

# Backstepping

In control theory, backstepping is a technique developed circa 1990 by Petar V. Kokotovic and others<sup>[1]</sup> for designing stabilizing controls for a special class of nonlinear dynamical systems. These systems are built from subsystems that radiate out from an irreducible subsystem that can be stabilized using some other method. Because of this recursive structure, the designer can start the design process at the known-stable system and "back out" new controllers that progressively stabilize each outer subsystem. The process terminates when the final external control is reached. Hence, this process is known as *backstepping*.<sup>[2]</sup>

The backstepping approach provides a recursive method for stabilizing the origin of a system in strict-feedback form. That is, consider a system of the form<sup>[3]</sup>

$$\begin{cases} \dot{\mathbf{x}} = f_x(\mathbf{x}) + g_x(\mathbf{x})z_1 \\ \dot{z}_1 = f_1(\mathbf{x}, z_1) + g_1(\mathbf{x}, z_1)z_2 \\ \dot{z}_2 = f_2(\mathbf{x}, z_1, z_2) + g_2(\mathbf{x}, z_1, z_2)z_3 \\ \vdots \\ \dot{z}_i = f_i(\mathbf{x}, z_1, z_2, \dots, z_{i-1}, z_i) + g_i(\mathbf{x}, z_1, z_2, \dots, z_{i-1}, z_i)z_{i+1} \quad \text{for } 1 \leq i < k-1 \\ \vdots \\ \dot{z}_{k-1} = f_{k-1}(\mathbf{x}, z_1, z_2, \dots, z_{k-1}) + g_{k-1}(\mathbf{x}, z_1, z_2, \dots, z_{k-1})z_k \\ \dot{z}_k = f_k(\mathbf{x}, z_1, z_2, \dots, z_{k-1}, z_k) + g_k(\mathbf{x}, z_1, z_2, \dots, z_{k-1}, z_k)u \end{cases}$$

where

- $\mathbf{x} \in \mathbb{R}^n$  with  $n \geq 1$ ,
- $z_1, z_2, \dots, z_i, \dots, z_{k-1}, z_k$  are scalars,
- $u$  is a scalar input to the system,
- $f_x, f_1, f_2, \dots, f_i, \dots, f_{k-1}, f_k$  vanish at the origin (i.e.,  $f_i(0, 0, \dots, 0) = 0$ ),
- $g_1, g_2, \dots, g_i, \dots, g_{k-1}, g_k$  are nonzero over the domain of interest (i.e.,  $g_i(\mathbf{x}, z_1, \dots, z_k) \neq 0$  for  $1 \leq i \leq k$ ).

Also assume that the subsystem

$$\dot{\mathbf{x}} = f_x(\mathbf{x}) + g_x(\mathbf{x})u_x(\mathbf{x})$$

is stabilized to the origin (i.e.,  $\mathbf{x} = \mathbf{0}$ ) by some **known** control  $u_x(\mathbf{x})$  such that  $u_x(\mathbf{0}) = 0$ . It is also assumed that a Lyapunov function  $V_x$  for this stable subsystem is known. That is, this  $\mathbf{x}$  subsystem is stabilized by some other method and backstepping extends its stability to the  $\mathbf{z}$  shell around it.

In systems of this *strict-feedback form* around a stable  $\mathbf{x}$  subsystem,

- The backstepping-designed control input  $u$  has its most immediate stabilizing impact on state  $z_n$ .
- The state  $z_n$  then acts like a stabilizing control on the state  $z_{n-1}$  before it.
- This process continues so that each state  $z_i$  is stabilized by the *fictitious* "control"  $z_{i+1}$ .

The **backstepping** approach determines how to stabilize the  $\mathbf{x}$  subsystem using  $z_1$ , and then proceeds with determining how to make the next state  $z_2$  drive  $z_1$  to the control required to stabilize  $\mathbf{x}$ . Hence, the process "steps backward" from  $\mathbf{x}$  out of the strict-feedback form system until the ultimate control  $u$  is designed.

## Recursive Control Design Overview

1. It is given that the smaller (i.e., lower-order) subsystem

$$\dot{\mathbf{x}} = f_x(\mathbf{x}) + g_x(\mathbf{x})u_x(\mathbf{x})$$

is already stabilized to the origin by some control  $u_x(\mathbf{x})$  where  $u_x(\mathbf{0}) = 0$ . That is, choice of  $u_x$  to stabilize this system must occur using *some other method*. It is also assumed that a Lyapunov function  $V_x$  for this stable subsystem is known. Backstepping provides a way to extend the controlled stability of this subsystem to the larger system.

2. A control  $u_1(\mathbf{x}, z_1)$  is designed so that the system

$$\dot{z}_1 = f_1(\mathbf{x}, z_1) + g_1(\mathbf{x}, z_1)u_1(\mathbf{x}, z_1)$$

is stabilized so that  $z_1$  follows the desired  $u_x$  control. The control design is based on the augmented Lyapunov function candidate

$$V_1(\mathbf{x}, z_1) = V_x(\mathbf{x}) + \frac{1}{2}(z_1 - u_x(\mathbf{x}))^2$$

The control  $u_1$  can be picked to bound  $\dot{V}_1$  away from zero.

3. A control  $u_2(\mathbf{x}, z_1, z_2)$  is designed so that the system

$$\dot{z}_2 = f_2(\mathbf{x}, z_1, z_2) + g_2(\mathbf{x}, z_1, z_2)u_2(\mathbf{x}, z_1, z_2)$$

is stabilized so that  $z_2$  follows the desired  $u_1$  control. The control design is based on the augmented Lyapunov function candidate

$$V_2(\mathbf{x}, z_1, z_2) = V_1(\mathbf{x}, z_1) + \frac{1}{2}(z_2 - u_1(\mathbf{x}, z_1))^2$$

The control  $u_2$  can be picked to bound  $\dot{V}_2$  away from zero.

4. This process continues until the actual  $u$  is known, and

- The *real* control  $u$  stabilizes  $z_k$  to *fictitious* control  $u_{k-1}$ .
- The *fictitious* control  $u_{k-1}$  stabilizes  $z_{k-1}$  to *fictitious* control  $u_{k-2}$ .
- The *fictitious* control  $u_{k-2}$  stabilizes  $z_{k-2}$  to *fictitious* control  $u_{k-3}$ .
- ...
- The *fictitious* control  $u_2$  stabilizes  $z_2$  to *fictitious* control  $u_1$ .
- The *fictitious* control  $u_1$  stabilizes  $z_1$  to *fictitious* control  $u_x$ .
- The *fictitious* control  $u_x$  stabilizes  $\mathbf{x}$  to the origin.

This process is known as **backstepping** because it starts with the requirements on some internal subsystem for stability and progressively *steps back* out of the system, maintaining stability at each step. Because

- $f_i$  vanish at the origin for  $0 \leq i \leq k$ ,
- $g_i$  are nonzero for  $1 \leq i \leq k$ ,
- the given control  $u_x$  has  $u_x(\mathbf{0}) = 0$ ,

then the resulting system has an equilibrium at the **origin** (i.e., where  $\mathbf{x} = \mathbf{0}$ ,  $z_1 = 0$ ,  $z_2 = 0$ , ...,  $z_{k-1} = 0$ , and  $z_k = 0$ ) that is globally asymptotically stable.

## Integrator Backstepping

Before describing the backstepping procedure for general strict-feedback form dynamical systems, it is convenient to discuss the approach for a smaller class of strict-feedback form systems. These systems connect a series of integrators to the input of a system with a known feedback-stabilizing control law, and so the stabilizing approach is known as *integrator backstepping*. With a small modification, the integrator backstepping approach can be extended to handle all strict-feedback form systems.

## Single-integrator Equilibrium

Consider the dynamical system

$$\begin{cases} \dot{\mathbf{x}} = f_x(\mathbf{x}) + g_x(\mathbf{x})z_1 \\ \dot{z}_1 = u_1 \end{cases} \quad (1)$$

where  $\mathbf{x} \in \mathbb{R}^n$  and  $z_1$  is a scalar. This system is a cascade connection of an integrator with the  $\mathbf{x}$  subsystem (i.e., the input  $u$  enters an integrator, and the integral  $z_1$  enters the  $\mathbf{x}$  subsystem).

We assume that  $f_x(\mathbf{0}) = 0$ , and so if  $u_1 = 0$ ,  $\mathbf{x} = \mathbf{0}$  and  $z_1 = 0$ , then

$$\begin{cases} \dot{\mathbf{x}} = f_x(\underbrace{\mathbf{0}}_{\mathbf{x}}) + (g_x(\underbrace{\mathbf{0}}_{\mathbf{x}}))(\underbrace{0}_{z_1}) = 0 + (g_x(\mathbf{0}))(0) = 0 & (\text{i.e., } \mathbf{x} = \mathbf{0} \text{ is stationary}) \\ \dot{z}_1 = \underbrace{0}_{u_1} & (\text{i.e., } z_1 = 0 \text{ is stationary}) \end{cases}$$

So the origin  $(\mathbf{x}, z_1) = (\mathbf{0}, 0)$  is an equilibrium (i.e., a stationary point) of the system. If the system ever reaches the origin, it will remain there forever after.

## Single-integrator Backstepping

In this example, backstepping is used to stabilize the single-integrator system in Equation (1) around its equilibrium at the origin. To be less precise, we wish to design a control law  $u_1(\mathbf{x}, z_1)$  that ensures that the states  $(\mathbf{x}, z_1)$  return to  $(\mathbf{0}, 0)$  after the system is started from some arbitrary initial condition.

- First, by assumption, the subsystem

$$\dot{\mathbf{x}} = F(\mathbf{x}) \quad \text{where} \quad F(\mathbf{x}) \triangleq f_x(\mathbf{x}) + g_x(\mathbf{x})u_x(\mathbf{x})$$

with  $u_x(\mathbf{0}) = 0$  has a Lyapunov function  $V_x(\mathbf{x}) > 0$  such that

$$\dot{V}_x = \frac{\partial V_x}{\partial \mathbf{x}}(f_x(\mathbf{x}) + g_x(\mathbf{x})u_x(\mathbf{x})) \leq -W(\mathbf{x})$$

where  $W(\mathbf{x})$  is a positive-definite function. That is, we **assume** that we have **already shown** that this **existing simpler  $\mathbf{x}$  subsystem** is **stable (in the sense of Lyapunov)**. Roughly speaking, this notion of stability means that:

- The function  $V_x$  is like a "generalized energy" of the  $\mathbf{x}$  subsystem. As the  $\mathbf{x}$  states of the system move away from the origin, the energy  $V_x(\mathbf{x})$  also grows.
- By showing that over time, the energy  $V_x(\mathbf{x}(t))$  decays to zero, then the  $\mathbf{x}$  states must decay toward  $\mathbf{x} = \mathbf{0}$ . That is, the origin  $\mathbf{x} = \mathbf{0}$  will be a **stable equilibrium** of the system – the  $\mathbf{x}$  states will continuously approach the origin as time increases.
- Saying that  $W(\mathbf{x})$  is positive definite means that  $W(\mathbf{x}) > 0$  everywhere except for  $\mathbf{x} = \mathbf{0}$ , and  $W(\mathbf{0}) = 0$ .
- The statement that  $\dot{V}_x \leq -W(\mathbf{x})$  means that  $\dot{V}_x$  is bounded away from zero for all points except where  $\mathbf{x} = \mathbf{0}$ . That is, so long as the system is not at its equilibrium at the origin, its "energy" will be decreasing.
- Because the energy is always decaying, then the system must be stable; its trajectories must approach the origin.

Our task is to find a control  $u$  that makes our cascaded  $(\mathbf{x}, z_1)$  system also stable. So we must find a *new* Lyapunov function **candidate** for this new system. That candidate will depend upon the control  $u$ , and by choosing the control properly, we can ensure that it is decaying everywhere as well.

- Next, by *adding and subtracting*  $g_x(\mathbf{x})u_x(\mathbf{x})$  (i.e., we don't change the system in any way because we make no *net* effect) to the  $\dot{\mathbf{x}}$  part of the larger  $(\mathbf{x}, z_1)$  system, it becomes

$$\begin{cases} \dot{\mathbf{x}} = f_x(\mathbf{x}) + g_x(\mathbf{x})z_1 + \underbrace{(g_x(\mathbf{x})u_x(\mathbf{x}) - g_x(\mathbf{x})u_x(\mathbf{x}))}_0 \\ \dot{z}_1 = u_1 \end{cases}$$

which we can re-group to get

$$\begin{cases} \dot{\mathbf{x}} = \underbrace{(f_x(\mathbf{x}) + g_x(\mathbf{x})u_x(\mathbf{x}))}_{F(\mathbf{x})} + g_x(\mathbf{x}) \underbrace{(z_1 - u_x(\mathbf{x}))}_{z_1 \text{ error tracking } u_x} \\ \dot{z}_1 = u_1 \end{cases}$$

So our cascaded supersystem encapsulates the known-stable  $\dot{\mathbf{x}} = F(\mathbf{x})$  subsystem plus some error perturbation generated by the integrator.

- We now can change variables from  $(\mathbf{x}, z_1)$  to  $(\mathbf{x}, e_1)$  by letting  $e_1 \triangleq z_1 - u_x(\mathbf{x})$ . So

$$\begin{cases} \dot{\mathbf{x}} = (f_x(\mathbf{x}) + g_x(\mathbf{x})u_x(\mathbf{x})) + g_x(\mathbf{x})e_1 \\ \dot{e}_1 = u_1 - \dot{u}_x \end{cases}$$

Additionally, we let  $v_1 \triangleq u_1 - \dot{u}_x$  so that  $u_1 = v_1 + \dot{u}_x$  and

$$\begin{cases} \dot{\mathbf{x}} = (f_x(\mathbf{x}) + g_x(\mathbf{x})u_x(\mathbf{x})) + g_x(\mathbf{x})e_1 \\ \dot{e}_1 = v_1 \end{cases}$$

We seek to stabilize this **error system** by feedback through the new control  $v_1$ . By stabilizing the system at  $e_1 = 0$ , the state  $z_1$  will track the desired control  $u_x$  which will result in stabilizing the inner  $\mathbf{x}$  subsystem.

- From our existing Lyapunov function  $V_x$ , we define the *augmented* Lyapunov function candidate

$$V_1(\mathbf{x}, e_1) \triangleq V_x(\mathbf{x}) + \frac{1}{2}e_1^2$$

So

$$\dot{V}_1 = \dot{V}_x(\mathbf{x}) + \frac{1}{2}(2e_1\dot{e}_1) = \dot{V}_x(\mathbf{x}) + e_1\dot{e}_1 = \dot{V}_x(\mathbf{x}) + e_1 \underbrace{\dot{e}_1}_{v_1} = \underbrace{\frac{\partial V_x}{\partial \mathbf{x}}(\mathbf{x})}_{\text{(i.e., } \frac{dV_x}{dt})} + e_1 v_1 = \underbrace{\frac{\partial V_x}{\partial \mathbf{x}}((f_x(\mathbf{x}) + g_x(\mathbf{x})u_x(\mathbf{x})) + g_x(\mathbf{x})e_1)}_{\dot{V}_x} + e_1 v_1$$

By distributing  $\partial V_x / \partial \mathbf{x}$ , we see that

$$\dot{V}_1 = \overbrace{\frac{\partial V_x}{\partial \mathbf{x}}(f_x(\mathbf{x}) + g_x(\mathbf{x})u_x(\mathbf{x}))}^{\leq -W(\mathbf{x})} + \frac{\partial V_x}{\partial \mathbf{x}}g_x(\mathbf{x})e_1 + e_1 v_1 \leq -W(\mathbf{x}) + \frac{\partial V_x}{\partial \mathbf{x}}g_x(\mathbf{x})e_1 + e_1 v_1$$

To ensure that  $\dot{V}_1 \leq -W(\mathbf{x}) < 0$  (i.e., to ensure stability of the supersystem), we **pick** the control law

$$v_1 = -\frac{\partial V_x}{\partial \mathbf{x}}g_x(\mathbf{x}) - k_1 e_1$$

with  $k_1 > 0$ , and so

$$\dot{V}_1 = -W(\mathbf{x}) + \frac{\partial V_x}{\partial \mathbf{x}}g_x(\mathbf{x})e_1 + e_1 \overbrace{\left(-\frac{\partial V_x}{\partial \mathbf{x}}g_x(\mathbf{x}) - k_1 e_1\right)}^{v_1}$$

After distributing the  $e_1$  through,

$$\dot{V}_1 = -W(\mathbf{x}) + \overbrace{\frac{\partial V_x}{\partial \mathbf{x}}g_x(\mathbf{x})e_1 - e_1 \frac{\partial V_x}{\partial \mathbf{x}}g_x(\mathbf{x}) - k_1 e_1^2}_0 = -W(\mathbf{x}) - k_1 e_1^2 \leq -W(\mathbf{x}) < 0$$

So our *candidate* Lyapunov function  $V_1$  is a true Lyapunov function, and our system is **stable** under this control law  $v_1$  (which corresponds the control law  $u_1$  because  $v_1 \triangleq u_1 - \dot{u}_x$ ). Using the variables from the original coordinate system, the equivalent Lyapunov function

$$V_1(\mathbf{x}, z_1) \triangleq V_x(\mathbf{x}) + \frac{1}{2}(z_1 - u_x(\mathbf{x}))^2 \quad (2)$$

As discussed below, this Lyapunov function will be used again when this procedure is applied iteratively to multiple-integrator problem.

- Our choice of control  $v_1$  ultimately depends on all of our original state variables. In particular, the actual feedback-stabilizing control law

$$\underbrace{u_1(\mathbf{x}, z_1)}_{\text{By definition of } v_1} = \underbrace{v_1 + \dot{u}_x}_{\substack{v_1 \\ \text{By definition of } v_1}} = \underbrace{-\frac{\partial V_x}{\partial \mathbf{x}} g_x(\mathbf{x}) - k_1(z_1 - u_x(\mathbf{x}))}_{e_1} + \underbrace{\frac{\partial u_x}{\partial \mathbf{x}} (f_x(\mathbf{x}) + g_x(\mathbf{x})z_1)}_{\substack{\dot{u}_x \text{ (i.e., } \frac{du_x}{dt}) \\ \mathbf{x} \text{ (i.e., } \frac{d\mathbf{x}}{dt})}} \quad (3)$$

The states  $\mathbf{x}$  and  $z_1$  and functions  $f_x$  and  $g_x$  come from the system. The function  $u_x$  comes from our known-stable  $\dot{\mathbf{x}} = F(\mathbf{x})$  subsystem. The **gain** parameter  $k_1 > 0$  affects the convergence rate of our system.

Under this control law, our system is stable at the origin  $(\mathbf{x}, z_1) = (\mathbf{0}, 0)$ .

Recall that  $u_1$  in Equation (3) drives the input of an integrator that is connected to a subsystem that is feedback-stabilized by the control law  $u_x$ . Not surprisingly, the control  $u_1$  has a  $\dot{u}_x$  term that will be integrated to follow the stabilizing control law  $\dot{u}_x$  plus some offset. The other terms provide damping to remove that offset and any other perturbation effects that would be magnified by the integrator.

So because this system is feedback stabilized by  $u_1(\mathbf{x}, z_1)$  and has Lyapunov function  $V_1(\mathbf{x}, z_1)$  with  $\dot{V}_1(\mathbf{x}, z_1) \leq -W(\mathbf{x}) < 0$ , it can be used as the upper subsystem in another single-integrator cascade system.

### Motivating Example: Two-integrator Backstepping

Before discussing the recursive procedure for the general multiple-integrator case, it is instructive to study the recursion present in the two-integrator case. That is, consider the dynamical system

$$\begin{cases} \dot{\mathbf{x}} = f_x(\mathbf{x}) + g_x(\mathbf{x})z_1 \\ \dot{z}_1 = z_2 \\ \dot{z}_2 = u_2 \end{cases} \quad (4)$$

where  $\mathbf{x} \in \mathbb{R}^n$  and  $z_1$  and  $z_2$  are scalars. This system is a cascade connection of the single-integrator system in Equation (1) with another integrator (i.e., the input  $u_2$  enters through an integrator, and the output of that integrator enters the system in Equation (1) by its  $u_1$  input).

By letting

$$\begin{aligned} \bullet \quad \mathbf{y} &\triangleq \begin{bmatrix} \mathbf{x} \\ z_1 \end{bmatrix}, \\ \bullet \quad f_y(\mathbf{y}) &\triangleq \begin{bmatrix} f_x(\mathbf{x}) + g_x(\mathbf{x})z_1 \\ 0 \end{bmatrix}, \\ \bullet \quad g_y(\mathbf{y}) &\triangleq \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}, \end{aligned}$$

then the two-integrator system in Equation (4) becomes the single-integrator system

$$\begin{cases} \dot{\mathbf{y}} = f_y(\mathbf{y}) + g_y(\mathbf{y})z_2 & (\text{ where this } \mathbf{y} \text{ subsystem is stabilized by } z_2 = u_1(\mathbf{x}, z_1) ) \\ \dot{z}_2 = u_2. \end{cases} \quad (5)$$

By the single-integrator procedure, the control law  $u_y(\mathbf{y}) \triangleq u_1(\mathbf{x}, z_1)$  stabilizes the upper  $z_2$ -to- $\mathbf{y}$  subsystem using the Lyapunov function  $V_1(\mathbf{x}, z_1)$ , and so Equation (5) is a new single-integrator system that is structurally

equivalent to the single-integrator system in Equation (1). So a stabilizing control  $u_2$  can be found using the same single-integrator procedure that was used to find  $u_1$ .

### Many-integrator backstepping

In the two-integrator case, the upper single-integrator subsystem was stabilized yielding a new single-integrator system that can be similarly stabilized. This recursive procedure can be extended to handle any finite number of integrators. This claim can be formally proved with mathematical induction. Here, a stabilized multiple-integrator system is built up from subsystems of already-stabilized multiple-integrator subsystems.

- First, consider the dynamical system

$$\dot{\mathbf{x}} = f_x(\mathbf{x}) + g_x(\mathbf{x})u_x$$

that has scalar input  $u_x$  and output states  $\mathbf{x} = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$ . Assume that

- $f_x(\mathbf{x}) = \mathbf{0}$  so that the zero-input (i.e.,  $u_x = 0$ ) system is stationary at the origin  $\mathbf{x} = \mathbf{0}$ . In this case, the origin is called an *equilibrium* of the system.
- The feedback control law  $u_x(\mathbf{x})$  stabilizes the system at the equilibrium at the origin.
- A Lyapunov function corresponding to this system is described by  $V_x(\mathbf{x})$ .

That is, if output states  $\mathbf{x}$  are fed back to the input  $u_x$  by the control law  $u_x(\mathbf{x})$ , then the output states (and the Lyapunov function) return to the origin after a single perturbation (e.g., after a nonzero initial condition or a sharp disturbance). This subsystem is **stabilized** by feedback control law  $u_x$ .

- Next, connect an integrator to input  $u_x$  so that the augmented system has input  $u_1$  (to the integrator) and output states  $\mathbf{x}$ . The resulting augmented dynamical system is

$$\begin{cases} \dot{\mathbf{x}} = f_x(\mathbf{x}) + g_x(\mathbf{x})z_1 \\ \dot{z}_1 = u_1 \end{cases}$$

This "cascade" system matches the form in Equation (1), and so the single-integrator backstepping procedure leads to the stabilizing control law in Equation (3). That is, if we feed back states  $z_1$  and  $\mathbf{x}$  to input  $u_1$  according to the control law

$$u_1(\mathbf{x}, z_1) = -\frac{\partial V_x}{\partial \mathbf{x}} g_x(\mathbf{x}) - k_1(z_1 - u_x(\mathbf{x})) + \frac{\partial u_x}{\partial \mathbf{x}} (f_x(\mathbf{x}) + g_x(\mathbf{x})z_1)$$

with gain  $k_1 > 0$ , then the states  $z_1$  and  $\mathbf{x}$  will return to  $z_1 = 0$  and  $\mathbf{x} = \mathbf{0}$  after a single perturbation.

This subsystem is **stabilized** by feedback control law  $u_1$ , and the corresponding Lyapunov function from Equation (2) is

$$V_1(\mathbf{x}, z_1) = V_x(\mathbf{x}) + \frac{1}{2}(z_1 - u_x(\mathbf{x}))^2$$

That is, under feedback control law  $u_1$ , the Lyapunov function  $V_1$  decays to zero as the states return to the origin.

- Connect a new integrator to input  $u_1$  so that the augmented system has input  $u_2$  and output states  $\mathbf{x}$ . The resulting augmented dynamical system is

$$\begin{cases} \dot{\mathbf{x}} = f_x(\mathbf{x}) + g_x(\mathbf{x})z_1 \\ \dot{z}_1 = z_2 \\ \dot{z}_2 = u_2 \end{cases}$$

which is equivalent to the *single*-integrator system

$$\begin{cases} \overbrace{\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{z}_1 \end{bmatrix}}^{\triangleq \dot{\mathbf{x}}_1} = \overbrace{\begin{bmatrix} f_x(\mathbf{x}) + g_x(\mathbf{x})z_1 \\ 0 \end{bmatrix}}^{\triangleq f_1(\mathbf{x}_1)} + \overbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}^{\triangleq g_1(\mathbf{x}_1)} z_2 \quad (\text{ by Lyapunov function } V_1, \text{ subsystem stabilized by } u_1(\mathbf{x}_1) ) \\ \dot{z}_2 = u_2 \end{cases}$$

Using these definitions of  $\mathbf{x}_1$ ,  $f_1$ , and  $g_1$ , this system can also be expressed as

$$\begin{cases} \dot{\mathbf{x}}_1 = f_1(\mathbf{x}_1) + g_1(\mathbf{x}_1)z_2 \quad (\text{ by Lyapunov function } V_1, \text{ subsystem stabilized by } u_1(\mathbf{x}_1) ) \\ \dot{z}_2 = u_2 \end{cases}$$

This system matches the single-integrator structure of Equation (1), and so the single-integrator backstepping procedure can be applied again. That is, if we feed back states  $z_1$ ,  $z_2$ , and  $\mathbf{x}$  to input  $u_2$  according to the control law

$$u_2(\mathbf{x}, z_1, z_2) = -\frac{\partial V_1}{\partial \mathbf{x}_1} g_1(\mathbf{x}_1) - k_2(z_2 - u_1(\mathbf{x}_1)) + \frac{\partial u_1}{\partial \mathbf{x}_1}(f_1(\mathbf{x}_1) + g_1(\mathbf{x}_1)z_2)$$

with gain  $k_2 > 0$ , then the states  $z_1$ ,  $z_2$ , and  $\mathbf{x}$  will return to  $z_1 = 0$ ,  $z_2 = 0$ , and  $\mathbf{x} = \mathbf{0}$  after a single perturbation. This subsystem is **stabilized** by feedback control law  $u_2$ , and the corresponding Lyapunov function is

$$V_2(\mathbf{x}, z_1, z_2) = V_1(\mathbf{x}_1) + \frac{1}{2}(z_2 - u_1(\mathbf{x}_1))^2$$

That is, under feedback control law  $u_2$ , the Lyapunov function  $V_2$  decays to zero as the states return to the origin.

- Connect an integrator to input  $u_2$  so that the augmented system has input  $u_3$  and output states  $\mathbf{x}$ . The resulting augmented dynamical system is

$$\begin{cases} \dot{\mathbf{x}} = f_x(\mathbf{x}) + g_x(\mathbf{x})z_1 \\ \dot{z}_1 = z_2 \\ \dot{z}_2 = z_3 \\ \dot{z}_3 = u_3 \end{cases}$$

which can be re-grouped as the *single*-integrator system

$$\begin{cases} \overbrace{\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{z}_1 \\ \dot{z}_2 \end{bmatrix}}^{\triangleq \dot{\mathbf{x}}_2} = \overbrace{\begin{bmatrix} f_x(\mathbf{x}) + g_x(\mathbf{x})z_2 \\ z_2 \\ 0 \end{bmatrix}}^{\triangleq f_2(\mathbf{x}_2)} + \overbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}^{\triangleq g_2(\mathbf{x}_2)} z_3 \quad (\text{ by Lyapunov function } V_2, \text{ subsystem stabilized by } u_2(\mathbf{x}_2) ) \\ \dot{z}_3 = u_3 \end{cases}$$

By the definitions of  $\mathbf{x}_1$ ,  $f_1$ , and  $g_1$  from the previous step, this system is also represented by

$$\begin{cases} \overbrace{\begin{bmatrix} \dot{\mathbf{x}}_1 \\ \dot{z}_2 \end{bmatrix}}^{\triangleq \dot{\mathbf{x}}_2} = \overbrace{\begin{bmatrix} f_1(\mathbf{x}_1) + g_1(\mathbf{x}_1)z_2 \\ 0 \end{bmatrix}}^{\triangleq f_2(\mathbf{x}_2)} + \overbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}^{\triangleq g_2(\mathbf{x}_2)} z_3 \quad (\text{ by Lyapunov function } V_2, \text{ subsystem stabilized by } u_2(\mathbf{x}_2) ) \\ \dot{z}_3 = u_3 \end{cases}$$

Further, using these definitions of  $\mathbf{x}_2$ ,  $f_2$ , and  $g_2$ , this system can also be expressed as

$$\begin{cases} \dot{\mathbf{x}}_2 = f_2(\mathbf{x}_2) + g_2(\mathbf{x}_2)z_3 \quad (\text{ by Lyapunov function } V_2, \text{ subsystem stabilized by } u_2(\mathbf{x}_2) ) \\ \dot{z}_3 = u_3 \end{cases}$$

So the re-grouped system has the single-integrator structure of Equation (1), and so the single-integrator backstepping procedure can be applied again. That is, if we feed back states  $z_1$ ,  $z_2$ ,  $z_3$ , and  $\mathbf{x}$  to input  $u_3$  according to the control law

$$u_3(\mathbf{x}, z_1, z_2, z_3) = -\frac{\partial V_2}{\partial \mathbf{x}_2} g_2(\mathbf{x}_2) - k_3(z_3 - u_2(\mathbf{x}_2)) + \frac{\partial u_2}{\partial \mathbf{x}_2}(f_2(\mathbf{x}_2) + g_2(\mathbf{x}_2)z_3)$$

with gain  $k_3 > 0$ , then the states  $z_1, z_2, z_3$ , and  $\mathbf{x}$  will return to  $z_1 = 0, z_2 = 0, z_3 = 0$ , and  $\mathbf{x} = \mathbf{0}$  after a single perturbation. This subsystem is **stabilized** by feedback control law  $u_3$ , and the corresponding Lyapunov function is

$$V_3(\mathbf{x}, z_1, z_2, z_3) = V_2(\mathbf{x}_2) + \frac{1}{2}(z_3 - u_2(\mathbf{x}_2))^2$$

That is, under feedback control law  $u_3$ , the Lyapunov function  $V_3$  decays to zero as the states return to the origin.

- This process can continue for each integrator added to the system, and hence any system of the form

$$\begin{cases} \dot{\mathbf{x}} = f_x(\mathbf{x}) + g_x(\mathbf{x})z_1 & (\text{ by Lyapunov function } V_x, \text{ subsystem stabilized by } u_x(\mathbf{x}) ) \\ \dot{z}_1 = z_2 \\ \dot{z}_2 = z_3 \\ \vdots \\ \dot{z}_i = z_{i+1} \\ \vdots \\ \dot{z}_{k-2} = z_{k-1} \\ \dot{z}_{k-1} = z_k \\ \dot{z}_k = u \end{cases}$$

has the recursive structure

$$\left\{ \left\{ \left\{ \left\{ \left\{ \begin{cases} \dot{\mathbf{x}} = f_x(\mathbf{x}) + g_x(\mathbf{x})z_1 & (\text{ by Lyapunov function } V_x, \text{ subsystem stabilized by } u_x(\mathbf{x}) ) \\ \dot{z}_1 = z_2 \\ \dot{z}_2 = z_3 \\ \vdots \\ \dot{z}_i = z_{i+1} \\ \vdots \\ \dot{z}_{k-2} = z_{k-1} \\ \dot{z}_{k-1} = z_k \\ \dot{z}_k = u \end{cases} \right. \right. \right. \right. \right.$$

and can be feedback stabilized by finding the feedback-stabilizing control and Lyapunov function for the single-integrator  $(\mathbf{x}, z_1)$  subsystem (i.e., with input  $z_2$  and output  $\mathbf{x}$ ) and iterating out from that inner subsystem until the ultimate feedback-stabilizing control  $u$  is known. At iteration  $i$ , the equivalent system is

$$\left\{ \begin{array}{l} \triangle \mathbf{x}_{i-1} \\ \dot{\mathbf{x}} \\ \dot{z}_1 \\ \dot{z}_2 \\ \vdots \\ \dot{z}_{i-2} \\ \dot{z}_{i-1} \\ \dot{z}_i = u_i \end{array} \right\} = \underbrace{\begin{bmatrix} \triangle f_{i-1}(\mathbf{x}_{i-1}) \\ f_{i-2}(\mathbf{x}_{i-2}) + g_{i-2}(\mathbf{x}_{i-1})z_{i-2} \\ 0 \end{bmatrix}}_{\triangle f_{i-1}(\mathbf{x}_{i-1})} + \underbrace{\begin{bmatrix} \triangle g_{i-1}(\mathbf{x}_{i-1}) \\ 0 \\ 1 \end{bmatrix}}_{\triangle g_{i-1}(\mathbf{x}_{i-1})} z_i \quad (\text{ by Lyap. func. } V_{i-1}, \text{ subsystem stabilized by } u_{i-1}(\mathbf{x}_{i-1}) )$$

The corresponding feedback-stabilizing control law is

$$u_i(\underbrace{\mathbf{x}, z_1, z_2, \dots, z_i}_{\triangle \mathbf{x}_i}) = -\frac{\partial V_{i-1}}{\partial \mathbf{x}_{i-1}} g_{i-1}(\mathbf{x}_{i-1}) - k_i(z_i - u_{i-1}(\mathbf{x}_{i-1})) + \frac{\partial u_{i-1}}{\partial \mathbf{x}_{i-1}} (f_{i-1}(\mathbf{x}_{i-1}) + g_{i-1}(\mathbf{x}_{i-1})z_i)$$

with gain  $k_i > 0$ . The corresponding Lyapunov function is

$$V_i(\mathbf{x}_i) = V_{i-1}(\mathbf{x}_{i-1}) + \frac{1}{2}(z_i - u_{i-1}(\mathbf{x}_{i-1}))^2$$



By this construction, the ultimate control  $u(\mathbf{x}, z_1, z_2, \dots, z_k) = u_k(\mathbf{x}_k)$  (i.e., ultimate control is found at final iteration  $i = k$ ).

Hence, any system in this special many-integrator strict-feedback form can be feedback stabilized using a straightforward procedure that can even be automated (e.g., as part of an adaptive control algorithm).

## Generic Backstepping

Systems in the special strict-feedback form have a recursive structure similar to the many-integrator system structure. Likewise, they are stabilized by stabilizing the smallest cascaded system and then *backstepping* to the next cascaded system and repeating the procedure. So it is critical to develop a single-step procedure; that procedure can be recursively applied to cover the many-step case. Fortunately, due to the requirements on the functions in the strict-feedback form, each single-step system can be rendered by feedback to a single-integrator system, and that single-integrator system can be stabilized using methods discussed above.

## Single-step Procedure

Consider the simple strict-feedback system

$$\begin{cases} \dot{\mathbf{x}} = f_x(\mathbf{x}) + g_x(\mathbf{x})z_1 \\ \dot{z}_1 = f_1(\mathbf{x}, z_1) + g_1(\mathbf{x}, z_1)u_1 \end{cases} \quad (6)$$

where

- $\mathbf{x} = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$ ,
- $z_1$  and  $u_1$  are scalars,
- For all  $\mathbf{x}$  and  $z_1$ ,  $g_1(\mathbf{x}, z_1) \neq 0$ .

Rather than designing feedback-stabilizing control  $u_1$  directly, introduce a new control  $u_{a1}$  (to be designed *later*) and use control law

$$u_1(\mathbf{x}, z_1) = \frac{1}{g_1(\mathbf{x}, z_1)} (u_{a1} - f_1(\mathbf{x}, z_1))$$

which is possible because  $g_1 \neq 0$ . So the system in Equation (6) is

$$\begin{cases} \dot{\mathbf{x}} = f_x(\mathbf{x}) + g_x(\mathbf{x})z_1 \\ \dot{z}_1 = f_1(\mathbf{x}, z_1) + g_1(\mathbf{x}, z_1) \overbrace{\frac{1}{g_1(\mathbf{x}, z_1)} (u_{a1} - f_1(\mathbf{x}, z_1))}^{u_1(\mathbf{x}, z_1)} \end{cases}$$

which simplifies to

$$\begin{cases} \dot{\mathbf{x}} = f_x(\mathbf{x}) + g_x(\mathbf{x})z_1 \\ \dot{z}_1 = u_{a1} \end{cases}$$

This new  $u_{a1}$ -to- $\mathbf{x}$  system matches the *single-integrator cascade system* in Equation (1). Assuming that a feedback-stabilizing control law  $u_x(\mathbf{x})$  and Lyapunov function  $V_x(\mathbf{x})$  for the upper subsystem is known, the feedback-stabilizing control law from Equation (3) is

$$u_{a1}(\mathbf{x}, z_1) = -\frac{\partial V_x}{\partial \mathbf{x}} g_x(\mathbf{x}) - k_1(z_1 - u_x(\mathbf{x})) + \frac{\partial u_x}{\partial \mathbf{x}} (f_x(\mathbf{x}) + g_x(\mathbf{x})z_1)$$

with gain  $k_1 > 0$ . So the final feedback-stabilizing control law is

$$u_1(\mathbf{x}, z_1) = \frac{1}{g_1(\mathbf{x}, z_1)} \left( \overbrace{-\frac{\partial V_x}{\partial \mathbf{x}} g_x(\mathbf{x}) - k_1(z_1 - u_x(\mathbf{x})) + \frac{\partial u_x}{\partial \mathbf{x}} (f_x(\mathbf{x}) + g_x(\mathbf{x})z_1) - f_1(\mathbf{x}, z_1)}^{u_{a1}(\mathbf{x}, z_1)} \right) \quad (7)$$

with gain  $k_1 > 0$ . The corresponding Lyapunov function from Equation (2) is

$$V_1(\mathbf{x}, z_1) = V_x(\mathbf{x}) + \frac{1}{2}(z_1 - u_x(\mathbf{x}))^2 \quad (8)$$

Because this *strict-feedback system* has a feedback-stabilizing control and a corresponding Lyapunov function, it can be cascaded as part of a larger strict-feedback system, and this procedure can be repeated to find the surrounding feedback-stabilizing control.

## Many-step Procedure

As in many-integrator backstepping, the single-step procedure can be completed iteratively to stabilize an entire strict-feedback system. In each step,

1. The smallest "unstabilized" single-step strict-feedback system is isolated.
2. Feedback is used to convert the system into a single-integrator system.
3. The resulting single-integrator system is stabilized.
4. The stabilized system is used as the upper system in the next step.

That is, any *strict-feedback system*

$$\begin{cases} \dot{\mathbf{x}} = f_x(\mathbf{x}) + g_x(\mathbf{x})z_1 \\ \dot{z}_1 = f_1(\mathbf{x}, z_1) + g_1(\mathbf{x}, z_1)z_2 \\ \dot{z}_2 = f_2(\mathbf{x}, z_1, z_2) + g_2(\mathbf{x}, z_1, z_2)z_3 \\ \vdots \\ \dot{z}_i = f_i(\mathbf{x}, z_1, z_2, \dots, z_i) + g_i(\mathbf{x}, z_1, z_2, \dots, z_i)z_{i+1} \\ \vdots \\ \dot{z}_{k-2} = f_{k-2}(\mathbf{x}, z_1, z_2, \dots, z_{k-2}) + g_{k-2}(\mathbf{x}, z_1, z_2, \dots, z_{k-2})z_{k-1} \\ \dot{z}_{k-1} = f_{k-1}(\mathbf{x}, z_1, z_2, \dots, z_{k-2}, z_{k-1}) + g_{k-1}(\mathbf{x}, z_1, z_2, \dots, z_{k-2}, z_{k-1})z_k \\ \dot{z}_k = f_k(\mathbf{x}, z_1, z_2, \dots, z_{k-1}, z_k) + g_k(\mathbf{x}, z_1, z_2, \dots, z_{k-1}, z_k)u \end{cases} \quad (\text{by Lyapunov function } V_x, \text{ subsystem stabilized by } u_x(\mathbf{x}))$$

has the recursive structure

$$\left\{ \left\{ \left\{ \left\{ \left\{ \left\{ \begin{cases} \dot{\mathbf{x}} = f_x(\mathbf{x}) + g_x(\mathbf{x})z_1 \\ \dot{z}_1 = f_1(\mathbf{x}, z_1) + g_1(\mathbf{x}, z_1)z_2 \\ \dot{z}_2 = f_2(\mathbf{x}, z_1, z_2) + g_2(\mathbf{x}, z_1, z_2)z_3 \\ \vdots \\ \dot{z}_i = f_i(\mathbf{x}, z_1, z_2, \dots, z_i) + g_i(\mathbf{x}, z_1, z_2, \dots, z_i)z_{i+1} \\ \vdots \\ \dot{z}_{k-2} = f_{k-2}(\mathbf{x}, z_1, z_2, \dots, z_{k-2}) + g_{k-2}(\mathbf{x}, z_1, z_2, \dots, z_{k-2})z_{k-1} \\ \dot{z}_{k-1} = f_{k-1}(\mathbf{x}, z_1, z_2, \dots, z_{k-2}, z_{k-1}) + g_{k-1}(\mathbf{x}, z_1, z_2, \dots, z_{k-2}, z_{k-1})z_k \\ \dot{z}_k = f_k(\mathbf{x}, z_1, z_2, \dots, z_{k-1}, z_k) + g_k(\mathbf{x}, z_1, z_2, \dots, z_{k-1}, z_k)u \end{cases} \right. \right. \right. \right. \right. \right. \quad (\text{by Lyapunov function } V_x, \text{ subsystem stabilized by } u_x(\mathbf{x}))$$

and can be feedback stabilized by finding the feedback-stabilizing control and Lyapunov function for the single-integrator  $(\mathbf{x}, z_1)$  subsystem (i.e., with input  $z_2$  and output  $\mathbf{x}$ ) and iterating out from that inner subsystem until the ultimate feedback-stabilizing control  $u$  is known. At iteration  $i$ , the equivalent system is

$$\begin{cases} \dot{\mathbf{x}} \\ \dot{z}_1 \\ \dot{z}_2 \\ \vdots \\ \dot{z}_{i-2} \\ \dot{z}_{i-1} \end{cases} \stackrel{\triangle}{=} \begin{cases} \mathbf{x}_{i-1} \\ \dot{\mathbf{x}}_{i-1} \end{cases} = \begin{matrix} \stackrel{\triangle}{=} f_{i-1}(\mathbf{x}_{i-1}) \\ \left[ \begin{array}{c} f_{i-2}(\mathbf{x}_{i-2}) + g_{i-2}(\mathbf{x}_{i-2})z_{i-2} \\ f_{i-1}(\mathbf{x}_i) \end{array} \right] \end{matrix} + \begin{matrix} \stackrel{\triangle}{=} g_{i-1}(\mathbf{x}_{i-1}) \\ \left[ \begin{array}{c} \mathbf{0} \\ g_{i-1}(\mathbf{x}_i) \end{array} \right] \end{matrix} z_i \quad (\text{by Lyap. func. } V_{i-1}, \text{ subsystem stabilized by } u_{i-1}(\mathbf{x}_{i-1}))$$

$$\dot{z}_i = f_i(\mathbf{x}_i) + g_i(\mathbf{x}_i)u_i$$

By Equation (7), the corresponding feedback-stabilizing control law is

$$u_i(\underbrace{\mathbf{x}, z_1, z_2, \dots, z_i}_{\triangleq \mathbf{x}_i}) = \frac{1}{g_i(\mathbf{x}_i)} \left( \overbrace{-\frac{\partial V_{i-1}}{\partial \mathbf{x}_{i-1}} g_{i-1}(\mathbf{x}_{i-1}) - k_i(z_i - u_{i-1}(\mathbf{x}_{i-1})) + \frac{\partial u_{i-1}}{\partial \mathbf{x}_{i-1}} (f_{i-1}(\mathbf{x}_{i-1}) + g_{i-1}(\mathbf{x}_{i-1})z_i) - f_i(\mathbf{x}_{i-1})}^{\text{Single-integrator stabilizing control } u_{a_i}(\mathbf{x}_i)} \right)$$

with gain  $k_i > 0$ . By Equation (8), the corresponding Lyapunov function is

$$V_i(\mathbf{x}_i) = V_{i-1}(\mathbf{x}_{i-1}) + \frac{1}{2}(z_i - u_{i-1}(\mathbf{x}_{i-1}))^2$$

By this construction, the ultimate control  $u(\mathbf{x}, z_1, z_2, \dots, z_k) = u_k(\mathbf{x}_k)$  (i.e., ultimate control is found at final iteration  $i = k$ ). Hence, any strict-feedback system can be feedback stabilized using a straightforward procedure that can even be automated (e.g., as part of an adaptive control algorithm).

## References

# Article Sources and Contributors

**Backstepping** *Source:* <http://en.wikipedia.org/w/index.php?oldid=484165373> *Contributors:* Chris the speller, David Eppstein, Encyclops, Giftlite, J Milburn, J04n, Jiuguang Wang, Jrriehl, Mild Bill Hiccup, Ohconfucius, Oleg Alexandrov, PKT, TedPavlic, Ukexpat, 7 anonymous edits

## License

---

Creative Commons Attribution-Share Alike 3.0 Unported  
[//creativecommons.org/licenses/by-sa/3.0/](http://creativecommons.org/licenses/by-sa/3.0/)

---