Implementing PID Control with Real World Features

Philipp Schuette
University of Paderborn

May 19, 2019

Abstract

The aim of this paper is to showcase a variety of phenomena, that appear in actual implementations of PID controllers, but are often omitted in theoretical considerations. To this end, the basic theory of abstract control systems is first review. After that it is shown, how the concrete PID control of the (linear and nonlinear) inverted pendulum fits into this abstract picture.

The next section covers the numerical implementation of this system, including features like bounded control output, limited controller speed, incomplete system knowledge and measurement imprecision. Additionally, several criteria for finding optimal control parameters are discussed and numerically implemented.

Finally, a comparison is made between this more realistic implementation and a second implementation of an ideal PID controller, as it is often presented in textbooks. An outlook contains further features, that could enhance the implementation.

Contents

1 Abstract Foundation of Control Theory and the Inverted Pendulum

The main reference for this rather abstract chapter is the excellent elementary textbook [Son98]. In subsection 1.1, an abstract control system is defined. Then, an already much more concrete definition of continuous time system is given, together with some general remarks.

Subsection 1.2 contains a short introduction to the (inverted) pendulum form a physical point of view. The linearized and fully non-linear cases are described and the subsection is concluded with a very elementary numerical integration scheme for these two ordinary differential equations (*ODE*s).

The final subsection 1.3 gives a short overview over the fundamentals of PID control. It introduces the necessary nomenclature and the basic parameters of a PID controller. Also, references on more theoretical issues like controllability are provided.

1.1 Abstract Control System Definition

The following basic notation shall be adapted: For sets I, U denote by U^{I} the set of mappings $\{\omega : \mathrm{I} \to \mathrm{U}\}$. A *timeset* is any subgroup of $(\mathbb{R}, +)$, and if T is such a timeset, then for any interval $[a, b] \subseteq \mathbb{R}$ its intersection $[a, b] \cap \mathrm{T}$ is abbreviated as $[a, b]_{\mathrm{T}}$. Finally, if $\omega_1 \in \mathrm{U}^{[a,b]_{\mathrm{T}}}$ and $\omega_2 \in \mathrm{U}^{[b,c]_{\mathrm{T}}}$, then their concatenation will just be written by juxtaposition, i.e. $\omega_1\omega_2 \in \mathrm{U}^{[a,c]_{\mathrm{T}}}$.

Definition 1.1: Abstract Control System

([Son98], p. 26) A system Σ is a four-tuple $\Sigma = (\mathrm{T}, \mathbf{X}, \mathrm{U}, \varphi)$, with T a timeset, $\mathbf{X}$ and U non-empty sets and $\varphi : \mathcal{D}_\varphi \to \mathbf{X}$ a map on a subset $\mathcal{D}_\varphi \subseteq \{(\tau, \sigma, x, \omega) \mid \sigma, \tau \in \mathrm{T}, \sigma \leq \tau, x \in \mathbf{X}, \omega \in \mathrm{U}^{[\sigma,\tau[_{\mathrm{T}}}\}$, satisfying the following conditions:

1. For every $x \in \mathbf{X}$ there exist $\sigma < \tau$ in T and $\omega \in \mathrm{U}^{[\sigma,\tau[_{\mathrm{T}}}$, such that $(\tau, \sigma, x, \omega) \in \mathcal{D}_\varphi$, i.e. ω is admissible for x.
2. If $\omega \in \mathrm{U}^{[\sigma,\tau[_{\mathrm{T}}}$ is admissible for $x \in \mathbf{X}$, then $\omega_1 = \omega\big|_{[\sigma,\mu[}$ is also admissible for any $\mu \in [\sigma, \tau[$, and $\omega\big|_{[\mu,\tau[}$ is admissible for $\varphi(\mu, \sigma, x, \omega_1)$.
3. Let $\sigma < \tau < \mu$ be elements of T, $\omega_1 \in \mathrm{U}^{[\sigma,\tau[_{\mathrm{T}}}$ and $\omega_2 \in \mathrm{U}^{[\tau,\mu[_{\mathrm{T}}}$. If $x \in \mathbf{X}$ is admissible for ω_1 and if ω_2 is admissible for $x_1 = \varphi(\tau, \sigma, x, \omega_1)$, then $\omega_1\omega_2$ is admissible for x and $\varphi(\mu, \sigma, x, \omega_1\omega_2) = \varphi(\mu, \tau, x_1, \omega_2)$.
4. For any $x \in \mathbf{X}$ and $\sigma \in \mathrm{T}$, the empty sequence $\square \in \mathrm{U}^{[\sigma,\sigma[_{\mathrm{T}}}$ is admissible for x and $\varphi(\sigma, \sigma, x, \square) = x$.

In this definition, $\mathbf{X}$ is called the *state space*, U is called the *control value space* and φ is the so-called *state transition map*. One extension of this definition is important when e.g. considering controllers with limited measurement precision. A *system with outputs* is a system $\Sigma = (\mathrm{T}, \mathbf{X}, \mathrm{U}, \varphi)$ together with a set $\mathbf{Y}$ and a function $h : \mathrm{T} \times \mathbf{X} \to \mathbf{Y}$. $\mathbf{Y}$ is called the *measurement value space* and h the *measurement map*.

The above definition is much more general than strictly needed, when considering problems of controlling physical systems. In such cases, the system description is almost always given by an ODE[1]. For this reason, practically oriented textbooks like [Fö90] directly start with an analysis of differential equations, without considering any abstract mathematical framework. For concrete examples this might work, but it makes proofing technical theorems mostly impossible. The reason for stating the definition here is different, though. It is the author's opinion, that the abstract formulation of definition 1.1 gives one another perspective on control problems. This abstract perspective can then complement any specific knowledge or intuition one has about a given controlled ODE.

As this paper is not interested in any general statements regarding systems according to definition definition 1.1 or any subclasses of such systems (c.f. chapter 2 of [Son98]), an immediate connection to the omnipresent ODEs will now be made.

[1] Or partial differential equation (*PDE*), which is generally even more complicated.

Definition 1.2: Continuous Time System Preparation

([Son98], p. 43) Let $\mathbf{X} \subseteq \mathbb{R}^n$ be open and $(\mathrm{U}, \mathrm{d}_\mathrm{U})$ a metric space. $f : \mathbb{R} \times \mathbf{X} \times \mathrm{U} \to \mathbb{R}^n$ is called a right-hand side for $\mathbf{X}$ and U, if there exist a metric space $(\mathrm{S}, \mathrm{d}_\mathrm{S})$ and maps $g : \mathrm{S} \times \mathbf{X} \times \mathrm{U} \to \mathbb{R}^n$, $\pi : \mathbb{R} \to \mathrm{S}$, such that

$$f(t, x, u) = g(\pi(t), x, u) \ \forall (t, x, u) \in \mathbb{R} \times \mathbf{X} \times \mathrm{U}, \tag{1}$$

and the following conditions hold:

1. $g(s, \cdot, u)$ is continuously differentiable for every $(s, u) \in \mathrm{S} \times \mathrm{U}$.
2. π is measurable and locally essentially bounded.
3. g and its derivative g_x are continuous.

It is now rather easy, if one has the correct version of Picard-Lindelöf available, to show, that for any right-hand side f and measurable, essentially bounded $u \in \mathrm{U}^{[t_0, t_1[}$ the initial value problem

$$\dot{x}(t) = f(t, x(t), u(t)), \quad x(t_0) = x_0 \tag{2}$$

possesses a unique maximal solution on a sub-interval of $[t_0, t_1[$ containing t_0. Thus, the flow corresponding to (2) yields a system as in definition definition 1.1 ([Son98], p. 43f.). If a system arises in this way from an ODE, it is called a *continuous-time system*. Again, one can add a measurement map, which again would usually be required to fulfill certain technical conditions like continuity.

Having made the connection to more abstract notions from mathematical control theory, the remainder of this paper will use system definitions as in (2). Also, the utilized right-hand sides will not require the full generality of definition definition 1.2.

1.2 The Inverted Pendulum

First, consider an idealized (mathematical) pendulum in the plane. The pendulum's equation of motion can be determined using either Newtonian, Hamiltonian or Lagrangian mechanics. As the Newtonian approach is certainly the most common, Lagrangian mechanics will be employed here. In this simple example it should pose no difficulties to change from the Lagrangian to the Hamiltonian picture, if the reader prefers the symplectic view.

The central ingredient is the so-called *Lagrangian*. This scalar function on the tangent bundle determines the *action*, which yields a variational problem. This problems Euler-Lagrange equations are the equations of motion for the physical system associated to the Lagrangian.

Given a Lagrangian $\mathcal{L}(t, x, \dot{x})$, its Euler-Lagrange equations turn out to be

$$\frac{\mathrm{d}}{\mathrm{dt}} \frac{\partial \mathcal{L}}{\partial \dot{x}} = \frac{\partial \mathcal{L}}{\partial x} \ . \tag{3}$$

The Lagrangian itself can be calculated from the potential energy V and the systems kinetic energy, specified by its mass m: $\mathcal{L}(t, x, \dot{x}) = \frac{1}{2} m \dot{x}^2 - V(t, x)$. Mathematically, $\mathcal{L} : \mathbb{R} \times \mathrm{T}\mathbf{M} \to \mathbb{R}$, with $\mathrm{T}\mathbf{M}$ the tangent bundle of the manifold $\mathbf{M}$ of generalized coordinates for the system. For the purposes of this paper it will be assumed, that $\mathrm{T}\mathbf{M} = \mathbb{R} \times \mathbb{R}$.[2]

Let m and l denote the pendulums mass and length, respectively. The acceleration due to gravity will be assumed constant and equal to $\mathrm{g} = 9.81 \frac{\mathrm{m}}{\mathrm{s}^2}$. If x denotes the angle between the pendulum and its stable equilibrium position, then the Lagrangian is given by

$$\mathcal{L}(t, x, \dot{x}) = \frac{1}{2} m \dot{x} + m l \mathrm{g} \cos(x). \tag{4}$$

The pendulums equation of motion is therefore

$$m \ddot{x} = -m l \mathrm{g} \sin(x). \tag{5}$$

[2] Even for the pendulum, this is actually incorrect. The angle coordinate x does not belong to the real line, but rather to the circle $\mathbb{R}/2\pi\mathbb{Z} = \mathbf{M}$. For a mathematically rigorous treatment, one should therefore work on $(\mathbb{R}/2\pi\mathbb{Z}) \times \mathbb{R}$.

One obtains the inverted pendulum with the substitution $x \mapsto -(\pi - x)$. The new angle coordinate, still denoted by x, represents the angle between the pendulum and its unstable equilibrium position. The transformed equations of motion become

$$\ddot{x} = lg\sin(x). \tag{6}$$

Despite common practice in more theoretical treatments of the subject, units will be kept here. For reference, table 1 contains the relevant units of measurement.

Table 1: Units appearing in the discussed equations of motion.

Dimension	Unit
Angle	[radiant] = [rad]
Time	[second] = [s]
Distance	[meter] = [m]

One final physics equation will be needed in the following discussions. The equation is obtained from (6) by the so-called *small-angle approximation*. This very common strategy originates in the Taylor expansion for the analytical sine function: $\sin(x) = \sum_0^\infty \frac{(-1)^n}{(2n+1)!} x^{2n+1}$. For small angles x, one assumes, that approximating $\sin(x) = x$ yields results in sufficient agreement with the actual solution. While equation (6) will be called the full or nonlinear pendulum, the following will be called the linearized pendulum:

$$\ddot{x} = lgx. \tag{7}$$

1.3 PID Controller Fundamentals

The free system dynamics are determined by (6) and (7). From equation (6) it is obvious, that $x = 0, \pi$ are the pendulum's two fixed points. While $x = \pi$ is stable, $x = 0$ is unstable: A pendulum, whose starting position differs only slightly from $x = 0$ will not return there asymptotically. If one wants to dictate asymptotic return to $x = 0$, or any other dynamical behavior, a feedback controller is necessary.

For the pendulum (7), a so-called PID controller can be shown to exhibit many desirable properties. These properties extend, at least locally, to the fully non-linear model (6). The PID controllers name already suggests its ingredients: It feeds the sum of three terms back into the system, each depending on the difference between desired *set point* and actual *system value*. The **P** terms is proportional to, the **I** term is the integral of and the **D** term is the derivative of this difference.

For simplicity in notation, assume the set point to be 0. To accommodate for the PID terms, one simply adds the control function $u(t) = -\mu \int_{t_0}^{t} x(\tau)\mathrm{d}t - \alpha x(t) - \beta \dot{x}(t)$ to the right hand side of either (6) or (7). According to definition definition 1.2, the system dynamics should be given as a system of first order differential equations. This can easily be achieved by introducing two additional variables, which encode the derivative and the integral of $x(t)$. The correct right hand side in the technical sense of definition definition 1.2 is then given by[3]

$$f(t, x_0, x_1, x_2, u) = \begin{pmatrix} x_1 \\ x_2 \\ \sin(x_1) + u \ , \end{pmatrix} \tag{8}$$

which yields the following system of first order differential equations:[4]

$$\begin{pmatrix} \dot{x}_0 \\ \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = f(t, x_0(t), x_1(t), x_2(t), u(t)) = \begin{pmatrix} x_1(t) \\ x_2(t) \\ lg\sin(x_1(t)) - \mu x_0(t) - \alpha x_1(t) - \beta x_2(t) \end{pmatrix} \tag{9}$$

[3]For the linearized pendulum one simply replaces $\sin(x)$ by x.

[4]Where $x_1(t)$ corresponds to the original $x(t)$!

As a final reference to definition definition 1.2, one recognizes from (8), that $n = 3$, $\mathrm{U} = \mathbb{R}$, $\mathbf{X} = \mathbf{R}^3$, $\pi = \mathrm{id}_{\mathbb{R}}$. Also, the system is time invariant, which means that f does not depend on t explicitly. For a technical definition of time invariance, c.f. [Son98], p. 29.

The difficulties in designing a good PID controller consist mainly of finding correct values for the parameters α, β, μ. After establishing suitable optimality criteria, one would usually notice, that there exist no parameter triples, that optimize more than one criterium simultaneously. This means, that the actual task consists of finding a compromise between several requirements in a real world system. Additionally, the accessible parameter set might be severely restricted because of system or controller specifications, cost considerations or similar points, that practice might dictate. This paper wants to highlight specifically these points, and how they alter the controller behavior in comparison with a straight forward solution to (9).

2 Implementation of the PID Controlled Pendulum

In this section, the concrete numerical implementation of the PID controller as stated in 9 is discussed. After an introduction to the numerical method and implementation in subsection 2.1, subsection 2.2 gives a short overview for all features, that were included. This especially encompasses results of numerical simulations, which illustrate the effects specific features have on the system dynamics.

2.1 Overview over the Numerical Implementation

The original numerical solution of the PID controlled pendulum can be found as a separate repository on the author's GitHub[5] It was written in **Python3** by the author and in a manner, which should make it accessible for experiments and extensions by others. The PID controller itself is implemented as a class named `PIDControl`.

```python
class PIDControl():
    def proportional_output(...):       # Computes proportional control output
    def derivative_output(...):         # Computes derivative control output
    def integral_output(...):           # Computes integral control output
    def total_output(...):              # Computes total control output
```

A detailed description of this class and its methods can be found in the documentation for this project[6], while a short structural description is presented in the above code listing. An instance of this class has to be initialized with several attributes, contained in table 2.

Table 2: Attributes of the PIDControl Python class.

Attribute	Description
`alpha`	Constant α for proportional control.
`beta`	Constant β for derivative control.
`mu`	Constant μ for integral control.
`frequency`	Controller output speed.
`max_control`	Bound for controller output's absolute value.
`set_point`	Set point for controller.
`deadband`	Deadband parameter for controller output change.

[5] `https://github.com/PhilippSchuette/PID_pendulum/`.

[6] A LaTeX documentation is located at `https://github.com/PhilippSchuette/PID_pendulum/docs/` and was formatted automatically using the **Spinx** documentation tool.

`PIDControl` has four main methods, as shown in the snipped: `proportional_output`, `derivative_output` and `integral_output` calculate the **P**, **D** and **I** terms. They take one or two system values and one or two corresponding time points and output the product of their respective constants with the system value (**P**), a simple difference quotient approximation (**D**) and a simple integral approximation (**I**). Thus, the controller calculates the three terms from system and time values alone, it does not have access to the actual derivative and integral values. The latter would be the case, if one solved (9) directly, but that is not very realistic. In most applications, a controller has limited system knowledge, and in the case of angles and velocities, this would most likely be just the angle.

The last method again takes two system values and two corresponding time values, but it does not output anything. After checking on conditions related to bounded controller output and speed, it changes some of the controllers attributes, most importantly the `output` attribute. It does this according to the **PID** values calculated with the three methods above.

```
class Pendulum():
    def solve(...):                     # Solves the controlled pendulum ODE
    def plot(...):                      # Plotes the solution curve of solve()
    def animate(...):                   # Animates the solution curve of solve()
```

Equation (9) can be integrated with a second class called `Pendulum`. For a rough overview, see the above code snippet. It solves the pendulum ODE based on the following simple Taylor series approximation:

Assume $\ddot{x} = f(x) + u$. If $x(t)$ is suitably regular, then $x(t \pm h) \approx x(t) \pm \dot{x}(t)h + \ddot{x}(t)\frac{h^2}{2}$. Adding the equations for both signs, one obtains the approximation $x(t+h) + x(t-h) \approx 2x(t) + \ddot{x}(t)h^2$. Substituting the ODE finally yields

$$x(t+h) \approx 2x(t) - x(t-h) + f(x(t))h^2 + u(t)h^2. \tag{10}$$

Equation (10) allows one to calculate the system value $x[n+1]$, if $x[n]$ and $x[n-1]$ are given. It is therefore necessary, to calculate $x[1]$ from $x[0]$ and $\dot{x}[0]$, which are the physically and analytically more suitable initial conditions. Again, a Taylor expansion yields $x(t+h) \approx x(t) + \dot{x}(t)h$. Thus, the discretized version of (10) can be completed with prescription of $x[0]$, $\dot{x}[0]$ and the equation $x[1] = x[0] + \dot{x}[0]h$.

2.2 Features Included in the Implementation

Features: Controller speed and bounded output, deadband, limited measurement precision.

Still to be implemented: Variable set point, system noise/perturbation, optimality criteria, anti-windup, friction, anti-aliasing filter.

→ Mention, how the number of time interpolation points and the controller speed have to be chosen with respect to the system's eigenvalues. Refer to Nyquist's criterion! Maybe also implement a version for the speed parameter, where you choose speed not with respect to the number of simulation points, but with respect to the system's eigenvalues!?

3 Comparison between Realistic and Ideal Implementations

In this section, the more detailed and thus more realistic implementation of a PID controlled pendulum presented in chapter 2 is compared with the most basic implementation possible. The latter, described briefly in subsection 3.1, solves the controller ODE directly, with full access to all system variables and without restrictions on parameters like speed or power.

Subsection 3.2 contains the actual comparison. Different numerical experiments illustrate, that the controller quality is massively influenced by the simplifications made in subsection 3.1. This highlights one of the main points of this paper: In regarding the (over-)simplified model, one loses almost all features of real world systems and has therefore no chance of exemplifying key points in practical control theory.

3.1 Simplified PID Control Implementation

3.2 Main Differences Between the two Implementations

4 Outlook

In this section, further improvements in realistically modeling PID control of an inverted pendulum are discussed. This includes physical features like friction (4.1) and controller features like additional optimality criteria (4.2).

4.1 Further Improvements of the Physical Model

4.2 Further Improvements of the Controller Model

References

[Fö90] Otto Föllinger, *Regelungstechnik: Einführung in die methoden und ihre anwendungen*, 6. ed., Hüthig Buch Verlag GmbH, Heidelberg, 1990.

[Son98] Eduardo D. Sontag, *Mathematical control theory: Deterministic finite dimensional systems*, second ed., Texts in Applied Mathematics, vol. 6, Springer, New York, NY, 1998.