Modeling control system in Pro/Mechanisms

The following paper describes how to model control systems, using the now retired "Mechanica-Motion" multi-body simulation software. This same procedure can be used by the current version of Pro/Mechanisms to achieve the same result. The differences are the following:

1. The use of "Parameters" should replace the reference to "Design Variables"
2. The polynomial coefficient "C", should be "B" for the current version of Mechanisms
3. Three (3) different measures and motors can be created – one for each portion of the PID control; "proportional", "derivative" and "integrated".
4. BMX is used to conduct sensitivity studies and optimizations to arrive at desirable parameters, KP, KD, & KI.

Chris Kaswer Nov. 22nd, 2005

MODELING CONTROL SYSTEMS IN APPLIED MOTION

We are commonly asked "Can you model feedback control systems in Applied Motion". The answer is "yes". This article covers the fundamentals of the widely used proportional-integral-derivative (PID) feedback controller and how to model this type of controller in Applied Motion.

1.0 Feedback Control Systems

Figure 1 - Feedback Control System with Proportional Control
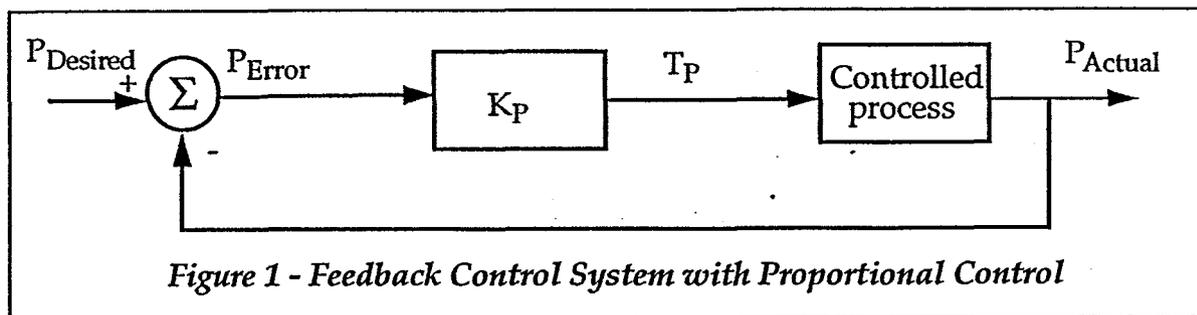
Proportional Control

Figure 1 shows a simple "proportional" feedback control system. This control system works by determining the error between the desired position and the actual position of a motor axis and multiplying the error by a constant gain K. The mathematical formula governing this control system is

$$T_P = K_P*(P_{Error})$$
$$= K_P*(P_{Desired}-P_{Actual})$$

where the variables are defined below:
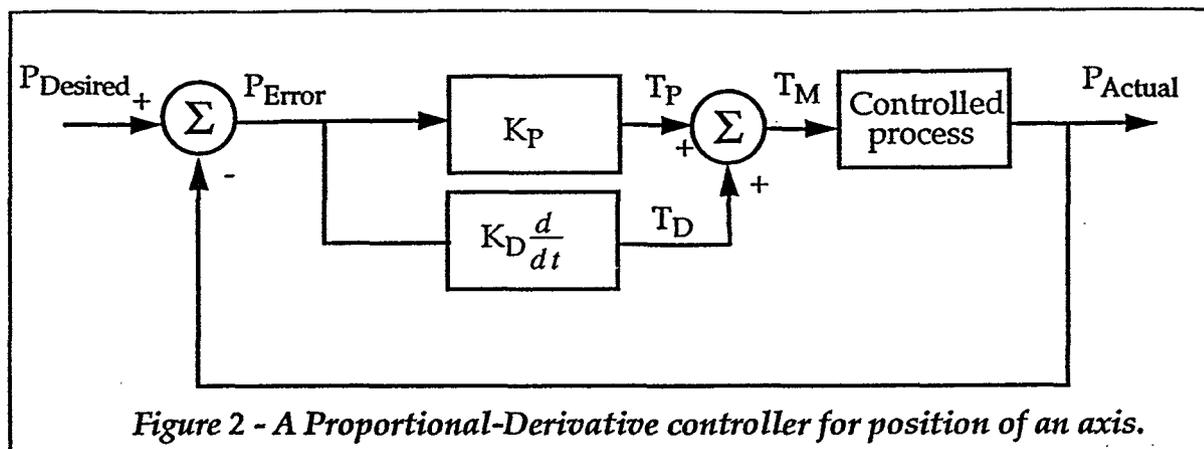
$P_{Desired}$	Desired Position of the axis of rotation
P_{Error}	Difference between $P_{Desired}$ and P_{Actual}
P_{Actual}	Actual Position of the axis of rotation
K_P	Proportional Control Gain
T_P	Command Torque from Proportional Control

Proportional control systems are analogous to adding springs to the system and can exhibit undesirable behavior such as overshooting the desired position. This behavior can be corrected by adding damping to the system.

Proportional-Derivative (PD) Control

Derivative control adds damping to reduce the overshoot problem. Derivative control measures the instantaneous slope of the error, predicts the large overshoot ahead of time and makes a proper correcting effort before the overshoot actually occurs. Figure 2 shows a PD controller.

Figure 2 - A Proportional-Derivative controller for position of an axis.

The torque generated by the derivative portion of the controller is governed by the equation:

$$T_D = K_D{}^*(d/dt(P_{Error}))$$
$$= K_D{}^*(d/dt(P_{Desired} - P_{Actual}))$$
$$= K_D{}^*(-V_{Actual})$$

The total control torque, T_M, is equal to $T_P + T_D$.

Integral Control

Integral control is added to PD control to remove any bias torque in the system. Bias torque will cause a PD type controller to exhibit steady state error. Figure 3 shows a complete PID controller..

Figure 3 - Feedback control system with proportional-integral-derivative control
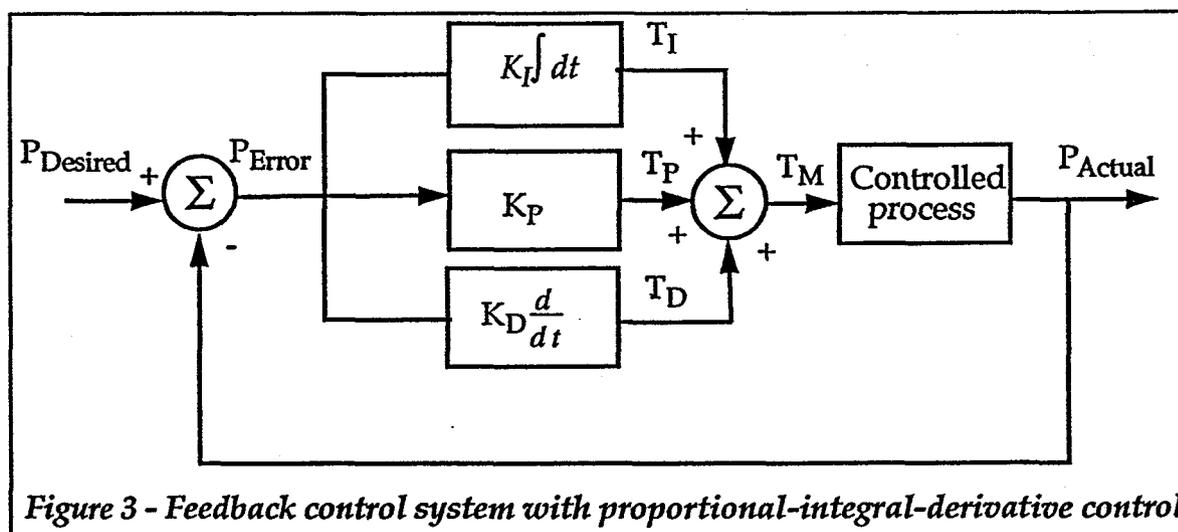
The integral control portion of the controller is governed by the equation:

$$T_I = K_I{}^* \int (P_{Desired} - P_{Actual})$$

The total control torque, T_M, is equal to $T_P + T_D + T_I$

APPLIED MOTION APPLICATION PAPER

2.0 "PID" Applied Motion Example

All the basic building blocks for control systems are available in Applied Motion 2.0. To illustrate modeling a PID controller in Applied Motion, lets look at controlling the position of a simple pendulum subjected to gravity. The force of gravity will act as a bias on the system.
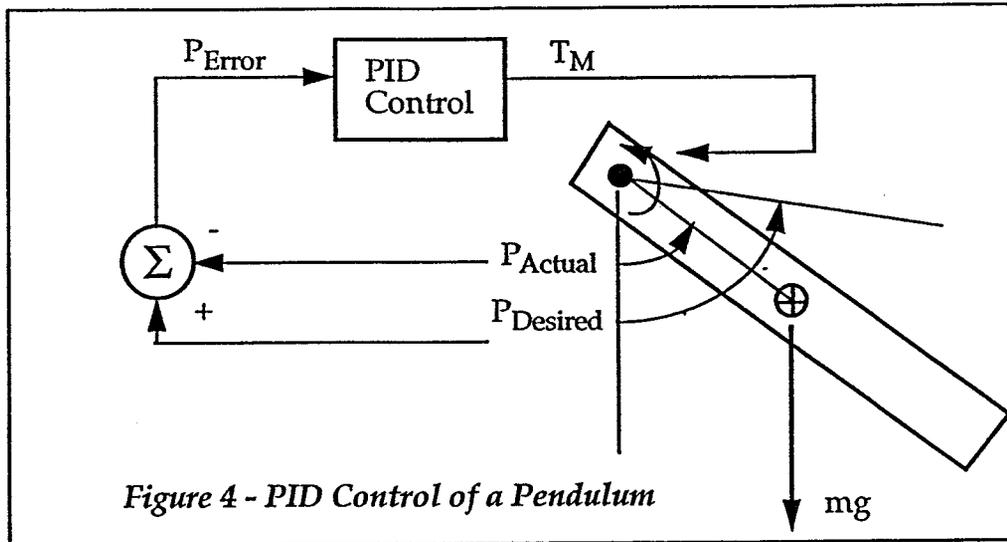

Figure 4 - PID Control of a Pendulum

Following is a step by step process to building the model.

1. Create the pendulum part which is connected to ground via a pin joint. The PID controller will act on the pin joint DOF.

2. Create two DOF measures on the pin joint. Call the first one "PActual". Set the DOF Measure Type to Position and the Evaluation Method to "Each Time Step". Call the second one "VActual". Set the DOF Measure Type to "Velocity" and the Evaluation Method to "Each Time Step".

3. Create design variables for the control gain constants and the desired position of the pin axis. Use the following names

Design Variable	Measure
KP	KProp
KD	KDeriv
KI	KInteg
PDesired	PosDesired

4. Create a computed measure named TP_TD representing the torques for the proportional and derivative control signals. Use the following definition

   KProp*(PosDesired-PActual)+KDeriv*(-VActual)

   Figure 5 shows what the computed measure form should look like.

Figure5 - Computed Measure for PD portion of controller

5. Create a JOINT force on the pin joint. Define the force function as a polynomial with the "C" coefficient equal to 1. Make the force a function of the measure TP_TD. At this point the PD portion of the controller is complete.

6. To add the integral control portion create a computed measure named TI with the following definition:

KInteg*(PosDesired-PActual)

Set the Evaluation Method to "Integrated".

7. As the final step, create a JOINT force on the pin joint. Define the polynomial force with the C coefficient set to 1 and a function of the measure TI.

The model of the PID controller is complete. Test it out by setting the design variables for gains and some desired position. Then run a motion analysis. Finding the proper gain constants is not always a trivial task. Using the power of Applied Motion, you can set up sensitivity analyses which vary the gains to find suitable values for your control system. The graph in Figure 6 compares the results of P, PD, and PID controllers on the pendulum, where the desired angle of the pendulum was set to 1.0 radians. Note the overshoot in the P controller, the bias error in the PD controller and the best results of the PID controller.
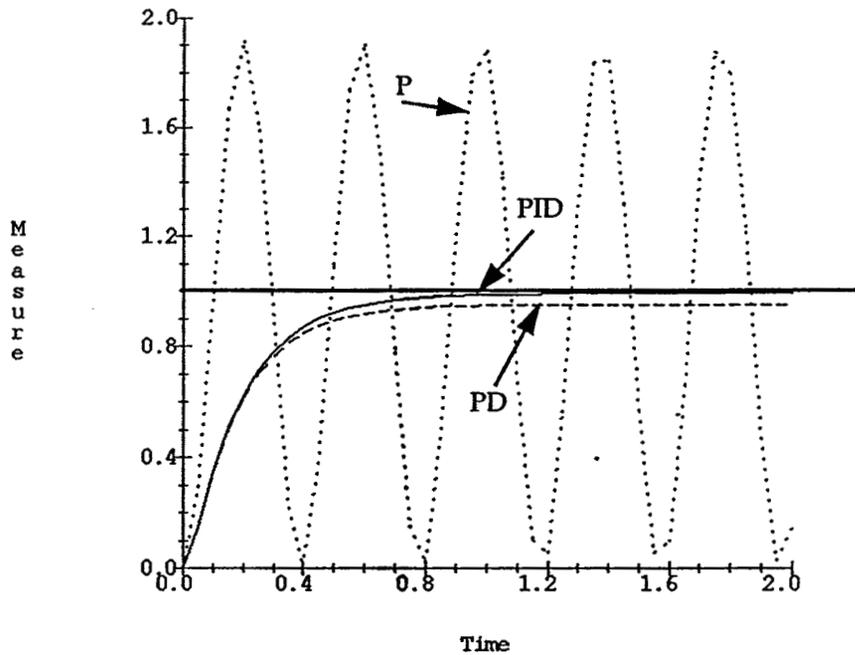
Figure 6 – DOF position compared for P, PD, and PID controllers

In summary, Applied Motion is an extremely versatile and powerful tool for analyzing the behavior of feedback control systems. Even though not shown here, Applied Motion is not limited to PID type control systems, but can also analyze models with Full State Feedback, Optimal Control with estimators, and transfer-function based controllers.