Robotics, Vision, & Mechatronics for Manufacturing.

Dynamics & Control

Xu Chen

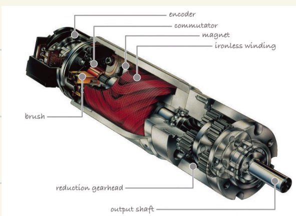
Sp 21

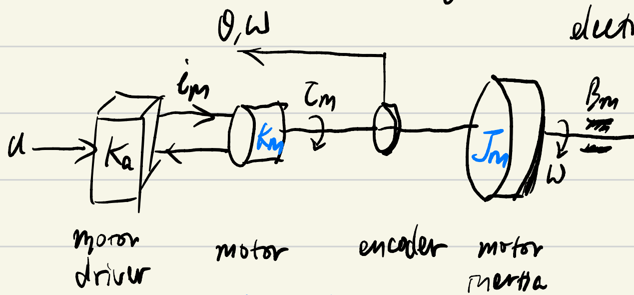
Overview:

— links are ultimately moved by forces & torques exerted by the joints.

— two schemes
- §1 independent joint control
- §2 Rigid-body equation of motion

— essentials of motor control + actuation mechanisms:
- large industrial robot: typically brushless servo motors
- small hobby robots: DC motors or stepper motors
- very large payloads e.g. in mining, forestry, construction: electro-hydraulic motors





B_m: friction coefficient ⇒ friction torque:

$$\tau_f = B\omega + \tau_c$$

$$\tau_c = \begin{cases} \tau_c^+ & \omega > 0 \\ 0 & \omega = 0 \\ \tau_c^- & \omega < 0 \end{cases}$$

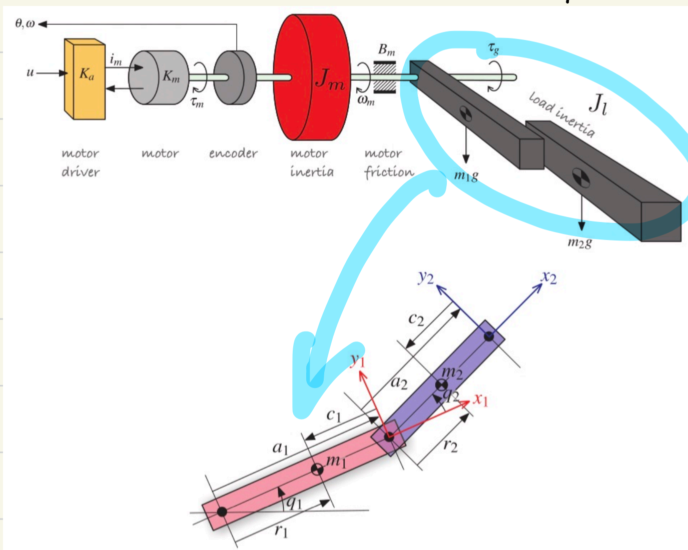
but can be more nonlinear.

$$\underbrace{i_m = K_a u \qquad \tau_m = K_m i_m \qquad \tau_m - \tau_f = J_m \dot{\omega}}$$

if no friction ⇒ $K_m K_a u = J_m \dot{\omega}$ ⇒ $\dfrac{\Omega(s)}{U(s)} = \dfrac{K_m K_a}{J_m s}$

§1 From motor control to independent robot joint control

— Effect of load mass : $\underbrace{\text{adds extra inertia \& reaction torque due to gravity}}_{\text{depends on robot configuration}}$



torque acting on joint 1: (from dynamics)

$$\tau_1 = M_{11}(q_2)\ddot{q}_1 + M_{12}(q_2)\ddot{q}_2 + c_1(q_2)\dot{q}_1\dot{q}_2 + c_2(q_2)\dot{q}_2^2 + g(q_1, q_2)$$

$$M_{11} = m_1(a_1^2 + 2a_1c_1 + c_1^2) + m_2(a_1^2 + (a_2+c_2)^2 + (2a_1a_2 + 2a_1c_2)\cos q_2)$$

$$M_{12} = m_2(a_2+c_2)(a_2+c_2+a_1\cos q_2)$$

$$c_1 = -2a_1m_2(a_2+c_2)\sin q_2$$

$$c_2 = -a_1m_2(a_2+c_2)\sin q_2$$

$$g = (a_1m_1 + a_1m_2 + c_1m_1)\cos q_1 + (a_2m_2 + c_2m_2)\cos(q_1+q_2)$$

The effect of joint motion is affected by all the other joints

From motor control to robot joint control

— Effect of gearbox

+ electric motors : compact & high speed, but low torque out of the box

+ reduction gear : trades of speed for torque

load side

motor side

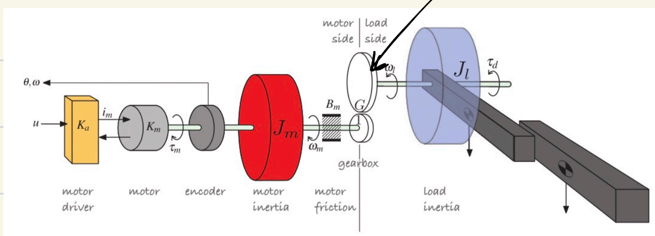
disturbance torque ⇑ reduced by G

A $G:1$ reduction drive : $\frac{\tau_l}{\tau_m} = G \iff \tau_m = \frac{1}{G}\tau_l$

$\left(\tau_l r_l = \tau_m r_m \Rightarrow \frac{r_l}{r_m} = \frac{1}{G} \right)$

⇓

inertia of the load reduced by a factor of G^2



Modeling the robot joint

Newton's law $\Rightarrow$

$$J'\dot{\omega} = \overbrace{K_m K_a u - B'\omega - \tau_c'(\omega) - \frac{\tau_{dl}(q)}{G}}^{\text{total torque}}$$

disturbance from load side

effective inertia $J' = J_m + \frac{J_l}{G^2}$

effective total viscous friction $B' = B_m + \frac{B_l}{G^2}$

effective Coulomb friction $\tau_c' = \tau_{cm} + \frac{\tau_{cl}'}{G}$

Linearized model : $J'\dot{\omega} + B'\omega = K_m K_a u$

$\xrightarrow{\text{Laplace}}$ $sJ'\Omega(s) + B'\Omega(s) = K_m K_a U(s)$

$\Rightarrow$ $$\frac{\Omega(s)}{U(s)} = \frac{K_m K_a}{J's + B'}$$

MATLAB : >> tf = p560.jointdynamics(qn)

>> tf(2)

6 transfer functions for 6 joints

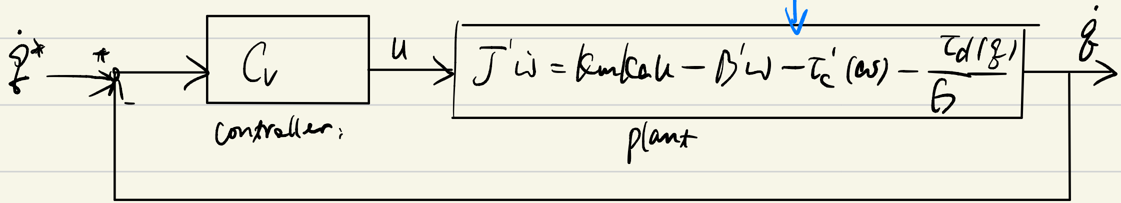
Analysis : $J' = J_m + \frac{J_l}{G^2}$: 1. inertia from the load side usually is comparable to the inertia of the motor itself.

2. $\frac{J_l}{G^2}$ varies based on robot pose.

Velocity control loop.

Now we have the joint model, we can do various controls:

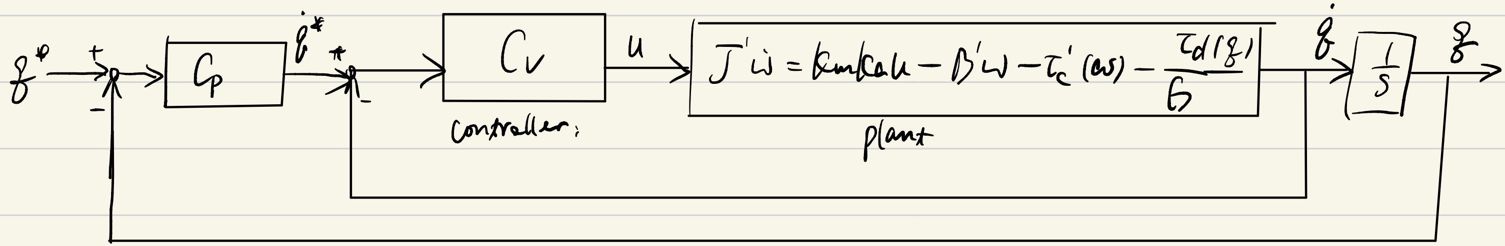
friction & disturbance compensation



e.g.

- P control : $u = K_v(\dot{q}^* - \dot{q})$
- PI control : $u = \left(K_v + \frac{K_i}{s}\right)(\dot{q}^* - \dot{q})$

Position Control loop



e.g. $\dot{q}^* = K_p(q^* - q)$

§ 2 control based on Rigid-body Equations

Matrix form of the coupled serial chain:

$$Q = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F(\dot{q}) + G(q) + J(q)^T W$$

generalized actuator force

joint-space inertia matrix

Coriolis & centripetal coupling matrix

friction

gravity

Jacobian

wrench applied at the end-effector

$$Q = D^{-1}(q, \dot{q}, \ddot{q})$$

zero vector with dimension equal to q.

MATLAB: >> Q = p560.rne(qn, qz, qz)

recursive Newton-Euler algorithm for solving the rigid-body dynamics

nominal pose

⇒ required torque to hold robot at nominal pose

>> p560.links(1).dyn