

# Lab 2: Time Domain System ID – First Order System

## Introduction

---

First order systems describe many physical phenomena. For example, the time history of a capacitor's voltage in a simple RC circuit is a first order system. The temperature history of an oven losing heat from conduction or convection is also a first order system. In this laboratory experiment, the step response behavior of the hydraulic motor system will be studied.

A major focus in engineering is to develop mathematical models of physical systems to predict the system's response to varying inputs. These plant models are then used as the basis for controller design. For example, after creating a mathematical model of oven temperature in terms of heat input you can design a controller to maintain a desired set temperature.

A critical step in development of mathematical models is the process of plant identification. In any model, there may be physical parameters such as thermal conductivity, time constant, or modulus of elasticity, whose values must be determined. As engineers, we use the plant identification process to experimentally estimate these parameters' values. Physicists may try to generate these values analytically.

The hydraulic motor system is assumed to have first order behavior, hence the title. You will determine the numerical values of the time constant and gain parameters of the first order system representing the hydraulic motor. Additionally, you will investigate any unexpected behavior in the physical system.

## Objectives

---

### Required Steps:

- Observe the characteristics of a first order system step response.
- Identify the time constant,  $\tau$ , and system gain,  $K$ , in your model of the hydraulic motor.
- Deduce the source of unexpected dynamics and signal noise.

## Experiment

---

### Hydraulic Hardware:

The Parker Hannifin Hydraulic Trainer will be used with Parker's BD90 portable valve/control-board subplate. The BD90 valve is a proportional directional control valve, which means it can control the direction of hydraulic fluid flow as well as the portion of available flow that gets transmitted through the valve.

This valve will be used to control a hydraulic actuator, the hydraulic motor for this lab. The valve will receive many open-loop step input commands of varying magnitudes and the motor's speed output will be measured. The B&R X20 Controller will be used to perform the open-loop control and data collection.

The system's simplified block diagram is displayed in Figure 1.

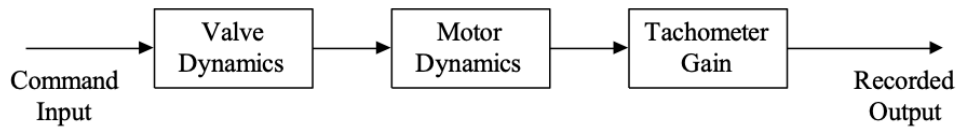


Figure 1: Hydraulic Motor System Block Diagram

Since the valve and tachometer dynamics are fast, it is assumed they do not affect the model's time constant,  $\tau$ , and system gain,  $K$ . Such assumptions may not always hold, so keep this in consideration when examining the collected data.

### Electrical Hardware:

The hydraulic motor's rotational speed is measured via a magnetic proximity sensor. These sensors are commonly used to measure a car's engine speed. The rotating gear of the motor induces a current in the sensor's circuit varying the proximity switch's signal. It is also referred to as a tachometer, an instrument that measures the operational speed of a rotating machine. The sensor's signal is available for measurement from a BNC connection located on the left side of the gray electrical box mounted on the top-left corner of the hydraulic trainer. It is labeled "Tachometer".

The tachometer described above is a frequency signal (a square wave at a specific frequency). It is not useful to directly measure the voltage of this frequency signal to quantify the motor's speed. The MW2000 is a signal processing device (the big blue electrical box). Analog Input Ch. 7 of the MW2000 will convert a frequency signal to a DC voltage proportional to the motor's speed. The converted analog voltage is then be read by the B&R X20 Controller.

The B&R X20 Controller runs on a real-time operating system (RTOS) to execute your control law code in a timed loop called a task cycle. An additional Simulink toolbox is used to generate C code from your Simulink model to be compiled and run in the task cycle. We will be using this Simulink toolbox with the B&R X20 Controller for the remainder of the labs.

## Procedure

---

### Simulink:

Copy the file 'N:\HydraulicsLab\ME460\Lab2\_MotorSysID.slx' and place the copy in your folder, 'C:\ME460\_SPxx\ABx\Lab2'. Run MATLAB and set your current directory to the location of your copy. Double-click the *Custom Step Input* block to change the start time to 1 second and duration to 5 seconds. Keep the initial value set to zero.

### Data Collection:

You will collect two sets of data. The first set will compare step inputs values for a constant system pressure. At 400 psi, you will give step values of 1 V, 2.5 V, and 5 V.

The second set of data will compare three different system pressures at a constant step input. You will give a step input value of 2.5 V at system pressures of 300, 400, and 500 psi.


### Construct Hydraulic Circuit:

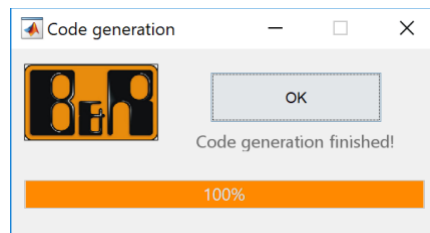
Construct a circuit to run the hydraulic motor using the BD90 servo valve.

1. Connect the valve's pressure and return lines (left and right ports when facing the bench) to the turn ports on the hydraulic motor. The hydraulic motor is bidirectional, so you can hook up the pressure and return lines to either port.
2. Plug in and power on the valve subplate control box. Connect the BNC terminal labeled *CMD* of the valve subplate control box to Analog Output Ch.1 of the MW2000 (blue box).
3. Power on the sensor box on the top left of the hydraulic trainer, you should see a red light go on. Make sure the BNC connection labeled *Tachometer* on the sensor box is connected to Analog Input Ch.7 of the MW2000.
4. Turn on the MW2000, the switch is on the back side (you will hear the fan turn on).
5. Turn on the B&R X20 Controller by flipping up the Eaton breaker switch.
6. **Have your TA check all connections before starting the hydraulic trainer.**
7. Turn on the hydraulic trainer following the startup steps written on the white board. Set the system pressure via the relief valve at 400 psi.


## Run Simulink Code:

### 1. Build

To run your Simulink model, click the *Build* button  in the top right corner of the model window to generate C code for the B&R Controller. This takes about 45 seconds and maybe longer for the first build. The small B&R pop-up will tell you when the code generation is finished, then click OK. You may close the diagnostics viewer.




### 2. Connect

Next, you must connect to your B&R Controller, click the *Connect to Target* button . Once connected you should see a green lit bar with the current run time in the bottom right corner of your model window. Now your scope blocks will display the input and output voltages in real-time.



### 3. Run

While connected, click the *RUN* rectangular push button in your Simulink model to execute your step input command. Once the step input's duration has been completed, you may click the *RUN* button again to initiate another step input command while remaining connected to the target. Note: You must Disconnect from Target,  for uploaded data to be logged to the MATLAB workspace.



#### 4. **Halt**

If at any point your control input causes the motor's behavior to go unstable, you may click the *HALT* button disabling the voltage output to the valve (re-enabled on the next *RUN*).

## Data Set 1

---

- Set the final value of the *Custom Step Input* block to 1 V.
- Build, connect, and run your input command with your Simulink model.
- Once your 5 second input command is completed, you must Disconnect from Target. This will log the scope data to the MATLAB workspace as *InputData* and *ResponseData*. Ignore any temp variables.
- Change the names of the input and response data in the MATLAB workspace to uniquely save the data.
  - The data is in the form of *Structure With Time*. To get the time vector, use the dot notation `ResponseData1V.time`. Similarly use `ResponseData1V.signals.values`, to get the output vector.
- The next time you connect and disconnect from target, new data will be saved to the MATLAB workspace under the original names.
- Plot the control input voltage and output response voltage on the same time plot.
- Repeat the above steps for final step values of 2.5 V and 5 V. You do not have to re-Build your code each time.

## Data Set 2

---

- Collect data for system pressures 300 psi and 500 psi with a step input of 2.5 V following the steps outlined in Data Set 1. Again, you should not have to re-Build your code.
- Once completed, shut down the hydraulic trainer following the instructions on the whiteboard. Power off the valve subplate control box, the feedback sensor box, and the MW2000.

## Report Questions

---

1. Determine the time constant  $\tau$  and the system (DC) gain  $K$  for each trial (gain is defined as output divided by input). You are conducting a time domain system identification of the first order hydraulic motor system.
2. Observe the small delay in the system response after the point in time when the step changes. What would cause the system response not to start exactly when the step changes? How does this effect the identified values  $\tau$  and  $K$ .

3. In Data Set 1, do the time constant and system gain change? Recall the hydraulic motor system model from Lab 1.
4. Compare Data Sets 1 and 2. Do the transfer functions change?
5. A system response to a step input that overshoots its final steady state value is an indication of a high order system. We assumed our hydraulic motor is a first order system. Why would there be overshoot in our system?
6. What physical parameters, such as pressure and inertia, affect the system's response?
7. How does signal noise affect your measurements of the identified parameters.