences in control characteristics between the two configurations.

**Results and Discussion**

**1. Open-Loop Control Observations**

* Describe the system’s behavior when manually controlling the solenoid valve in open-loop mode.
* Note the response time, cylinder movement characteristics, and any limitations or challenges you observed.
* Comment on how the cylinder responds when actuating the **rod side** versus the **piston side** under open-loop control.

**2. Closed-Loop System Response**

**a. PID Controller Tuning**

* List the final PID gain values you selected (Kp, Ki, Kd).
* Briefly explain your tuning approach (e.g., trial and error, increasing proportional gain first, etc.).
* Describe how changes in each gain affected system performance (rise time, overshoot, steady-state error, settling time).

**b. Measured Performance**

* Provide a summary of the key performance metrics for your final controller:
  + **Steady-State Error (mm):**
  + **Percent Overshoot (%):**
  + **±2% Settling Time (s):**
* Include screenshots or plots of the system response curves for both **rod side** and **piston side** actuation.
* Indicate whether your system met the specified performance criteria.

**3. Comparative Analysis: Rod Side vs. Piston Side**

* Discuss any differences in control behavior and system dynamics when controlling the **rod side** versus the **piston side**.
* Address aspects such as:
  + Speed of response
  + Overshoot tendency
  + Required control effort (PWM duty cycle or valve switching frequency)
  + Position holding ability when in the closed-center position

**4. Sources of Error and System Limitations**

* Identify possible sources of error in your experiment:
  + Sensor calibration inaccuracies
  + Air leaks or pressure fluctuations
  + Valve switching delays
  + Air compressibility effects
* Discuss how these might have affected your results and performance measurements.

**5. Recommendations for Improvement**

* Suggest at least two ways to improve system performance or experimental setup.
* Examples might include:
  + Using proportional valves for finer control
  + Improving sensor resolution
  + Adjusting cycle timing
  + Enhancing the control algorithm (e.g., implementing feedforward control)