Plan:
- Independent Joint Modeling
- Dynamic model recap
- Laplace transforms
- Transfer functions (short)
- Block diagrams
- P, PD, PID setpoint controllers

Independent Joint Modeling
- Controlling a whole manipulator is fairly difficult
  - We focus instead on controlling only one joint at a time
- All interaction between joints (dynamic coupling) will be classified as noise

Dynamic Model of a Robot
From the equations of motion from dynamics we have:
$J(q)\ddot{q} + C(q,\dot{q})\dot{q} + B\dot{q} + g(q) = f$

$J(q)\ddot{q}$ - inertia
$C(q,\dot{q})\dot{q}$ - coriolis/centrifugal forces
$B\dot{q}$ - viscous friction (damping)
$g(q)$ - gravitational forces
f - torque/force from actuators

Coriolis/centrifugal
Gravity
Coupling ($J(q)\ddot{q}$ -> $J\ddot{q}$)
} -> D - disturbance

$J\ddot{q} + B\dot{q} + D = f$

- Not the same notations as in velocity kinematics: J is not Jacobian, D matrix is not the one from dynamics
- Inertia and inertial forces are not the same thing, they are loosely related, if at all
- Coriolis and centrifugal forces are often classified as inertial (fictitious) forces

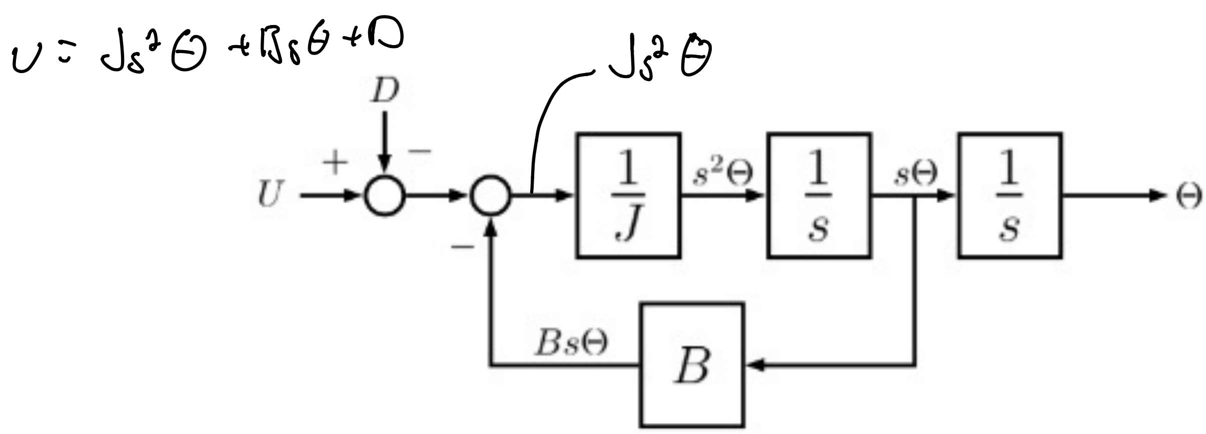
Laplace Transform
  - Time -> frequency
  - Ordinary differential equation -> linear equation
  - Differentiation in time -> multiplication by s in Laplace
  - Integration in time -> division by s in Laplace

E.g: $J\ddot{q} + B\dot{q} + D = f$ -> $Js^2\Theta + Bs\Theta + D = f$

Transfer functions
- Y(s) = H(s)X(s)
- Y(s) - system output
- X(s) - system input
- H(s) - transfer function (transforms input into output)
- Rearrange equations to find: H(s) = Y(s)/X(s)
- If numerator (Y) is set to 0, we can find the poles of the system by solving for s
- If denominator (X) is set to 0, we can find the zeros
  - Useful for determining stability of the system

Block Diagrams
- Helpful for visualizing equations and feedback/feedforward loops

Block diagram for the generic system:



Block diagram with basic building blocks

- On the left side, we start with the full torque equation denoted as U (control effort)
- As the signal proceeds through the blocks, we "peel" terms off one by one, until we end up with only Θ
- The whole diagram represents a robotic system
  - Can be used as a building block for a larger diagram of controller blocks

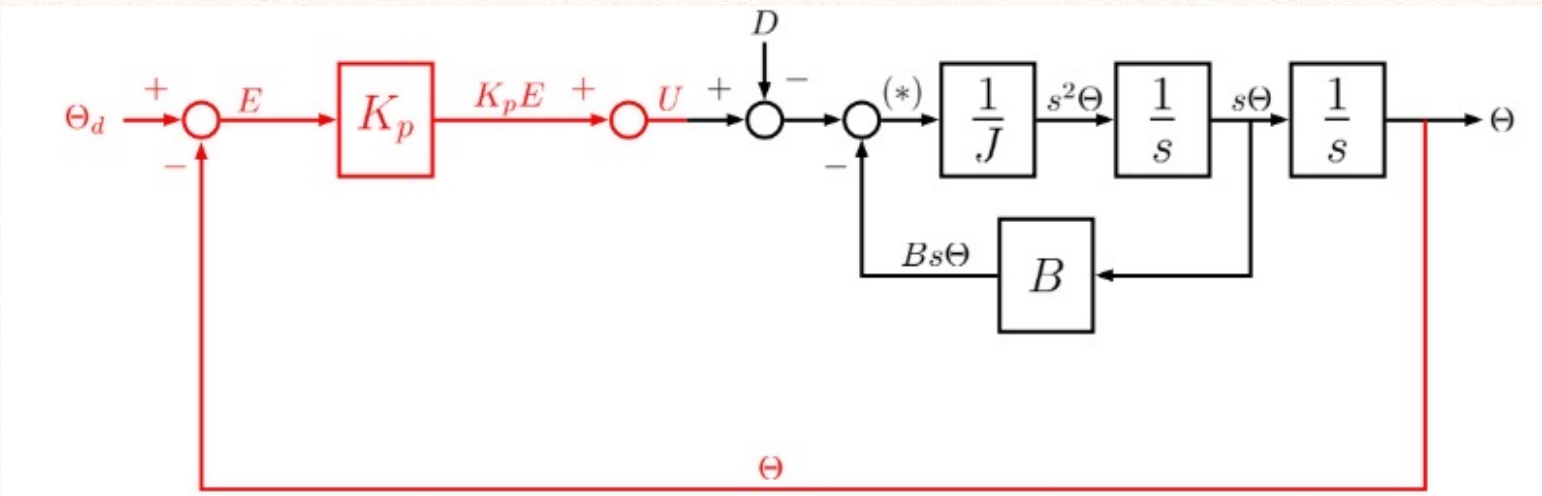
Setpoint controllers

- Controllers that drive a robot to a set point
- Current angle: Θ
- Desired angle: Θ_d
- Error: $e(t) = \Theta_d - \Theta$
- Controllers use the error term to calculate the control effort U (output torque/force)
- Controllers try to reduce the error to 0

- Types of setpoint controllers
  - Proportional (P)
  - Proportional Derivative (PD)
  - Proportional Integral Derivative (PID)
  (- Proportional Integral (PI))

P controller

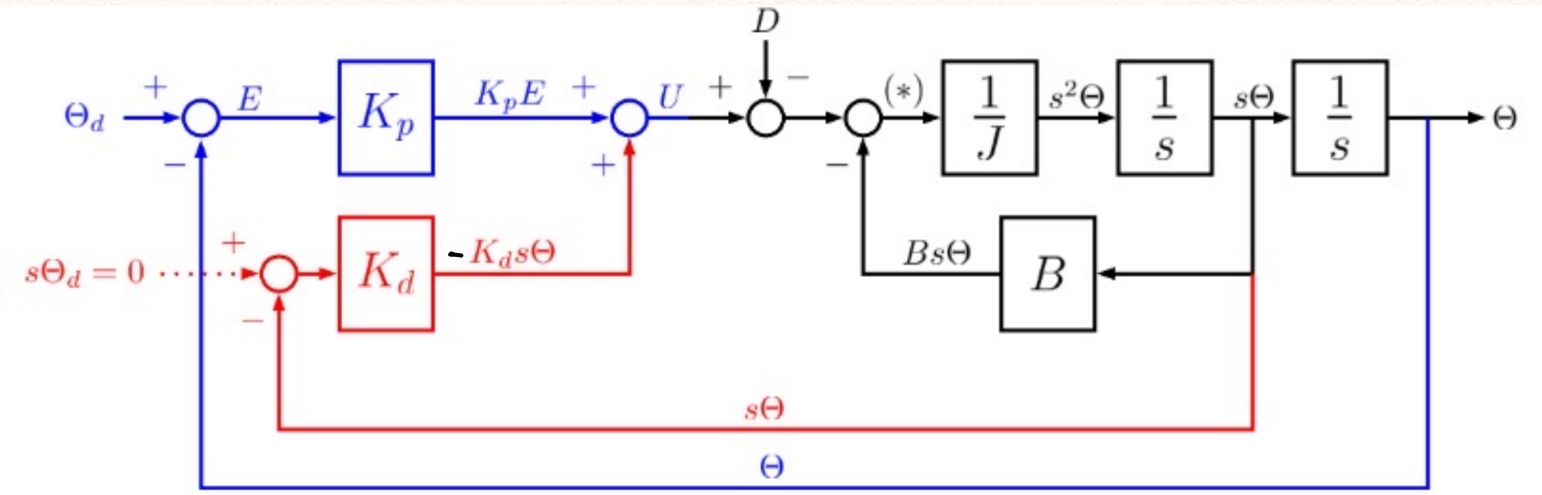
- $U(t) = K_p e(t)$
- Control effort proportional to the controller
- Laplace: $U(s) = K_p E(s)$
- Block diagram:



- Increased K_p gives:
  - Faster response
  - Decrease in steady state error
  - Increased oscillations
- Proportional term by itself will not eliminate the error

PD controller

- Builds on the P controller by adding a derivative term
- Derivative term represents how fast the error changes - used to "predict" future error
- $U(t) = K_p e(t) + K_d \dot{e}(t)$ $\quad \dot{e} = \dot{\theta}_d - \dot{\theta}$ (with $\dot{\theta}_d = 0$) $\quad K_d \dot{e} = -K_d \dot{\theta}$
- Laplace: $U(s) = K_p(\Theta_d - \Theta) - K_d s\Theta$
  - Write out the error terms, since the desired velocity for theta is 0 (no oscillations) we can simplify
- Block diagram:

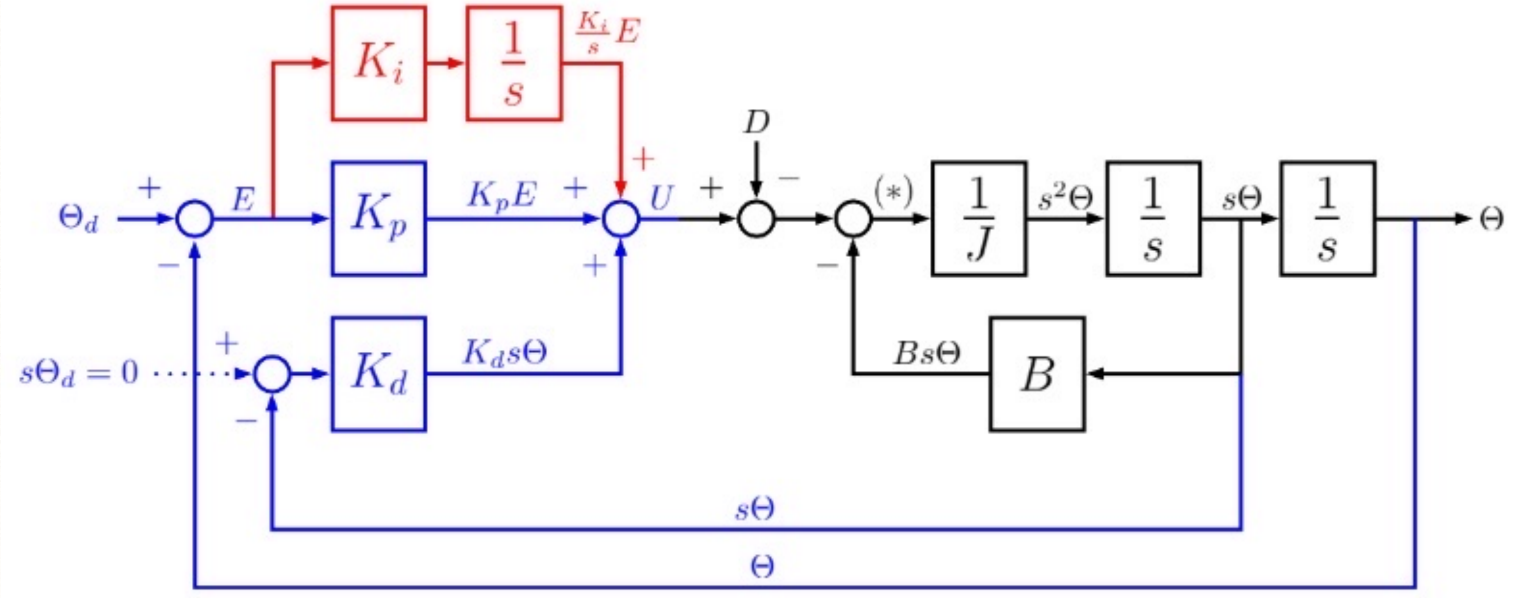


- Increased K_d reduces oscillations, but can make the robot stop before reaching the desired point

PID controller

- Builds on the PD controller by adding an integral term
- Integral term accumulates past errors over time
- $U(t) = K_p e(t) + K_d \dot{e}(t) + K_i \int e(t)dt$
- Laplace: $U(s) = (K_p + K_d s + \frac{K_i}{s})E(s)$

- Block diagram:



- Good choice of K_i gives the necessary push to prevent early stopping due to the derivative term, eliminating the error
- If K_i is too big, expect an overshoot, oscillations and instability