3 Motion Control with Velocity Inputs

- There are two kinds of control inputs, e.g., velocity control and torque control. The joint velocity will be commanded when
  - the stepper motors are used
  - the amplifier for an electric motor is placed in velocity control mode
- Here we can assume that there is direct control of the joint velocities, instead of joint torques.
- Also we will assume that the control inputs are joint velocities.
- The motion control task can be expressed in joint space or task space.
  - When the trajectory is expressed in task space, the controller is fed a steady stream of end-effector configurations $X_d(t)$, and the goal is to command joint velocities that cause the robot to track this trajectory.
  - In joint space, the controller is fed a steady stream of desired joint positions $\theta_d(t)$.

3.1 Motion Control of a Single Joint

Feedforward Control

- Given a desired joint trajectory $\theta_d(t)$, the simplest type of control would be to choose the commanded velocity $\dot{\theta}(t)$ as

$$\dot{\theta}(t) = \dot{\theta}_d(t)$$

- This is called a feedforward or open-loop controller, since no feedback (sensor data) is needed to implement it.

Feedback Control

- In practice, position errors will accumulate over time under the feedforward control law.
- An alternative strategy is to measure the actual position of each joint continually and implement a feedback controller.

P Control and First-Order Error Dynamics

- The simplest (feedforward plus) feedback controller is

$$\dot{\theta}(t) = \dot{\theta}_d(t) + K_p(\theta_d(t) - \theta(t)) = \dot{\theta}_d(t) + K_p\theta_e(t)$$

  where $K_p > 0$.

- It is would be preferable to use our knowledge of the desired trajectory $\theta_d(t)$ to initiate motion before any error accumulates.
- This controller is called a proportional controller, or P controller, because it creates a corrective control proportional to the position error $\theta_e(t) = \theta_d(t) - \theta(t)$.
- In other words, the constant control gain K_p acts somewhat like a virtual spring that tries to pull the actual joint position to the desired joint position.
- The error dynamics

$$\dot{\theta}_e(t) = \dot{\theta}_d(t) - \dot{\theta}(t)$$

  is written as follows after substituting in the P controller $\dot{\theta}(t) = \dot{\theta}_d(t) + K_p\theta_e(t)$:

$$\dot{\theta}_e(t) = -K_p\theta_e(t) \qquad \rightarrow \qquad \dot{\theta}_e(t) + K_p\theta_e(t) = 0$$

- This is a first-order error dynamic equation with time constant $\mathrm{t} = \frac{1}{K_p}$.
- The steady-state error is zero, there is no overshoot, and the 2% settling time is $\frac{4}{K_p}$.
- A larger K_p means a faster response.

PI Control and Second-Order Error Dynamics

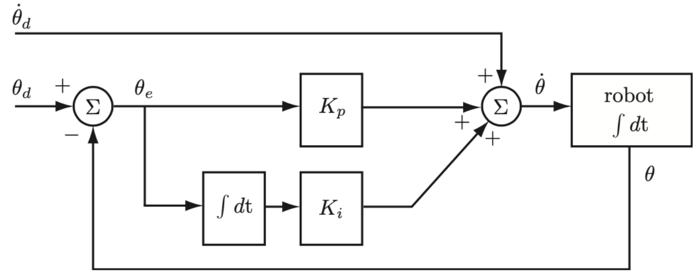

Figure 11.9: The block diagram of feedforward plus PI feedback control that produces a commanded velocity $\dot{\theta}$ as input to the robot.

- An alternative to using a large gain K_p is to introduce another term in the control law.
- A (feedforward plus) proportional-integral controller, or PI controller, adds a term that is proportional to the time-integral of the error:

$$\dot{\theta}(t) = \dot{\theta}_d(t) + K_p\theta_e(t) + K_i \int_0^t \theta_e(\sigma)d\sigma$$

  where t is the current time and σ is the variable of integration.

- With this controller, the error dynamics becomes

$$\dot{\theta}_e(t) = \dot{\theta}_d(t) - \dot{\theta}(t)$$

  is written as follows after substituting in the PI controller $\dot{\theta}(t) = \dot{\theta}_d(t) + K_p\theta_e(t) + K_i \int_0^t \theta_e(\sigma)d\sigma$:

$$\dot{\theta}_e(t) = -K_p\theta_e(t) - K_i \int_0^t \theta_e(\sigma)d\sigma \qquad \rightarrow \qquad \ddot{\theta}_e(t) + K_p\dot{\theta}_e(t) + K_i\theta_e(t) = 0$$

- We can rewrite this equation in the standard second-order form, with

$$\text{natural frequency}: \omega_n = \sqrt{K_i}$$
$$\text{damping ratio}: \zeta = \frac{K_p}{2\sqrt{K_i}}.$$

where the gain K_p plays the role of $\frac{b}{\mathfrak{m}}$ for the mass-spring-damper (a larger K_p means a larger damping constant b), and the gain K_i plays the role of $\frac{k}{\mathfrak{m}}$ (a larger K_i means a larger spring constant k).

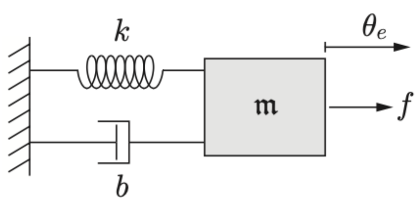

Figure 11.3: A linear mass–spring–damper.

- The PI-controlled error dynamics equation is stable if $K_i > 0$ and $K_p > 0$, and the roots of the characteristic equation are

$$s_{1,2} = -\frac{K_p}{2} \pm \sqrt{\frac{K_p^2}{4} - K_i}$$

- Let's hold $K_p = 20$ and plot the roots in the complex plane as K_i grows from zero. This plot, or any plot of the roots as one parameter is varied, is called a root locus.

- (Case I) For $K_i = 0$, the characteristic equation $s^2 + 20s = s(s + 20) = 0$ has roots at $s_1 = 0$ and $s_2 = -20$.

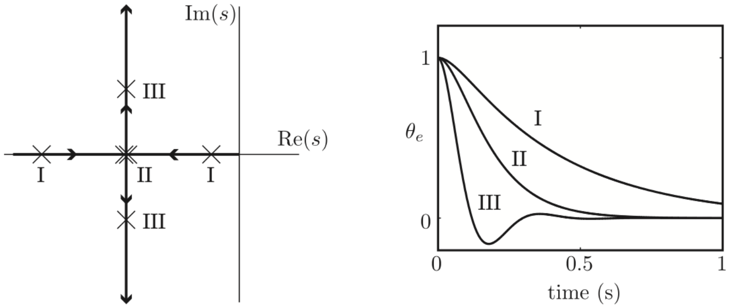

Figure 11.7: (Left) The complex roots of the characteristic equation of the error dynamics of the PI velocity-controlled joint for a fixed $K_p = 20$ as K_i increases from zero. This is known as a root locus plot. (Right) The error response to an initial error $\theta_e = 1$, $\dot{\theta}_e = 0$, is shown for overdamped ($\zeta = 1.5$, $K_i = 44.4$, case I), critically damped ($\zeta = 1$, $K_i = 100$, case II), and underdamped ($\zeta = 0.5$, $K_i = 400$, case III) cases.

  - As K_i increases, the roots move toward each other on the real axis of the s-plane as shown in the left-hand panel in the figure.
  - Because the roots are real and unequal, the error dynamics equation is overdamped ($\zeta = \frac{K_p}{2\sqrt{K_i}} > 1$, case I) and the error response is sluggish due to the time constant $\mathrm{t}_1 = -\frac{1}{s_1}$ of the exponential corresponding to the "slow" root.
  - As K_i increases, the damping ratio decreases, the "slow" root moves left (while the "fast" root moves right), and the response gets faster.

- (Case II) When $K_i = 100$, the two roots meet at $s_{1,2} = -10 = -\omega_n = -\frac{K_p}{2}$
  - The error dynamics equation is critically damped ($\zeta = 1$, case II).
  - The error response has a short 2% settling time of $4\mathrm{t} = \frac{4}{\omega_n} = 0.4s$ and no overshoot or oscillation.

- (Case III) As $K_i > 100$ continues to grow, the damping ratio $0 < \zeta < 1$
  - The roots move vertically off the real axis, becoming complex conjugates at $s_{1,2} = -10 \pm j\sqrt{K_i - 100}$ (case III).

- The error dynamics is underdamped, and the response begins to exhibit overshoot and oscillation as K_i increases.
- The settling time is unaffected as the time constant $\mathrm{t} = \frac{1}{\zeta\omega_n} = \frac{2}{K_p} = 0.1$ remains constant.

- According to our simple model of the PI controller, we could always choose K_p and K_i for critical damping ($K_i = \frac{K_p^2}{4}$) and increase K_p and K_i without bound to make the error response arbitrarily fast.
- As described above, however, there are practical limits. Within these practical limits, K_p and K_i should be chosen to yield critical damping.
- A well-designed PI controller can be expected to provide better tracking performance than a P controller.

3.2 Motion Control of a Multi-joint Robot

- The single-joint PI feedback plus feedforward controller

$$\dot{\theta}(t) = \dot{\theta}_d(t) + K_p\theta_e(t) + K_i \int_0^t \theta_e(\sigma)d\sigma$$

  generalizes immediately to robots with n joints.

- The reference position $\theta_d(t) \in \Re^n$ and actual position $\theta(t) \in \Re^n$ are now n-vectors, and the gains K_p and K_i are diagonal $n \times n$ matrices of the form k_pI and k_iI, where the scalars k_p and k_i are positive and I is the $n \times n$ identity matrix.

$$\theta_d(t) = \begin{bmatrix} \theta_{1,d}(t) \\ \theta_{2,d}(t) \\ \vdots \\ \theta_{n,d}(t) \end{bmatrix} \in \Re^n \quad \theta(t) = \begin{bmatrix} \theta_1(t) \\ \theta_2(t) \\ \vdots \\ \theta_n(t) \end{bmatrix} \in \Re^n \quad K_p = \begin{bmatrix} k_p & 0 & \cdots & 0 \\ 0 & k_p & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & k_p \end{bmatrix} \in \Re^{n\times n} \quad K_i = \begin{bmatrix} k_i & 0 & \cdots & 0 \\ 0 & k_i & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & k_i \end{bmatrix} \in \Re^{n\times n}$$

- Each joint is subject to the same stability and performance analysis as the single joint in Section 11.3.1.

3.3 Task-Space Motion Control

- Let us express the feedforward plus feedback control law in task space.
- Let $X_{sb}(t) \in SE(3)$ be the configuration of the end-effector as a function of time and $\mathcal{V}_b(t)$ be the end-effector twist expressed in the end-effector frame {b}, i.e., $[\mathcal{V}_b] = X_{sb}^{-1}\dot{X}_{sb}$.
- The desired motion is given by $X_{sd}(t)$ and $[\mathcal{V}_d] = X_{sd}^{-1}\dot{X}_{sd}$.
- A task-space version of the control law is

$$\mathcal{V}_b(t) = [Ad_{X_{sb}^{-1}X_{sd}}]\mathcal{V}_d(t) + K_p X_e(t) + K_i \int_0^t X_e(\sigma)d\sigma$$

  - The term $[Ad_{X_{sb}^{-1}X_{sd}}]\mathcal{V}_d(t)$ expresses the feedforward twist $\mathcal{V}_d$ in the actual end-effector frame at X_{sb} rather than the desired end-effector frame X_{sd}.
  - When the end-effector is at the desired configuration ($X_{sb} = X_{sd}$), this term reduces to $\mathcal{V}_d$.
  - The configuration error $X_e(t)$ is not simply $X_d(t) - X(t)$, since it does not make sense to subtract elements of $SE(3)$.
  - X_e should refer to the twist which, if followed for unit time, takes X_{sb} to X_{sd}.
  - The $se(3)$ representation of this twist, expressed in the end-effector frame, is $[X_e] = \log(X_{sb}^{-1}X_{sd})$.
  - Diagonal gain matrices $K_p, K_i \in \Re^{6\times 6}$ take the form $k_p I$ and $k_i I$, respectively, where $k_p, k_i > 0$.
  - The commanded joint velocities $\dot{\theta}(t)$ realizing $\mathcal{V}_b$ from the control law can be calculated using the inverse velocity kinematics,

$$\dot{\theta}(t) = J_b^{+}(t)\mathcal{V}_b = J_b^{+}(t)\left[[Ad_{X_{sb}^{-1}X_{sd}}]\mathcal{V}_d(t) + K_p X_e(t) + K_i \int_0^t X_e(\sigma)d\sigma\right]$$

    where $J_b^{+}(t)$ is the pseudoinverse of the body Jacobian.

- Motion control in task space can be defined using other representations of the end-effector configuration and velocity.
- For a minimal coordinate representation of the end-effector configuration $x \in \Re^m$, the control law can be written

$$\dot{x}(t) = \dot{x}_d(t) + K_p(x_d(t) - x(t)) + K_i \int_0^t (x_d(\sigma) - x(\sigma))d\sigma$$

- For a hybrid configuration representation $X_{sb} = (R_{sb}, p)$, with velocities represented by $(\omega_b, \dot{p})$:

$$\begin{bmatrix} \omega_b(t) \\ \dot{p}(t) \end{bmatrix} = \begin{bmatrix} R_{sb}^T(t)R_{sd}(t) & 0_{3\times 3} \\ 0_{3\times 3} & I_{3\times 3} \end{bmatrix} \begin{bmatrix} \omega_d(t) \\ \dot{p}_d \end{bmatrix} + K_p X_e(t) + K_i \int_0^t X_e(\sigma)d\sigma$$

where

$$X_e(t) = \begin{bmatrix} \log(R_{sb}^T(t)R_{sd}(t)) \\ p_d(t) - p(t) \end{bmatrix}$$

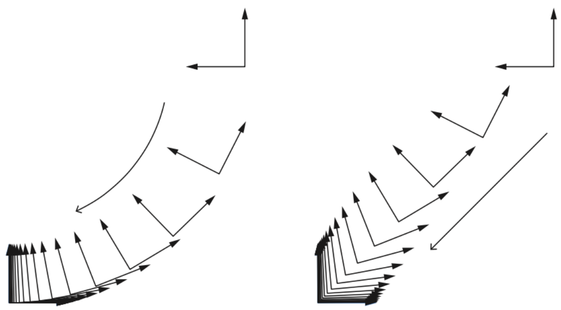

Figure 11.10: (Left) The end-effector configuration converging to the origin under the control law (11.16), where the end-effector velocity is represented as the body twist $\mathcal{V}_b$. (Right) The end-effector configuration converging to the origin under the control law (11.18), where the end-effector velocity is represented as $(\omega_b, \dot{p})$.

- Figure shows the performance of the control law (11.16), where the end-effector velocity is the body twist $\mathcal{V}_b$, and the performance of the control law (11.18), where the end-effector velocity is $(\omega_b, \dot{p})$.
- The control task is to stabilize X_{sd} at the origin from the initial configuration

$$R_0 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad p_0 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

- The feedforward velocity is zero and $K_i = 0$.
- Figure shows the different paths followed by the end-effector.
- The decoupling of linear and angular control in the control law (11.18) is visible in the straight-line motion of the origin of the end-effector frame.