Fundamentals of Robot Control:

Lecture 4: Introduction to Nonlinear Control: Inverse Dynamics, Fully, Over and Under Actuated systems. Feedback Linearization

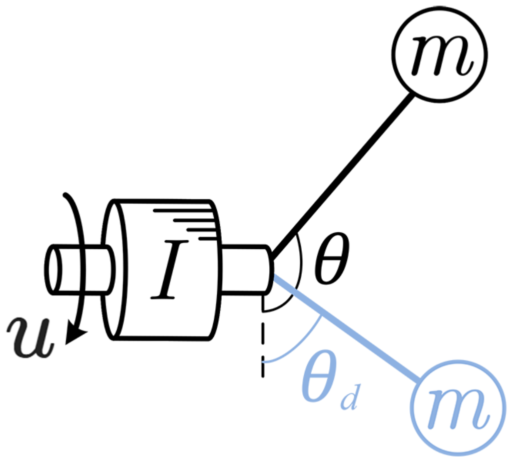
Goals for today:

- Study effect of nonlinearities and how to compensate them
- Introduce the notions of fully, over and under actuated systems
- Discuss how to use the power of optimization in order to tackle some issues in over-actuated control
- Perform exact linearization with help of feedback to achieve truly linear response without approximations

Effect of Nonlinearities

Let us now test the linear feedback techniques to regulate the angle of nonlinear pendulum, $\mathbf{x}_d = [\theta_d, 0]^T$. Control error is then defined as:$\tilde{\mathbf{x}} = [\tilde{\theta}, -\dot{\theta}]^T$.

Here, we assume that we can accurately measure or estimate full state of system $\mathbf{x} = [\theta, \dot{\theta}]^T$



Dynamics of this system is given as:

$$(mL^2 + I)\ddot{\theta} + mgL\sin\theta + b\dot{\theta} = u = k_p\tilde{\theta} - k_d\dot{\theta}$$

Substitution of the second order full state feedback control law yields the following closed loop system:

$$(mL^2 + I)\ddot{\theta} - k_p\tilde{\theta} + mgL\sin\theta + (k_d + b)\dot{\theta} = 0$$

What do you think does this system imply convergence of $\tilde{\theta} \to 0$ for any θ_d ?

Substistution of the $\dot{\theta} = 0, \ddot{\theta} = 0$ yields the following expression for equilibriums states of closed loop response:

$$k_p\tilde{\theta} = mgL\sin\theta$$

So there are infinite amount of equilibrium points provided by the expression above. One can also check the stability of these points or simulate the response (do this at home).

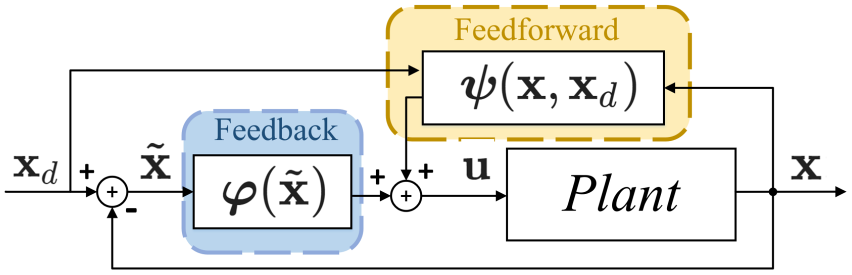
So the linear state feedback on itself is not able to produce a single desired equilibrium i.e: $\tilde{\theta} = 0$?. A natural question to ask - can we do better?

Feedforward Action

As we have shown above feedback alone can't always guarantee that control error is going to converge to zero, thus one may need to incorporate the feedforward action inside the control loop.

It is always convenient to represent the controller as a combination of nonlinear feedback and feedforward terms:

$$\mathbf{u}(\mathbf{x}, \mathbf{x}_d) = \boldsymbol{\varphi}(\tilde{\mathbf{x}}) + \boldsymbol{\psi}(\mathbf{x}, \mathbf{x}_d)$$


Feedforward
$\boldsymbol{\psi}(\mathbf{x}, \mathbf{x}_d)$
Feedback
$\boldsymbol{\varphi}(\tilde{\mathbf{x}})$
$\mathbf{x}_d$
+
-
$\tilde{\mathbf{x}}$
+
+
$\mathbf{u}$
Plant
$\mathbf{x}$

In practice, feedback control is often implemented as a linear combinations of control errors:

$$\mathbf{u}(\mathbf{x}, \mathbf{x}_d) = \mathbf{K}\tilde{\mathbf{x}} + \boldsymbol{\psi}(\mathbf{x}, \mathbf{x}_d)$$

Incorporating such control action greatly increases the accuracy of control without affecting stability!

Example: Pendulum angle regulation

First, let us describe the feedforward part of the regulator. A good idea would be cancelling out nonlinear gravity effects as well as the friction term by the following control model:

$$\boldsymbol{\psi}(\mathbf{x}, \mathbf{x}_d) = mgL\sin\theta + b\dot{\theta}$$

Substitution of controler $u = u_{fb} + u_{ff}$ with feed forward term described above into the dynamics of the pendulum yields:

$$(mL^2 + I)\ddot{\theta} + mgL\sin\theta + b\dot{\theta} = \boldsymbol{\psi}(\mathbf{x}, \mathbf{x}_d) + k_p\tilde{\theta} - k_d\dot{\theta} \to (mL^2 + I)\ddot{\theta} = k_p\tilde{\theta} - k_d\dot{\theta}$$

Note how **nonlinear** system was transformed into the **linear** one:

$$(mL^2 + I)\ddot{\theta} = k_p\tilde{\theta} - k_d\dot{\theta}$$

Now we can use powerful tools from **linear control theory** to obtain the desired response of the system.

The equation above implies that error $\tilde{\theta} \to 0 \iff \theta \to \theta_d; t \to \infty$ as desired.

Let's define our controller and simulate controlled motion.

Control Over Mechanical Systems. Fully, Over, and Underactuated

Consider the dynamic equations of an mechanical system in matrix form:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{B}(\mathbf{q})\mathbf{u}$$

where $\mathbf{B} \in \mathbb{R}^{n\times m}$ is input matrix which connect the number of control inputs (actuators) of the system comparing to the number of generalized coordinates (joints).

Then we can do the following classification:

- number of control inputs are equal to the number generilized coordinate ($n = m$) systems is said to be **fully actuated**
- if number of control inputs are greater then of generilized coordinates ($n < m$) system is **over actuated**
- and finnaly if there is more coordinates then actuators ($n > m$) system is called **under actuated**

This seemingly innocent classification in fact produce a substantial differences in the control techniques we can apply to fully/over and under-actuated systems. Let us begin with simplest case of fully-actuated systems.

Inverse Dynamics over Fully Actuated Mechanical Systems

In the fully actuated case the number of control inputs is equal to control channels and matrix $\mathbf{B}$ is invertable (usually identity $\mathbf{I}_n$), so without lose of generality we can write the dynamics in form:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{u}$$

The idea of inverse dynamics is to seek a nonlinear feedback control law in form:

$$\mathbf{u} = \boldsymbol{\phi}(\mathbf{q}, \dot{\mathbf{q}}, t)$$

which, when substituted into dynamics results in a truly **linear closed-loop system** (do not confuse with local linearization).

By inspecting we see that there is two things that can potentially make the response of the system nonlinear, first is in the inertia matrix $\mathbf{M}$ while second is in the term $\mathbf{h}$ so lets try to compensate for both of them by chosing control law as follows:

$$\mathbf{u} = \mathbf{M}(\mathbf{q})\mathbf{v} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$$

then, since the inertia matrix M is invertible, the combined system given by:

$$\ddot{\mathbf{q}} = \mathbf{v}$$

The term $\mathbf{a}_q$ represents a new input that is yet to be chosen, and the resulting closed loop system is known as the **double integrator** as it represents n
uncoupled double integrators.

This is rather remarkable result, namely that the
trully nonlinear ssystem now become linear and decoupled. This means that each input v_i can be designed to control a scalar linear system. Moreover,
assuming that v_i is a function only of qi and $\dot{q}_i$ then the closed-loop system will be fully decoupled and we can use a simple PD like regulator:

$$v_i = \ddot{q}_d(t) + k_i^d \dot{\tilde{q}}_i + k_i^p \tilde{q}_i$$

And the closed loop will be:

$$\ddot{\tilde{q}}_i + k_i^d \dot{\tilde{q}}_i + k_i^p \tilde{q}_i = 0$$

which is stable provided $k^p, k^d > 0$. Moreover you can make system **critically damped** with $k_p^i = \omega_i^2$ and $k_d^i = 2\omega_i$. In fact you can use whatever linear control technique on the double integrate, all of them will work equally well.

And that's it! Once you know the model and your system is fully actuated, the control is pretty straightforward. Let us now use the very same ideas to build the similar controllers for **over actuated systems**.

Inverse Dynamics of Over-actuated Mechanical Systems

The discussed above ideas are easily applicable to over actuated systems. To see it let us consider the dynamics of mechanical system and introduce auxiliary variable $\mathbf{Q}$ (generalized forces):

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{B}(\mathbf{q})\mathbf{u} = \mathbf{Q}$$

So if one can produce required $\mathbf{Q}$ the problem will be identical to the fully actuated case. However, the inputs $\mathbf{Q}$ may not be actually present to the system by means of physical actuators that are tighly associated with $\mathbf{u}$.

Thus one need to satisfy the following:

$$\mathbf{B}(\mathbf{q})\mathbf{u} = \mathbf{Q}_d$$

With $\mathbf{Q}_d$ being your desired controller for the fully actuated system.

This in turn implyis that one need just to find the $\mathbf{u}$ that will satisfy equation above. For over actuated systems $\dim{(\mathbf{Q}_d)} < \dim{(\mathbf{u})}$, thus matrix $\mathbf{B}$ is tall and the simplest solution of the linear equations above is straight forward:

$$\mathbf{u} = \mathbf{B}^+(\mathbf{q})\mathbf{Q}_d(\mathbf{q}, \dot{\mathbf{q}}, t)$$

and $\mathbf{B}^+$ is pseudo inverse of input matrix.

Let's stand here for a while and think about the physical interpretation of this idea. In fact what we are trying to do is to produce the desired force $\mathbf{Q}_d$ which in turn will move our system as it is desired in fully actuated circumstances. However, now we have more actuators u then DoF in our system. so there's infinitely many ways how we can produce the desired force $\mathbf{Q}_d$ with help of actuators $\mathbf{u}$ (infinetly many solutions of the linear system $\mathbf{Bu} = \mathbf{Q}_d$), and is fully your responsibility to chose one of the appropriate one.

There is a lot of possibilities in there:

- use regular pseudo inverse which in fact find the solution of $\mathbf{Bu} = \mathbf{Q}_d$ with minimal norm of $\mathbf{u}$
- **wieght** the pseudo inverse: , which in practice is required if some of actuators are weaker then others.
- You may minimize some quantity which is connected to the performance of your system, for instance, **energy, power** etc, which usually result in the convex optimization problems
- there is even possibility to include some constraints on $\mathbf{u}$, for instance take to account the unidirectional actuation $\mathbf{u}$ or the physical capability of the actuators $\mathbf{u}$

Force/Torque Optimization

The all of the notions discussed above may be compactly formulated as an optimization problems:

$$\begin{aligned} \underset{\mathbf{u}}{\text{minimize}} \quad & \mathcal{J}_c(\mathbf{u}, \mathbf{q}, \dot{\mathbf{q}}, t) \\ \text{subject to} \quad & \mathbf{B}(\mathbf{q})\mathbf{u} = \mathbf{Q}_d \\ & \mathbf{A}(\mathbf{q}, \dot{\mathbf{q}}, t)\mathbf{u} \leq \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}, t) \end{aligned}$$

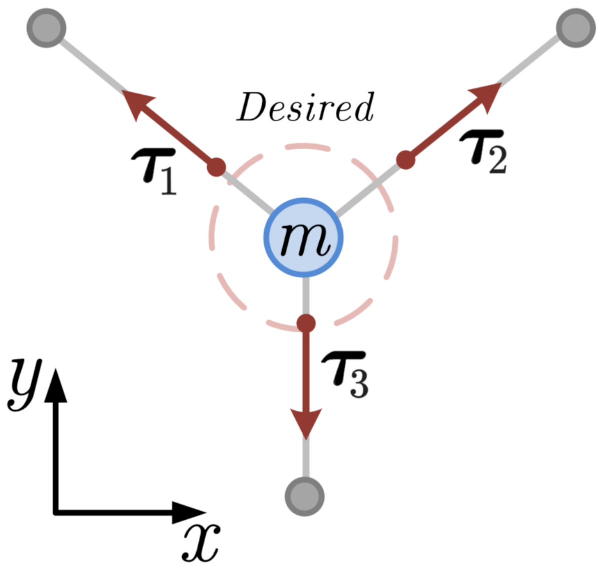
So the controller is then defined as a solution of optimization problem above in this particular time instant.

If the cost J_c is quadratic or linear, thus resulting in respectively linear or quadratic programming with mature ready to use and free solvers, like osqp, ecos etc.

The formulation above is fairly general and can encounter many different practical constraints, like unidirectional actuators $\mathbf{u} \geq 0$, limited capabilities in form $|\mathbf{u}| \leq \mathbf{u}_{max}$ and many others. In fact with help of so called **barrier techniques** one can even introduce the safety constraints on states $\mathbf{q}, \dot{\mathbf{q}}$ while preserving optimization problem convex.

Example:

Consider a cable (unstretchable) driven manipulator:



A goal is to find **positive** tensions $\boldsymbol{\tau} > \mathbf{0}$ on cables such that end effector **track the desired trajectory** end overall tension effort is minimized:

$$\mathcal{J}_c = \|\boldsymbol{\tau}\|_1 \quad \text{or} \quad \mathcal{J}_c = \|\boldsymbol{\tau}\|_2^2$$

Underactuated Systems and Partial Feedback Linearization:

It seems that the controll over fully and underactuated mechanical systems are in general similar, in fact if we have more actuators they provide some flexibility in to control design by means of proper optimization over control inputs.

However, control over underactuated systems is **drastically different and much more challenging**. First of all we cannot always hope to convert our dynamics in to the decoupled double integrator and even not all trajectories are **dynamically feasible** and it is out of scope of this class.

However there is some similar techniques, that are strongly related to the inverse dynamics and feedback linearization. Although we cannot always simplify the full dynamics of the system, it is still **possible to linearize a portion** of the system dynamics. The technique is called partial feedback linearization.

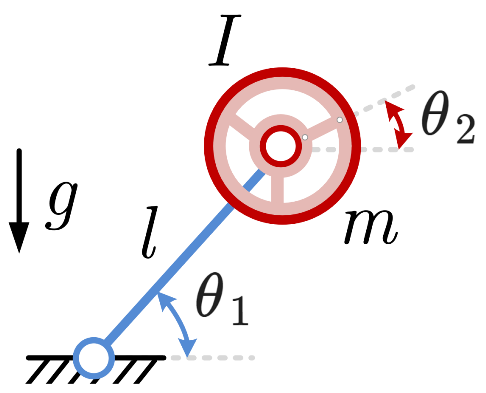
Example:

Consider for the so called reaction wheel pendulum (1, 2).

The dynamics of such system is given by:

$$\begin{cases} (I + ml^2)\ddot{\theta}_1 + I\ddot{\theta}_2 + mgl\cos\theta_1 + b\dot{\theta}_2 = 0 \\ I(\ddot{\theta}_1 + \ddot{\theta}_2) = u \end{cases}$$

The goal is to control the angle θ_1 just by applying the control torque u



General Form of PFL of Mechanical Systems

For systems that are trivially underactuated (torques on some joints, no torques on other joints), we can, without loss of generality,

reorganize the joint coordinates in any underactuated system described by
the manipulator equations into the form:

$$\begin{gathered}\mathbf{M}_{11}\ddot{\mathbf{q}}_1 + \mathbf{M}_{12}\ddot{\mathbf{q}}_2 = \mathbf{h}_1 \\ \mathbf{M}_{21}\ddot{\mathbf{q}}_1 + \mathbf{M}_{22}\ddot{\mathbf{q}}_2 = \mathbf{h}_2 + \mathbf{u}\end{gathered}$$

with $\mathbf{q}_1 \in \mathbb{R}^{n-m}$ representing all the passive joints and $\mathbf{q}_2 \in \mathbb{R}^m$.
all actuated joints.

Fortunately, because $\mathbf{M}$ is uniformly (e.g.
$\forall \mathbf{q}$) positive definite, $\mathbf{M}_{11}$ and $\mathbf{M}_{22}$ are also positive definite

Now one can actually solve the equations above for either $\ddot{\mathbf{q}}_1$ or $\ddot{\mathbf{q}}_2$ resulting in so called **collocated** and **non-collocated** linearization.

Collocated and Non-Collocated PFL

Collocated linearization:
solving the equations above for passive joints $\ddot{\mathbf{q}}_1$ and substituting to the dynamics yields:

$$(\mathbf{M}_{22} - \mathbf{M}_{21}\mathbf{M}_{11}^{-1}\mathbf{M}_{12})\ddot{\mathbf{q}}_2 - \mathbf{h}_2 + \mathbf{M}_{21}\mathbf{M}_{11}^{-1}\mathbf{h}_1 = \mathbf{u}$$

The collocated PFL holds globally since $\mathbf{M}_{22} - \mathbf{M}_{21}\mathbf{M}_{11}^{-1}\mathbf{M}_{12}$.

Non-Collacted linearization:
One can also solve the equations above for active joints $\ddot{\mathbf{q}}_2$ which yields:

$$(\mathbf{M}_{21} - \mathbf{M}_{22}\mathbf{M}_{12}^{+}\mathbf{M}_{11})\ddot{\mathbf{q}}_1 - \mathbf{h}_2 + \mathbf{M}_{22}\mathbf{M}_{12}^{+}\mathbf{h}_1 = \mathbf{u}$$

The pseudo inverse $\mathbf{M}_{12}^{+}$ provides a unique solution when the rank of $\mathbf{M}_{12}$ is equals to the $n-m$ (the number of passive degrees of freedom). This condition is sometimes called **"strong inertial coupling"**. It is state dependent. A system is said to has **global strong inertial coupling** if it exhibits strong inertial coupling $\forall \mathbf{q}$

Generilized Feedback Linearization

Consider a control affine class of nonlinear systems in state space form:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}$$

where:

- $\mathbf{x} \in \mathbb{R}^n$ is state of the system

- $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^n$ nonlinear smooth function of state evolution (vector field)
- $\mathbf{G}(\mathbf{x}) \in \mathbb{R}^{n\times m}$ nonlinear smooth input matrix

Let's focus on single-input for a while ($m = 1$):

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})u$$

The system above is said to be locally **feedback linearizable** if there exists transformation $\mathbf{T}$ (diffeomorphism):

$$\mathbf{z} = \mathbf{T}(\mathbf{x})$$

Together with nonlinear feedback:

$$u = \alpha(\mathbf{x}) + \beta(\mathbf{x})v$$

Such that the transformed state $\mathbf{z}$ satisfy a **controllable linear system** (n-th order integrator):

$$\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{b}v$$

Where:

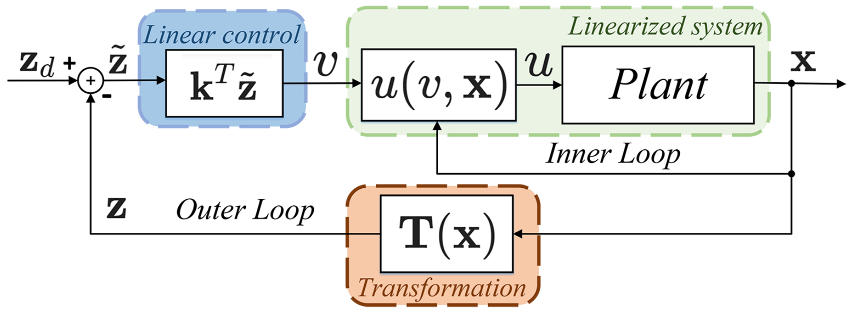
$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & \cdot & 0 \\ 0 & 0 & 1 & \cdot & 0 \\ \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & \cdot & & 1 \\ 0 & 0 & \cdot & \cdot & 0 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ 1 \end{bmatrix},$$

Choosing desired state to be $\mathbf{x}_d$ and respective error $\tilde{\mathbf{x}} = \mathbf{x}_d - \mathbf{x}$ one can design **linear feedback controller** v such that $\mathbf{z}$ follow $\mathbf{z}_d = \mathbf{T}(\mathbf{x}_d)$.

Indeed, defining control error as $\tilde{\mathbf{z}} = \mathbf{z}_d - \mathbf{z} = \mathbf{T}(\mathbf{x}_d) - \mathbf{T}(\mathbf{x})$ or $\tilde{\mathbf{z}} = \frac{\partial \mathbf{T}}{\partial \mathbf{x}}\tilde{\mathbf{x}}$ if $\tilde{\mathbf{x}}$ is known to be small, we may choose full state linear feedback controller:

$$v = \mathbf{k}^T\tilde{\mathbf{z}}$$

Then, applying control $u = \alpha(\mathbf{x}) + \beta(\mathbf{x})v$ to original system will ensure that $\mathbf{x} \to \mathbf{x}_d$ if such $\mathbf{x}_d$ is feasible.



Example:

Let us find transformation $\mathbf{T}(x)$ and feedback linearization controller u for following system:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 + x_1^2 \\ -x_1^3 + u \end{bmatrix}$$

First let's choose the following state transformation:

$$\mathbf{z} = \mathbf{T}(\mathbf{x}) = \begin{bmatrix} x_1 \\ x_2 + x_1^2 \end{bmatrix}$$

Differentiating with respect to time yields:

$$\dot{\mathbf{z}} = \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 + 2x_1\dot{x}_1 \end{bmatrix} = \begin{bmatrix} x_2 + x_1^2 \\ -x_1^3 + u + 2x_1(x_2 + x_1^2) \end{bmatrix}$$

Thus linearizing control law u:

$$u = -2x_1(x_2 + x_1^2) + x_1^3 + v$$

The inner loop controller above transform our nonlinear system to the linear with respect to $\mathbf{z}$ and v:

$$\dot{\mathbf{z}} = \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} z_2 \\ v \end{bmatrix}$$

now we can easily design the stabilizing controller (regulation to the origin $\mathbf{x}_d = [0,0]^T$) using linear tools!

ne can easely check that corresponding transformed desired state $\mathbf{z}_d = \mathbf{T}(\mathbf{x}_d) = [0,0]^T$ therefore stabilizing controller is given by:

$$v = -\mathbf{k}^T\mathbf{z}$$

However, this process is not well formulated, one of the problems how to get the proper transformation $\mathbf{T}(\mathbf{x})$?

There are some techniques to do so, the general idea - choose some state $\mathbf{x}_i$ you want ro regulate and differentiate it until input u appears.

Exercise:

Consider the following system:

$$\begin{cases} \dot{x}_1 = -x_1 + a x_2 + \sin x_1 \\ \dot{x}_2 = -x_2 \cos x_1 + u(\cos x_1 + b) \end{cases}$$

where a, b are constants

Try to find the proper transformation and inner loop controller