MATLAB Examples

Modeling a PWM Driven Hydraulic Servomechanism

This example shows the use of Simulink® and Stateflow® to model a hydraulic servomechanism controlled by a pulse-width modulated (PWM) solenoid. This type of motion control system is used in industrial, manufacturing, automotive and aerospace applications.

In this example, nonlinear differential equations are used to model the magnetic, hydraulic and mechanical components of the system. Discrete-time difference equations are used to represent the controller. A behavioral model in Stateflow implements the electronic circuit generating the PWM waveforms and regulating the solenoid current. Figure 1 shows a schematic of this mechanism.

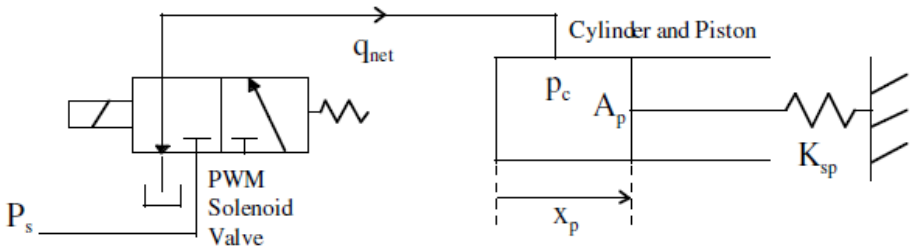

Figure 1: Schematic diagram of the Solenoid valve and hydraulic actuator

- **Note**: This is a basic hydraulics example. You can more easily build hydraulic models using Simscape™ and Simscape Fluids™. Simscape Fluids extends Simulink with tools for modeling and simulating hydraulic power and control systems. It enables you to describe multi-domain systems containing connected hydraulic and mechanical components as physical networks.

Contents

Modeling the System Using Nonlinear Differential Equations

The model of the servomechanism valve is divided in four parts:

- Magnetic Circuit
- Solenoid Armature Motion
- Hydraulic System
- Piston Motion

Magnetic Circuit

Consider first the magnetic circuit. The derivative of the magnetic flux is determined by Faraday's law as:

$$\frac{d\phi}{dt} = \frac{v_{sol} - iR}{N}$$

where v_{sol} is the solenoid voltage, i is the current, R is the winding resistance and N the number of turns. This equation assumes that fringing and leakage flux are negligible, as are eddy currents.

We assume that the cross-sectional area of the air gap A which relates the flux ϕ and the flux density B at the air gap applies uniformly for the steel path so that:

$$B = \frac{\phi}{A}$$

The flux density is linked to the magnetic field intensity in the air and the steel. In the air, magnetic field intensity is computed by:

$$H_{air} = \frac{B}{\mu_0}$$

where μ_0 is the permeability of air. In the steel, the magnetic field intensity H_{steel} is linked to the flux density B through a nonlinear function so that:

$$H_{steel} = f(B)$$

Note that the hysteresis is not modeled in this example. The combination of the magnetic fields in the air and the steel gives us the magnetomotive force:

$$MMF = H_{air}g + H_{steel}L_{steel}$$

where g is the length of the air gap and L_{steel} is the length of the steel magnetic circuit.

One last equation is now required to close the loop and obtain the current i in the magnetic circuit by:

$$i = \frac{MMF}{N}$$

On the mechanical side, the force developed by the solenoid is computed as:

$$F_{sol} = \frac{AB^2}{2\mu_0}$$

Servomechanism Tuni

This example shows how t
Simulink® Design Optimiza
optimize the position contr
parameters for a servomot

Implemented in Simulink, magnetic circuit equations look like:

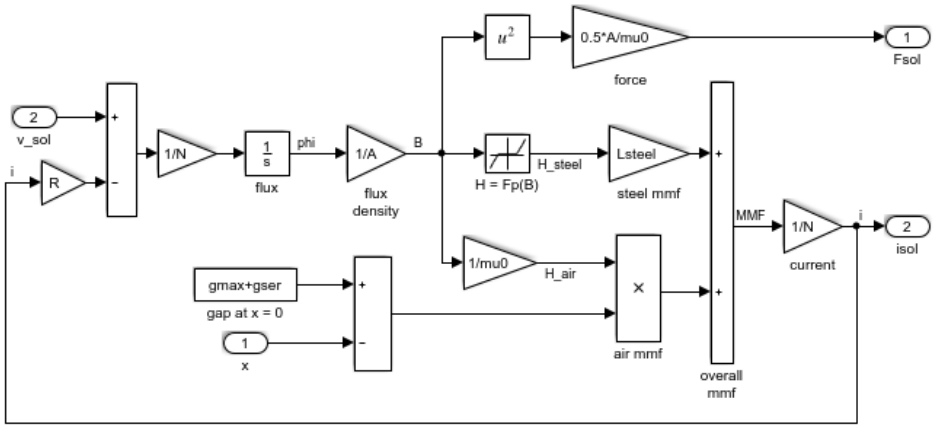

Figure 2: Solenoid magnetic circuit subsystem

Solenoid Armature Motion

The motion of the solenoid moving part, the armature, is generated by a combination of magnetic, hydraulic and mechanical forces:

$$m\frac{d^2x}{dt^2} = F_{sol} + A_0 P_s - F_{s0} - K_s x - C_v \frac{dx}{dt}$$

where A_0 is the supply orifice area, P_s is the hydraulic circuit supply pressure, F_{s0} is the spring preload, K_s is the return spring rate and C_v is the damping rate.

In Simulink this equation is implemented using the Second-Order Integrator block as shown in figure 3.

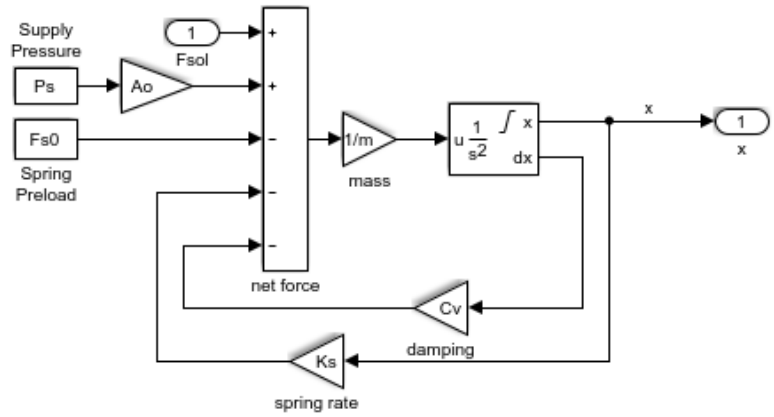

Figure 3: Armature motion subsystem

Hydraulic Circuit

The flow in the hydraulic circuit is controlled by the motion of the solenoid armature. A schematic of this system is displayed in the following diagram

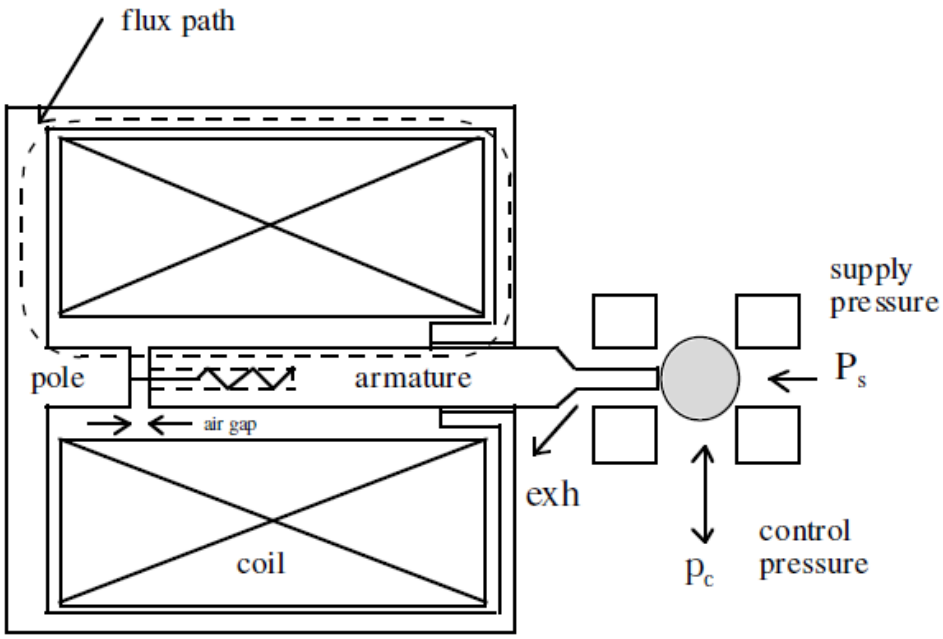

Figure 4: Solenoid Schematic

This diagram shows that when no current is applied to the magnetic circuit, the internal spring pushes the armature and ball to block the supply pressure P_s. When the supply pressure P_s is blocked, the control pressure P_c is directed toward the exhaust. When the solenoid is energized, the armature and pole come together and the supply pressure P_s pushes the ball to open the supply port and block the exhaust port.

The oil flow directed from the supply system to the piston, q_{net}, is the combination of the supply flow and the exhaust flow.

$$q_{net} = q_s - q_{ex}$$

The supply flow is computed has:

$$q_s = K_0 A_0 sgn(P_s - P_c)\sqrt{|P_s - P_c|}$$

where P_c is the control pressure and K_0 is the flow coefficient. The exhaust flow is computed has

$$q_{ex} = K_0 A_0 \sqrt{P_c}$$

In Simulink, the supply and exhaust equations are implemented using If Action Subsystems. These subsystems are activated based on the armature position.

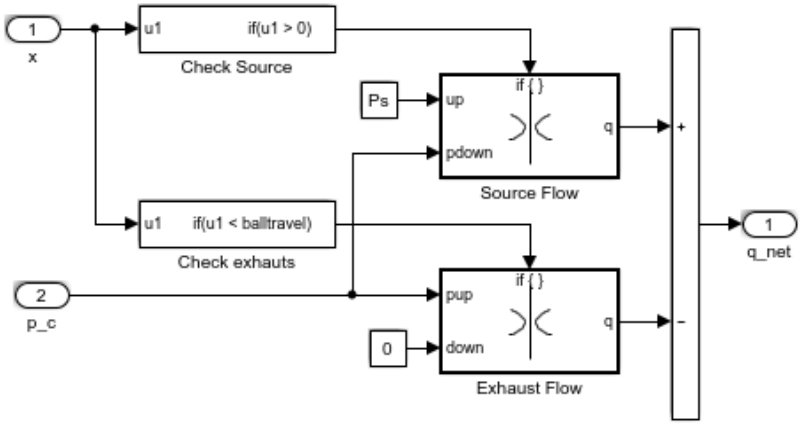

Figure 5: Valve flows subsystem

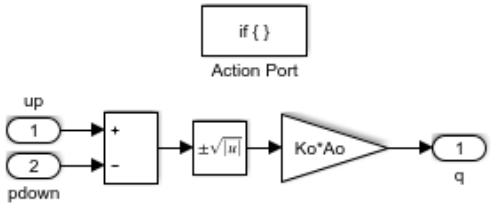

Figure 6: Orifice flow subsystem

To complete the hydraulic circuit, the control pressure must be computed. This pressure is a function of the net oil flow in the circuit q_{net} and the piston position x_p as:

$$\frac{dP_c}{dt} = \frac{\beta}{x_p A_p}(q_{net} - \frac{dx_p}{dt}A_p)$$

where β is the oil bulk modulus and A_p is the cross-sectional area of the cylinder.

In Simulink this is implemented as:

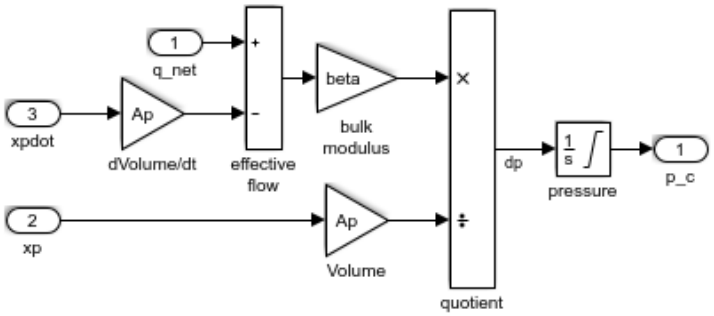

Figure 7: Cylinder Pressurization Subsystem

Piston Motion

The piston's equation of motion is:

$$M_p \frac{d^2x}{dt^2} = p_c A_p - K_{sp} x_p$$

where M_p is the net actuator mass and K_{sp} is the spring rate.

In Simulink this equation is implemented using the Second-Order Integrator block as:

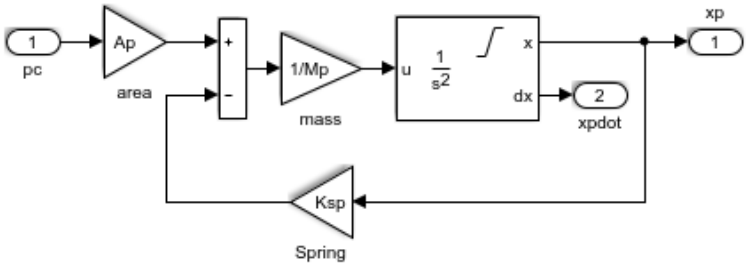

Figure 8: Piston Motion Subsystem

Implementing the Controller Using Discrete-Time Difference Equations

The objective of the system is to position the load x_p so that it follows a time-varying set point r_{set}. An electronic controller compares the command and set point to generate a PWM control signal at a rate of 50 Hz. We employ a discrete-time PI control law:

$$dutycycle = (K_p + \frac{K_I}{z-1})(r_{set} - x_p)$$

The control signal is applied to a 50 Hz pulse train and the power electronics converts the pulse signal to the solenoid current.

Implementing the Behavioral Model of a PWM Generating Circuit Using Stateflow

Digital and analog integrated circuits are available to perform this function, so we use a behavioral model rather than a highly detailed physical model. The behavior is best described in terms of the circuit's reaction to the commands and the response of its load. The voltage and current for one typical cycle of the control circuit is shown in figure 9.

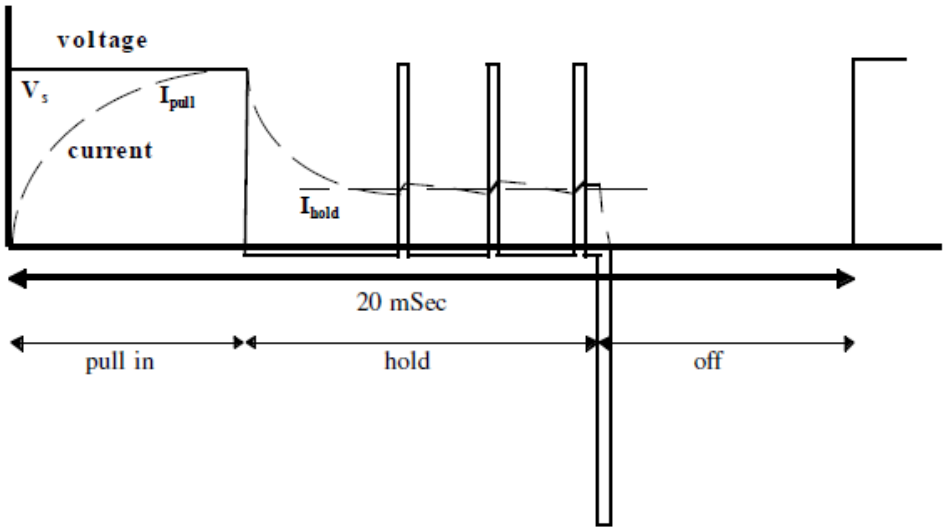

Figure 9: Voltage and current behavior of the PWM signal within one pulse.

At the beginning of each cycle, the PWM pulse turns on to push the armature against the pole and open the valve supplying pressure. At that time the driver circuit applies the full voltage to achieve the fastest initial rise in current. The solenoid maintains this condition until the current has risen to the level at which the magnetic and hydraulic forces overcome the spring and move the armature.

Once the armature has been pulled in, the air gap is very small and less current is needed to hold the armature in place. The driver regulates the current at a lower level for the remainder of the "on" portion of the cycle.

At the end of each pulse, the armature releases so that the ball returns to its original position to open the exhaust. This is achieved by opening the solenoid circuit so that the magnetic field collapses quickly. The current then remains at zero for the duration of the "off" time until the next cycle begins.

This way, the "on" portion of the PWM cycle is composed of the "pull in" and "hold". The "off" portion is characterized by the initial rapid decay, followed by zero voltage and current.

This can be implemented in Stateflow as:

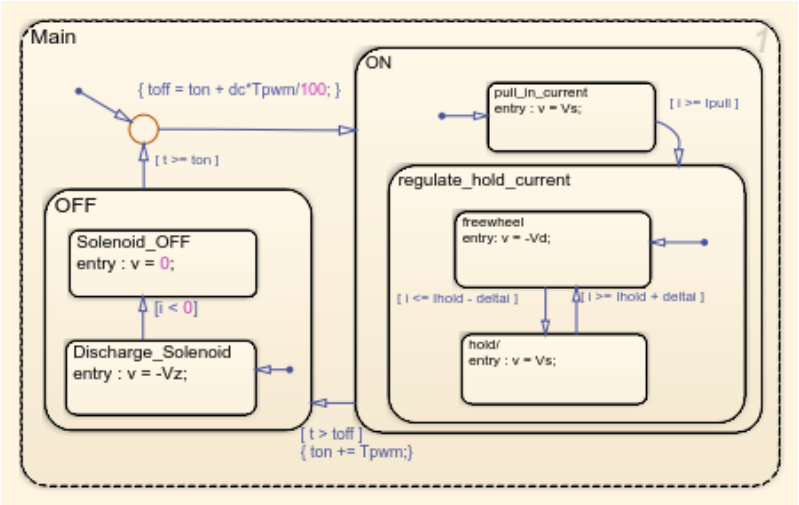

Figure 10: Stateflow implementation of the PWM driver circuit

mathworks.com