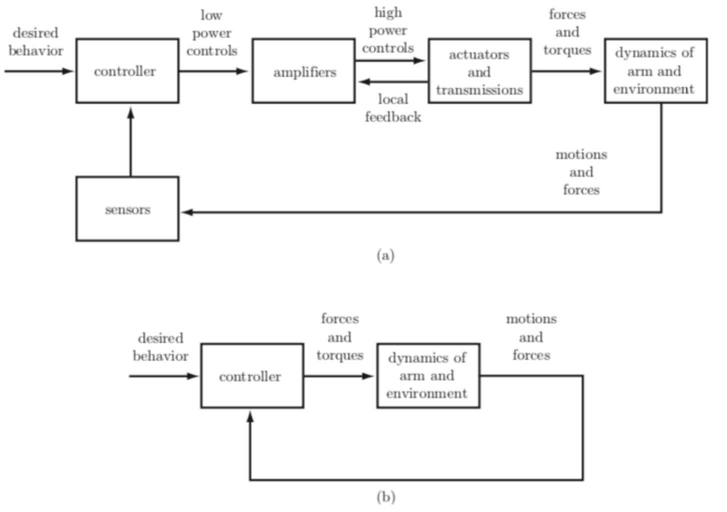
Introduction to Robotics
Lec 16: Basics of Robot control

Control diagram

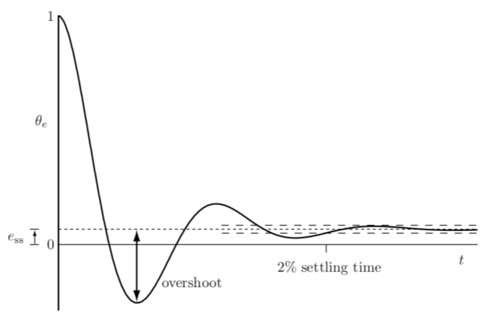


Given sensor readings, how to design actuator torques?

Error dynamics

- Denote a desired sequence of joint values by $\theta_d(t)$, and the actual joint values by $\theta(t)$. We define the joint error as
$$\theta_e := \theta_d(t) - \theta(t).$$
- The differential equation governing the dynamics of $\theta_e(t)$ is called the *error dynamics*.
- We analyze/design a controller based on the error dynamics here (other point of views are possible!)
- An ideal controller would be so that whenever $\theta_e(t) \neq 0$, it is brought to zero instantaneously. This ideal scenario is of course not realisable.
- The commonly used/standard situation to design/analyze a controller performance is the following: we assume that at $t = 0$, we have
$$\theta_e(0) = 1 \text{ and } \dot{\theta}_e(0) = \ddot{\theta}_e(0) = \cdots = 0.$$
We benchmark controllers' performances from that initial state.
- We assume here for now that the error dynamics is *linear*. Robots have nonlinear dynamics, but if the error is small, the linear approximation yields good results. Methods to obtain linear approximations are given later in the lecture.

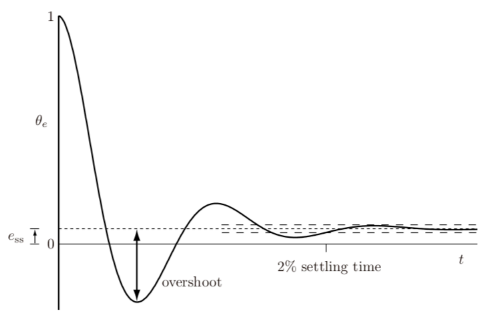
Error dynamics



- We plot a typical *linear* system error response above.
- We can split the response into a *transient response* (time during which the dynamics is not negligible) and a *steady-state* response (when we are almost at equilibrium).
- The steady-state response is characterized by the steady-state error

$$\theta_{e,ss} := \lim_{t \to \infty} \theta_e(t).$$

Error dynamics



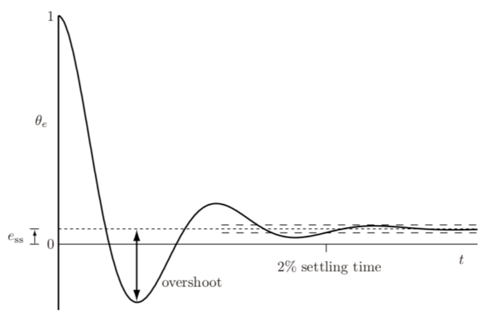
- The transient response is characterized by the *overshoot* (how far the error goes past its steady-state):

$$\text{overshoot} = \left| \frac{\theta_{e,min} - \theta_{e,ss}}{\theta_e(0) - \theta_{e,ss}} \right| \times 100\%$$

and the *2% settling-time*: the time needed so that

$$\|\theta_e(t) - \theta_{e,ss}\| \leq 0.02(\theta_e(0) - \theta_{e,ss})$$

Error dynamics



- A good error response is characterized by
  1. small steady-state error
  2. small overshoot
  3. short settling time

Linear Error dynamics

- A general linear error dynamics is given by

$$a_p\theta_e^{(p)} + a_{p-1}\theta_e^{(p-1)} + \cdots + a_2\ddot{\theta}_e + a_1\dot{\theta}_e + a_0\theta_e = c.$$

- The above equation is called homogeneous if $c = 0$ and non-homogeneous if $c \neq 0$.
- Introducing new variables, we can write the equation as a first order *system* of equations:

$$\begin{aligned}
x_1 &:= \theta_e \\
x_2 &:= \dot{x}_1 = \dot{\theta}_e \\
&= \cdots \\
x_p &:= \dot{x}_{p-1} = \theta_e^{(p-1)} \\
\dot{x}_p &= -a_0/a_p x_1 - a_1/a_p x_2 - \cdots - a_{p-1}/a_p x_p
\end{aligned}$$

Linear Error dynamics

- We can write the previous equation in matrix/vector form: $\dot{x} = Ax$ where $x = (x_1, \ldots, x_p)$ and

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ -a'_0 & -a'_1 & -a'_2 & \ddots & -a'_{p-1} & -a'_{p-1} \end{bmatrix}$$

  with $a'_i := a_i / a_p$. This matrix is called a *companion* matrix.
- Recall that the solution of $\dot{x} = Ax$ is $x(t) = e^{At}x(0)$, as we saw earlier in the course. This can be verified by plugging the definition of e^{At} into the differential equation.

Stability of linear systems

- The linear autonomous system $\dot{x} = Ax$ is said to be *stable* if the real parts of the eigenvalues of A are *negative*. It is unstable otherwise.
- The state of an unstable system is asymptotically infinite for some initial conditions: there exists x_0 so that

$$\lim_{t \to \infty} \|x(t)\| = \lim_{t \to \infty} \|e^{At} x_0\| = \infty$$

- It is easy obtain a stability criterion in case A is *diagonalizable*, that is A can be written as $A = PDP^{-1}$ for some diagonal matrix D and invertible matrix P. In this case, D has the eigenvalues of A in its diagonal and

$$e^{At} = Pe^{Dt}P^{-1}.$$

Now recall that e^{dt}, for $d = a + bi \in \mathbb{C}$ is $e^{dt} = e^{at}(\cos bt + i \sin bt)$. If $a > 0$, e^{at} grows large. If $a < 0$, e^{at} goes to zero.

A requirement of any controller: the error dynamics is **stable**

First order dynamics

- We now consider the angle $\theta(t)$ at a joint, omitting the index i for now.
- If the error dynamics is of first order, then it can be written as

$$\dot{\theta}_e(t) + \frac{k}{b}\theta_e(t) = 0$$

- You can think of this as a P controller (P is for proportional), with parameter k: if $\theta_e > 0$, then change θ_e so that it decreases, hence choose $k > 0$. We come back to this later.
- The solution of this equation is

$$\theta_e(t) = e^{-k/bt}\theta_e(0).$$

  Set $\tau = b/k$. There is no overshoot, and

$$\frac{\theta_e(t)}{\theta_0} = 0.02 = e^{-t/\tau} \Rightarrow t = 3.91\tau$$

Second order dynamics

- A second order dynamics $\ddot{\theta}_e + c_1\dot{\theta}_e + c_2\theta_2 = 0$ is written in the following standardized form:

$$\ddot{\theta}_e(t) + 2\zeta\omega_n\dot{\theta}_e(t) + \omega_n^2\theta_e(t) = 0,$$

  where ω_n is called the natural frequency of the system, and ζ is called the damping ratio.

- We can write this system in matrix form. The characteristic polynomial of the corresponding A matrix is then

$$s^2 + 2\zeta\omega_n s + \omega_n^2.$$

  The roots of this polynomial are the eigenvalues of A, and are

$$s_1 = -\zeta\omega_n + \omega_n\sqrt{\zeta^2 - 1} \text{ and } s_2 = -\zeta\omega_n - \omega_n\sqrt{\zeta^2 - 1}.$$

  The dynamics is stable if and only if ω_n and ζ are positive.

Second order dynamics: three cases

- **overdamped:** This is the case $\zeta > 1$. In this case, the roots s_1, s_2 are real, distinct, and the solution is

$$\theta_e(t) = c_1 e^{s_1 t} + c_2 e^{s_2 t}$$

  for some constants c_1, c_2 that we can compute from the knowledge of the initial conditions $\theta_e(0), \dot{\theta}_e(0)$.
  The time constant is given by the less negative root (real part) s_1 or s_2. Precisely, if the real parts are a_1 and a_2 with $a_1 < a_2 < 0$, then $t = |3.91/a_2|$
- **critically damped:** This is the case $\zeta = 1$. In this case, the roots s_1, s_2 are equal and real, and the solution is

$$\theta_e(t) = (c_1 + c_2 t)e^{-\omega_n t}$$

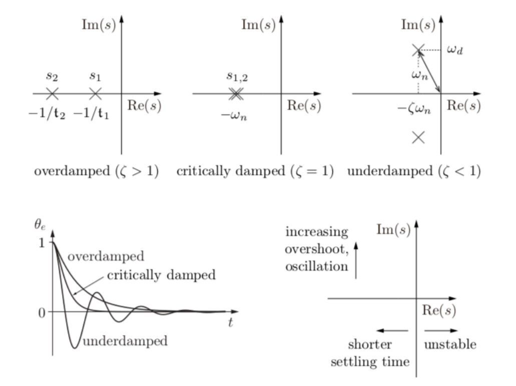
  for some constants c_1, c_2 that we can compute from the knowledge of the initial conditions $\theta_e(0), \dot{\theta}_e(0)$.
  The time constant is $\frac{1}{\omega_n}$.

- **underdamped:** This is the case $\zeta < 1$. The roots are complex conjugate

$$s_{1,2} = -\zeta\omega_n \pm j\omega_d,$$

where $\omega_d = \omega_n\sqrt{1-\zeta^2}$ is the *natural frequency*. The solution is

$$\theta_e(t) = (c_1 \cos\omega_d t + c_2 \sin\omega_d t)e^{-\zeta\omega_n t}.$$



Feedback/feedforward control with Velocity inputs

- A *feedback* controller is a controller that uses sensors to obtain information about the current state of the system, and act depending on it.
- A *feedforward* controller applies a predetermined control sequence without taking into account sensor measurements.
- Feedforward control is advisable only when sensing is impossible or unavailable, as it can yield large errors.
- For example: Consider the model

$$\tau(t) = M(\theta(t))\ddot{\theta}(t) + h(\theta, \dot{\theta}).$$

In order to find the desired torques, it suffices to plug-in $\theta(t) \leftarrow \theta_d(t)$:

$$\tau_c(t) = M(\theta_d(t))\ddot{\theta}_d(t) + h(\theta_d, \dot{\theta}_d).$$

If there are modelling errors, or small errors in what we believe the current state of the system is, these errors can grow.

Motion Control with Velocity inputs

- We assume here that we have a direct control over the velocity of the joints angles. Many off-the-shelf motors/actuators have a velocity input that can be used for that purpose (amplifiers are used to try to make this assumption true).
- If the inertia is large relative to the power of the motor, we have to consider torque control.
- We assume that we have computed a desired trajectory for the joint variables. Denote it $\theta_d(t) \in \mathbb{R}^n$. This is a vector whose entries are the desired joint values at time t.
- We thus want to have

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

  i.e. the actual velocities should be equal to the desired ones.
- We now focus on the control a single joint with angle θ. Since we assume that the joints are independent, the procedure to control several joint is to control each of them individually.

Feedback control with Velocity inputs

- Assume that we have a sensor reading the current position of the joint: $\theta(t)$.
- A *proportional* or P controller uses this information to control the system according to

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

where K_p is a real parameter to determine. Recall that we defined $\theta_e := \theta_d - \theta$.

Feedback control with Velocity inputs

- To analyze the P controller, we first assume that $\theta_d(t) \equiv 0$. Taking another constant value besides zero does not change the analysis.
- Recall that $\dot{\theta} = K_p \theta_e$. Adding $\dot{\theta}_d = 0$, we get

$$\dot{\theta}_e(t) = -K_p \theta_e(t) \Rightarrow \theta_e(t) = e^{-K_p} \theta_e(0).$$

- For any $K_p > 0$, we see that $\theta_e(\infty) = 0$.
- Using the results derived in the previous lecture, we get that the 2% settling time is $4/K_p$: choosing the feedback gain K_p large will improve the reaction time.

Feedback control with Velocity inputs

- We now return to the case of a time-varying $\theta_d(t)$. The general rule of thumb is that if $\theta_d(t)$ varies slowly, and if K_p is large enough, the P controller will be able to track $\theta_d(t)$ well.
- Let us try to quantify this. Assume that $\dot{\theta}_d(t) = c$, for a fixed constant c.
- The dynamics is then

  $$\dot{\theta}_e(t) + K_p\theta_e(t) = c,$$

  which has a solution

  $$\theta_e(t) = \frac{c}{K_p} + \left(\theta_e(0) - \frac{c}{K_p}\right) e^{-K_p t}.$$

  If K_p is large, we see that $\theta_e(t)$ indeed converges to a small value, and we have good tracking.

PI with Velocity inputs

- The previous analysis tells us to choose K_p as large as possible in order to reduce the value $\theta_e(\infty) = \frac{c}{K_p}$. However, it is not practical to take K_p too large. Is there a way to cancel this error?
- A PI controller uses, in addition to $\theta_e = \theta_d - \theta$, the *integral* of the error: $\int_0^t \theta_e(s)ds$. The resulting system is (assuming θ_d is constant)

$$\dot{\theta}_e(t) = K_p\theta_e(t) + K_i \int_0^t \theta_e(s)ds,$$

  for constants K_i and K_p to be determined.
- Assuming $\dot{\theta}_d(t)$ constant, we get

$$\dot{\theta}_e + K_p\theta_e + K_i \int_0^t \theta_e(s)ds = c.$$

- Taking the derivative, we obtain

$$\ddot{\theta}_e + K_p\dot{\theta}_e + K_i\theta_e = 0$$

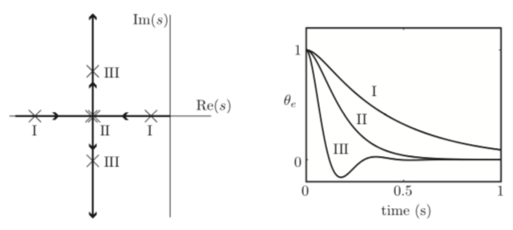
PI with Velocity inputs

- Rewrite the previous equation in canonical form $\ddot{\theta}_e(t) + 2\zeta\omega_n\dot{\theta}_e(t) + \omega_n^2\theta_e(t) = 0$, we get $\omega_n = \sqrt{K_i}$ and $\zeta = K_p/(2\sqrt{K_i})$.
- The roots of the characteristic equation are

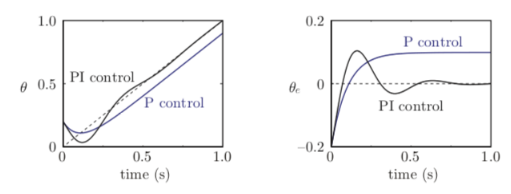
$$s_{1,2} = -\frac{K_p}{2} \pm \sqrt{\frac{K_p^2}{4} - K_i}.$$

  If both $K_p, K_i > 0$, the dynamics is stable.
- We have two parameters to shape the behavior of the controlled system. A common practice is to fix one parameter, and vary the other until we obtain the desired behavior.
- For example, we can fix $K_p = 20$, and look at what the roots are as K_i varies from 0 to $+\infty$. This is called a *root locus* analysis, which we do not cover here.

PI with Velocity inputs

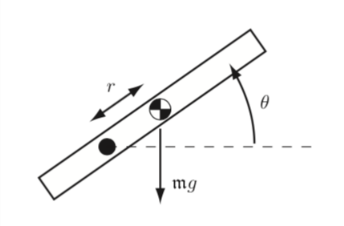


- The plot above-left is called the root locus.
- The solutions $\theta_e(t)$ are plotted on the right. We see that in all cases, $\theta_e(\infty) = 0$, as desired.



Motion Control with T inputs

- We assumed so far that we could directly control the velocity of the joint angles.
- This assumption is valid only when the masses/inertia involved are relatively small compared to the power of the actuators.
- We look at a simple case: one link with an R joint, as shown below



- The dynamics is given by

$$\tau = M\ddot{\theta} + mgr\cos\theta,$$

with *M* the inertia of the link, *m* its mass.

Motion Control with T inputs

- Adding friction proportional to the joint velocity $\tau_{fric} = b\dot{\theta}$, the model becomes

$$\tau = M\ddot{\theta} + mgr\cos\theta + b\dot{\theta}$$

- We assume given a desired $\theta_d(t)$, and set $\theta_e(t) := \theta_d(t) - \theta(t)$.
- Let us assume for now that the robot moves in a horizontal plane, and thus we can set $g = 0$:

$$\tau = M\ddot{\theta} + b\dot{\theta}.$$

- A *PD controller*, which stands for *proportional-derivative*, uses knowledge of $\theta(t)$ *and* its derivative $\dot{\theta}(t)$. A *PD* controller is of the form

$$\tau = K_p(\theta_d - \theta) + K_d(\dot{\theta}_d - \dot{\theta})$$

Motion Control with T inputs

- The system dynamics with a *PD* controller is

$$M\ddot{\theta} + b\dot{\theta} = K_p(\theta_d - \theta) + K_d(\dot{\theta}_d - \dot{\theta})$$

- We assume that $\theta_d = c$ a constant. The dynamics is then

$$M\ddot{\theta}_e + (b + K_d)\dot{\theta}_e + K_p\theta_e = 0.$$

- This yields the coefficients $\omega_n = \sqrt{\frac{K_p}{M}}$ and $\zeta = \frac{b+K_d}{2\sqrt{K_p M}}$. For stability, both $b + K_d$ and K_p need to be positive.
- Similarly to what we did for the *PI* controller, we can set the values of the coefficients K_p and K_i to obtain a desired dynamic behavior.

Motion Control with T inputs

- We now return to the case where the link moves in the vertical plane:

$$M\ddot{\theta}_e + (b + K_d)\dot{\theta}_e + K_p\theta_e = mgr\cos\theta.$$

- Let us assume that we want to stabilize the link at a configuration θ_d. At an equilibrium, $\dot{\theta}_e = 0$, which implies that

$$\theta_e = \frac{1}{K_p} mgr\cos\theta_e.$$

  We see that *all* PD controllers will have a tracking error (of a constant)! That is, for any choice of coefficients K_p, K_d, we get that when $\dot{\theta}_e = 0$, $\theta_e \neq 0$, unless $\theta_d = \pm\pi/2$.

- We now show how to handle the nonlinear term, through linearization.

System linearization

- To linearize a system in the variables x, y, say

$$\frac{d}{dt}\begin{pmatrix} x \\ y \end{pmatrix} = F(x, y),$$

replace every occurence of x by $x + \Delta x$ and every occurence of y by $y + \Delta y$:

$$\frac{d}{dt}\begin{pmatrix} x + \Delta x \\ y + \Delta y \end{pmatrix} = F(x + \Delta x, y + \Delta y).$$

- Then choose a trajectory $x(t), y(t)$ (i.e., a solution of the equation) around which to linearize. Expand every nonlinear terms using a first order Taylor expansion around $x(t)$ and $y(t)$:

$$\frac{d}{dt}\begin{pmatrix} x(t) + \Delta x \\ y(t) + \Delta y \end{pmatrix} = F(x(t), y(t)) + \begin{pmatrix} \frac{\partial F_1}{\partial x} & \frac{\partial F_1}{\partial y} \\ \frac{\partial F_2}{\partial x} & \frac{\partial F_2}{\partial y} \end{pmatrix} |_{x=x(t), y=y(t)} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} + hot.$$

System linearization

- We obtain from the previous equation that

$$\frac{d}{dt}\begin{pmatrix}\Delta x(t)\\ \Delta y(t)\end{pmatrix} = \begin{pmatrix}\frac{\partial F_1}{\partial x} & \frac{\partial F_1}{\partial y}\\ \frac{\partial F_2}{\partial x} & \frac{\partial F_2}{\partial y}\end{pmatrix}\Big|_{x=x(t),y=y(t)}\begin{pmatrix}\Delta x\\ \Delta y\end{pmatrix}.$$

  The above system is called the *linearized* system. It describes how small variations $\Delta x, \Delta y$ around a nominal trajectory evolve.
- The most common case is to linearize around an *equilibrium*: an equilibrium is a point x_0, y_0 for which the derivative vanishes. Said otherwise, a point so that $F(x_0, y_0) = 0$.

System linearization

- We now return to our system:

$$M\ddot{\theta}_e + (b + K_d)\dot{\theta}_e + K_p\theta_e = mgr\cos\theta.$$

- We look for an equilibrium, that is a point so that $\dot{\theta}_e = 0$. We saw earlier that this implies $\theta_e = \frac{1}{K_p} mgr\cos\theta_e$. We can solve this equation numerically to obtain an equilibrium point. Call it θ_0.
- We linearize the system around this point:

$$M\frac{d^2}{dt^2}(\theta_0 + \Delta\theta_e) + (b + K_d)\frac{d}{dt}(\theta_0 + \Delta\theta_e) + K_p(\theta_0 + \Delta\theta_e) = mgr\cos(\theta_0 + \Delta\theta_e),$$

  which yields the linearized system

$$M\frac{d^2}{dt^2}\Delta\theta_e + (b + K_d)\frac{d}{dt}\Delta\theta_e + K_p\Delta\theta_e = -(mgr\sin\theta_0)\Delta\theta_e$$

  Set $\bar{K}_p = K_p + mgr\sin\theta_0$. The linearized system is then:

$$M\Delta\ddot{\theta}_e + (b + K_d)\Delta\dot{\theta}_e + \bar{K}_p\Delta\theta_e = 0$$

- We could perform an analysis of the linearized system and choose the coefficients K_P, K_D to stabilize $\Delta\theta_e$ at zero. Note that this would imply that we stabilize around $\theta_e \neq 0$ as found above.
- Adding an integral term to this design would not help! The design enforces correctly that $\Delta\theta_e = 0$, so from the point of view of our design, there is no tracking error.

PID controller and nonlinearity as a disturbance

- Another approach (that bypasses linearization) is to consider the nonzero term to be a *disturbance*, and use an integral controller to cancel it. We start with

$$M\ddot{\theta}_e + (b + K_d)\dot{\theta}_e + K_p\theta_e = mgr\cos\theta.$$

- We add an integral term and consider nonlinear terms to be *disturbances*:

$$M\ddot{\theta}_e + (b + K_d)\dot{\theta}_e + K_p\theta_e + K_i\int_0^t \theta_e(s)ds = \tau_{dist}.$$

- Taking derivatives on both sides, and assuming that $\dot{\tau}_{dist} = 0$, we get

$$M\dddot{\theta}_e + (b + K_d)\ddot{\theta}_e + K_p\dot{\theta}_e + K_i\theta_e = 0.$$

- The corresponding characteristic equation is

$$s^3 + \frac{b + K_d}{M}s^2 + \frac{K_p}{M}s + \frac{K_i}{M} = 0$$

PID controller and nonlinearity as a disturbance

- We need to choose the coefficients K_i, K_d, K_p so that all the roots have negative real parts. This will make the system stable, i.e. so that $\theta_e(t) \to 0$.
- The conditions for the roots of a third order polynomial to have negative real parts are:

$$s^3 + a_2 s^2 + a_1 s + a_0 = 0$$

has roots with negative real parts if and only if:

$$a_2, a_1, a_0 > 0 \text{ and } a_2 a_1 > a_0$$

This is a particular case of the more general Routh-Hurwitz criterion. We do not cover it.

- This yields here: $K_d > -b$, $K_p > 0$ and $\frac{(b+K_d)K_p}{M} > K_i > 0$.
- This approach is not guaranteed to work in general (the assumption of considering nonlinear term to be constant disturbances is quite strong). There exist methods to check that it holds, which we do not cover.