summary: robot dynamics, LQR

a linear quadratic regulator has the form of a *state feedback controller* with gain *K:*

$$\boldsymbol{u} = -K\boldsymbol{q}$$

it minimizes the quadratic cost function given by

$$J = \int_0^\infty (\boldsymbol{q}^T Q \boldsymbol{q} + \boldsymbol{u}^T R \boldsymbol{u}) dt$$

shifts the design problem from gain choices to weight choices

---

dynamics with pure vectors

$$\Sigma \boldsymbol{f} = m\dot{\boldsymbol{v}}$$

$$\Sigma \boldsymbol{\tau} = \mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega}$$

dynamics for simulation

$\Sigma \boldsymbol{f} = m\dot{\boldsymbol{v}}$	**world coordinates**
$\Sigma \boldsymbol{\tau}' = \mathbf{J}\dot{\boldsymbol{\omega}}' + \boldsymbol{\omega}' \times \mathbf{J}\boldsymbol{\omega}'$	**body-attached coordinates**
$\dot{\mathbf{R}} = \mathbf{R}\boldsymbol{\omega}'^{\times}$	**orientation**
$\dot{\boldsymbol{p}} = \boldsymbol{v}$	**world coordinates**

remarks on linear quadratic regulator

- computing *K* requires solving "algebraic riccati equation"
  - tricky to solve & requires numerical iteration ⇒ best to use software from experts

    $$K = -R^{-1}B^TP$$
    $$0 = PA + A^TP - PBR^{-1}B^TP + Q$$

    `ct.lqr(A,B,Q,R)` or MATLAB `lqr(A,B,Q,R)`
- LQR is a special case of general optimization problem: find ***u*** that minimizes given cost function and satisfies constraints (e.g. max throttle). This can be used to guide system to desired state/trajectory
- sketch for how LQR is solved:
  - use pontryagin's maximum principle (variational calculus)
  - in special case of linear, time-invariant system, quadratic cost, and infinite time horizon, result is that input is a linear function of state: ***u***=-*K**q***
- for control far away from equilibrium (e.g. aggressive maneuvers), need full nonlinear *trajectory tracking*
  - common "engineering" approach is receding horizon control (a.k.a. "model predictive control"): repeatedly calculate optimal ***u*** over a short horizon
  - biology-inspired: explore the solution space, reward if success (reinforcement learning). parameterize controller/"policy" with a neural network