Lecture 1

Introduction to Nonlinear Systems

We begin by exploring the fundamental differences between linear and nonlinear systems; both at the level of dynamics and control. In essence, most aspects of linear systems are now fully understood while nonlinear systems are substantially more complex as demonstrated through even the simplest examples. This fundamental difference is what renders nonlinear systems intrinsically interesting: their complexity reflects that of the real-world. Yet we will be able, throughout these lecture notes, to develop methods that simplify the analysis of nonlinear (control) systems. This is the art of nonlinear systems: finding simplicity on the far side of complexity. This art will be illustrated in practice on robotic systems (as seen in Figure 1.1) throughout the lectures to demonstrate the ability to translate one's understanding of nonlinear systems on real-world applications. Given the introductory nature of this lecture, the discussion will lack the formality of subsequent lectures where all the concepts and results will be revisited in a more detailed and rigorous manner.

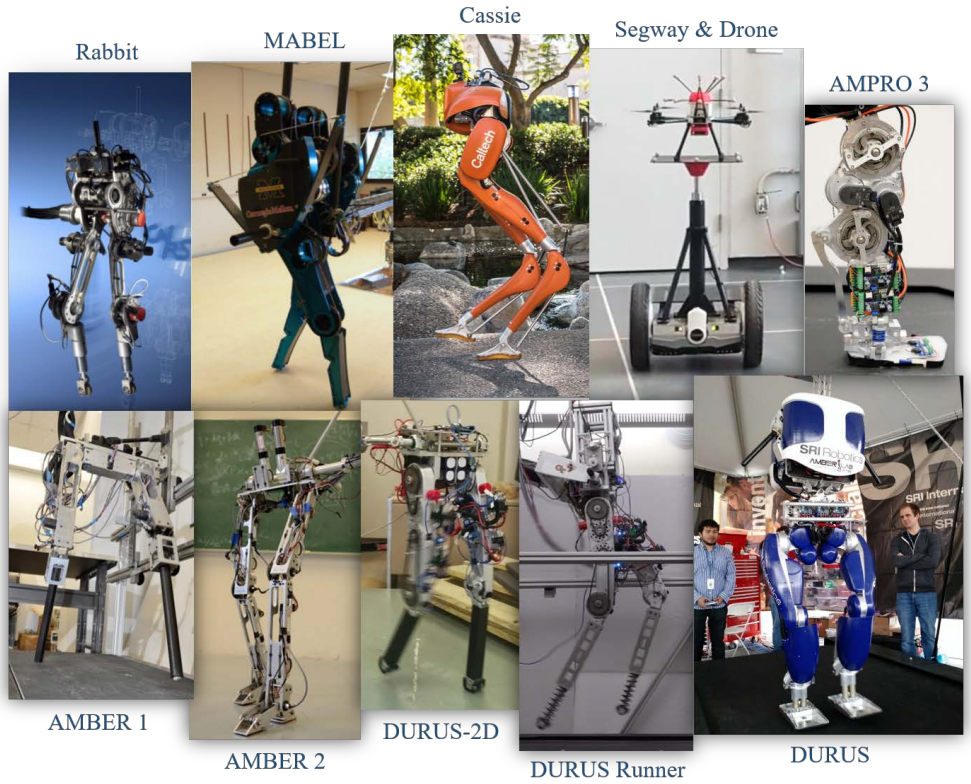


Figure 1.1. The collection of robots that will be considered throughout the lectures. The instantiation of the theory on these platforms will illustrate the dynamic behaviors achievable through nonlinear control. This will be highlighted in the form of "Applications" that will summarize the translation of theory to practice.

1.1 Dynamical Systems: Linear vs. Nonlinear

Before discussing nonlinear systems, consider a linear system described by the linear Ordinary Differential Equation (ODE):

$$\dot{x} = Ax, \quad x(t_0) = x_0 \in \mathbb{R}^n, \quad t_0 \in \mathbb{R},$$

where $A \in \mathbb{R}^{n\times n}$ is an $n \times n$ matrix. Linear systems can be completely characterized in many respects:

1. Solutions always exist and are given in closed form by $x(t) = e^{A(t-t_0)}x_0$ for all $t \geq t_0$, $t \in \mathbb{R}$.
2. Solutions also exist for all $t < t_0$ and are given by the same expression $x(t) = e^{A(t-t_0)}x_0$. Hence, solutions exist for all $t \in \mathbb{R}$.
3. Solutions are unique: for two solutions $x_1(t)$ and $x_2(t)$, if $x_1(t_0) = x_2(t_0)$ then $x_1(t) = x_2(t)$ for all $t \in \mathbb{R}$.
4. The set of equilibrium points is the nullspace of A (all those $x^* \in \mathbb{R}^n$ such that $Ax^* = 0$) which is a linear subspace of $\mathbb{R}^n$. Hence, equilibrium points are "connected" and never "isolated" (except for the trivial case where the only solution to $Ax^* = 0$ is $x^* = 0$).
5. Periodic solutions are only marginally stable, but never asymptotically nor exponentially stable.

We now show through a few simple examples that nonlinear systems are much richer (and more complex). We consider nonlinear systems governed by a nonlinear ODE:

$$\dot{x} = f(x), \quad x(t_0) = x_0 \in \mathbb{R}^n, \quad t_0 \in \mathbb{R},$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is a nonlinear function. We illustrate the complexity (and hence interesting) aspects of nonlinear systems with a few key examples that differentiate them from linear systems.

Example 1.1. Finite Escape Time: Consider the 1-dimensional nonlinear ODE $\dot{x} = 1+x^2$ with $x \in \mathbb{R}$. The solution can be obtained by direct integration:

$$\begin{array}{ccccc} \frac{dx}{dt} = 1 + x^2 & & \iff & & x(t) = \tan(t + c) \\ \Updownarrow & & & & \Updownarrow \\ \frac{dx}{1+x^2} = dt & \Leftrightarrow & \int \frac{dx}{1+x^2} = \int dt & \Leftrightarrow & \arctan(x) = t + c. \end{array}$$

If we now take $t_0 = 0$ and $x(t_0) = 0$, we obtain $c = 0$ and thus $x(t) = \tan(t)$. It is clear that $\lim_{t\to\pi/2} x(t) = +\infty$ as depicted in Figure 1.2. We conclude, in contrast with linear systems, that solutions are not defined for all time. For $t_0 = 0$ and $x(t_0) = 0$ the solution is only defined on the interval $[0, \pi/2)$ since it escapes to infinity as time approaches the *finite escape time*: $\pi/2$.

Example 1.2. Non-uniqueness of solutions: Consider the 1-dimensional nonlinear ODE $\dot{x} = x^{2/3}$, $x(t_0) = 0 \in \mathbb{R}$. There are two scalar functions of time, $x(t)$, that satisfy the differential equation and the initial condition:

$$x(t) = 0, \quad t \geq t_0 \qquad \text{and} \qquad x(t) = \left(\frac{t - t_0}{3}\right)^3, \quad t \geq t_0.$$

Hence, both of these functions are solutions which shows that solutions are not unique (see Figure 1.2). This phenomenon is not particular to scalar differential equations and occurs for all dimensions.

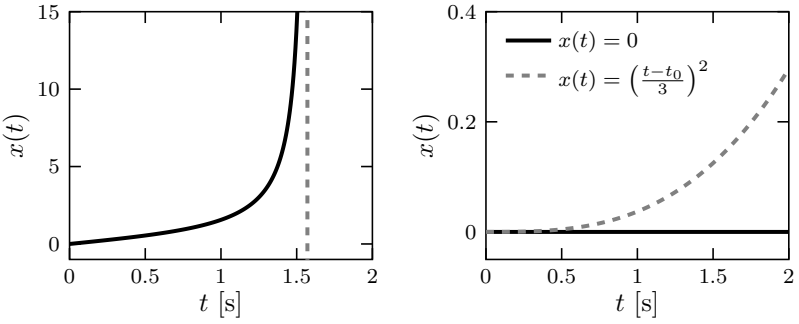


Figure 1.2. Two examples of behavior unique to nonlinear systems: *Finite Escape Time (left):* The solution exists only on a bounded interval of time. This phenomenon occurs for all dimensions. *Non-unique solutions (right):* There is more than one solution that satisfies the ODE! This phenomenon occurs for all dimensions.

Lyapunov Functions and Stability: Nonlinear systems, despite the fact that closed-form solutions cannot be explicitly found and may not exist for all time, can often be understood through certificates on their behavior. The quintessential examples are Lyapunov functions (introduced in Lecture 7 and developed in depth in Lectures 8-12), which provide certificates of stability, i.e., functional conditions that guarantee (local) stable behavior of the system. We illustrate this certification process utilizing Lyapunov functions with a simple example that will be returned to in this lecture in the context of controller synthesis.

Consider a downward pendulum, as depicted in Figure 1.3, where the angle θ is measured from the bob's bottommost position. The dynamics are obtained from the Lagrangian:

$$\mathtt{L}(\theta, \dot\theta) = \underbrace{\frac{1}{2} m l^2 \dot\theta^2}_{\text{Kinetic Energy}} - \underbrace{- mgl\cos(\theta)}_{\text{Potential Energy}}, \tag{1.1}$$

where l is the length of the pendulum, m is the mass, and g is the acceleration due to gravity. For simplicity of notation, throughout the remainder of this lecture we will take $m = l = g = 1$, yielding:

$$\mathtt{L}(\theta, \dot\theta) = \frac{1}{2}\dot\theta^2 + \cos(\theta). \tag{1.2}$$

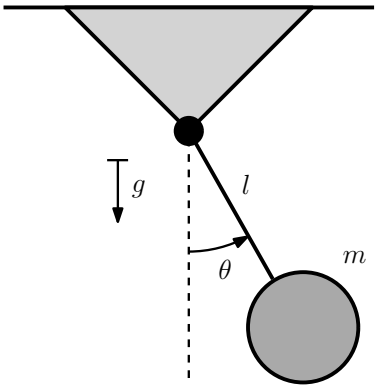


Figure 1.3. The classic "downward" pendulum.

Utilizing the Euler-Lagrange equations (see Lecture 10):

$$\frac{d}{dt}\left(\frac{\partial \mathtt{L}}{\partial \dot\theta}\right) - \frac{\partial \mathtt{L}}{\partial \theta} = 0, \tag{1.3}$$

we obtain the equations of motion for the pendulum system:

$$\ddot\theta + \sin(\theta) = 0. \tag{1.4}$$

Moreover, by defining a state vector $x \in \mathbb{R}^2$ consisting of the angle, θ, and angular velocity, $\dot\theta$:

$$x \triangleq \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \theta \\ \dot\theta \end{bmatrix},$$

we can rewrite the dynamics as a nonlinear ODE in $\mathbb{R}^2$ of the form:

$$\dot x = \underbrace{\begin{bmatrix} x_2 \\ -\sin(x_1) \end{bmatrix}}_{f(x)}. \tag{1.5}$$

This system has an equlibrium point at $x = 0$, i.e., at $\theta = 0$ and $\dot{\theta} = 0$, representing the stationary downward position of the pendulum. This can be seen by noting that $f(0) = 0$ and thus any solution starting at $x = 0$ remains there for all future time. Note that this system has a collection of isolated equilibrium points, unlike linear systems where isolated equilibrium points are unique.

To certify stability we consider the energy $\mathtt{E}$ of the system, obtained by adding the potential and kinetic energy introduced in (1.1):

$$\mathtt{E}(x_1, x_2) = \underbrace{\frac{1}{2}x_2^2}_{\text{Kinetic Energy}} + \underbrace{-\cos(x_1).}_{\text{Potential Energy}}$$

We are interested in certifying the stability of the downward position of the pendulum, i.e., around the equilibrium point $x = 0$. Thus we consider the following function that is zero at this point:

$$\mathtt{V}(x) \triangleq \mathtt{E}(x_1, x_2) - \mathtt{E}(0, 0) = \frac{1}{2}x_2^2 - \cos(x_1) + 1, \tag{1.6}$$

obtained from $\mathtt{E}$ by subtracting the value of the energy at the equilibrium point. The function $\mathtt{V}$ is termed a *candidate Lyapunov function* and has the property that it is a *positive definite function*, i.e., $\mathtt{V}(x_1, x_2) \geq 0$ and $\mathtt{V}(x_1, x_2) = 0$ if and only if $(x_1, x_2) = (0, 0)$ (locally, in a neighborhood of the origin). To informally see this, we note that near the origin we can utilize the small angle approximation, $\cos(x_1) \approx 1 - \frac{x_1^2}{2}$, wherein:

$$\mathtt{V}(x) \approx \frac{1}{2}\left(x_2^2 + x_1^2\right), \tag{1.7}$$

and this function is clearly always positive except at $(0, 0)$.

Let $x(t)$ be a solution to (1.5) with initial condition $x(t_0)$ at time t_0. Differentiating the "energy-like" function (1.6) along these solutions with respect to time yields:

$$\begin{aligned} \dot{\mathtt{V}}(x(t)) = \frac{d}{dt}\mathtt{V}(x(t)) &= \frac{\partial \mathtt{V}}{\partial x_1}\Big|_{x(t)}\dot{x}_1(t) + \frac{\partial \mathtt{V}}{\partial x_2}\Big|_{x(t)}\dot{x}_2(t) \\ &= \underbrace{\frac{\partial \mathtt{V}}{\partial x_1}\Big|_{x(t)}}_{\sin(x_1(t))} x_2(t) + \underbrace{\frac{\partial \mathtt{V}}{\partial x_2}\Big|_{x(t)}}_{x_2(t)}(-\sin(x_1(t))) \\ &= 0. \end{aligned} \tag{1.8}$$

Thus, $\dot{\mathtt{V}}$ is zero and $\mathtt{V}$ is constant. Therefore $\mathtt{V}(x(t)) = \mathtt{V}(x(t_0))$ for all $t \geq t_0$ and so solutions must stay in the set defined by $\mathtt{V}(x) \leq \mathtt{V}(x(t_0))$, i.e., for "small" $c \in \mathbb{R}_{\geq 0}$ the set:

$$\Omega_c = \left\{x \in \mathbb{R}^2 \mid \mathtt{V}(x) \leq c\right\},$$

is forward invariant (see Lecture 12) by virtue of the fact that $\dot{\mathtt{V}} = 0$:

$$x(t_0) \in \Omega_c \implies V(x(t_0)) \leq c \overset{\dot{\mathtt{V}}(x)=0}{\implies} \mathtt{V}(x(t)) \leq c, \ \forall\, t \geq t_0 \implies x(t) \in \Omega_c, \ \forall\, t \geq t_0.$$

Additionally, since $\mathtt{V}$ is a positive definite function, Ω_c is also compact (closed and bounded) for sufficiently small $c \in \mathbb{R}_{\geq 0}$ and therefore forms a neighborhood of the origin. This allows us to conclude that the system is stable: if it starts in a neighborhood of the origin, $x(t_0) \in \Omega_c$, it will stay in a neighborhood of the origin for all time: $x(t) \in \Omega_c, \ \forall\, t \geq t_0$.

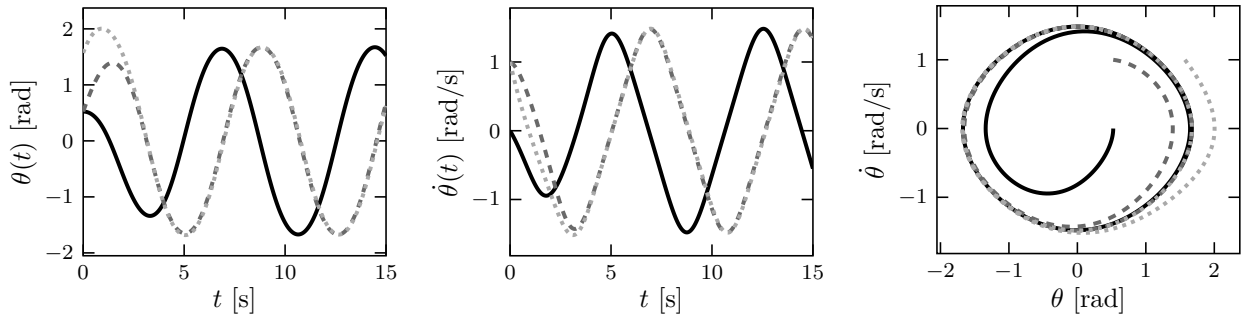


Figure 1.4. Solutions converge to a stable periodic orbit, but do not converge to the origin (which is unstable). This phenomenon occurs in nonlinear systems for all dimensions above 1, but does not occur in linear systems. In fact, in this case, the linearization of the nonlinear system at the origin would only say the origin is an unstable equilibrium point but would say nothing about the existence of stable periodic motions.

Example 1.3. Stable periodic orbits: Nonlinear systems can display *asymptotically stable* periodic behaviors (see Lecture 6 for notions of stability and Lectures 13-15 for characterizations of stable periodic motion). To provide an example of this, consider a pendulum being influenced by an external force, i.e., the system governed by the differential equation:

$$\ddot{\theta} + \sin(\theta) + \underbrace{\alpha\dot{\theta}\left(\frac{\dot{\theta}^2}{2} - \cos(\theta) - c\right)}_{\text{External Forcing}} = 0, \qquad c \in (0,1), \qquad \alpha \in \mathbb{R}_{>0}.$$

By defining a state vector[1] $x = (x_1, x_2) \in \mathbb{R}^2$ with $x_1 = \theta$ and $x_2 = \dot{\theta}$, as in the previous example, we can rewrite the dynamics as a nonlinear system in $\mathbb{R}^2$ of the form:

$$\dot{x} = f(x) = \begin{bmatrix} x_2 \\ -\sin(x_1) - \alpha x_2 \left(\frac{x_2^2}{2} - \cos(x_1) - c\right) \end{bmatrix}. \tag{1.9}$$

Unlike for linear systems, there exists no simple closed form solution for the solutions of this system. Additionally, and very importantly, this system has a periodic orbit, i.e., a solution $x(t)$ satisfying $x(t) = x(t+T)$ for some $T > 0$, that is asymptotically stable. This implies that nearby solutions will converge to the periodic orbit (see Figure 1.4). We will be able to prove this in later chapters (e.g., Lecture 13) with tools from nonlinear dynamics *without* needing to explicitly solve for the solution. This type of stability cannot occur in linear systems; periodic orbits do exist, but cannot be asymptotically stable, i.e., nearby solutions cannot converge to periodic orbits.

1.2 Nonlinear Control: Motivating Simple Example

To motivate nonlinear controller design, and hence nonlinear systems more generally, we begin by considering one of the simplest examples of a nonlinear control problem: stabilizing an inverted pendulum upright. While this example is simple to the point of apparent triviality, it effectively illustrates the large class of available nonlinear control design techniques that will be developed throughout the lectures. In many ways, this can be viewed as the simplest representation of more

[1]Note that, for simplicity, we often denote column vectors inline without the necessary transposes, i.e., the proper notation here would be $x = (x_1, x_2)^T \in \mathbb{R}^2$. We assume the reader can infer the proper collection of transposes to yield the appropriate column vectors.

complex systems of interest ranging from cars to bipedal robots. In the latter case, we view the base of the pendulum as the "ankle" where the torque is applied to the robot and the pendulum bob as the center of mass of the robot (as illustrated in Figure 1.5).

Control System Modeling. We return to the example of a pendulum considered in the previous section, except now inverted. This inverted pendulum is depicted in Figure 1.5, where the angle θ is measured from bob's topmost position. As in the case of the pendulum (and again taking $m = l = g = 1$ for simplicity), the control system dynamics is obtained from the corresponding Lagrangian:

$$\mathtt{L}(\theta, \dot{\theta}) = \frac{1}{2}\dot{\theta}^2 - \cos(\theta), \tag{1.10}$$

and the (forced) Euler-Lagrange equations:

$$\frac{d}{dt}\left(\frac{\partial \mathtt{L}}{\partial \dot{\theta}}\right) - \frac{\partial \mathtt{L}}{\partial \theta} = u. \tag{1.11}$$

Here u denotes the torque applied to the base of the pendulum, i.e., the "non-conservative forces" (in this case torque produced by the actuator) appear on the right hand side of the Euler-Lagrange equations. The result is the equations of motion for the controlled pendulum:

$$\ddot{\theta} - \sin(\theta) = u. \tag{1.12}$$

Choosing $x = (x_1, x_2) \in \mathbb{R}^2$, where $x_1 = \theta$ and $x_2 = \dot{\theta}$, results in a nonlinear control system in the standard *control affine* form:

$$\dot{x} = \underbrace{\begin{bmatrix} x_2 \\ \sin(x_1) \end{bmatrix}}_{f(x)} + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{g(x)} u. \tag{1.13}$$

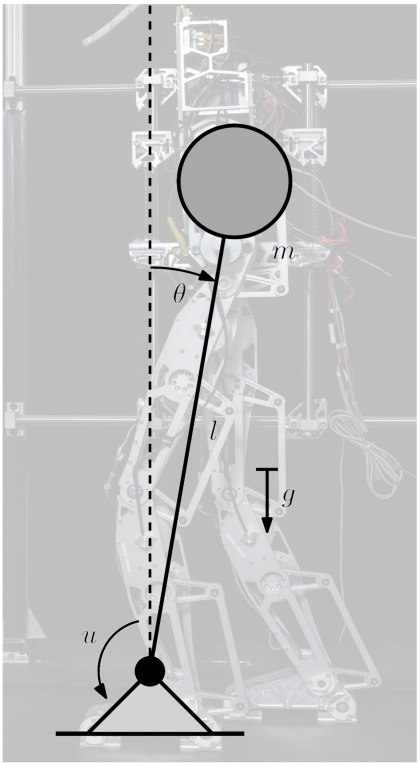


Figure 1.5. An inverted pendulum, viewed as a low-dimensional representation of a humanoid robot (in this case AMBER 3M) consisting of the center of mass and an ankle joint above the foot.

I.e., the right-hand side of (1.13) is the affine function of the control input u given by $f(x) + g(x)u$. This is the central form of systems that will be considered since most systems of interest are either in this form or can be converted to this form (see Lecture 18).

Linear Control Design. One could approach the above nonlinear system from the perspective of linear systems. In essence, this would amount to first linearizing the system, then designing a linear controller, and finally applying that linear controller to the nonlinear system. Linearizing the system (in this case, via the small angle assumption: $\sin(x_1) \approx x_1$) results in the linear control system:

$$\dot{x} = \underbrace{\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}}_{F} x + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{G} u. \tag{1.14}$$

This system is controllable (as a simple check of the controllability matrix shows) and its eigenvalues can be freely placed by a suitably designed linear controller. A specific example of a control law stabilizing the pendulum in its upright position is given by placing all the closed loop eigenvalues at $-\varepsilon$ for $\varepsilon > 0$ via the controller:

$$u = k(x) = \underbrace{\begin{bmatrix} -(1+\varepsilon^2) & -2\varepsilon \end{bmatrix}}_{K} x = -x_1 - \varepsilon^2 x_1 - 2\varepsilon x_2, \tag{1.15}$$

which yields a stable closed loop system:

$$\dot{x} = Fx + GKx = \begin{bmatrix} 0 & 1 \\ -\varepsilon^2 & -2\varepsilon \end{bmatrix} x. \tag{1.16}$$

Results from nonlinear control actually guarantee that this procedure works for more general nonlinear systems (see Lecture 8); yet, in all cases, these results are local and provide no information about the region where they are guaranteed to hold. Less formally, there is something unsatisfying about reducing nonlinear systems to linear ones as it removes what makes nonlinear systems interesting—their complexity—and prevents the use of this complexity to design better controllers.

Feedback Linearizing Control Design. Rather than approximating the nonlinear dynamics of the controlled pendulum with its linearization (through the small angle assumption) we can, instead, avoid approximations and use feedback to make the system linear. This is the approach taken by several constructive methods in nonlinear control and is termed *feedback linearization* (detailed in Lectures 18-22). In particular, noting that the nonlinearities appear in the equation governing the evolution of x_2, along with the input, allows one to cancel them out and then apply a linear controller to the resulting linear system. As an illustration of this technique, consider the controller:

$$u = k(x) + v = -\sin(x_1) + v, \tag{1.17}$$

where $v \in \mathbb{R}$ is an auxiliary control input, which results in the linear control system:

$$\dot{x} = f(x) + g(x)k(x) + g(x)v = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v. \tag{1.18}$$

In fact, it results in a very special linear control system termed a *double integrator*:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= v, \end{aligned} \tag{1.19}$$

and the controller $v = k'(x) = -\varepsilon^2 x_1 - 2\varepsilon x_2$ yields the closed loop dynamics given in (1.16); except, in this case, these dynamics are the actual dynamics of the nonlinear system after applying the "feedback linearizing" control law:

$$\dot{x} = f(x) + g(x)(k(x) + k'(k)) = (F + GK)x, \tag{1.20}$$

and, as a result, the control law:

$$u = k(x) + k'(x) = -\left(\sin(x_1) + \varepsilon^2 x_1 + 2\varepsilon x_2\right), \tag{1.21}$$

stabilizes the pendulum upright.

Energy (Lyapunov) Based Controller Design. The methods presented so far rely on essentially "forcing" the nonlinear system to act like a linear system. An alternative approach is to use the pendulum's energy to shape its dynamics—this motivates the use of a powerful tool in nonlinear control, *control Lyapunov functions (CLFs)* (introduced in Lecture 24 and central to Lectures 25-30 both in the context of stability and safety). CLFs allow for the synthesis of entire classes of controllers that can be generated via optimization-based methods and implemented in real-time. This will be the core method used on applications, such as the robotic systems shown in Fig 1.1, that will be highlighted throughout Part 2 of this book.

The main idea in the context of the pendulum is to virtually rotate it, via control, so that it acts like a pendulum stable to the downward position that was introduced in the previous section. Consider the Lypaunov function $\mathtt{V}$ for the downward pendulum, as given in (1.6), derived from the energy, $\mathtt{E}$, of that system:

$$\mathtt{V}(x) = \frac{1}{2}x_2^2 - \cos(x_1) + 1. \tag{1.22}$$

While this Lyapunov function satisfied $\dot{\mathtt{V}} \leq 0$ for the downward pendulum, the inverted pendulum will not be stable without control. Thus, the goal is to synthesize a controller that will cause $\dot{\mathtt{V}}$ to decrease and thus stabilize the system. That is, we wish to use the energy-based Lyapunov function $\mathtt{V}$ to make the unstable pendulum act like a stable pendulum.

Differentiating the "energy-like" function (1.22) with respect to time along solutions of the system (1.13) (as in (1.8), except in this case we make the dependence on time implicit) yields:

$$\begin{aligned}
\dot{\mathtt{V}}(x) &= \frac{\partial \mathtt{V}}{\partial x_1}\Big|_x \dot{x}_1 + \frac{\partial \mathtt{V}}{\partial x_2}\Big|_x \dot{x}_2 \\
&= \frac{\partial \mathtt{V}}{\partial x_1}\Big|_x x_2 + \frac{\partial \mathtt{V}}{\partial x_2}\Big|_x (\sin(x_1) + u) \\
&= x_2(u + 2\sin(x_1)).
\end{aligned} \tag{1.23}$$

This is often written in the form of *Lie derivatives* (introduced in Lecture 18):

$$\dot{\mathtt{V}}(x) = \underbrace{2x_2\sin(x_1)}_{L_f\mathtt{V}(x) \triangleq \frac{\partial \mathtt{V}}{\partial x} f(x)} + \underbrace{x_2}_{L_g\mathtt{V}(x) \triangleq \frac{\partial \mathtt{V}}{\partial x} g(x)} u. \tag{1.24}$$

Therefore, with the goal of picking a control law that causes $\dot{\mathtt{V}}$ to decrease (and thereby ultimately making its value converge to its minimum, which is the point that we are stabilizing the system to: $(0,0)$), i.e., with the goal of enforcing the inequality:

$$\dot{\mathtt{V}}(x) = L_f\mathtt{V}(x) + L_g\mathtt{V}(x)u \leq 0,$$

we pick the control law:

$$u = k(x) = -2\sin(x_1) - \gamma x_2, \tag{1.25}$$

for some $\gamma > 0$, yielding the *negative semi-definite* function:

$$\dot{\mathtt{V}}(x) = -\gamma x_2^2.$$

Although $\dot{\mathtt{V}}$ is only negative semi-definite (i.e., $\dot{\mathtt{V}}(x_1, x_2) \leq 0$, yet $\dot{\mathtt{V}}(x_1, x_2) = 0$ does not imply $(x_1, x_2) = (0, 0)$), asymptotic stability is still guaranteed by the use of the *LaSalle's invariance principle* (introduced in Lecture 10). This can be best seen by noting that the control law (1.25)

transforms the original system (1.13) to the dynamics of a downward pendulum (see (1.5)) with damping:

$$\dot{x} = \begin{bmatrix} x_2 \\ -\sin(x_1) - \gamma x_2 \end{bmatrix}. \tag{1.26}$$

To see that this system is stable (beyond the physical intuition) one can look at the resulting linearized system via the small angle assumption (as was done to obtain (1.14)):

$$\dot{x} = \underbrace{\begin{bmatrix} 0 & 1 \\ -1 & -\gamma \end{bmatrix}}_{A_s} x, \tag{1.27}$$

and check the corresponding eigenvalues, $\frac{1}{2}\left(-\gamma \pm \sqrt{\gamma^2 - 4}\right)$, which have negative real part for any value of $\gamma > 0$.

Lyapunov Based Controller Design. To design controllers, we can take inspiration from the two presented nonlinear control design methods—feedback linearizing control design and energy based control design—in order to try to achieve the best of both worlds; in many ways, this "two-pronged" approach has proven to be the most successful in nonlinear control design. That is, it provides a systematic way of constructing control Lyapunov functions that, whenever possible, take inspiration from the natural dynamics of the system (see Lecture 25).

We begin by considering a quadratic Lyapunov function candidate of the form $\mathtt{V}(x) = x^T P x$. The matrix P is to be chosen so that $\dot{\mathtt{V}}(x)$ becomes a negative-definite function when considering the stable linearized system (1.27) resulting from the application of the energy based controller. The inequality $\dot{\mathtt{V}}(x) < 0$ can be written as a matrix equation by first choosing the matrix:

$$Q = \gamma I = \begin{bmatrix} \gamma & 0 \\ 0 & \gamma \end{bmatrix}, \quad \gamma > 0, \tag{1.28}$$

and then solving the *Lyapunov equation* (as will be seen in Lecture 8):

$$A_s^T P + P A_s = -Q, \tag{1.29}$$

where A_s is the matrix in (1.27) describing the linearized dynamics of the pendulum in closed loop with the energy based controller. The end result is the *Lyapunov matrix* P given by:

$$P = \frac{1}{2}\begin{bmatrix} (\gamma^2 + 2) & \gamma \\ \gamma & 2 \end{bmatrix}. \tag{1.30}$$

The importance of the Lyapunov equation is that, since Q is positive definite, it implies that:

$$\dot{\mathtt{V}}(x) = x^T A_s^T P x + x^T P A_s x = x^T (A_s^T P + P A_s) x = -x^T Q x < 0 \quad \text{for all} \quad x \neq 0.$$

Therefore $\dot{\mathtt{V}}(x)$ is negative definite and $\mathtt{V}$ decays in the desired fashion. In fact, $\mathtt{V}$ decreases exponentially since the same equation implies that:

$$\dot{\mathtt{V}}(x) \leq -\lambda \mathtt{V}(x) \qquad \Longrightarrow \qquad \mathtt{V}(x(t)) \leq e^{-\lambda t}\mathtt{V}(x(0)) \qquad \Longrightarrow \qquad \mathtt{V}(x(t)) \overset{t\to\infty}{\longrightarrow} 0,$$

where[2] $\lambda = \frac{\lambda_{\min}(Q)}{\lambda_{\max}(P)}$ and the first implication follows from the Comparison Lemma discussed in Lecture 3.

We can now use the Lyapunov function $\mathtt{V}(x) = x^T P x$ to obtain a stabilizing controller. Taking the time derivative of this Lyapunov function along solutions of the nonlinear system (1.13) yields:

$$\dot{\mathtt{V}}(x,u) = \underbrace{\left(x_2\left(\left(\gamma^2+2\right)x_1+\gamma x_2\right)+\sin(x_1)(\gamma x_1+2x_2)\right)}_{L_f\mathtt{V}(x)=2x^TPf(x)} + \underbrace{(\gamma x_1+2x_2)}_{L_g\mathtt{V}(x)=2x^TPg(x)}\ u. \tag{1.31}$$

With the goal of achieving *exponential stability*, we can seek a control law $u = k(x)$ that satisfies:

$$L_f\mathtt{V}(x) + L_g\mathtt{V}(x)k(x) \leq -\lambda \mathtt{V}(x),$$

for an appropriate constant $\lambda > 0$ (in this case, we take $\lambda = \gamma$). An example of such a control law is given by[3]:

$$k(x) = \frac{-\lambda \mathtt{V}(x) - L_f\mathtt{V}(x)}{L_g\mathtt{V}(x)}, \tag{1.32}$$

but this controller still does not optimally leverage the dynamics. The final controller development will utilize these ideas in the context of optimization-based controllers.

Optimization-based Control. Recent developments in nonlinear control have established that one can go beyond previously discussed explicitly defined controllers and, instead, view controllers as being implicitly defined by the result of an optimization problem (that can be solved online in real-time). This allows one to greatly extend the class of considered controllers and incorporate physical constraints into controller design (this is the core idea underlying Lectures 24-30).

To provide a concrete example, we return to the Lyapunov based controller. Suppose that, instead of picking a specific feedback control law $u = k(x)$, we search for any u that satisfies:

$$\dot{\mathtt{V}}(x) = L_f\mathtt{V}(x) + L_g\mathtt{V}(x)u \leq -\lambda \mathtt{V}(x). \tag{1.33}$$

We know that such a u exists since we can simply pick $u = k(x)$ as given in (1.32). Yet, we might be able to find a "better" u. For example, we could solve the following optimization problem:

$$\begin{aligned} k^*(x) = \underset{u\in\mathbb{R}}{\operatorname{argmin}} \quad & u^2 \\ \text{s.t.} \quad & L_f\mathtt{V}(x) + L_g\mathtt{V}(x)u \leq -\lambda \mathtt{V}(x), \end{aligned} \tag{1.34}$$

where $L_f\mathtt{V}$ and $L_g\mathtt{V}$ are given in (1.31). The solution of this optimization problem is, therefore, a feedback control law $u = k^*(x)$ that finds the minimum torque needed to achieve the convergence rate provided by the original control law $u = k(x)$ as described by the decay of $\mathtt{V}$ (mathematically defined by $\dot{\mathtt{V}}(x)$). Importantly, (1.34) is affine in its constraints (in terms of the argument u) and quadratic in its cost, and is therefore a *quadratic program* (QP); hence, this problem can be solved very fast (in real-time) using a variety of existing solvers. Moreover, the solution to (1.34) has a closed form solution, termed the *minimum norm controller*, given by:

$$k^*(x) = \begin{cases} \frac{-L_f\mathtt{V}(x)-\lambda\mathtt{V}(x)}{L_g\mathtt{V}(x)} & \text{if} \quad L_f\mathtt{V}(x)+\lambda\mathtt{V}(x) > 0 \\ 0 & \text{if} \quad L_f\mathtt{V}(x)+\lambda\mathtt{V}(x) \leq 0 \end{cases}. \tag{1.35}$$

[2] $\lambda_{\min}(Q)$ and $\lambda_{\max}(P)$ denote, respectively, the smallest and largest eigenvalues of the matrices Q and P.

[3] Note that this control law is only well defined for points x such that $L_gV(x) \neq 0$.

Therefore, the controller does nothing when the natural dynamics of the system satisfy the convergence criteria, and produces a minimal control value to achieve the desired convergence otherwise.

The advantage of the optimization-based formulation of controllers is that additional constraints can be added to create a richer set of behaviors. For example, the actuator controlling the pendulum has a maximum torque it can exert, $-u_{\min} \leq u \leq u_{\max}$, and if the controller requires more torque than this maximum value the actuator will not be able to achieve the desired behavior. Optimization-based controllers allow for this constraint to be taken into account when computing a desired control action. In particular, we can consider the following modification of (1.34):

$$k^*(x) = \underset{(x,\delta)\in\mathbb{R}^2}{\operatorname{argmin}} \quad u^2 + p\delta^2 \tag{CLF-QP}$$

$$\text{s.t.} \quad L_f\mathtt{V}(x) + L_g\mathtt{V}(x)u \leq -\lambda\mathtt{V}(x) + \delta. \tag{1.36}$$

$$u \leq u_{\max} \tag{1.37}$$

$$-u \leq u_{\min} \tag{1.38}$$

This is again a quadratic program, but in this case there is no closed form solution. Note that δ is a relaxation term; intuitively, it allows for the convergence of the controller to be temporarily relaxed in order to satisfy the desired torque constraint. The constant $p \in \mathbb{R}_{>0}$ is a penalty (typically chosen to be a large number) that encourages the relaxation term δ to go to zero.

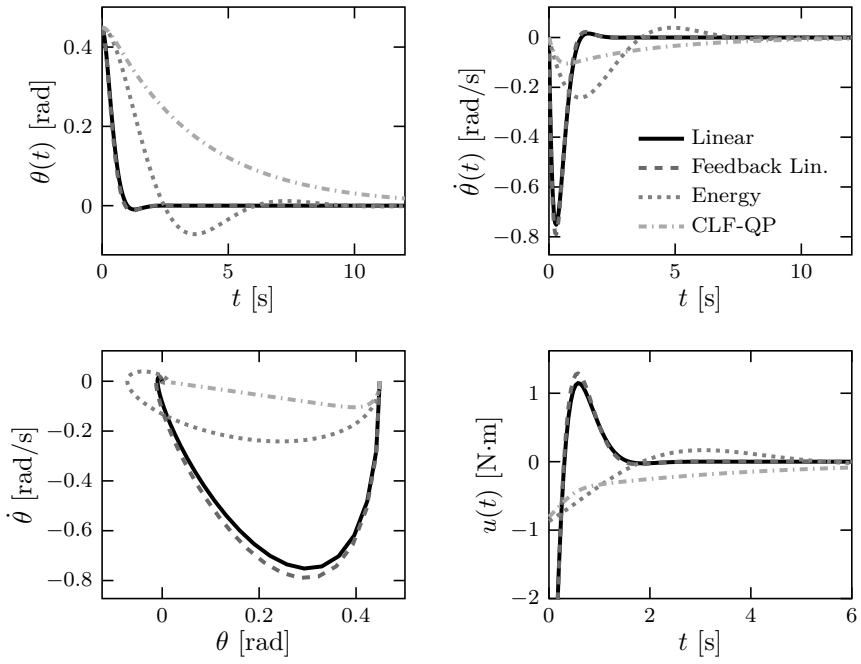


Figure 1.6. Comparison of the behaviors of all four controllers: the linear (1.15), feedback linearizing (1.21), energy shaping (1.25), and optimization-based (CLF-QP). Top: Angles over time as they converge (left), and velocities over time (right). Bottom left: phase portraits for the different controllers. Bottom right: torque over time.

Performance of Controllers. It is informative to compare all the introduced controllers to stabilize the pendulum upright. Even with such a simple example, one can see differences in the performance of the controllers—this is indicative of nonlinear systems, as different controllers can have dramatically different behavior even when they achieve the same overall objective. Let us denote the four controllers introduced in (1.15), (1.21), (1.25), (CLF-QP) by $k_{(1.15)}(x)$, $k_{(1.21)}(x)$, $k_{(1.25)}(x)$, and $k_{(\text{CLF-QP})}(x)$, respectively. As it can be seen in Figure 1.6, all four controllers stabilize the pendulum from an initial condition of $(\pi/7, 0)$ (with parameters $m = l = 1$, $g = 1$, control

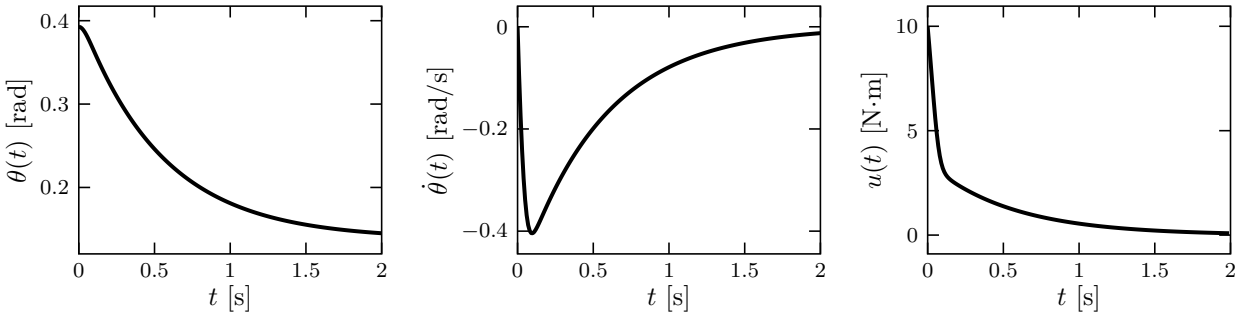


Figure 1.8. Simulation results for the Segway-type robot (show in Figure 1.7). These show the robot converging to the upright position from an initial condition of $x = (0, \frac{\pi}{8})$.

parameters $\varepsilon = 3$, $\lambda = 1$, $p = 10^4$, and minimum/maximum torque $u_{\min} = u_{\max}$ =1Nm). The true test of these controllers, since they all achieve the objective of stabilization, is to study the torque they require. In particular, there are the following largest torque values produced by the different controllers in the simulation results depicted in Figure 1.6:

$$\max(|k_{(1.15)}(x)|) = 4.488 > \max(|k_{(1.21)}(x)|) = 4.4731 > \max(|k_{(1.25)}(x)|) = 0.8678, \quad \max(|k_{(\text{CLF-QP})}(x)|) = 1. \tag{1.39}$$

Therefore, the more advanced nonlinear controllers—due to the fact that they naturally utilize the natural dynamics of the system—have progressively lower torque requirements. Note, however, that lower torques result in slower convergence towards the upright position. The one exception is $k_{(\text{CLF-QP})}$ due to the introduction of the torque bounds—it matches the torque bound briefly (since that is all that is required, i.e., it is the only hard constraint in the QP).

1.3 Application to Hardware

Nonlinear controllers are ultimately judged by their ability to be realized on hardware and, as such, affect the behavior of highly dynamic systems so as to render them safe and stable. To demonstrate the power of the theoretic concepts developed throughout these lectures, we will highlight their application on a diverse set of hardware platforms. This includes everything form ground robots to legged and flying robots. These platforms are illustrated in Figure 1.1. It is important to note that all of the implementation details cannot be given in a reasonable fashion, but the reader is encouraged to refer to the associated papers for this additional content.

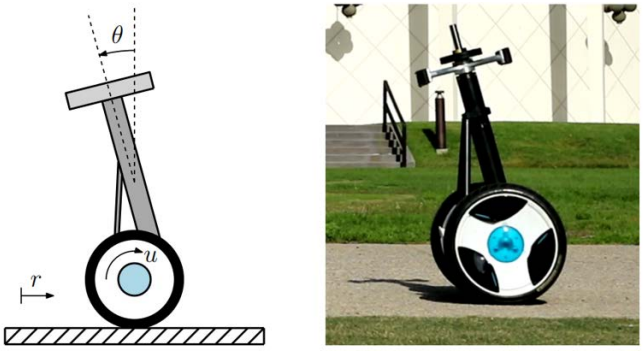


Figure 1.7. A wheeled inverted pendulum (Segway) restricted to the plane, i.e., moving in two-dimensions. Coordinates for this system can be (locally) described by a vertical angle, θ, and an horizontal position r. The control input to the system, u, consists of voltages applied to motors at the wheels.

To give a feel for the application on nonlinear control on hardware, we outline two applications where the controllers developed in this section where applied to admittedly more complex models: a wheeled inverted pendulum robot (Segway) and the bipedal robot Cassie.

Appication 1.1: Wheeled Inverted Pendulum Robot (Segway)

To illustrate (a subset) of the concepts presented in this lecture, and that will be developed throughout this book in more detail, consider a wheeled inverted pendulum robot, i.e., a "Segway". This robot consists of two actuated wheels that hold up a pendulum component as shown in Figure 1.7. The general goal is to keep this pendulum upright—much like the inverted pendulum exampled considered throughout this lecture. As such, while the dynamics are more complex, the same general principles can be applied; this will be formalized in subsequent lectures.

A Segway-type robot moving in the plane (i.e., restricted to moving in two-dimensional cartesian space) with wheels driven by a direct current motor can be modeled through a lateral displacement r, a lateral velocity $\dot{r}$, a pendulum angle θ, and a pendulum angular velocity $\dot{\theta}$. The control system describing the time evolution of these variables is more complex than that of the pendulum but can be written in the general "normal" form (see Lecture 20):

$$\begin{aligned} \dot{x} &= f(x,z) + g(x,z)u \\ \dot{z} &= \omega_0(x,z) + \omega_1(x,z)u, \end{aligned} \tag{1.40}$$

where $x = (\theta, \dot{\theta}) \in \mathbb{R}^2$ is the component of the system describing the evolution of the angle of the pendulum and $z = (r, \dot{r}) \in \mathbb{R}^2$ describes the evolution of lateral position of the robot. The input, $u \in \mathbb{R}$, is the commanded voltage at the motors.

Despite the additional complexity of these system dynamics, the same general methodology applied to the pendulum can be applied in this case; it will be seen in Part 2 of the book that these concepts generalize in a similar fashion to a large class of nonlinear control systems. The control objective is to drive $\theta \to 0$, i.e., stabilize the pendulum on the wheeled platform upright. Under the assumption that $g(x,z) \neq 0$, we can "feedback linearize" part of the system dynamics to stabilize the pendulum component of the dynamics (per the methods that will be introduced in Lecture 21):

$$u = k(x,z) = \frac{1}{g(x,z)}(-f(x,z) + A_s x) \qquad \Longrightarrow \qquad \dot{x} = A_s x, \tag{1.41}$$

with A_s as in (1.27). The result is a stable linear system on the θ dynamics, hence the pendulum is stabilized upright. Note that while the pendulum is stabilized, this says nothing of the stability of the translational component of the system, encoded by the dynamics of z in (1.40).

Instead of using the "feedback linearizing" controller above, we can instead leverage the natural dynamics of the system to stabilize the pendulum upright. In particular, we can consider the Lyapunov matrix P in (1.30) obtained by solving the Lyapunov equation. This yields a control Lyapunov function $\mathtt{V}(x) = x^T P x$ defined on the pendulum state (this is termed a control Lyapunov function on the output dynamics, and will be introduced in Lecture 25). One can calculate $\dot{\mathtt{V}}(x,z,u)$ as in (1.31) but with respect to the dynamics in (1.40). It can be verified that under the assumption that $g(x,z) \neq 0$ there exists a control law $u = k(x,z)$ such that $\dot{\mathtt{V}}(x,z,k(x,\dot{r})) \leq -\lambda \mathtt{V}(x)$. As a result, we can synthesize an optimization-based controller as in (1.34) and, additionally, include input bounds in the controller via a CLF-QP. Simulations illustrating the application of a CLF-QP to the Segway are shown in Figure 1.8, wherein the pendulum angle $\theta \to 0$ as the theory indicates.

These ideas and corresponding CLF-based optimization controllers can also be realized on hardware. In particular, rather than simply stabilizing the pendulum upright we can instead track a desired pendulum angle over time: $\theta_d(x, \dot{r}, t) \triangleq \theta_e - K_v(\dot{r} - \dot{r}_d(t))$, where θ_e is a constant, K_v is a control gain, and $\dot{r}_d(t)$ is the desired translational velocity of the Segway. The control objective is now to drive $\theta \to \theta_d$, or to drive the error: $e \triangleq \theta - \theta_d \to 0$. As such, we can modify the previous Lyapunov function to be defined on the dynamics of the error e:

$$\mathtt{V}(e, \dot{e}) = \begin{bmatrix} e & \dot{e} \end{bmatrix} P \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \qquad \Longrightarrow \qquad \textbf{Goal: } \text{Find } u \text{ s.t. } \quad \dot{\mathtt{V}}(e, \dot{e}, z, u) \leq -\lambda \mathtt{V}(e). \tag{1.42}$$

One can verify that such a u can be found and the existence of such an input implies that one can utilize an optimization-based controller, and specifically a CLF-QP, that causes the pendulum angle, θ, to track the desired pendulum angle θ_d. This was realized experimentally on the hardware platform shown in Figure 1.7 with the result being tracking of θ_d as shown in Figure 1.9. Model mismatch and the realities of implementation results in small errors between the actual and desired pendulum angles. These issues will be addressed in more detail in this book in the context of input-to-state stability, robust, and adaptive control in Lecture 28 and Lecture 29.

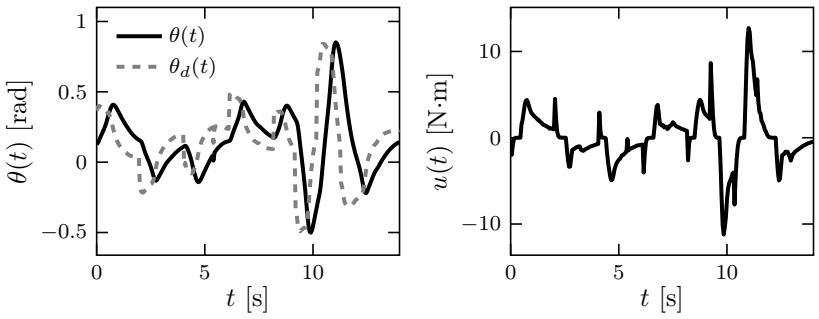
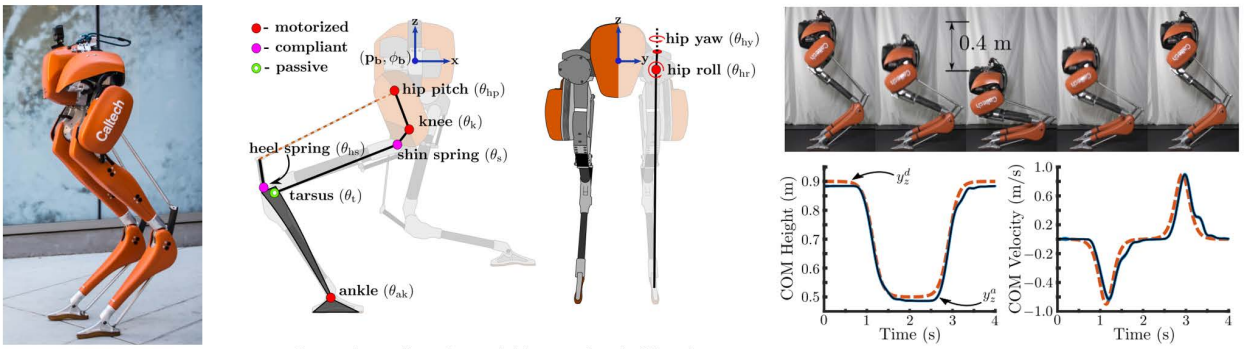


Figure 1.9. Experimental results for the Segway-type system (see Figure 1.7). The pendulum angle, θ, tracks the desired signal θ_d (subject to a time-delay due to implementation). Also shown are the inputs required to achieve this tracking.

Appication 1.2: Bipedal Robot Cassie

To demonstrate these ideas on a more complex robotic system, consider the bipedal robot Cassie as shown in Figure 1.10. The general goal is to achieve dynamic behaviors on the system ranging from stable walking to jumping. In this setting, the complexity of the robotic system results in a high dimensional nonlinear control system that represents passive joints, joints with compliance induced by springs, and joints actuated by motors. Despite the complexity of this system, the same general methodology applied to the pendulum throughout this lecture, and the Segway-type robot in Application 1.1, can be applied to Cassie.

To provide a specific example, consider the goal of achieving stable squatting on the robot. To this end, a control Lyapunov function of the general form of (1.42) can be synthesized. In this case, the error e now consists of the difference between the actual and desired behaviors. In the context of the target behavior, squatting, the height of the center of mass must move up and down in a stable fashion. The error is thus of the form: $e = y_z^a(\theta) - y_z^d(t)$ where y_z^a is the height of the center of mass and y_z^d is the desired behavior of the center of mass (this encodes an up and down motion). The Lyapunov function $\mathtt{V}$ obtained from this error can be shown to satisfy the goal in (1.42), i.e., there exists an input such that $\dot{\mathtt{V}} \leq -\lambda \mathtt{V}$. Thus this



(a) Cassie

(b) Cassie configuration variables associated with active (motorized), passive and compliant joints.

(c) Tracking error with CLF-QP while squatting

Figure 1.10. The bipedal robot Cassie (a), together with its configuration space (b). Illustrated in (c) is squatting, experimentally achieved through Lyapunov functions and optimization-based controllers, and the tracking error for the center of mass height and velocity.

error can be driven to zero exponentially (since `V` can be chosen to be quadratic in e and $\dot{e}$ as in (1.42)). Importantly, this was used to synthesize an optimization-based controller that achieved the overall control objective experimentally on Cassie: the squatting behavior shown in Figure 1.10. The tracking error is also depicted in Figure 1.10 showing how the height of the center of mass achieves the desired behavior.

Additional Reading

Many textbooks on nonlinear control devote the first chapter to discussing nonlinear systems, and behavior unique to these systems, through the use of simple systems. Good examples include Khalil [110], Sastry [183] and Perko [166]. We refer the reader to these textbooks, and others, for additional examples of nonlinear systems and the corresponding motivation for studying them.

The robot AMBER 3M shown in Figure 1.5 is described in detail in [7]. The applications to hardware considered in this lecture follow from [80] in the case of the Segway-type robot and [178] for Cassie. Additional optimization-based nonlinear controllers have been implemented on these platforms, including QP controllers on the Segway [85, 209, 206], and CLF-QP based controllers for achieving walking on Cassie [176]. Finally, a general overview of nonlinear control methods, including CLFs, for bipedal robots can be found in [175]; the use nonlinear control ideas in robotic walking will also be highlighted throughout this book—especially in later lectures.

Problems for Lecture 1

[P1.1] For the nonlinear system in Example 1.3, what happens when $c < 0$? When $c > 1$? How does setting $\alpha < 0$ change the behavior of the system?

[P1.2] Prove that the Lyapunov function candidate given in (1.22) is a positive definite function (without linearizing the function) on an appropriately defined domain.

[P1.3] *[Requires knowledge of linear control systems]* Consider the linearization of the inverted pendulum, as given in (1.14). For this linear system, in (1.15) we generated a controller that stabilizes this system by placing all of the poles on the real axis. There are "better" linear

feedback controllers that achieve the goal of stabilizing the pendulum. To this end, we can consider the linear quadratic regulator (LQR) (see Lecture 25, and specifically Remark 25.3). In particular, let P be the positive definite matrix solving the continuous-time algebraic Riccati equation (CARE):

$$F^T P + PF - PGG^T P = -Q,$$

for some positive definite matrix Q, and for F and G given in (1.14).

(a) From the matrix P solving the CARE, show that the LQR controller: $u = -\frac{1}{2}G^T Px$ stabilizes the pendulum upright. How does this behavior compare with the behavior of the system under the feedback controller given in (1.15)?

(b) Show that the feedback linearizing controller (1.17) can be coupled with the LQR controller to stabilize the nonlinear control system (1.13). How does this controller compare with the controller in (1.21)?

[P1.4] Show that the Lyapunov matrix P in (1.30) solves the Lyapunov equation (1.29) for the given Q in (1.28). Additionally, show that the matrix P is positive definite for all $\gamma > 0$.

[P1.5] Suppose that the goal for the inverted pendulum is no longer to stabilize it to the upright position, but rather to create a stable periodic orbit, i.e., to have the pendulum oscillate in a periodic motion such that solutions are stable to that motion. Use the system in Example 1.3, coupled with a feedback control law of the form:

$$u = k(x, v) = -\sin(x_1) + v,$$

to achieve this objective, where v is a new auxiliary control input that can be chosen to achieve the desired objective. That is, find a controller that "shapes" the dynamics to be that of the desired system.

[P1.6] Consider the controlled version of the Van der Pol oscillator (see Lecture 13, and specifically Example 13.3 and corresponding Figure 13.3), described by a control system in $\mathbb{R}^2$:

$$\dot{x} = f(x) + g(x)u = \begin{bmatrix} x_2 \\ \mu(1 - x_1^2)x_2 - x_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \tag{1.43}$$

for $\mu > 0$. For this system, synthesize a family of controllers as was done for the inverted pendulum. Specifically:

(a) *Linearization:* Linearize the system, and construct a controller that stabilize the resulting linear system to the origin: $(x_1, x_2) = (0, 0)$.

(b) *Feedback Linearization:* Use feedback linearization to linearize the dynamics through feedback and thereby synthesize a controller that stabilizes the nonlinear system to the origin.

(c) *Lyapunov Based Controller Design:* Use the stable linear system generated through feedback linearization, together with the Lyapunov equation (1.29), to generate a control Lyapunov function that exponentially stabilizes the system. Instantiate this controller via a quadratic program as in (1.33) and justify why this controller stabilizes the system.

(d) Compare the controllers synthesized in Parts (a)-(c). Which has the best performance relative to the amount of control effort needed to stabilize the system.

[P1.7] Consider an iron ball levitating within a magnetic field created by a single electromagnet. A simplified model describing the vertical position of the ball is given by:

$$\dot{x}_1 = x_2 \tag{1.44}$$
$$\dot{x}_2 = \frac{1}{2m}\lambda^2 - g \tag{1.45}$$
$$\dot{\lambda} = -\frac{R}{c}(1 - x_1)\lambda + u, \tag{1.46}$$

where x_1 denotes the ball's height, x_2 the ball's velocity, m is the ball's mass, g is gravity's constant, λ is the magnetic flux linkage in the electromagnet, R is the resistance of the electromagnet's coil, c is a positive constant modeling the electromagnet's geometry and construction, and u is the voltage applied at the coil's terminals constituting the input to this system. This simplified model describes, e.g., a magnetically levitated train where the train mass has been lumped into the ball.

(a) If the objective is to stabilize the ball's height at $x_1^* \in \mathbb{R}$, linearize the equations of motion at a suitable equilibrium.

(b) Design a linear controller stabilizing the ball at x_1^* using the linearized model.

(c) By drawing inspiration from how we used feedback linearization with the inverted pendulum, can you feedback linearize this control system and design a controller to stabilize the ball's height at $x_1^* \in \mathbb{R}$?

[P1.8] [**Advanced Problem**] Following Application 1.1, a wheeled inverted pendulum, i.e., a Segway-type robot, moving in a plane with wheels driven by a DC motor can be described by the the following control system with the tripe $(\dot{r}, \theta, \dot{\theta}) \in \mathbb{R}^3$ as the state [85]:

$$\begin{bmatrix} \ddot{r} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \frac{1}{\cos(\theta) - b}\left(\begin{bmatrix} \cos(\theta)(c_1\dot{r} + g\sin(\theta)) + c_2\dot{r} - c_3\dot{\theta}^2\sin(\theta) \\ \dot{\theta} \\ -d_1\dot{r}\cos(\theta) - d_2\dot{r} - \sin(\theta)(d_3 + \dot{\theta}^2\cos(\theta)) \end{bmatrix} + \begin{bmatrix} -\cos(\theta)c_4 - c_5 \\ 0 \\ \cos(\theta)d_4 + d_5 \end{bmatrix} u \right),$$

and where the input u is the commanded voltage to the motors, and $b, c_i, d_i \in \mathbb{R}_{>0}$, $i = 1, \ldots, 5$ and positive constants associated with the physical properties of the system.

(a) Transform the above dynamics into the form in (1.40). That is, find f, g, ω_0, and ω_1.

(b) Show that the controller in (1.41) stabilizes x to zero exponentially. Find the domain of pendulum angles, θ, such that the assumption $g(x, \dot{r}) \neq 0$ is valid. What does this say about nonlinear controllers developed with this assumption?

(c) Utilizing the Lyapunov function $\mathtt{V}$ defined in (1.22), calculate $L_f\mathtt{V}(x, \dot{r})$ and $L_g\mathtt{V}(x, \dot{r})$ as in (1.31) but for the dynamics describing the Segway. Use this to calculate a feedback control law $u = k(x)$ that satisfies:

$$\dot{\mathtt{V}}(x, \dot{r}, u) = L_f\mathtt{V}(x, \dot{r}) + L_g\mathtt{V}(x, \dot{r})u \leq 0.$$

Does this controller make $\dot{\mathtt{V}}$ negative definite or negative semi-definite?

(d) The goal is to now synthesize a different Lyapunov function utilizing the approach in Application 1.1. Specifically, define $\mathtt{V}(x) = x^T P x$ with $x = (\theta, \dot{\theta})$ and P given in (1.30).

Calculate $\dot{\mathtt{V}}(x, \dot{r}, u)$ as in (1.31) but with the Segway dynamics. Use this to demonstrate that the feedback controller given in (1.41) results in:

$$\dot{\mathtt{V}}(x, \dot{r}, k(x, \dot{r})) \leq -\lambda \mathtt{V}(x).$$

Does this controller make $\dot{\mathtt{V}}$ negative definite or negative semi-definite?

(e) Use the Lyapunov function $\mathtt{V} = x^T P x$ to defined an optimization-based controller as in (1.34). What are the benefits of this controller when compared to the explicitly defined controller $u = k(x, \dot{r})$ in (1.41)?

(f) Finally, consider the CLF-QP which factors in both the Lyapunov constraint and input bounds as via (1.36)-(1.38). What are the benefits and drawbacks of this controller relative to the controller of the form in (1.34)?