Kuwait University
College of Engineering and Petroleum

جامعة الكويت
KUWAIT UNIVERSITY

ME417 CONTROL OF MECHANICAL SYSTEMS

Part II: Controller Design Using Root-Locus

Lecture 6: Improving Transient & Steady State Response

Spring 2021

Ali AlSaibie

- Objectives:
  - Combine the ideal integral compensator and ideal derivative compensator (PID Control)
  - Introduce the concept of cascaded control loops.
- Reading:
  - *Nise: 9.4-9.5*

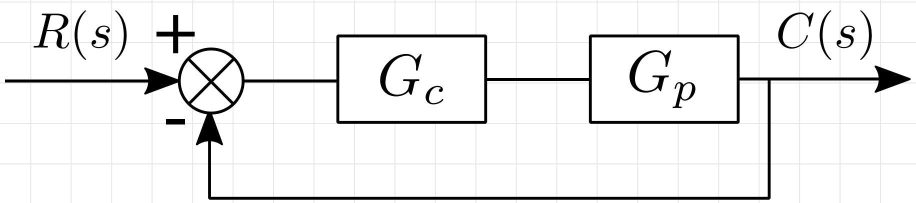
- We have discussed applying either a PI or a PD controller independently
  - A PI Controller helps eliminate steady-state error
    - But can also improve transient response via gain adjustment
  - A PD Controller helps improve the transient response and stabilize an unstable system.
    - It can dramatically affect the shape of the root-locus: the possible closed-loop pole locations
- What if we want to achieve zero steady-state error AND achieve a specific transient response behavior?
  - We can combine the PD and PI controllers into a PID controller.

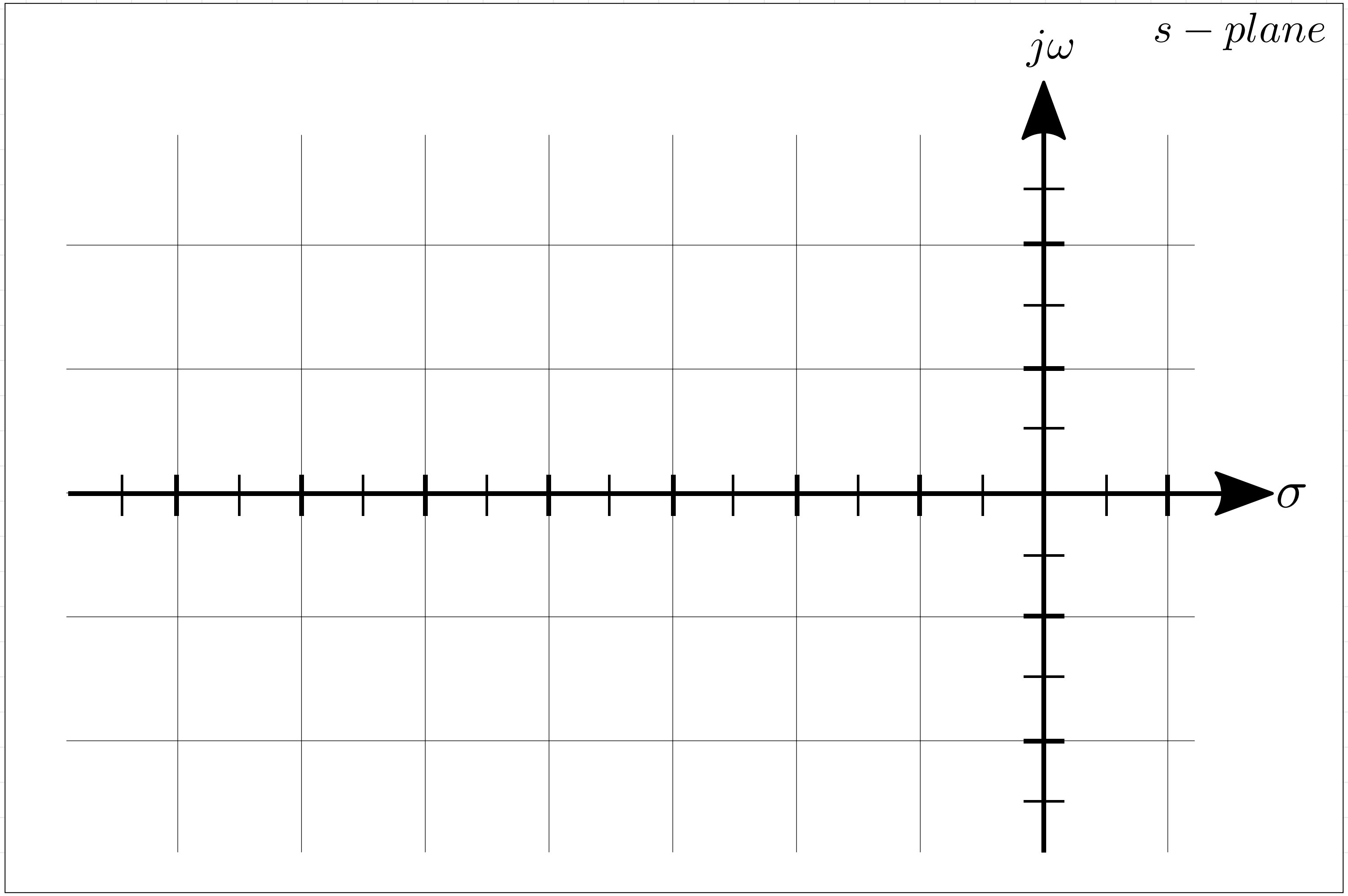
- The design process is as follows:
  1. Starting with an uncompensated system, draw the root-locus
  2. Specify your desired transient response by locating the desired location of the dominant closed-loop poles
  3. If the uncompensated root-locus does not intersect our desired point, add a compensator zero (PD Controller) to achieve the intersection
     - Apply the angle condition to find the compensator zero location
  4. Find the gain of the PD controller to place the closed-loop poles at the desired location
     - Apply the magnitude condition to find K
  5. Apply a PI controller (with unity gain), with a zero close to the origin, to increase the system type and eliminate steady-state error.

جامعة الكويت
KUWAIT UNIVERSITY

Design a controller for the given system to achieve: a critically damped response, settling time of $T_s = .25s$ and zero steady-state error

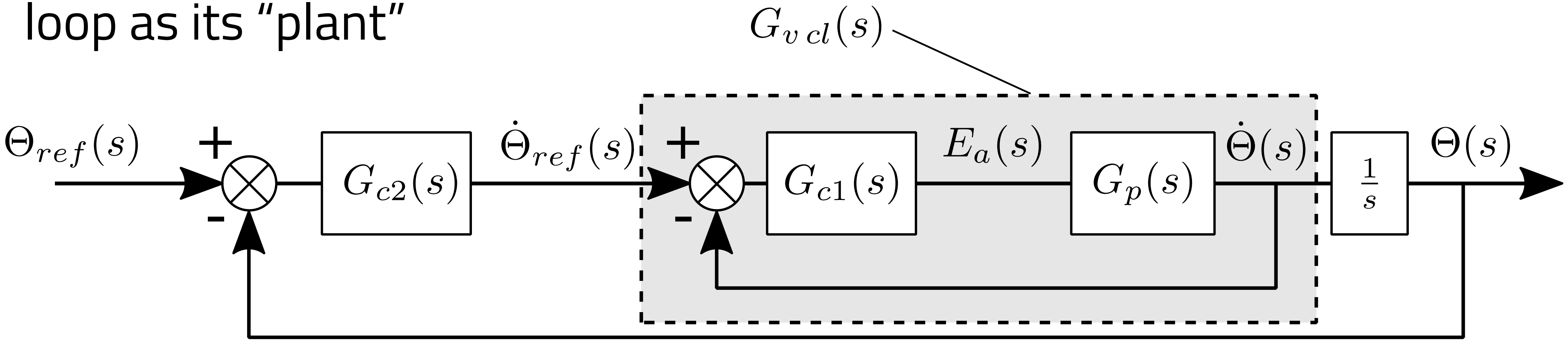
$$G_p(s) = \frac{16}{(s+1)(s-2)}$$





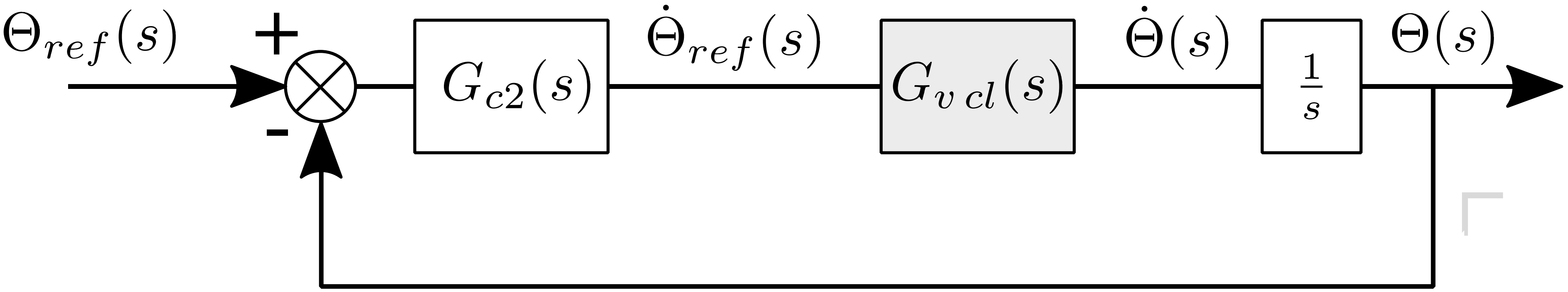
- In real-world applications, controllers are often designed around other feedback control systems
- An autonomous car may have a navigation controller built on top of a road position controller, and the position controller is built on top a car velocity controller, and the latter is built on top an engine controller or electric motor controller.
- A drone/multirotor will have a rate (angular velocity) controller, inside an attitude (angular position) controller, inside a position controller, inside a mission navigator.
- It is more practical to break-down components, and treat them at a manageable level then proceed to integrate them with higher level components and systems.

- Consider the two motor control loops
  - The inner loop is a velocity control loop, which can be tuned to achieve a specific performance, in other words, a specific closed-loop transfer function
  - The outer loop is the position control loop, which treats the inner closed-loop as its “plant”



- Note that the position controller design becomes a new problem with a new plant transfer function $G_{v\,cl}(s)$, which itself can be designed to have a specific form
- Assume we tune the velocity controller to achieve a critically damped response, the closed loop transfer function will then be

$$G_{v\,cl}(s) = \frac{K}{(s+a)^2} \approx \frac{K}{s+b},$$ an approximately first order response.



Cascaded Control Loops

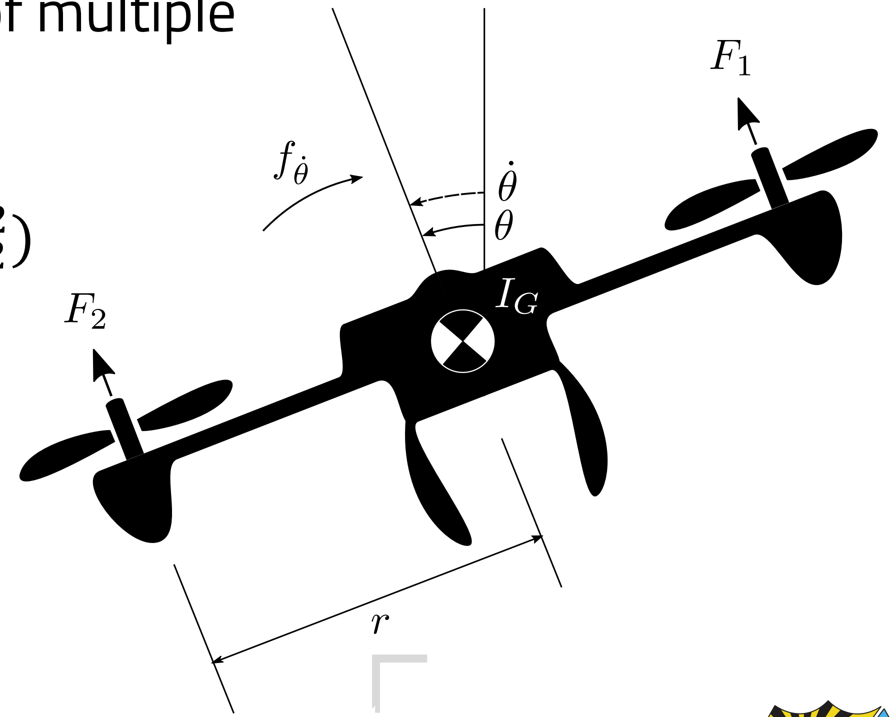
- Another great benefit of this design approach is the ability to confidently design, by software simulation, the outer (higher level) control loops
  - Assuming we can achieve, on the real world system, a desired closed-loop response.
- Example: We can tune the velocity controller of the motor on the actual motor until we are satisfied, then given the real closed-loop velocity response
  - We can confidently design, offline, the position controller and other higher level control loops.
  - Then integrate them rapidly on the real system, with minimal troubleshooting and testing.
  - The design cycle and complexity will be greatly reduced.
- How can cascading control loops help in designing space rocket launch control systems?

جامعة الكويت
KUWAIT UNIVERSITY

- Consider a simplified quadrotor pitch θ control model
- Thrust F is a function of the rotor's speed $F = k_T \omega^2$
  - k_T the thruster constant: a function of the propeller and motor
- $f_{\dot{\theta}}$ is the drag force, a nonlinear component that is a function of multiple parameters
- The equation of motion for the pitch system

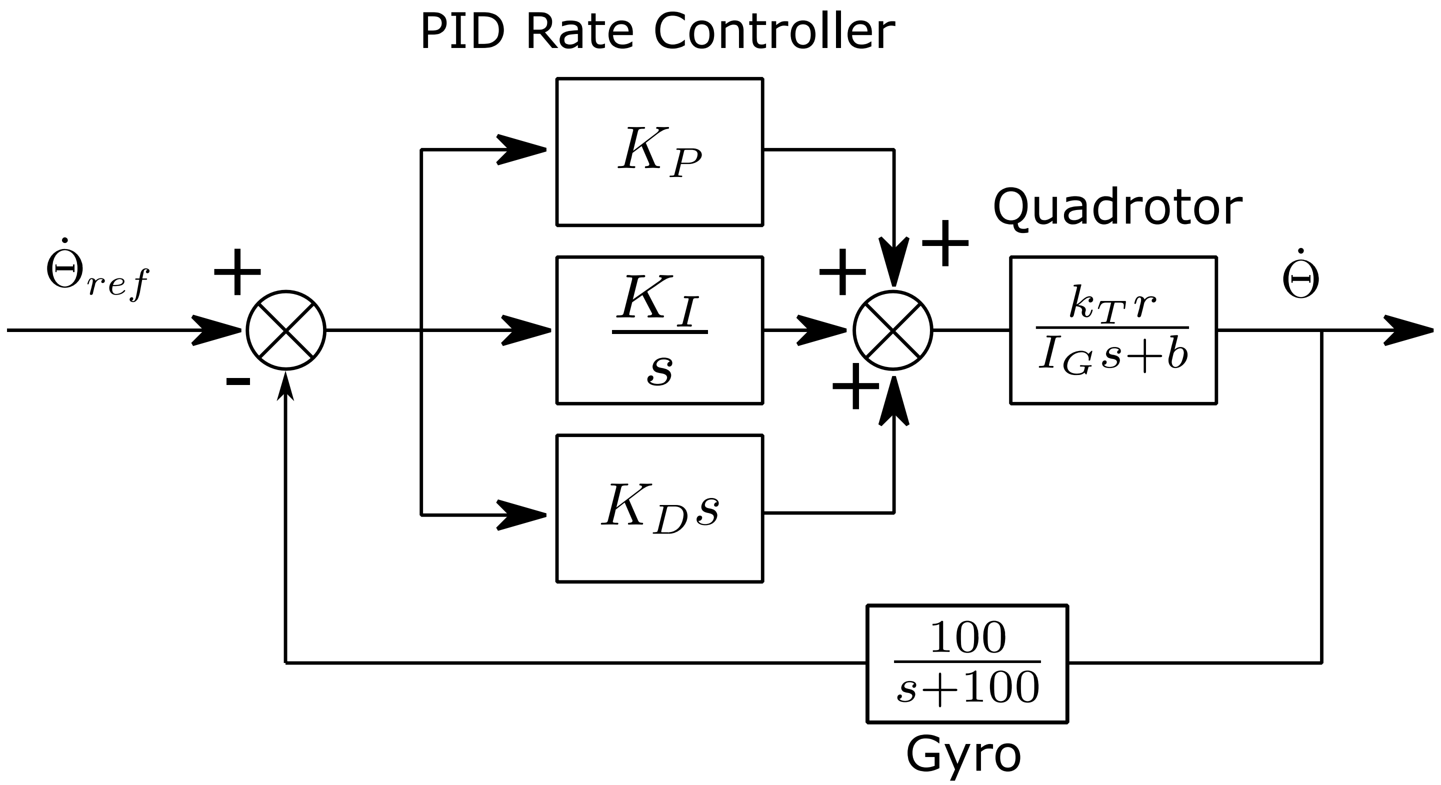
$$I_G \ddot{\theta} + f_{\dot{\theta}} = M_G = (F_1 - F_2)r = k_T r(\omega_1^2 - \omega_2^2)$$

- Assuming linear damping around an operating point: $f_{\dot{\theta}} = b\dot{\theta}$
- Simplifying the input speed to be $\Delta\omega_{sq} = (\omega_1^2 - \omega_2^2)$
- The transfer function can be written as

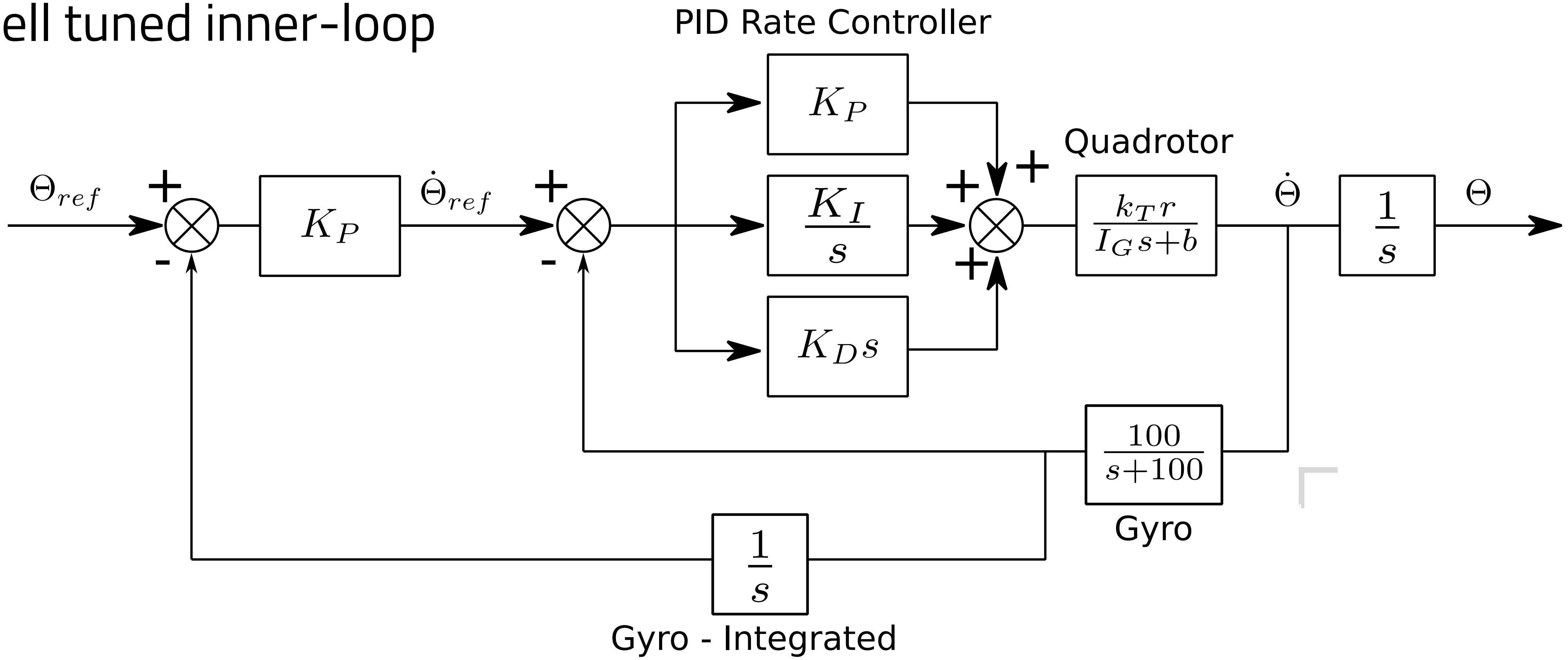


$$G(s) = \frac{\dot{\Theta}(s)}{\Delta\omega_{sq}} = \frac{k_T r}{(I_G s + b)}$$

- A gyroscope is a sensor that provides angular rotation rates, which can be modeled as a fast first order system.
- We can design a PI controller to eliminate steady-state error, increase the settling time and maximize the damping ratio.



- To control the pitch angle, we design an outer loop around the rate control loop.
- A proportional (gain only) controller is sufficient as the outer loop controller, for a well tuned inner-loop



- Note that the inner loop can be treated as just a plant transfer function (The closed-loop transfer function)