MCE/EEC 647/747: Robot Dynamics and Control

Lecture 6, Part I: Independent Joint Control: Classical Methods

Reading: SHV Chapter 6

Mechanical Engineering

Hanz Richter, PhD

Servomotor Model

A direct-current motor (DC motor or servomotor) for robotic applications is usually controlled using armature voltage. These motors typically use permanent magnets. As shown in elementary courses, the electromagnetic torque produced by the motor is given by

$$\tau_m = K_m i_a$$

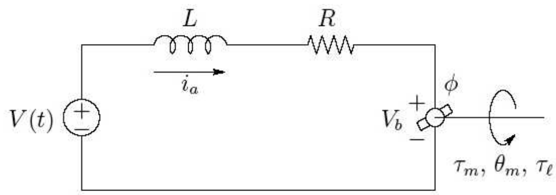
where K_m is the *torque constant* (a function of the motor's constructive characteristics) and i_a is the current circulating in the armature circuit. When the motor rotates, it also acts as a generator, producing a voltage which opposes the armature current. This voltage, called *back-electromotive force*, or *back-emf* is given by

$$V_b = K_b w_m$$

where K_b is the *back-emf constant* and w_m is the angular velocity of the motor (rad/sec).

Servomotor Model...

The units of K_m are N-m/A and the units of K_b are V-sec, but it's easy to show that 1 N-m/A=1 V-sec, so K_b and K_m are numerically equal if expressed in the same units. Fig. 6.3 in SHV shows the circuit diagram of the DC motor:



τ_l is the load torque, used to represent the effect of all mechanical loads connected to the shaft.

Independent Joint Model

To derive this model, it is assumed that the DC motor is connected to a gear reduction of ratio $r:1$ and moment of inertia J_g. The reduced-speed shaft drives a rotational inertia J_l, which represents the link driven by the motorized joint.
Of course, the motion of the other links should influence the DC motor as well. For the independent joint model, however, we treat these influences as disturbances and design the controller to be robust (tolerant) against them. The only reason for this this gross assumption is simplicity (ignorance) and low performance requirements.

A general principle when designing control systems is: "the performance of the controller increases with the accuracy of the mathematical model". Electromechanical systems are well-understood and lead to accurate models. Higher performance will be afforded with the MIMO techniques to be studied later in the course.

Independent Joint Model...

Analyzing the electrical side of the model, we obtain

$$V - i_a R - L\frac{di_a}{dt} - V_b = 0$$

Using $V_b = K_b \dot{\theta}_m$ and taking the Laplace transform we get

$$(Ls + R)I_a(s) = V(s) - K_b s \Theta_m(s)$$

On the mechanical side, we use $\theta_s = \theta_m / r$ to describe the rotation of the link and assume viscous damping (proportional to speed) in the gear reduction, with coefficient B_m. The torque balance equation gives

$$J_m \ddot{\theta}_m + B_m \dot{\theta}_m + \tau_l / r = \tau_m = K_m i_a$$

where $J_m = J_a + J_g$. Note (Fig. 6.5) that τ_l has been used to represent the disturbance torque, reduced by a factor of r before it makes it to the motor shaft. In Laplace form:

$$(J_m s^2 + B_m s)\Theta(s) = K_m I_a(s) - \tau_l(s)/r$$

Independent Joint Model...

In what follows, we will use Simulink (or Scilab/Scicos) to simulate the performance of our controllers. The block diagram of Fig. 6.6 gives access to important internal signals like the net torque and the current (must not exceed mechanical and electrical ratings, respectively).

HW3: Build a Simulink (optionally Scicos) model corresponding to Fig. 6.6. Write an m-file loading a set of values for the DC motor parameters.

Very often, the ratio L/R is very small. This can be used as the basis for a model simplification, leading to

$$J\ddot{\theta} + B\dot{\theta} = u - d$$

where now $J = J_m$, $B = B_m + K_b K_m / R$, $u = (K_m/R)V$ and $d = \tau_l / r$ is the disturbance.

This model can be used for control design. The designed controllers will be simulated against the full model, however.

PD Control

The transfer function from V to Θ has a pole at the origin and a negative pole. A quick root locus indicates that the simplest controller stabilizing the plant is a proportional gain. However, the settling time cannot be modified using only P-control.
By adding derivative action, we can obtain complex poles with varying time constants and damping. The control transfer function is

$$U(s) = (K_p + K_d s)E(s)$$

where $E(s) = \Theta^d(s) - \Theta(s)$ is the tracking error. It is straightforward to show that any two positive values of K_p and K_d drives the error to zero asymptotically when $d = 0$. However, the quality of the response is highly dependent on the actual gain selections.

PD Control...

If a step disturbance is applied, stability of the system is preserved, however the error converges to a nonzero value (called *steady-state error*):

$$e_{ss} = -\frac{D}{K_p}$$

Since $d = \tau_l/r$, a smaller offset can be obtained for larger gear reductions and higher values of proportional gain. The practical limitation comes from control saturation and overcurrent limitations.

In Example 6.1, we see that the PD compensator introduces a zero at K_p/K_d. The closed-loop poles have been forced to be real and equal by setting $\zeta = 1$. This strategy prevents the performance degradation observed whenever zeroes have time constants similar to the dominant poles.

PID Control

When the limits on the P gain imposed by control saturation and current limits are incompatible with the level of disturbance, an I-term can be used to reject the disturbance completely without resorting to high gains. The control transfer function is now

$$U(s) = (K_p + K_d s + \frac{K_i}{s})E(s)$$

As shown in SHV, the closed-loop poles are the solutions of a third-order characteristic polynomial. The Routh-Hurwitz criterion can be used to derive a condition for stability:

$$K_i < \frac{(B + K_d)K_p}{J}$$

in addition to the requirement that all gains be positive.

Tuning of the PID gains can be done, for instance, using the root locus or by trial-and-error simulations.

Integrator Windup

By far, the main limitation of integrating control is the *windup* phenomenon. When a large and persistent disturbance acts on the system so that an offset results, the integrating term in the control law is likely to reach large values which saturate the actuator (power amplifier in robotics). Although the control input cannot increase beyond the saturation limit, the integrating term keeps growing as long an offset is present (winds up).
Even upon sudden removal of the disturbance, the integral term takes a long time to reduce its value below saturation. This typically results in a large bump in the response.

We run a sample Simulink model demonstrating the windup phenomenon and a common anti-windup strategy.

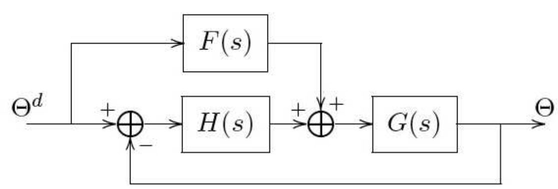
Trajectory Tracking-Feedforward+Feedback

When the reference input is not constant (point-to-point vs. continuous path) but rather an arbitary function of time, additional complexity is required in the control strategy.
A drawback of the classical error feedback control structure is that an error must necessarily occur before the controller takes corrective action (the controller will not generate output if the error has been zero for a while). Imagine driving a car by just looking at the rearview mirror and steering only in response to deviations observed in the mirror.
On the other hand, purely predictive control based on plant model will fail due to model errors and unexpected disturbances. Imagine pre-programming the steering actions and not taking any corrective action.

Feedforward can be used in combination with feedback to improve tracking and maintain disturbance rejection and robustness.

Trajectory Tracking...

The diagram in Fig.6-16 illustrates the feedforward+feedback arrangment:



Suppose the plant, feedback compensator and feedforward compensator TFs are $G(s) = \frac{q(s)}{p(s)}$, $H(s) = \frac{c(s)}{d(s)}$ and $F(s) = \frac{a(s)}{b(s)}$ with $G(s)$ *strictly proper* (more poles than zeroes) and $H(S)$ proper (#poles $\geq$ #zeroes).

Trajectory Tracking...

A simple block diagram reduction shows that the closed-loop transfer function is

$$\frac{\Theta(s)}{\Theta_d(s)} = T(s) = \frac{q(s)(c(s)b(s) + a(s)d(s))}{b(s)(p(s)d(s) + q(s)c(s))}$$

Therefore $H(s)$ and $F(s)$ must be chosen so that $b(s)(p(s)d(s) + q(s)c(s))$ has poles in the left half of the complex plane (Hurwitz). Obviously, the factors $b(s)$ and $p(s)d(s) + q(s)c(s)$ must be themselves Hurwitz.

Since $b(s)$ is the denominator of $F(s)$, we require $F(s)$ to be a stable TF.

If $G(s)$ has all its zeroes in the lhp (a *minimum-phase system*) , then we can choose $F(s) = \frac{1}{G(s)}$. By doing this, it's easy to show that the tracking error is identically zero at all times if no disturbance is present and the model is perfect.

Trajectory Tracking...

If a disturbance $D(s)$ is present at plant input, there will be some error:

$$E(s) = \frac{q(s)d(s)}{p(s)d(s) + q(s)c(s)} D(s)$$

The feedforward+feedback approach can be applied to the robot model operating under PD control. In this case $G(s)$ is minimum-phase, so we may take $F(s) = Js^2 + Bs$. Note that although $F(s)$ includes double and single differentiation, these operations do not need to be carried out online (signal differentiation is noisy) if $\theta_d(t)$ is known in advance. The derivatives can be pre-computed and stored in the control program. See Eqs. 6.36 thru 6.38.

Harmonic Drives and Joint Flexibility

A harmonic drive is a compact gear mechanism achieving large reduction ratios with high torque capacity and low backlash. For more information on harmonic drives see

`http://www.harmonicdrive.net/reference/operatingprinciples/`

`http://www.powertransmission.com/issues/0706/harmonic.htm`

A side-effect of using a harmonic drive is the flexibility introduced in the drive train. SHV develops a simplified model which takes motor torque as opposed to voltage as the control input. This model is used to show the limitations of PD control.

As a homework project, you will design a PD or PID compensator assuming voltage input and testing for performance against the full-order motor model.