Article:
Jin, S., Nie, Y. and Jones, M. (2026) Feedback linearisation with state constraints. International Journal of Robust and Nonlinear Control. ISSN: 1049-8923

https://doi.org/10.1002/rnc.70670

International Journal of Robust and Nonlinear Control

WILEY

RESEARCH ARTICLE OPEN ACCESS

Feedback Linearisation with State Constraints

Songlin Jin | Yuanbo Nie | Morgan Jones

School of Electrical and Electronic Engineering, University of Sheffield, Sheffield, UK

Correspondence: Songlin Jin (sjin16@sheffield.ac.uk)

Received: 13 September 2025 | **Revised:** 30 June 2026 | **Accepted:** 2 July 2026

Keywords: asymptotically tracking control | feedback linearisation | nonlinear systems | state constraints

ABSTRACT

Feedback Linearisation (FBL) is a widely used technique that applies feedback laws to transform input-affine nonlinear control systems into linear control systems, allowing for the use of linear controller design methods such as pole placement. However, for problems with state constraints, controlling the linear system induced by FBL can be more challenging than controlling the original system. This is because simple state constraints in the original nonlinear system become complex nonlinear constraints in the FBL induced linearised system, thereby diminishing the advantages of linearisation. To avoid increasing the complexity of state constraints under FBL, this paper introduces a method to first augment system dynamics to capture state constraints before applying FBL. We show that our proposed augmentation method leads to ill-defined relative degrees at state constraint boundaries. However, we show that ill-defined relative degrees can be overcome by using a switching FBL controller. Numerical experiments illustrate the capabilities of this method for handling state constraints within the FBL framework.

1 | Introduction

Nonlinear systems pervade every facet of our world, from turbulent flows [1] to the sophisticated operations of human-made cybernetic infrastructure [2]. Unlike linear systems, nonlinear systems do not obey the solution superposition property leading to phenomena such as non-global stability, limit cycles and sensitivity to small initial changes (chaos). These properties make the control of nonlinear systems significantly more challenging than that of linear systems. Nevertheless, there exist several techniques that approximate or transform challenging nonlinear problems into simpler linear problems, such as Jacobian/Taylor linearisation, Koopman operators [3], Carleman linearisation [4] and Feedback Linearisation. FBL is a fairly mature topic based on the early works [5, 6] from the seventies and has been extensively treated in textbooks [7].

FBL, also known as Nonlinear Dynamic Inversion in special cases [8], is widely used in aerospace applications such as flight control [9]. FBL works by designing a controller to cancel the nonlinearities within the system dynamics, enabling the application of well-established linear control techniques to the remaining "virtual input." For a system with a single affine input u and single state x, $\dot{x}(t) = f(t, x(t)) + g(t, x(t))u$, assuming $g(t, x) \neq 0$, FBL is as simple as applying the controller $u = \frac{\nu - f(t,x)}{g(t,x)}$, where ν is the virtual input designed to control the induced linear system $\dot{x}(t) = \nu$. For Multiple Input Multiple Output (MIMO) systems, FBL involves the use of geometric techniques to find the relationship between the output and the input by taking the time derivatives of the output. Derivatives are taken until the system input appears explicitly within the expression, this order of derivative is known as the **relative degree** of the system. Input-output linearisation (IOL) is one of the most common formulations of FBL [10]. In IOL, the output function is fixed, and the linearisation is carried out on this predetermined output, and hence the system may only be partially linearised (that is, there may not exist a diffeomorphism [11] mapping between the state space of the linearised system and the original state variables). In classical FBL, there is more freedom in the choice of output function, and full linearisation of the system, via a

diffeomorphism, can be achieved by selecting an appropriate output. The method presented here is general enough to encompass both settings. Although the problem formulation (Section 2) includes an output function for concreteness, this function is not predetermined and may be freely chosen. We therefore adopt the term "FBL" throughout the paper.

For nonlinear control, FBL provides an optimisation-free analytical controller whose stability properties can be analysed [12]. In contrast, other nonlinear optimal control methods usually require significant computational resources involving solving a nonlinear programming problem [13] or a nonlinear PDE such as the Euler-Lagrange equation [14] or HJB PDE [15]. However, arguably the popularity of FBL when compared to alternative nonlinear control techniques has waned in recent times due to the following challenges: (1) FBL requires exact knowledge of the model of the system and is sensitive to uncertainties [16]. (2) The feedback linearisation transformation maps simple inputs and state constraints to complex nonlinear constraints, limiting the direct use of linear control methods to the FBL induced system [17].

Significant effort has been made to overcome the first challenge of improving FBL robustness to model uncertainty, adaptive FBL controllers have been developed, and the stability impact of model uncertainties has been studied through a robust control theory framework [18–20]. This work focuses on the second challenge of using FBL under state constraints.

To overcome the second challenge, several methods have been developed to handle input constraints, such as computing inner polytope approximations [21], integral invariant controllers [22] adding hedging to the reference signal, known as the pseudo control hedging [23], and transforming nonlinear control allocation problems into linear ones to handle actuator saturation [24]. However, very few results have been published regarding the use of FBL in the presence of state constraints. Existing results [25–29] use optimisation based control, such as Model Predictive Control (MPC) or Linear Matrix Inequalities (LMIs), with inner approximations of the transformed state constraint and/or time discretisation, leading to significant computational burdens that limit feasibility for online applications. Alternatively, under certain assumptions, a PID controller can be derived such that the system is guaranteed to satisfy box state constraints [30]. In contrast to these methods, our proposed method does not require the solution of an optimisation problem during online implementation and is not limited to simple box constraints, it can be implemented for general time-varying state constraints.

The primary contribution of this paper is to propose a general FBL method capable of handling state constraints without the need for any online optimisation.

The following three aspects summarise the specific contributions of the paper:

- **Scalable Constraint Handling:** We extend the work of [31] to cases where the constraint dimension is greater than the input dimension by introducing the novel method of sequentially capturing the constraints into system dynamics, and then an unconstrained augmented system is derived. For the first time, an analytic expression for the transformation in [31] is derived and shown in Lemma 1.
- **Fundamental Controller Design Principles:** Our main result, Theorem 1, establishes the fundamental principles for designing implementable controllers that obey state constraints. This theorem reveals that the properties of *integrability* and *invertibility* of virtual controllers are key when certifying constraint satisfaction under our state augmentation approach.
  Firstly, we develop an integral controller framework in Section 3.1 to ensure a bounded controller. Next, we resolve an inherent limitation of the transformation technique in [31], more specifically, in Proposition 1, we theoretically prove that the transformation may lead to ill-defined relative degrees at the boundary of the constraints. Therefore, in Sections 5.1 and 5.2, a controller switching strategy [32] [33] is introduced to overcome the invertibility challenge caused by ill-defined relative degrees.
- **Establishing a Practical Implementation Workflow:** Moving beyond existing results in the literature [32] [33], we provide a definitive controller implementation method in Section 5.3 that bridges the gap between theory and practice, in which back-substitution is employed to extinguish the dependency on augmented states, thereby enabling a switchable FBL controller to be implemented on the original system.

Furthermore, our method is summarised in the flowchart given in Figure 1, follows three steps: (1) Augment the dynamics to capture state constraints and modify with integral control structure, (2) Apply FBL to the integral augmented system to derive an FBL controller, (3) Implement the FBL controller where augmented state dependencies are removed via algebraic substitution.

Extending the state-space by augmenting the system dynamics has previously been used to solve discrete-time problems with non-additively separable cost functions [34, 35]. Our work focuses on tracking problems in continuous time rather than minimising general cost functions, using state augmentation to encapsulate state constraints. We follow and extend the approach of [31] by introducing a slack variable to represent the square root of the proximity to the state constraint boundary. Analogous to the first step of FBL, we differentiate this equation to establish a relationship between the slack variable and the system input. Solving for the input, we obtain a new higher-dimensional system where the virtual input corresponds to the highest derivative of the slack variable. We show that the augmented system will not violate state constraints for any integrable input. We overcome the limitation of [31], that the number of input dimensions must match the number of constraints required, by proposing a sequential approach where we iteratively augment the state space to capture constraints. We also differ from [31] by imposing an integral controller structure to ensure any input used is integrable and hence cannot violate the state constraints. Moreover, unlike in [31], which uses HJB equations to solve for the controller, we synthesise a closed-loop controller using FBL and analyse the associated challenges that arise from this approach.

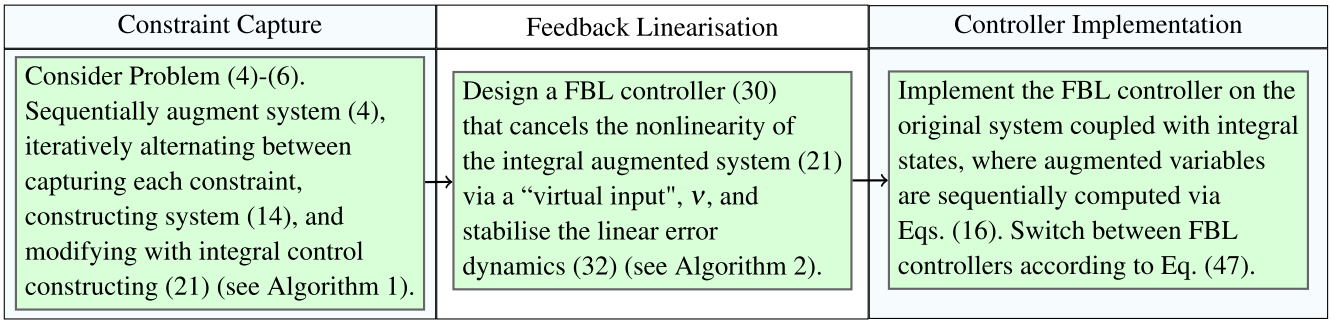


FIGURE 1 | Flowchart for closed-form controller synthesis and implementation: Augment the system to incorporate constraints while introducing an integral controller structure (Section 3). Design an FBL controller to cancel system nonlinearities and stabilise the linearised tracking error dynamics (Section 4). Finally, implement the FBL controller that depends only on the original states and the integral states, not the augmented state (Section 5).

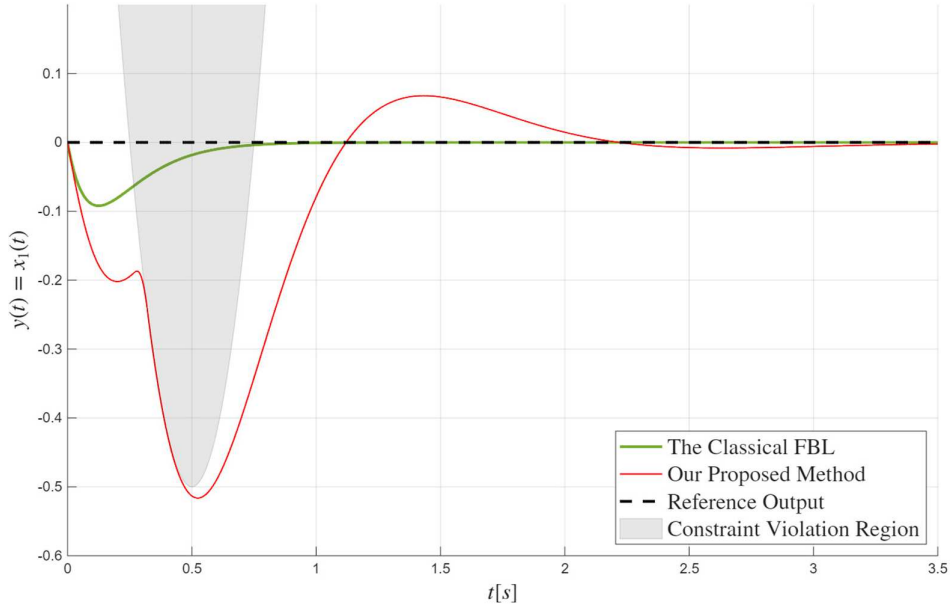


FIGURE 2 | Figure associated with Example 1 showing the trajectory deduced by our proposed method against that of the classical FBL method.

Unfortunately, it is non-trivial to apply FBL out of the box to the resulting augmented system, since we show that the augmented system has an ill-defined relative degree, changing its value along the state constraint boundary. Therefore, in order to apply FBL, we use the switching controller from [32] [33] that was designed to handle cases of ill-defined relative degree. Finally, we synthesise a state constraint feasible closed-loop controller for the original system coupled with integral states by removing the augmented states such that the controller only depends on the states of the original system and the induced integral states. In Figure 2, we implement the proposed method on a Single-Input Single-Output (SISO) system studied in [31], and stabilise the state trajectory to the origin while guaranteeing that the state constraint is satisfied, in contrast to the classical FBL method, where the constraints are violated. We show more implementation details in Example 1 and manually derive the controller following our proposed method.

While our underlying methodology is applicable to MIMO systems, the implementation involving a switching controller is restricted to SISO systems. Accordingly, Sections 2 to 4, which extend the state augmentation approach of [31] and analyse the complications that arise when combining this method with FBL, are presented in the general MIMO setting. From Section 5 onwards, however, we focus exclusively on SISO systems, where the practical implementation of the FBL controller, particularly the switching logic, is more tractable.

Notation: For $v \in \mathbb{R}^n$ we denote each component of the vector as v_i or $[v]_i$. If $v \in \mathbb{R}^n$ we say $v \leq 0$ if $v_i \leq 0$ for each $i \in \{1, \ldots, n\}$, $v \neq 0$ if $v_i \neq 0$ for each $i \in \{1, \ldots, n\}$ and $v = 0$ if $v_i = 0$ for each $i \in \{1, \ldots, n\}$. Let $||\cdot||_2$ denotes Euclidean norm or L^2 norm operator. We define I_m as a m by m identity matrix and $0_{p\times m}$ as a p by m matrix with all elements being zero. For $v \in \mathbb{R}^n$ we denote $D = \text{diag}(v^\top)$ to be the diagonal such that $D_{i,i} = v_i$ and $D_{i,j} = 0$ for all $i \neq j$. Similarly, we overload the notation $D = \text{diag}(A_1, \ldots, A_n)$ to denote the block diagonal matrix such that matrices $A_1, \ldots, A_n$ lying along the main diagonal and other elements of the matrix D are equal to zero. For $i \in \{1, \ldots, p\}$ the vector zeros with only the i-th component of value 1 is defined by

$$\mathcal{I}_{p,i} := [0_{1\times(i-1)}, 1, 0_{1\times(p-i)}] \in \mathbb{R}^{1\times p}. \quad (1)$$

We denote the set of k-differentiable functions, $f : X \to Y$ by $C^k(X, Y)$, with $C^\infty(X, Y)$ denoting the set of infinitely differentiable functions. Along the same lines, we denote $L^1(X, Y)$ as the set of Lebesgue integrable functions. Given $F \in C^1([0, \infty) \times \mathbb{R}^n, \mathbb{R}^r)$ and $x \in C^1([0, \infty), \mathbb{R}^n)$ we define $\frac{dF(t,x(t))}{dt} = \frac{\partial F(t,x(t))}{\partial x}\frac{\partial x(t)}{\partial t} + \frac{\partial F(t,x(t))}{\partial t}$, where $\frac{\partial F(t,x)}{\partial x} = \left[\frac{\partial F(t,x)}{\partial x_1}, \ldots, \frac{\partial F(t,x)}{\partial x_n}\right] \in \mathbb{R}^{1\times n}$ is a row vector. We define the 0-th derivative as $\frac{d^0}{dt^0}F(t, x) = F(t, x)$ and $\frac{\partial^0}{\partial t^0}F(t, x) = F(t, x)$. Consider $\psi \in C^i([0, \infty) \times \mathbb{R}^n, \mathbb{R}^m)$, $\psi^{(i)}(t, x(t))$ denotes the ith order time derivative of $\psi(t, x(t))$. Given $f \in C^1([0, \infty) \times \mathbb{R}^n, \mathbb{R}^n)$, $g \in C^1([0, \infty) \times \mathbb{R}^n, \mathbb{R}^{n\times m})$ and $h \in C^1([0, \infty) \times \mathbb{R}^n, \mathbb{R}^r)$, for $i \in \{1, \ldots, r\}$, we denote the Lie derivatives of the ith component in h by $L_f^k h_i \in \mathbb{R}$ and $L_{g_j} L_f^k h_i \in \mathbb{R}$ that are given in Equations (2),

$$
\begin{aligned}
&L_f^0 h_i(t, x) = h_i(t, x),\\
&L_f^k h_i(t, x) = \frac{\partial L_f^{k-1} h_i(t, x)}{\partial x} f(t, x) + \frac{\partial L_f^{k-1} h_i(t, x)}{\partial t} \text{ for } k \in \mathbb{N},\\
&L_{g_j} L_f^j h_i(t, x) = \frac{\partial L_f^k h_i(t, x)}{\partial x} g_j(t, x) \text{ for } k \in \mathbb{N} \cup \{0\},
\end{aligned} \tag{2}
$$

where g_j denotes the j-th column in g for $j \in \{1, \ldots, m\}$. Then, we define a collection vector $L_g L_f^k h_i(t, x) := [L_{g_1} L_f^k h_i(t, x), \ldots, L_{g_m} L_f^k h_i(t, x)] \in \mathbb{R}^{1\times m}$. This definition of Lie derivative is useful when dealing with time-varying systems, see [36] for more details.

2 | Problem Formulation: Asymptotic Constrained Tracking

Consider a square input-affine system, that is, the number of inputs equals the number of outputs, denoted by the tuple

$$\Sigma = (f, g, h, x_0), \tag{3}$$

where $f \in C^\infty([t_0, \infty) \times \mathbb{R}^n, \mathbb{R}^n)$, $g \in C^\infty([t_0, \infty) \times \mathbb{R}^n, \mathbb{R}^{n\times m})$, $h \in C^\infty([t_0, \infty) \times \mathbb{R}^n, \mathbb{R}^m)$, $t_0 \geq 0$ and $x_0 \in \mathbb{R}^n$. The state, input, and output at time $t \geq 0$ of Σ are denoted by $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, and $y(t) \in \mathbb{R}^m$ respectively, and the dynamics are governed by the following Initial Value Problem (IVP):

$$
\begin{aligned}
&\dot{x}(t) = f(t, x(t)) + g(t, x(t))u(t), \quad x_0 = x(t_0),\\
&y(t) = h(t, x(t)).
\end{aligned} \tag{4}
$$

ODE (4) is commonly expanded as $\dot{x}(t) = f(t, x(t)) + \sum_{i=1}^m g_i(t, x(t))u_i(t)$ [37] to demonstrate the correspondence between the input matrix and control vector fields, where $g_i \in \mathbb{R}^n$ denotes the i-th column in g.

For simplicity, throughout this paper we will assume that there exists a unique continuous solution, $x(t)$, to the ODE (4) for all $t \geq 0$ and $u \in L^1([t_0, \infty), \mathbb{R}^m)$. This assumption is non-restrictive, for instance, when the vector field of the ODE is Lipschitz continuous, the solution map exists over a finite time interval. Moreover, this interval can be extended arbitrarily if the solution remains within a compact set, as elaborated in [7].

Problem (Asymptotic Tracking with state constraints): Given a feasible initial condition $(t_0, x_0) \in [0, \infty) \times \mathbb{R}^n$, satisfying $\phi(t_0, x_0) \leq 0$, where $\phi \in C^\infty([t_0, \infty) \times \mathbb{R}^n, \mathbb{R}^r)$, and a smooth reference signal $y_r \in C^\infty([t_0, \infty), \mathbb{R}^m)$ find a feedback controller that ensures the output of ODE (4) tracks the reference asymptotically, while the closed loop trajectory, $x(t)$, satisfies the state constraints:

$$\lim_{t\to\infty} ||y(t) - y_r(t)||_2 = 0, \tag{5}$$

$$\phi(t, x(t)) \leq 0 \text{ for all } t \in [0, \infty]. \tag{6}$$

This problem is formulated to accommodate controller synthesis methods that introduce auxiliary integral states during design, such as the method proposed in this paper. These states, which are not present a priori, are constructed alongside the controller to enforce properties such as input boundedness.

The above tracking control problem is also a generalisation of the classical tracking problem [38], finding a controller that drives the output of the system to asymptotically track a reference signal while keeping the internal states, $x(t)$, of the closed-loop system bounded. Clearly, when $\phi(t, x) = ||x||_2^2 - R^2$, for sufficiently large $R > 0$, the above problem becomes the classical tracking problem. Classically, the bounded state constraint is not directly addressed, rather, a tracking controller is derived, and then, under various assumptions, such as stable zero dynamics [39], it can be proven that the closed-loop system has bounded internal states. In this paper, we propose a fundamentally different approach where we design a controller based on the knowledge of the system constraints, captured by ϕ. In the next section, we detail how to augment the state space to include information about the state constraints.

3 | State Augmentation: From Constrained to Unconstrained

In this section, we recall and extend the approach of [31] that introduced a technique of transforming a constrained tracking control problem into an unconstrained tracking control problem by augmenting/expanding the state dynamics to capture the constraints. More specifically, a slack variable is introduced that captures how close the system's constraint is to being violated. Then, the relationship between the slack variable and system input is derived by taking sufficiently many time derivatives. The number of time derivatives required to take is exactly equal to the number of time derivatives that can be taken of the constraint function before the system's input appears explicitly in the expression. This is called the relative degree of the function and is defined next.

Definition 1 (Relative Degree). We say $\psi \in C^\infty([t_0, \infty) \times \mathbb{R}^n, \mathbb{R}^r)$ has relative degree equal to $\rho = [\rho_1, \ldots, \rho_r] \in \mathbb{N}^r$, with respect to the system given by the tuple $\Sigma = (f, g, h, x_0)$ (3), if each of the $k \in \{1, \ldots, r\}$ components of ρ satisfies the following two conditions:

1. For all $(t, x) \in [t_0, \infty) \times \mathbb{R}^n$ we have $L_g L_f^i \psi_k(t, x) = 0$ for $i \in \{0, \ldots, \rho_k - 2\}$,
2. For all $(t, x) \in [t_0, \infty) \times \mathbb{R}^n$ we have $L_g L_f^{\rho_k - 1} \psi_k(t, x) \neq 0$.

Recalling that Lie derivatives were defined in Equation (2).

Let us now consider a system $\Sigma = (f, g, h, x_0)$ (3) with state constraint given in Equation (6). For simplicity, we initially assume that the dimension of the constraint function, ϕ, is equal to the dimension of the input, u, that is

$$r = m. \tag{7}$$

See Section 3.2 for the generalisation to $r = \alpha m$, where $\alpha \in \mathbb{N}$. To augment the dynamics and capture the state constraint, we follow [31] by introducing a time-varying slack variable given by the following equation,

$$\phi(t, x(t)) + Z(t) = 0 \tag{8}$$

where $x(t)$ is the solution to IVP (4) and $Z := \left[\frac{1}{2}z_1^2, \ldots, \frac{1}{2}z_r^2\right]^\top \in \mathbb{R}^r$. Note, if $Z(t) = 0$ for some $t > 0$, then $\phi(t, x(t)) = 0$, implying the state trajectory, $x(t)$, is on the boundary of the state constraint.

The slack variable, Z, depends explicitly on the state of the system through Equation (8) and thus implicitly depends on the input of the system. By computing the explicit expression that relates the system input to the slack variable, we may substitute the input variable for the slack variable, treating the derivative of the slack variable as a pseudo-input variable of the transformed system. Then, the transformed system will remain state-feasible for any value taken by the pseudo-input, since by the quadratic nature of the slack variable it follows from Equation (8) that $\phi(t, x(t)) \geq 0$ for any $t \in [t_0, \infty)$. The slack variables z_i for all $i \in \{1, \ldots, r\}$ are augmented to form new state-space coordinates for the transformation system. In the work of [31], the time derivatives of Equation (8) were not analytically derived. Next, we extend this work by deriving an analytical expression for the time derivatives of Equation (8) to obtain an expression that relates the slack variable to the system input.

Lemma 1. *Consider the IVP (4) and a smooth function $\phi \in C^\infty([t_0, \infty) \times \mathbb{R}^n, \mathbb{R}^r)$. Suppose the relative degree (Definition 1) of ϕ is $\rho \in \mathbb{N}^r$ and $z_k \in C^{\rho_k}([t_0, \infty), \mathbb{R})$ for each $k \in \{1, \ldots, r\}$. Then, for $k \in \{1, \ldots, r\}$ we have that,*

$$\begin{aligned}
&\frac{d^i}{dt^i}\left(\phi_k(t, x(t)) + \frac{z_k^2(t)}{2}\right) \\
&\quad = L_f^i \phi_k(t, x(t)) + z_k(t) z_k^{(i)}(t) + \frac{1}{2}\sum_{j=1}^{i}\binom{i}{j} z_k^{(i-j)}(t) z_k^{(j)}(t) \\
&\qquad \text{for } i \in \{1, \ldots, \rho_k - 1\}
\end{aligned} \tag{9}$$

$$\begin{aligned}
&\frac{d^{\rho_k}}{dt^{\rho_k}}\left(\phi_k(t, x(t)) + \frac{z_k^2(t)}{2}\right) \\
&\quad = L_g L_f^{\rho_k - 1}\phi_k(t, x(t))u(t) + L_f^{\rho_k}\phi_k(t, x(t)) + z_k(t) z_k^{(\rho_k)}(t) \\
&\qquad + \frac{1}{2}\sum_{j=1}^{\rho_k}\binom{\rho_k}{j} z_k^{(\rho_k - j)}(t) z_k^{(j)}(t).
\end{aligned} \tag{10}$$

Proof. From Definition 1, it follows that the derivatives of $\phi_k(t, x(t))$ satisfy

$$\begin{aligned}
\frac{d^i \phi_k(t, x(t))}{dt^i} &= \frac{\partial L_f^{i-1}\phi_k(t, x(t))}{\partial x(t)}\dot{x}(t) + \frac{\partial L_f^{i-1}\phi_k(t, x(t))}{\partial t} \\
&= L_f^i \phi_k(t, x(t)) \quad \text{, for } i \in \{1, \ldots, \rho_k - 1\}, \\
\frac{d^{\rho_k}\phi_k(t, x(t))}{dt^{\rho_k}} &= L_f^{\rho_k}\phi_k(t, x(t)) + L_g L_f^{\rho_k - 1}\phi_k(t, x(t))u(t).
\end{aligned}$$

By General Leibniz rule [40, p. 318], $\frac{d^n}{dx^n}(u(x)v(x)) = \sum_{k=0}^{n}\binom{n}{k}\frac{d^k u(x)}{dx^k}\frac{d^{n-k}v(x)}{dx^{n-k}}$, it follows that the $i \geq 0$ derivatives of $\frac{1}{2}z_k(t)^2$ satisfy

$$\begin{aligned}
\frac{d^i}{dt^i}\left(\frac{1}{2}z_k(t)^2\right) &= \frac{1}{2}\sum_{j=0}^{i}\binom{i}{j} z_k^{(i-j)}(t) z_k^{(j)}(t) \\
&= z_k(t) z_k^{(i)}(t) + \frac{1}{2}\sum_{j=1}^{i-1}\binom{i}{j} z_k^{(i-j)}(t) z_k^{(j)}(t).
\end{aligned}$$

□

For any feasible trajectory, there must exist slack variables such that Equation (8) is satisfied. Thus, the quantity $\phi_k(t, x(t)) + \frac{1}{2}z_k^2(t)$ must remain constant over time, that is, the Right Hand Side (RHS) of Equation (10) is equal to zero. The RHS of Equation (10) is affine in the system input variable for each $r \in \mathbb{N}$, thus if we make the following invertibility assumption, we can find an expression for the system input in terms of the slack variable.

Assumption 1. The *constraint decoupling matrix* defined as,

$$\Omega_g(t, x) := [L_g L_f^{\rho_1 - 1}\phi_1(t, x)^\top, \ldots, L_g L_f^{\rho_r - 1}\phi_r(t, x)^\top]^\top \in \mathbb{R}^{r \times m}$$

is invertible for all $(t, x) \in [t_0, \infty) \times \mathbb{R}^n$. Note, $r = m$ (see Equation 7) implying Ω_g is a square matrix.

Defining $S_{k,i-1}(t) := \frac{1}{2}\sum_{j=1}^{i-1}\binom{i}{j} z_k^{(i-j)}(t) z_k^{(j)}(t)$, $\Omega_f(t, x) := [S_{1,\rho_1 - 1} + L_f^{\rho_1}\phi_1(t, x), \ldots, S_{r,\rho_r - 1} + L_f^{\rho_r}\phi_r(t, x)]^\top \in \mathbb{R}^r$, $w(t) := [z_1^{(\rho_1)}(t), \ldots, z_r^{(\rho_r)}(t)]^\top \in \mathbb{R}^r$, $D(z) := \operatorname{diag}(z_1, \ldots, z_r) \in \mathbb{R}^{r \times r}$ allows us to write the system of equations arising from setting the RHS of Equation (10) to be zero as:

$$\Omega_g(t, x(t))u(t) + \Omega_f(t, x(t)) + D(z(t))w(t) = 0. \tag{11}$$

We now define the extended augmented state space:

$$x_A := \begin{bmatrix} x \\ \tilde{z} \end{bmatrix} \in \mathbb{R}^{n_A}, \tag{12}$$

where $\tilde{z} := [z_1, \ldots, z_1^{(\rho_1 - 1)}, \ldots, z_r, \ldots, z_r^{(\rho_r - 1)}]^\top \in \mathbb{R}^{\sum_{k=1}^r \rho_k}$ and $n_A := n + \sum_{k=1}^r \rho_k$.

Under Assumption 1, we solve Equation (11) to derive a relationship between the full state feedback controller, u, and the pseudo feedback controller, w, induced by the state augmentation:

$$u(t, x_A) = -\Omega_g^{-1}(t, x)\big(\Omega_f(t, x) + D(z)w(t, x_A)\big). \tag{13}$$

For simplicity, throughout the paper, we overload the notations u and w to denote both input signals, depending only on time (i.e.,

open-loop controllers), and closed-loop controllers, depending on both time and space.

Upon substituting the controller $u(t, x_A(t))$, from Equation (13), into ODE (4), treating w as a time-varying "virtual input" and noting $\frac{d}{dt} z_i^{(j)}(t) = z_i^{(j+1)}(t)$, we derive the following unconstrained input-affine augmented system $\Sigma_A = (f_A, g_A, h_A, x_A(t_0))$ associated with the following IVP,

$$\dot{x}_A(t) = f_A(t, x_A(t)) + g_A(t, x_A(t))w(t), \quad x_A(t_0) = \begin{bmatrix} x_0 \\ \tilde{z}(t_0) \end{bmatrix},$$
$$y(t) = h_A(t, x_A(t)) := h(t, x(t)), \tag{14}$$

where recalling the definition of $\mathcal{I}_{r,k} \in \mathbb{R}^{1\times r}$ in Equation (1) we get,

$$f_A(t, x_A) := \begin{bmatrix} f(t,x) - g(t,x)\Omega_g^{-1}(t,x)\Omega_f(t,x) \\ [z_1^{(1)}, \ldots, z_1^{(\rho_1-1)}]^\top \\ 0 \\ \vdots \\ [z_r^{(1)}, \ldots, z_r^{(\rho_r-1)}]^\top \\ 0 \end{bmatrix} \in \mathbb{R}^{n_A},$$
$$g_A(t, x_A) := \begin{bmatrix} -g(t,x)\Omega_g^{-1}(t,x)D(z) \\ 0_{(\rho_1-1)\times r} \\ \mathcal{I}_{r,1} \\ \vdots \\ 0_{(\rho_r-1)\times r} \\ \mathcal{I}_{r,r} \end{bmatrix} \in \mathbb{R}^{n_A\times r}. \tag{15}$$

Definition 2. Given a system $\Sigma = (f, g, h, x_0)$ (Equation (3)) and a constraint $\phi \in C^\infty([t_0, \infty) \times \mathbb{R}^n, \mathbb{R}^r)$ we define the following system augmentation map

$$\mathcal{T}_\phi(\Sigma) = (f_A, g_A, h_A, x_A(t_0)),$$

where $(f_A, g_A, h_A, x_A(t_0))$ is associated with the IVP given in Equation (15).

The feedback controller given in Equation (13) provides the relationship between the original system input and the pseudo input resulting from augmentation. By a similar argument, we can find the relationship between the augmented states, z_i, and the original system states by equating the RHS of Equation (9) to zero. Sequentially solving these equations, starting with Equation (8), gives us:

$$z_k(t) = \sqrt{-2\phi_k(t, x(t))}$$
$$z_k^{(i)}(t) = -\frac{L_f^i \phi_k(t, x(t)) + S_{k,i-1}(t)}{z_k(t)} \text{ for } i \in \{1, \ldots, \rho_k - 1\}. \tag{16}$$

The initial condition, $x_A(t_0)$, of the augmented system can then be computed by simply iterating Equation (16) to compute $z_k^{(j)}(t_0)$ for each j and k.

We next show that for any integrable input, w, the controller given in Equation (13) yields a state trajectory that obeys the state constraint given in Equation (6). This is an intuitive result since as we approach the boundary of the state constraint, the slack variable becomes zero, $z = 0_{r\times 1}$. Since the virtual input, w, is multiplied by $D(z) := \text{diag}(z_1, \ldots, z_r)$ in Equation (13), it follows that there is a loss of actuation as we approach the constraint boundary making violation impossible.

Theorem 1. *Consider the IVP given in Equation (14). Under Assumption 1,* **for any integrable input** $w \in L^1([t_0, T], \mathbb{R}^r)$ *it follows that the first component, $x(t)$, of the resulting state trajectory, $x_A(t)$, satisfies the state constraint,*

$$\phi(t, x(t)) \le 0 \text{ for all } t \in [t_0, T]. \tag{17}$$

Proof. By Assumption 1 it follows ϕ has relative degree ρ (Definition 1). Given an integrable input, $w \in L^1([t_0, T], \mathbb{R}^r)$, let $z_i^{(\rho_i)}(t) := w_i(t)$. Since the virtual input w is integrable, it follows by the fundamental theorem of calculus that each of the anti-derivatives of w is absolutely continuous (see corollary 3.33 in [41, p. 105]). Hence, for each $1 \le i \le \rho_j$ there exists an absolutely continuous function $z_i^{(j-1)}(t) := \int_{t_0}^t z_i^{(j)}(s)ds + z_i^{(j-1)}(t_0)$ almost everywhere. Define u according to Equation (13). By rearranging Equation (13), we get Equation (11). From Lemma 1 it then follows

$$\frac{d^{\rho_k}}{dt^{\rho_k}}\left(\phi_k(t, x(t)) + \frac{1}{2}z_k^2(t)\right) = 0 \text{ for each } k \in \{1, \ldots, r\}.$$

By integrating the above equation ρ_k times over $[t_0, T]$ for any $t_0 < t < T$ and applying the initial conditions derived from setting $t = t_0$ in Equations (16), we deduce that the equation given in Equation (8) holds almost everywhere. Since $Z := \left[\frac{1}{2}z_1^2, \ldots, \frac{1}{2}z_r^2\right]^\top \in \mathbb{R}^r$ and any real number squared is positive, it follows that the state constraint (17) holds almost everywhere.

Now, since $x(\cdot)$ and ϕ are continuous it follows $\phi(\cdot, x(\cdot))$ is continuous. By contradiction suppose there exists $t_0 \in [t_0, T]$ such that the state constraint (17) does not hold, that is $\phi(t_0, x(t_0)) > 0$. By continuity there exists $\varepsilon > 0$ and a time interval $I_t \subset [t_0, T]$ such that $\phi(t, x(t)) \ge \varepsilon > 0$ for all $t \in I_t$. However, since any interval has non-zero measure, this contradicts that the state constraint (17) holds almost everywhere. Therefore, it must follow that the state constraint (17) holds for all $t \in [t_0, T]$. □

Note, Theorem 1 applies to the original system (4) after the augmented states are eliminated via back-substitution using Equation (16).

3.1 | Ensuring an Integrable Input

Theorem 1 shows that the augmented system, Σ_A, defined in Equation (14) satisfies the constraint $\phi(t, x(t)) \le 0$ for any integrable input. To ensure any applied controller is integrable, we further modify the augmented system by introducing an integral controller structure. Specifically, we expand the state space by augmenting it with a new integrator state, defined as $\dot{\xi}(t) = \tilde{w}(t)$, and replace the input in the augmented system Σ_A with a bounded function of the integral state. That is

$$w(t) = s_\beta(\xi(t)), \tag{18}$$

where $s_\beta : \mathbb{R} \to [-\beta, \beta]^r$ is some bounded function. In this work, we consider the following choice:

$$s_{\beta,i}(x) := 2\beta\Big(\frac{1}{e^{-x}+1} - 0.5\Big) \text{ for } 1 \le i \le r. \tag{19}$$

The selection of $\beta > 0$ is equivalent deciding on an upper bound of the input and will have an impact on controller synthesis, this is discussed later in Section 4.2.

We now further enlarge the augmented state space in Equation (12) to include the integral state:

$$x_I := \begin{bmatrix} x_A \\ \xi \end{bmatrix} = \begin{bmatrix} x \\ \tilde{z} \\ \xi \end{bmatrix} \in \mathbb{R}^{n_A+m}, \tag{20}$$

The new virtual input is given by $\tilde{w}$, and we denote the resulting input-affine system with integral controller structure by $\Sigma_I = (f_I, g_I, h_I, x_I(t_0))$. The associated IVP with this system is:

$$\begin{aligned} \dot{x}_I(t) &= f_I(t, x_I(t)) + g_I(t, x_I(t))\tilde{w}(t), \quad x_I(t_0) = \begin{bmatrix} x_A(t_0) \\ \xi(t_0) \end{bmatrix}, \\ y(t) &= h_I(t, x_I(t)) := h(t, x(t)), \end{aligned} \tag{21}$$

where

$$f_I(t, x_I) = \begin{bmatrix} f_A(t, x_A) + g_A(t, x_A)s_\beta(\xi) \\ 0_{m\times 1} \end{bmatrix}, \quad g_I(t, x_I) = \begin{bmatrix} 0_{n_A\times m} \\ I_{m\times m} \end{bmatrix},$$

f_A and g_A are given in Equation (15).

Definition 3. Given a system $\Sigma_A = (f_A, g_A, h_A, x_A(t_0))$ (Equation (14)) and a integration bound β, we define the following system integral controller map

$$\mathcal{I}_\beta(\Sigma_A) = (f_I, g_I, h_I, x_I(t_0)),$$

where $(f_I, g_I, h_I, x_I(t_0))$ is associated with the IVP given in Equation (21).

After synthesising the tracking controller, $\tilde{w}$, for the IVP given in the integral augmented system Σ_I (Equation (21)), we can then apply the integral control (18) and the feedback controller (13) to derive a closed-loop feedback controller corresponding to the input of the original ODE (4):

$$u(t, x_I) = -\Omega_g^{-1}(t, x)\big(\Omega_f(t, x) + D(z)s_\beta(\xi)\big) \tag{22}$$

where the ξ evolves according to the FBL controller:

$$\dot{\xi}(t) = \tilde{w}(t, x_I). \tag{23}$$

Note, later in Section 4.1, we will see that FBL controller synthesis is formulated independently of the initial conditions. In particular, altering the initial condition of the integrator state, $\xi(t_0)$, results only in an initial offset in the controller input signal in Equation (22) and will not alter the asymptotic tracking FBL induces under Assumption 2. Therefore, $\xi(t_0)$ is a free variable that the user can tune to bias the initial input signal, $u(t_0)$, and thus the initial direction of the state trajectory. For simplicity, we will select a zero offset by setting $u = 0$ in Equation (22) to derive

$$\xi_0 := \xi(t_0) = s_\beta^{-1}\big(-D(z(t_0))^{-1}\Omega_f(t_0, x(t_0))\big), \tag{24}$$

where the inverse of s_β is applied component wise, $z(t_0)$ is found through setting $t = t_0$ in Eqs. (16) and $x(t_0) = x_0$ from ODE (4).

Now, from the feedback controller (22), the virtual input $w(t)$ given in Equation (13) becomes $w(t) = s_\beta(\xi(t))$. Since s_β is a bounded function, it follows that this virtual input is an element of $L^1([t_0, T], \mathbb{R}^r)$. Hence, we can apply Theorem 1 to show that the system trajectory, $x(t)$, resulting from the application of the feedback controller given in Equation (22) does not violate the state constraint, that is, Equation (17) is satisfied.

3.2 | Sequential Augmentation

In this subsection, we consider the case when the dimension of the state constraints, r, exceeds the dimension of the input space, m. If $r > m$, the system of equations in Equation (11), which must be solved to derive the augmented system Σ_A in Equation (14), becomes overdetermined, meaning there are more constraints than free variables. Consequently, this equation cannot be solved, and Assumption 1 no longer holds.

However, when $r = \alpha m$ for some $\alpha \in \mathbb{N}$, the state space can still be augmented by employing a sequential strategy. Instead of encapsulating all constraints simultaneously, we consider only a subset of m constraints at a time, momentarily ignoring the rest. Specifically, we partition the state constraints as

$$\phi = [\phi_1, \ldots, \phi_\alpha]^\top, \quad \text{where each } \phi_i : [t_0, \infty) \times \mathbb{R}^n \to \mathbb{R}^m. \tag{25}$$

The augmentation process begins by applying the transformation $\mathcal{T}_{\phi_1}$, defined in Definition 2, to obtain the first augmented system $\Sigma_A = \mathcal{T}_{\phi_1}(\Sigma)$. Next, we introduce an integral controller structure by applying $\mathcal{I}_\beta$, as defined in Definition 3, yielding $\Sigma_I = \mathcal{I}_\beta(\Sigma_A)$. Treating Σ_I as the new base system, we repeat this process for each subsequent subset of constraints until all have been incorporated. The complete augmentation process is given by

$$\Sigma_I = \mathcal{I}_\beta(\mathcal{T}_{\phi_\alpha}(\ldots \mathcal{I}_\beta(\mathcal{T}_{\phi_1}(\Sigma))\ldots)). \tag{26}$$

4 | Feedback Linearisation

Transformations $\mathcal{T}_\phi$ and $\mathcal{I}_\beta$, defined in Definitions 2 and 3, preserve the input-affine structure of system Σ given in Equation (4), which allows us to perform the Feedback Linearisation (FBL) method to achieve the tracking goal given in Equation (5). Thus, we first review the classical FBL method for general unconstrained input-affine systems in Section 4.1, and then analyse the challenges of applying FBL to the augmented systems Σ_A (Equations (14)) and Σ_I (Equations (21)) in Section 4.2.

Incorporating the augmented structure from the previous section, given in Equation (14), to handle state constraints.

4.1 | FBL with Pole Placement

Considering a system represented by Equation (4) and a reference signal, $y_r \in C^\infty([t_0, \infty), \mathbb{R}^m)$, we next recall the standard FBL approach that derives an expression that relates the tracking error $E(t) := y(t) - y_r(t)$ to the system input by taking successive time derivatives of tracking error. Having obtained this expression, we can design a feedback law to cancel out any nonlinearity as well as terms not involving the tracking error. The remaining system will be linear allowing us to utilise linear controllers for stabilisation to ensure that tracking errors asymptotically approach zero. In this paper, we choose the pole-placement strategy as it is frequently used for linear system stabilisation, however, it does not mean that the pole-placement is the only standard strategy.

We introduce new notation, $\sigma = [\sigma_1, \ldots, \sigma_m]^\top \in \mathbb{N}^m$, for the relative degree of the output function, $h \in C^\infty([t_0, \infty) \times \mathbb{R}^n, \mathbb{R}^m)$. This allows us to distinguish between $\rho \in \mathbb{R}^r$, the relative degree of the constraint function, $\phi \in C^\infty([t_0, \infty) \times \mathbb{R}^n, \mathbb{R}^r)$, that is defined in Section 3 with possibly differing dimensions to σ.

For each $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, \sigma_i - 1\}$, it follows by repeatedly taking time derivatives of the tracking error, $E(t) := y(t) - y_r(t)$, and by the definition of the relative degree (Definition 1) we have that:

$$\begin{aligned} E_i^{(j)}(t) &= L_f^j h_i(t, x) - y_{r_i}^{(j)}(t) \\ E_i^{(\sigma_i)}(t) &= L_f^{\sigma_i} h_i(t, x) + L_g L_f^{\sigma_i - 1} h_i(t, x(t))u(t) - y_{r_i}^{(\sigma_i)}(t). \end{aligned} \tag{27}$$

We next construct the expression relating the tracking error to the system input. To do this, we introduce the following notation for the stacked Lie derivatives,

$$\begin{aligned} \Gamma_f(t, x) &:= [L_f^{\sigma_1} h_1(t, x), \ldots, L_f^{\sigma_m} h_m(t, x)]^\top \in \mathbb{R}^m, \\ \Gamma_g(t, x) &:= [L_g L_f^{\sigma_1 - 1} h_1(t, x)^\top, \ldots, L_g L_f^{\sigma_m - 1} h_m(t, x)^\top]^\top \in \mathbb{R}^{m \times m}, \\ \Gamma_r(t) &:= [y_{r_1}^{(\sigma_1)}(t), \ldots, y_{r_p}^{(\sigma_m)}(t)]^\top \in \mathbb{R}^m. \end{aligned} \tag{28}$$

Our vector and matrix notation in Equations (28) now allows us to collect the derivatives of the tracking error given in Equations (27) into the following matrix equation,

$$\begin{bmatrix} E_1^{(\sigma_1)}(t) \\ \vdots \\ E_m^{(\sigma_m)}(t) \end{bmatrix} = \Gamma_f(t, x(t)) + \Gamma_g(t, x(t))u(t) - \Gamma_r(t). \tag{29}$$

We can now eliminate the nonlinearities if we make the following assumption,

Assumption 2. The *output decoupling matrix* defined as,

$$\Gamma_g(t, x) := [L_g L_f^{\sigma_1 - 1} h_1(t, x)^\top, \ldots, L_g L_f^{\sigma_m - 1} h_m(t, x)^\top]^\top \in \mathbb{R}^{m \times m}$$

is invertible for all $(t, x) \in [t_0, \infty) \times \mathbb{R}^n$.

The tracking error dynamics can now be linearised by using the controller $u = w(t, x)$, where

$$w(t, x) = -\Gamma_g(t, x)^{-1}\big(\Gamma_f(t, x) - \Gamma_r(t) - \nu(t)\big), \tag{30}$$

and $\nu \in \mathbb{R}^m$ is a "virtual input" that we will design using pole placement later. Substituting Equation (30) into Equation (29), we get that $[E_1^{(\sigma_1)}(t), \ldots, E_m^{(\sigma_m)}(t)]^\top = [\nu_1(t), \ldots, \nu_m(t)]^\top$.

To construct the state space of our linearised tracking error dynamics we define a column vector consisting of stacked tracking error derivatives, denoted by

$$\mathbf{E}(t) := [\mathbf{E}_1^\top(t), \ldots, \mathbf{E}_m^\top(t)]^\top \in \mathbb{R}^{\sum_{i=1}^m \sigma_i}, \tag{31}$$

where the ith component is given by $\mathbf{E}_i(t) := [E_i(t), \ldots, E_i^{(\sigma_i - 1)}(t)]^\top \in \mathbb{R}^{\sigma_i}$ and $E_i(t) := y_i(t) - y_{r,i}(t)$. Applying the controller given in Equation (30) and noting $\dot{E}_i^{(j)} = E_i^{(j+1)}$ we derive the full linearised tracking error dynamics:

$$\dot{\mathbf{E}}(t) = A\mathbf{E}(t) + B\nu(t), \tag{32}$$

where $A \in \mathbb{R}^{\sum_{i=1}^m \sigma_i \times \sum_{i=1}^m \sigma_i}$ is a block diagonal matrix of the form $A = \mathrm{diag}(A_1, \ldots, A_m)$ with $A_i := \begin{bmatrix} 0_{(\sigma_i - 1) \times 1} & I_{\sigma_i - 1} \\ 0 & 0_{1 \times (\sigma_i - 1)} \end{bmatrix} \in \mathbb{R}^{\sigma_i \times \sigma_i}$ and $B \in \mathbb{R}^{\sum_{i=1}^m \sigma_i}$ is a block diagonal matrix of the form $B = \mathrm{diag}(B_1, \ldots, B_m)$ with $B_i := [0_{1 \times (\sigma_i - 1)}, 1]^\top \in \mathbb{R}^{\sigma_i}$.

Determining Gains By Pole-Placement: We now recall the classical pole-placement technique in the context of FBL. A wider more in-depth overview of this technique can be found in [42]. To stabilise ODE (32) we use a feedback controller $\nu(t) = -K\mathbf{E}(t)$ where $K \in \mathbb{R}^{m \times \sum_{i=1}^m \sigma_i}$ is a gain matrix that has the block diagonal structure $K = \mathrm{diag}(K_1, \ldots, K_m)$ with $K_i \in \mathbb{R}^{1 \times \sigma_i}$. Because of the block structure of ODE (32), the system can be decoupled into m-subsystems with states $\mathbf{E}_i$. Each of these subsystems is in "canonical/companion" form. Substituting

$$\nu(t) = -K\mathbf{E}(t), \tag{33}$$

we get the following $i \in \{1, \ldots, m\}$ subsystems $\dot{\mathbf{E}}_i(t) = (A_i - B_i K_i)\mathbf{E}_i(t)$, where

$$A_i - B_i K_i = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \\ -K_{i,1} & -K_{i,2} & -K_{i,3} & \ldots & -K_{i,\sigma_i} \end{bmatrix}. \tag{34}$$

Because of the "canonical/companion" form, it then follows that the characteristic equation is $\det(\lambda I - (A_i - BK_i)) = \lambda^{\sigma_i} + K_{i,\sigma_i}\lambda^{\sigma_i - 1} + \ldots + K_{i,2}\lambda + K_{i,1}$. Then, to select the poles of this system to be at

$$\overline{\lambda}_i := [\overline{\lambda}_{i,1}, \ldots, \overline{\lambda}_{i,\sigma_i}] < 0, \tag{35}$$

we simply expand $(\lambda - \overline{\lambda}_{i,1}) \ldots (\lambda - \overline{\lambda}_{i,\sigma_i})$ and equate with the characteristic equation, $\det(\lambda I - (A_i - BK_i))$. By equating the coefficients of the same powers of λ, we solve for the $K_{i,j}$'s, selecting the appropriate gains for this pole placement. We will later discuss the selection criteria for poles in Section 5.4.

Under Assumptions 1 and 2, the feedback controller (Equation (22)) and FBL controller (Equation 30) are synthesised offline from state augmentation and FBL, and then

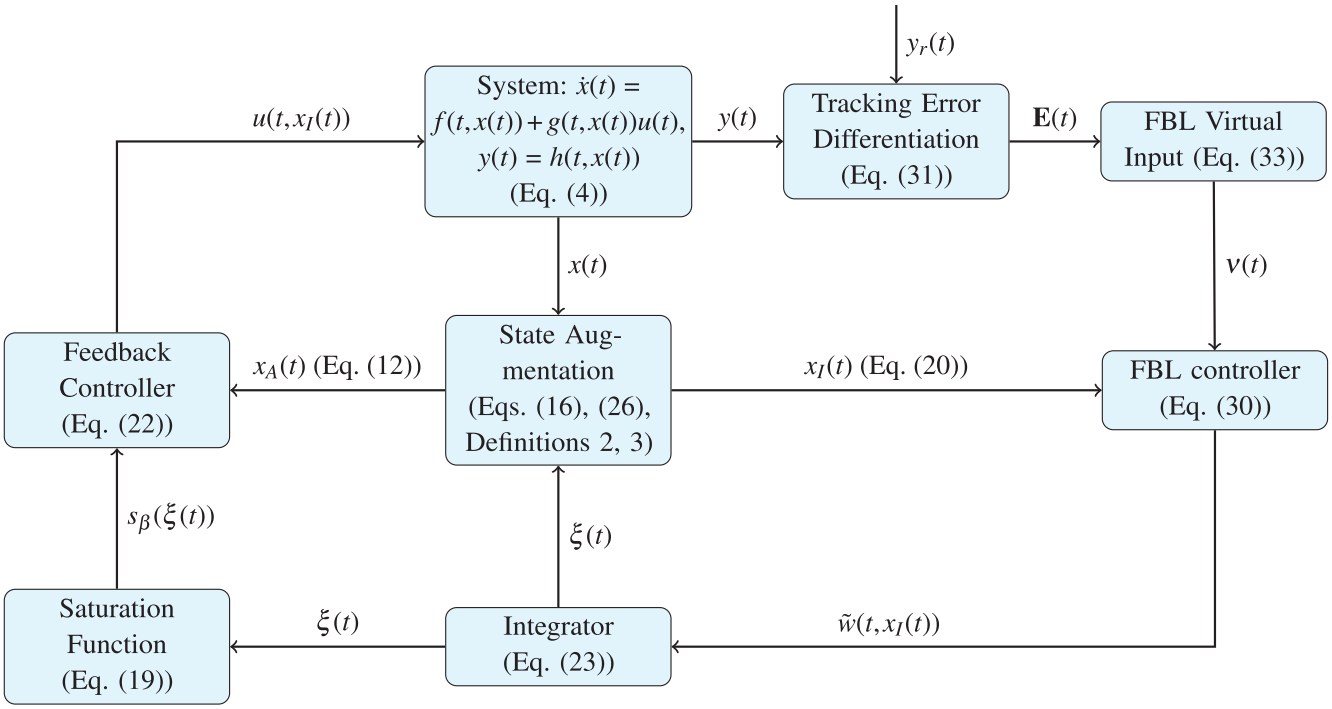


FIGURE 3 | Block diagram showing the closed loop online implementation of the feedback controller $u(t, x_I) \in \mathbb{R}^m$ (Equation 22). The augmented state space x_A and the integral augmented state space x_I are constructed from x via Equations (16). Taking derivatives of the tracking error and substituting them into Equation (33), the virtual input $v(t) \in \mathbb{R}^m$ is obtained, which together with x_I, determines the FBL controller $\tilde{w}(t, x_I) \in \mathbb{R}^m$ via Equation (30). This FBL controller is integrated through ODE (23) to yield the integral state ξ. The feedback controller $u(t, x_I)$ is obtained by substituting $s_\beta(\xi)$ and x_A into Equation (22), and is then implemented in the original system (4).

employed for online implementation. Figure 3 summarises the online implementation of the feedback controller in our method. More specifically, the FBL controller $\tilde{w}(t, x_I(t))$ is integrated via ODE (23) to compute the integral state $\xi(t)$, which is eventually coupled into the feedback controller $u(t, x_I(t))$. The feedback controller $u(t, x_I(t))$ depends on the augmented state x_A, constructed from the original state x through Equation (16), which is then implementable on the original system (4). Recall that the initial condition $\xi_0 = \xi(t_0)$ is computed by using Equation (24) after setting the feedback controller u to zero manually.

4.2 | Challenges: Augmentation with FBL

As we have seen in Section 4.1, FBL can be used to construct an asymptotically tracking controller, provided Assumption 2 holds, that is, the matrix Γ_g is invertible. In practice, FBL only requires Γ_g to be invertible along the closed-loop trajectory.

However, when applying FBL to the integral augmented system Σ_I defined in Equation (21), this assumption fails to hold due to two key structural issues that we list next, but leave the full details to the Appendix for brevity:

a. **Boundary singularity.** At the edge of the constraint set, some slack variables z_k become zero, leading to a loss of rank in Γ_{g_I}, that is, Γ_{g_I} is not invertible along the state constraint boundary. This results in an ill-defined relative degree and prevents direct application of FBL (see Corollary 1).

b. **β-dependent ill-conditioning.** As the bound $\beta > 0$ on the integral controller is reduced, the matrix Γ_{g_I} becomes increasingly ill-conditioned (see Lemma 2).

Similarly to the invertibility issues discussed above, where Assumption 2 may fail to hold, preventing the application of FBL, it is also possible, depending on the system structure, that Assumption 1 does not hold. In such cases, the matrix Ω_g becomes non-invertible, thereby preventing state augmentation and constraint capture.

Even when Ω_g or Γ_{g_I} remains theoretically invertible, they may be difficult to invert under finite precision arithmetic computation if their values approach zero, leading to numerical instability. However, since both augmentation and FBL only require invertibility of Ω_g and Γ_{g_I} along the closed-loop trajectory, it may still be feasible to apply these methods provided that the trajectory avoids regions where these matrices are non-invertible. If the system approaches a non-invertibility region, increasing β can help mitigate ill-conditioning of Γ_{g_I}, as Lemma 2 shows that Γ_{g_I} depends linearly on β. In cases where the system trajectory passes directly through singular regions, we introduce in Section 5 a switching strategy that adapts to the system's *local* relative degree. Note that we will consider SISO systems only in Section 5. If the singularity issue never occurs, our work prior to this section remains valid for MIMO systems, and an asymptotic tracking controller can be synthesised and implemented following Figure 3.

5 | A Practical Approach for State Constrained FBL

In this section, we will present a practical framework for the implementation of state constrained FBL for SISO systems, addressing the two challenges given in Section 4.2.

The key idea is to synthesise different closed-form controllers for regions with varying "local" relative degrees and switch between them as the trajectory transitions through these regions. Following [32, 33], we restrict our focus to SISO systems ($m = 1$), where $\Omega_g(t,x) := L_g L_f^{\rho_1 - 1}\phi_1(t,x)$ and $\Gamma_g(t,x) = L_g L_f^{\sigma-1}h(t,x)$ are both scalar. For brevity, we refer to [32, 33] for conditions ensuring switching FBL controllers result in an asymptotic approximate tracking with a bounded tracking error based on the magnitude of the switching threshold. This work focuses on implementing the switching strategy in the context of our state constrained FBL framework, where we apply a localised relative degree (Definition 4) to augment the system, capturing constraints, and perform FBL controller synthesis.

Definition 4 (The ϵ-Numerical Relative Degree Function). Given a system $\Sigma = (f, g, h, x_0)$ (Equation 3), a point $(t,x) \in [t_0, \infty) \times \mathbb{R}^n$, a differentiable function $\psi : [t_0, \infty) \times \mathbb{R}^n \to \mathbb{R}$ and a positive sequence $\{\varepsilon_i\}_{i=1}^{\infty} \subset (0, \infty)$. We say the ϵ-Numerical Relative Degree (ϵ-NRD) at (t,x) is $\hat{\rho} \in \mathbb{N}$ if the following two conditions are satisfied:

(1) $|L_g L_f^i \psi(t,x)| \leq \varepsilon_i$ for $i \in \{0, \ldots, \hat{\rho} - 2\}$.

(2) $|L_g L_f^{\hat{\rho}-1}\psi(t,x)| > \varepsilon_{\hat{\rho}-1}$.

Note, when $\varepsilon_i = 0$ holds for all $i \in \mathbb{N}$ and when the two conditions in Definition 4 are enforced globally for all $(t,x) \in [t_0, \infty) \times \mathbb{R}^n$, the ϵ-NRD becomes equivalent to the relative degree (Definition 1).

In the following subsections, we synthesise a tracking controller following the exact same steps detailed in the previous sections, replacing relative degrees with ϵ-NRDs as follows:

1. Augment system $\Sigma = (f, g, h, x_0)$ (Equation (3)) to capture constraints based on ϵ-NRD, $\hat{\rho}$.
2. Apply FBL to system Σ_I (Equation (21)) based on ϵ-NRD, $\hat{\sigma}$, to find the FBL controller (44).
3. Sequentially use Equations (16) to remove the augmented states in the FBL controller (44), obtaining a closed-loop controller. Couple the integral states to the original ODE (4) and implement the derived closed-loop system.

5.1 | Constraint Capture Using ϵ-Numerical Relative Degree

To address potential numerical invertibility issues caused by an ill-defined relative degree when capturing state constraints, we follow the same differentiation procedure as in Equations (9) and (10), but now discard terms where the input is multiplied by a coefficient smaller than a user defined ϵ-threshold. The number of derivatives required to ensure the coefficient of the input is sufficiently large, that is, satisfying $|L_g L_f^{j-1}\phi_k(t,x)| > \epsilon_j$, matches the $\hat{\rho}$ in the ϵ-NRD Definition 4, and varies with (t,x).

At a given temporal state-space point (t_0, x_0), and after neglecting input-related terms smaller than the threshold, we obtain:

$$\frac{d^{\hat{\rho}_k}}{dt^{\hat{\rho}_k}}\left(\phi_k(t,x(t)) + \frac{1}{2}z_k^2(t)\right) \approx \frac{1}{2}\sum_{j=1}^{\hat{\rho}_k}\binom{\hat{\rho}_k}{j} z_k^{(\hat{\rho}_k - j)}(t) z_k^{(j)}(t) + z_k(t) z_k^{(\hat{\rho}_k)}(t) + L_g L_f^{\hat{\rho}_k - 1}\phi_k(t,x)u(t) + L_f^{\hat{\rho}_k}\phi_k(t,x). \tag{36}$$

We overload our notation, $\hat{\rho}$, to define the ϵ-NRD Function, $\hat{\rho} : [t_0, \infty) \times \mathbb{R}^n \to \mathbb{N}^r$, for each point in time and state space. The ϵ-NRD, $\hat{\rho} = [\hat{\rho}_1(t_0, x_0), \ldots, \hat{\rho}_r(t_0, x_0)]^\top \in \mathbb{N}^r$, is denoted as the ϵ-NRD evaluated at (t_0, x_0), which represents the current system state.

Given a constraint $\phi \in \mathbb{R}^r$, similarly to Equation (12), we start by extending the state space to capture the first component ϕ_1,

$$\hat{x}_A := \begin{bmatrix} x \\ \hat{\bar{z}} \end{bmatrix} \in \mathbb{R}^{\hat{n}_A} \tag{37}$$

where $\hat{\bar{z}} := [z_1, \ldots, z_1^{(\hat{\rho}_1(t_0,x_0)-1)}]^\top$ and $\hat{n}_A = n + \hat{\rho}_1(t_0, x_0)$. Note, the state space dimension now depends on a fixed temporal space coordinate (t_0, x_0). For SISO systems, we are only able to capture a single state constraint at a time, for other state components $k > 1$, we later sequentially capture according to Section 3.2.

In the same manner as Equation (11), we set the RHS of Equation (36) to zero, starting with $k = 1$ (and then later for $k > 1$ during sequential augmentation). We solve the resulting system of equations to derive a controller as a function of the pseudo input, $\hat{u}(t) = z_1^{(\hat{\rho}_k(t_0,x_0))}(t)$:

$$\hat{u}_{(t_0,x_0)}(t, \hat{x}_A) = -\hat{\Omega}_g^{-1}(t,x)\left(\hat{\Omega}_f(t,x) + D(z)\hat{u}(t)\right), \tag{38}$$

where $\hat{\Omega}_g(t,x) = L_g L_f^{\hat{\rho}_1(t_0,x_0)-1}\phi_1(t,x)$, $\hat{\Omega}_f(t,x) = L_f^{\hat{\rho}_1(t_0,x_0)}\phi_1(t,x)$ and $D(z) = z_1$. Note, Equation (38) is well defined within a neighbourhood centred at (t_0, x_0) since $|\Omega_g(t_0,x_0)| = |L_g L_f^{\hat{\rho}_1(t_0,x_0)-1}\phi_1(t_0,x_0)| > \epsilon_{\hat{\rho}_1 - 1}$ follows from the definition of the ϵ-NRD. Hence, $\Omega_g(t_0, x_0)$ is invertible.

Given a point (t_0, x_0) and a threshold ϵ, we construct an augmented system by extending the state space according to Equation (37) and substituting the input according to Equation (38) to derive the system,

$$\hat{\Sigma}_A = (\hat{f}_A, \hat{g}_A, \hat{h}_A, \hat{x}_A(t_0)). \tag{39}$$

Although the system $\hat{\Sigma}_A$ given in Equation (39) has the same structure as the augmented system Σ_A in Equation (14), we use hat notation here to emphasise that $\hat{\Sigma}_A$ is constructed based on the ϵ-NRD (Definition 4, which depends on a local point (t_0, x_0) and threshold ϵ), rather than the relative degree (Definition 1).

Analogous to Definition 2 we next define this augmentation transformation based on the ϵ-NRD:

Definition 5. Given a system $\Sigma = (f, g, h, x_0)$ (Equation 3), a constraint $\phi \in C^\infty([t_0, \infty) \times \mathbb{R}^n, \mathbb{R})$ a local point (t_0, x_0) and threshold ε we define the following system augmentation map

$$\hat{\mathcal{T}}_{\phi,\varepsilon,(t_0,x_0)}(\Sigma) = \hat{\Sigma}_A$$

where $\hat{\Sigma}_A = (\hat{f}_A, \hat{g}_A, \hat{h}_A, \hat{x}_A(t_0))$ is defined in Equation (39).

After constructing $\hat{\Sigma}_A$ using state constraint capture, analogous to Section 3.1, we perform the operator $\mathcal{I}_\beta(\cdot)$ from Definition 3 on system $\hat{\Sigma}_A$ to obtain an integral augmented system $\hat{\Sigma}_I = (\hat{f}_I, \hat{g}_I, \hat{h}_I, \hat{x}_I(t_0))$, where state space is given by

$$\hat{x}_I := \begin{bmatrix} \hat{x}_A \\ \xi \end{bmatrix} \in \mathbb{R}^{\hat{n}_A + r}. \tag{40}$$

Recall that if $r = 1$, $\xi = \xi_1 \in \mathbb{R}$ evolves according to the FBL controller via ODE (23) but the state space is updated to $\hat{x}_I$. Substituting the integral control $\imath\hat{v} = s_\beta(\xi_1)$ (Equation 18) into the feedback controller (Equation 38), we derive the following feedback controller based on ε-NRD about (t_0, x_0):

$$\hat{u}_{(t_0,x_0)}(t, \hat{x}_I) = \hat{\Omega}_g^{-1}(t, x)\Big(\hat{\Omega}_f(t, x) + D(z)s_\beta(\xi_1)\Big). \tag{41}$$

If the dimension of the constraint is greater than one, $r > 1$, analogous to Section 3.1, we sequentially capture the remaining components $\phi_2, \ldots, \phi_r$. The complete augmentation process is given by $\hat{\Sigma}_I = \mathcal{I}_\beta(\hat{\mathcal{T}}_{\phi_r}(\ldots \mathcal{I}_\beta(\hat{\mathcal{T}}_{\phi_1}(\Sigma))\ldots))$. Then, the FBL controller is coupled to ξ_1 via an r layers of ODEs analogous to (23).

If the ε-NRD changes along the system trajectory, we will need to update the controller used in Equation (41). We detect changes in the ε-NRD by collecting the previously computed Lie derivatives and monitoring whether individual components pass the ε defined threshold. To this end, we define:

$$\omega_{k,(t_0,x_0)}(t, x) := \Big[|L_g L_f^0 \phi_k(t, x)|, \ldots, |L_g L_f^{\hat{\rho}_k(t_0,x_0)-1} \phi_k(t, x)|\Big] \tag{42}$$

For brevity in Equation (42) we have abused notation and ignored integral states, $\xi_1, \ldots, \xi_{k-1}$, dependencies when $k > 2$. This dependency occurs due to sequential augmentation, where we treat the integral augmented system as the base system for the next augmentation, and introduce a new integral state in each augmentation.

Our method for capturing state constraints by numerically sequentially augmenting a system about (t_0, x_0) is summarised in Algorithm 1.

ALGORITHM 1 | Sequential constraint capture.

Input: System Σ (3), Constraints ϕ (25), Threshold ε, Integration Bound β
Current system state (t_0, x_0).
Output: System $\hat{\Sigma}_I$ (21),
Lie derivatives $\{\omega_{k,(t_0,x_0)}\}_{k=1}^r$ (42).

$\hat{\Sigma}_A = \hat{\mathcal{T}}_{\phi_1,\varepsilon,(t_0,x_0)}(\Sigma)$ obtaining $\omega_{1,(t_0,x_0)}$ ▷ $\hat{\mathcal{T}}$ from Definition 5
$\hat{\Sigma}_I = \mathcal{I}_\beta(\hat{\Sigma}_A)$ ▷ $\mathcal{I}_\beta$ from Definition 3.
for $k \in \{2, \ldots, r\}$ **do**
  $\hat{\Sigma}_A = \hat{\mathcal{T}}_{\phi_k,\varepsilon,(t_0,x_0)}(\hat{\Sigma}_I)$obtaining $\omega_{k,(t_0,x_0)}$
  $\hat{\Sigma}_I = \mathcal{I}_\beta(\hat{\Sigma}_A)$
end for

5.2 | FBL with ε-Numerical Relative Degree

Following the same approach used in Section 5.1 for constraint capture, we now adapt classical FBL (Section 4.1) by replacing the relative degree (Definition 1) with the ϵ-NRD (Definition 4). Lie derivatives are stored to detect changes in the local ϵ-NRD and to update the controller accordingly.

Taking time derivatives of the tracking error, as described in Equations (27), and ignoring terms with magnitude less than $\varepsilon_i > 0$ multiplied by the input around a given point (t_0, x_0), we get:

$$E^{(\hat{\sigma})}(t) \approx L_f^{\hat{\sigma}} h(t, x) + L_g L_f^{\hat{\sigma}-1} h(t, x)u(t) - y_r^{(\hat{\sigma})}(t), \tag{43}$$

where $\hat{\sigma} = \hat{\sigma}(t_0, x_0)$ denotes the ε-NRD of the output map at evaluated (t_0, x_0).

Now, following the same process used to derive the FBL controller given in Equation (30) but using the approximated derivative of the tracking error given in Equation (43), we derive the following FBL controller associated with the point (t_0, x_0),

$$\imath\hat{v}_{(t_0,x_0)}(t, x) = -\hat{\Gamma}_g(t, x)^{-1}\big(\hat{\Gamma}_f(t, x) - \hat{\Gamma}_r(t) + \hat{v}(t)\big), \tag{44}$$

where $\hat{\Gamma}_g(t, x) = L_g L_f^{\hat{\sigma}(t_0,x_0)-1} h(t, x)$, $\hat{\Gamma}_f(t, x) = L_f^{\hat{\sigma}(t_0,x_0)} h(t, x)$, $\hat{\Gamma}_{y_r}(t) = y_r^{(\hat{\sigma}(t_0,x_0))}(t_0, x_0)$ and $\hat{v}(t)$ is the "virtual" input.

To detect changes in the ε-NRD during implementation, we collect the Lie derivatives in the following row vector:

$$\gamma_{(t_0,x_0)}(t, x) := \Big[|L_g L_f^0 h(t, x)|, \ldots, |L_g L_f^{\hat{\sigma}(t_0,x_0)-1} h(t, x)|\Big]. \tag{45}$$

We use $\omega_{k,(t_0,x_0)}(t, x)$ (Equation 42) and $\gamma_{(t_0,x_0)}$ (Equation 45) to form switching sets to detect the change of ε-NRDs, with details presented in Section 5.3 with Equations (46) and (47). If the ε-NRD changes, then we must update Equation (43) and synthesise a new FBL controller based on the current temporal and state space coordinates. We have summarised how to numerically synthesise such an FBL for a given state (t_0, x_0) and threshold ε for a general SISO system Σ in Algorithm 2.

5.3 | Controller Implementation

Eliminating Augmented States in Controller Design: For practical implementation, it is preferable for controllers to act directly on the original system rather than an augmented system. Although the augmented states introduced during our proposed controller synthesis method are algebraically dependent on the original states and hence require no additional measurements to evaluate, we eliminate these auxiliary variables to simplify implementation. Our controller is thus reformulated purely in terms of the original system and integral states.

ALGORITHM 2 | The FBL controller synthesis.

Input: System Σ (3), System Poles $\bar{\lambda}$, Threshold ϵ, Current system state (t_0, x_0).

Output: Controller $\hat{\iota\omega}_{(t_0,x_0)}$ (44), Lie derivatives $\gamma_{(t_0,x_0)}$ (45).

1: For ODE (4), take ϵ-NRD, $\hat{\sigma}(t_0, x_0)$ (Definition 4), derivatives of the tracking error, E, as in Eq. (43) and obtain $\gamma_{(t_0,x_0)}$ from Eq. (45).
2: Construct the linearised tracking error dynamics (32).
3: Compute the virtual input $\hat{\nu}$ using pole placement by equating coefficients of the characteristic equation of the linearised error dynamics in Eq. (32).
4: Compute the FBL controller, $\hat{\iota\omega}_{(t_0,x_0)}$, using Eq. (44).

Given a system Σ (3) with state constraints ϕ (25), a feasible initial point (t_0, x_0), and a threshold $\varepsilon > 0$, our controller synthesis proceeds in three stages. First, we apply Algorithm 1 to sequentially encode the constraints, yielding an integral-augmented system $\hat{\Sigma}_I$ and an associated feedback controller $\hat{\iota\omega}_{(t_0,x_0)}$(41), which depends on a virtual input to be determined next via FBL. Second, we invoke Algorithm 2 to perform FBL on $\hat{\Sigma}_I$, obtaining a controller $\hat{\iota\omega}_{(t_0,x_0)}$ (44). Finally, using Equations (16), we sequentially eliminate all augmented states, $\tilde{z}$, in $\hat{\iota\omega}_{(t_0,x_0)}(t, \hat{x}_I)$ (Equation (44)) and $\hat{\iota u}_{(t_0,x_0)}(t, \hat{x}_I)$ (Equation (41)) producing the controllers $\pi_{(t_0,x_0)} : [0, \infty) \times \mathbb{R}^{n+r} \to \mathbb{R}^m$ and $\hat{\iota u}_{(t_0,x_0)}(t, [x, \xi])$, which depend only on the original and integral states.

Unlike the algebraic relations in Equation (16), which allow direct substitution of the augmented states $\tilde{z}$, the integral state ξ evolves according to its own nonlinear ODE (23) and therefore remains part of the closed-loop system. This ODE depends only on ξ and the original system state x, but in general it cannot be solved analytically, preventing the elimination of ξ from the controller. Consequently, the controller possesses internal dynamics in ξ, a standard feature of integral control, where the state accumulates information over time. The dynamics of ξ, with its initial value ξ_0 determined explicitly from x_0 via Equation (24), are thus coupled with the original ODE (4) in which the synthesised controllers $\pi_{(t_0,x_0)}$ and $\hat{\iota u}_{(t_0,x_0)}$ are implemented. We will next see the implementation details of the controllers in Example 1.

Controller Switching Based on ϵ-NRD Changes: During closed-loop simulation, as the state evolves, the ϵ-NRDs may change, necessitating an update of the controller. To detect these changes, we use the outputted Lie derivatives $\{\omega_{k,(t_0,x_0)}\}_{k=1}^{r}$ (Equation 42) from Algorithm 1 and $\gamma_{(t_0,x_0)}$ (Equation 45) from Algorithm 2, to define the sets

$$
\begin{aligned}
C_1 :=& \bigcap_{k=1}^{r} \Big\{(t, x, \xi) : [\omega_{k,(t_0,x_0)}(t, [x, \xi])]_i < \epsilon_i \text{ for} \\
& 1 \le i \le \hat{\rho}_k(t_0, x_0) - 1, \text{ and} \\
& [\omega_{k,(t_0,x_0)}(t, [x, \xi])]_{\hat{\rho}_k(t_0,x_0)} \ge \epsilon_{\hat{\rho}_k(t_0,x_0)}\Big\}, \\
C_2 :=& \Big\{(t, x, \xi) : [\gamma_{(t_0,x_0)}(t, [x, \xi])]_i < \epsilon_i \text{ for } 1 \le i \le \hat{\sigma}(t_0, x_0) \\
& -1, \text{ and } [\gamma_{(t_0,x_0))}(t, [x, \xi])]_{\hat{\sigma}(t_0,x_0)} \ge \epsilon_{\hat{\sigma}(t_0,x_0)}\Big\}.
\end{aligned} \tag{46}
$$

The controller remains valid while ϵ-NRDs remains constant:

$$(t, [x(t), \xi(t)]) \in C_1 \cap C_2,$$

and a switch is triggered when a change in the ϵ-NRD is detected, that is:

$$(t, [x(t), \xi(t)]) \notin C_1 \cap C_2. \tag{47}$$

When a switch is activated, Algorithms 1 and 2 are executed to synthesise a new controller based on the current temporal state coordinate $(t, x(t))$. Since controller synthesis entails some computation, in practice, it is best to leverage memory storage to retain previously computed controllers indexed by their ϵ-NRD. When the system enters a region of the state space where the ϵ-NRD matches a previously encountered value, we retrieve the corresponding precomputed controller from memory rather than synthesising a new one.

5.4 | Selection of Controller Synthesis Parameters

Parameters threshold ϵ, the integration bound β, and the system poles $\bar{\lambda}$, are required for Algorithms 1 and 2. In practice, a grid search may be used to select the controller parameters that yield the desired performance. We next provide some guidelines on why you should not select these parameters to be too large or too small.

1. The switching region (the compliment of $C_1 \cap C_2$ from Equation (46)) is a function of the threshold ϵ. Increasing ϵ shrinks $C_1 \cap C_2$, which increases the size of the switching region. By the same logic, decreasing ϵ reduces the size of the switching region. Selecting a small ϵ threshold gives a small switching region, requiring a small numerical ODE step size to avoid inadvertently jumping too far in the direction of the vector field, causing constraint violation. Small numerical step size implies that the implementation of the controller on a physical system requires a high sensor refresh rate. On the other hand, selecting a ϵ-threshold that is too large results in significant approximation errors when evaluating ϵ-NRDs (Definition 4), and may eventually degrade the performance of the controller (see [32] that provides a bound on asymptotic tracking error depending on the magnitude of ϵ threshold).
2. Lemma 2 demonstrates that the integration bound β depend on Γ_{g_I} linearly, where Γ_{g_I} is defined in Corollary 1. Thus, β is selected not to be too small to improve the invertibility of Γ_{g_I}. At the same time, β should not be chosen too large, considering the controller effort, which increases with β as seen in Equation (22).
3. If the real parts of the poles of a linear system are selected as negative, then it is well known that the system is exponentially stable. In the context of ODE (32), this means that the tracking error, as well as the tracking error time derivatives, exponentially tend to zero. Poles can be adjusted so that criteria such as settling time, percentage overshoot, etc., are within a specified range. In general, poles placed further to the left on the real axis result in faster convergence but

greater overshoot. We present a numerical experiment with different pole placement schemes in Example 1.

6 | Numerical Experiments

We next provide two numerical examples to demonstrate our proposed FBL algorithm with state constraints. In the first example, we will manually derive the analytical controller to illustrate the algorithm. In the second numerical example, the algorithmic steps are carried out using Matlab's symbolic toolbox. In both examples, after the controller is synthesised, the closed-loop system is simulated using Matlab's ODE solver.

Example 1 (Illustrative Example). Let us consider the following problem adapted from [31],

$$\begin{aligned}
&\text{Find } u \text{ such that } \lim_{t\to\infty}||y(t)-y_r(t)||_2 = 0 \text{ where,}\\
&\begin{bmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\end{bmatrix} = \begin{bmatrix}x_2(t)\\ -x_2(t)\end{bmatrix} + \begin{bmatrix}0\\1\end{bmatrix}u(t),\ x(0) = \begin{bmatrix}0\\-2\end{bmatrix}\\
&y(t) = x_1(t) \text{ with } y_r(t) = 0\\
&\phi(t,x(t)) := x_1(t) - 8(t-0.5)^2 + 0.5 \le 0.
\end{aligned} \tag{48}$$

We first execute Algorithm 1 to augment the state space capturing $\phi(t,x) \le 0$. This necessitates the introduction of a slack variable, $z(t)$, and for us to compute the ϵ-NRD of the constraint function as in Equation (36):

$$\begin{aligned}
\frac{d^0}{dt^0}\left(\phi(t,x(t)) + \frac{1}{2}z(t)^2\right) &= x_1(t) - 8(t-0.5)^2 + 0.5 + \frac{1}{2}z(t)^2\\
\frac{d}{dt}\left(\phi(t,x(t)) + \frac{1}{2}z(t)^2\right) &= x_2(t) - 16(t-0.5) + z(t)z^{(1)}(t)\\
\frac{d^2}{dt^2}\left(\phi(t,x(t)) + \frac{1}{2}z(t)^2\right) &= u(t) - x_2(t) - 16 + z^{(1)}(t)^2 + z(t)z^{(2)}(t).
\end{aligned} \tag{49}$$

It is clear from Equation (49) that the relative degree (Definition 1) and the ϵ-NRD (Definition 4) of the constraint function both correspond to $\rho = 2$ when $0 < \epsilon_1 < 1$. Define our augmented states as $\tilde{z}(t) := [z(t), z^{(1)}(t)]^\top$ and pseudo input as $w(t) = z^{(2)}(t)$, we can set the RHS of Equation (49) to zero to derive the algebraic relationship between the original and augmented states that is also given for the general case in Equations (16) as well as the relationship between the original system input and augmented input given in Equation (13):

$$\begin{aligned}
\tilde{z}_1 &= \sqrt{16(t-0.5)^2 - 2x_1 - 1},\\
\tilde{z}_2 &= \frac{16t - x_2 - 8}{\tilde{z}_1} = \frac{16t - x_2 - 8}{\sqrt{16(t-0.5)^2 - 2x_1 - 1}},
\end{aligned} \tag{50}$$

$$u = x_2 + 16 - \tilde{z}_2^2 - \tilde{z}_1 w. \tag{51}$$

We next augment the state space to include $\tilde{z}$, find the initial conditions of the slack variables using Equations (50) and introduce integral controller structure ($w = s_\beta(\xi)$ with $\dot{\xi}(t) = \tilde{w}(t)$) to derive a system of the same form as Σ_I in Equation (21) with state space $x_I := [x_1, x_2, \tilde{z}_1, \tilde{z}_2, \xi]^\top$:

$$\dot{x}_I(t) = \begin{bmatrix} x_2(t)\\ 16 - \tilde{z}_2(t)^2 - \tilde{z}_1(t)s_\beta(\xi(t))\\ \tilde{z}_2(t)\\ s_\beta(\xi(t))\\ 0\end{bmatrix} + \begin{bmatrix}0\\0\\0\\0\\1\end{bmatrix}\tilde{w}(t), \tag{52}$$

$$\begin{aligned}
x_I(0) &= \left[0,\ -1,\ \sqrt{3},\ -\tfrac{7}{3}\sqrt{3},\ \log\left(\tfrac{3\beta+2\sqrt{3}}{3\beta-2\sqrt{3}}\right)\right]^\top\\
y(t) &= x_1(t) \text{ with } y_r(t) = 0.
\end{aligned} \tag{53}$$

We now execute Algorithm 2 to compute an FBL controller for the integral augmented system given in Equation (52). To do this, we first take time derivatives of the system output until the input first appears as in Equation (43):

$$\frac{d^3}{dt^3}y(t) = -3\tilde{z}_2(t)s_\beta(\xi(t)) - \tilde{z}_1(t)s'_\beta(\xi(t))\tilde{w}(t), \tag{54}$$

where $s'_\beta(\xi(t)) := \frac{\partial}{\partial\xi}s_{\beta,i}(\xi(t)) = \frac{2\beta e^{-\xi(t)}}{(e^{-\xi(t)}+1)^2}$.

Hence, the ϵ-NRD of the system output is 3 in the region $\{(t, x_I) \in [t_0,\infty)\times\mathbb{R}^5 : |\tilde{z}_1 s'_\beta(\xi)| > \epsilon_2\}$. The stacked Lie derivatives from Equation (45) can also be stored as follows, $\gamma = [0, 0, |\tilde{z}_1 s'_\beta(\xi)|]^\top$. While $|\tilde{z}_1 s'_\beta(\xi)| > \epsilon_2$ the FBL controller is given by

$$\tilde{w}(t, x_I) = -\frac{1}{\tilde{z}_1 s'_\beta(\xi)}(3\tilde{z}_1 s'_\beta(\xi) + \nu(t, x_I)), \tag{55}$$

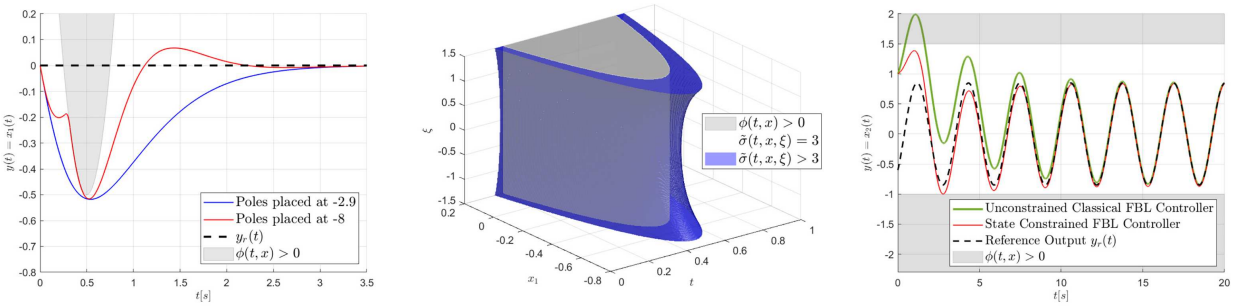
where ν is the virtual input designed to stabilise the linearised system. Selecting poles $\lambda \in (-\infty, 0)^3$, the gains of the linearised system are determined by ensuring the matrix given in Equation (34) has eigenvalues equal to the selected poles. For this example this is equivalent to finding $\lambda^3 + K_3\lambda^2 + K_2\lambda + K_1 = (\lambda - \lambda_1)(\lambda - \lambda_2)(\lambda - \lambda_3)$ leading to gains of

$$\begin{aligned}
K_1 &= -\lambda_1\lambda_2\lambda_3\\
K_2 &= \lambda_1\lambda_2 + \lambda_1\lambda_3 + \lambda_2\lambda_3\\
K_3 &= -\lambda_1 - \lambda_2 - \lambda_3.
\end{aligned} \tag{56}$$

The virtual input is then given by $\nu(t, x_I) = -K[y, \dot{y}, \ddot{y}]^\top = -[K_1, K_2, K_3][x_1, x_2, 16 - \tilde{z}_2^2 - \tilde{z}_1 s_\beta(\xi)]^\top$.

We now synthesise the controller π by substituting ν into the FBL controller $\tilde{w}$ (55) and eliminating the augmented slack variables using Equations (50), we derive a closed-loop controller π that only depends on t, x and ξ:

$$\begin{aligned}
\pi(t,[x,\xi]) = &\frac{-1}{s'_\beta(\xi(t))\sqrt{16(t-0.5)^2 - 2x_1(t) - 1}}\\
&\times\Bigg(3\sqrt{16(t-0.5)^2 - 2x_1(t) - 1}s'_\beta(\xi(t))\\
&- K\left[x_1(t), x_2(t), 16 - \frac{(16t - x_2(t) - 8)^2}{16(t-0.5)^2 - 2x_1(t) - 1}\right.\\
&\left.-s_\beta(\xi(t))\sqrt{16(t-0.5)^2 - 2x_1(t) - 1}\right]^\top\Bigg),
\end{aligned} \tag{57}$$



(a) (b) (c)

FIGURE 4 | Figures associated with Examples 1 and 2 showing the 3D visualisation of switching regions and produced output trajectories. (a) Output trajectories for Example 1. (b) 3D visualisation of switching regions for Example 1. (c) Output trajectories for Example 2.

where $K \in \mathbb{R}^3$ is given in Equation (56).

We also substitute the Equations (50) into the original system input (51) to remove the augmented slack variables,

$$u(t,[x,\xi]) = x_2 + 16 - \frac{(16t - x_2 - 8)^2}{16(t-0.5)^2 - 2x_1 - 1} - s_\beta(\xi)\sqrt{16(t-0.5)^2 - 2x_1 - 1}, \tag{58}$$

recalling the integral control $w = s_\beta(\xi)$.

Coupling the integral state ξ to the original ODE and substituting the closed-loop controller $\pi(t,[x,\xi])$ (57) and the original system input $u(t,[t,\xi])$ (58),

We eventually derive the following closed-loop implementation ODE, which is sent to Matlab's ODE solver later for solving.

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{\xi}(t) \end{bmatrix} = \begin{bmatrix} x_2(t) \\ -x_2(t) \\ \pi(t,[x(t),\xi(t)]) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} u(t,[x,\xi]).$$

Note that the controller π given in Equation (58) is only valid when the switching condition given in Equation (47) is not triggered. For this system the calculated Lie derivatives were $\gamma = [0, 0, |\tilde{z}_1 s'_\beta(\xi)|]$. Therefore, using the algebraic relationship between the augmented states and the original states given in Equations (50), the switching is activated for this particular problem whenever the trajectory satisfies

$$\left|\tilde{z}_1 s'_\beta(\xi)\right| = \left|s'_\beta(\xi)\sqrt{16(t-0.5)^2 - 2x_1 - 1}\right| \le \epsilon_2. \tag{59}$$

When the trajectory enters the switching region (states satisfying Equation 59), the ϵ-NRD increases beyond three. This necessitates synthesising a new FBL controller based on the updated ϵ-NRD at the current state. In this region, input coefficients less than ϵ_i are assumed to be negligible. Thus the output derivative simplifies to $\frac{d^3}{dt^3}y(t) \approx -3\tilde{z}_2(t)s_\beta(\xi(t))$, which now involves no input terms. Thus, higher-order derivatives of the output are required for the input to appear and for an FBL controller to be derived. For brevity, we omit the full derivation, as it mirrors that of Equation (58).

To illustrate the influence of pole placement on controller performance, we performed numerical experiments using Algorithms 1 and 2 to synthesise closed-loop controllers for different pole locations. The resulting simulations of system trajectories are shown in Figure 4a. Each simulation applied the derived controller to the system while monitoring the switching sets from Equation (46), which simplifies to Equation (59) in this example. Whenever a change in the ϵ-NRD was detected, the controller was re-synthesised about the current state trajectory using the same procedure. The parameters were fixed as $\beta = 100$ and $\epsilon = [0.01, \ldots, 0.01]^\top$, with poles assigned either to -2.9 or -8.

Since all poles are negative, both controllers are expected to achieve asymptotic tracking. However, placing poles further left on the real axis typically results in faster convergence at the expense of potentially worse transient behaviour. Figure 4a confirms this: both trajectories respect the constraint and achieve successful asymptotic tracking. The controller with poles at -2.9 (blue curve) applied the controller given in Equation (58) exactly throughout, without requiring any switching. In contrast, the controller with poles at -12 (red curve) achieved faster convergence but approached the constraint boundary, triggering a change in the ϵ-NRD and a subsequent re-synthesis of the controller around the affected state.

In a second numerical experiment of this example, we visualise the regions where the ϵ-NRD changes, the switching regions of our FBL controllers. For this example, the switching region is derived analytically in Equation (59). To make these regions more visible, we select parameters $\beta = 1$ and $\epsilon = [0.2, \ldots, 0.2]^\top$. The resulting plot in Figure 4b shows that the ϵ-NRD remains mostly constant at three, except near the boundary of the state constraint set. This behaviour is consistent with Corollary 1 (found in the appendix), which shows that the relative degree becomes ill-defined on the constraint boundary.

Example 2. Let us consider the following constrained nonlinear tracking problem with a time-varying reference,

Find u such that $\lim_{t\to\infty} ||y(t) - y_r(t)||_2 = 0$ where,

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{bmatrix} = \begin{bmatrix} 10(x_1(t) - x_2(t)) \\ 28x_1(t) - x_2(t) - x_1(t)x_3(t) \\ x_1(t)x_2(t) - \frac{8}{3}x_3(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} u(t), \quad x(0) = \begin{bmatrix} 0.1 \\ 1 \\ 16 \end{bmatrix}$$

$$y(t) = x_2(t) \text{ with } y_r(t) = 0.6(\sin(2t) - \cos(2t))$$

$$\phi(t,x) = [\phi_1(t,x), \phi_2(t,x)]^\top = [-x_2 - 1, x_2 - 1.5]^\top \le 0. \tag{60}$$

The tracking problem given in Equations (60) is a SISO system, however, it has multiple constraints requiring sequential augmentation (see Section 3.2).

Numerical experiments were conducted using an integral parameter of $\beta = 100$, ϵ-NRD threshold parameter of $\epsilon = [0.01, \ldots, 0.01]$ and positioning the poles at -0.3. By initialising the feedback controller to have zero initial value according to Equation (24), the initial condition of the integral states was found to be $\xi(0) = [\xi_1(0), \xi_2(0)]^\top = [0.002, -0.001]^\top$. We execute Algorithms 1 and 2 to synthesise controllers and simulate the closed-loop system. During simulation, we monitor the switching condition in Equation (47). When triggered, we re-execute two algorithms to synthesise a new controller, switch to the updated system, and continue the simulation, again tracking the switching condition.

For comparison, we also applied the classical FBL controller from Section 4.1, using the same poles and parameters but ignoring the state constraint (without augmenting the system at all). Figure 4c shows the trajectories generated by the two controllers. The constrained switching controller (red) respects the feasible region $\phi(t, x) \leq 0$, remaining entirely outside the shaded area, while still achieving asymptotic tracking of the reference signal $y_r(t)$. In contrast, the classical FBL controller (green) exhibits a significant overshoot early on, violating the constraint.

7 | Conclusion

In this paper, we introduced a state augmentation procedure and an integral controller structure modification to incorporate state constraints within the FBL framework. We demonstrated that the augmented system exhibits an ill-defined relative degree at the boundary of the original system's state constraints. To address this challenge and enable the application of FBL, we proposed a switching condition based on a locally valid relative degree, facilitating controller synthesis as we approach the region along the boundary of the state constraint. The proposed method enhances the practical applicability of FBL for systems with state constraints, as illustrated by our numerical experiments. Future work will explore ways to extend our switching framework to systems with multiple inputs and multiple outputs.

Funding

The authors have nothing to report.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

1. X. Lin, D. Xiao, and F. Fang, "Data Discovery of Low Dimensional Fluid Dynamics of Turbulent Flows," arXiv Preprint arXiv:2401.05449 (2024).

2. A. Dolgui, D. Ivanov, S. P. Sethi, and B. Sokolov, "Scheduling in Production, Supply Chain and Industry 4.0 Systems by Optimal Control: Fundamentals, State-Of-The-Art and Applications," *International Journal of Production Research* 57, no. 2 (2019): 411–432.

3. B. O. Koopman, "Hamiltonian Systems and Transformation in Hilbert Space," *Proceedings of the National Academy of Sciences* 17, no. 5 (1931): 315–318.

4. T. Carleman, "Application de la théorie des équations intégrales linéaires aux systèmes d'équations différentielles non linéaires," (1932).

5. R. W. Brockett, "Feedback Invariants for Nonlinear Systems," *IFAC Proceedings Volumes* 11, no. 1 (1978): 1115–1120.

6. A. J. Krener, "A Decomposition Theory for Differentiable Systems," *SIAM Journal on Control and Optimization* 15, no. 5 (1977): 813–829.

7. H. K. Khalil, *Control of Nonlinear Systems* (Prentice Hall, 2002).

8. G. Looye, "Design of Robust Autopilot Control Laws with Nonlinear Dynamic Inversion," (2001), https://api.semanticscholar.org/CorpusID:109344906.

9. C. Miller, "Nonlinear Dynamic Inversion Baseline Control Law: Flight-Test Results for the Full-Scale Advanced Systems Testbed f/a-18 Airplane," in *AIAA Guidance, Navigation, and Control Conference*, 6468, (2011).

10. H. K. Khalil and J. W. Grizzle, *Nonlinear Systems*, vol. 3 (Prentice hall, 2002).

11. M. W. Hirsch, *Differential Topology* (Springer Science & Business Media, 2012).

12. C. Schumacher, P. Khargonekar, and N. McClamroch, "Stability Analysis of Dynamic Inversion Controllers Using Time-Scale Separation," in *Guidance, Navigation, and Control Conference and Exhibit*, 4322, (1998).

13. J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, Advances in Design and Control, Second ed. (Society for Industrial and Applied Mathematics, 2010).

14. S. Liu, Y. Fan, and M. A. Belabbas, "Geometric Motion Planning for Affine Control Systems With Indefinite Boundary Conditions and Free Terminal Time," arXiv Preprint arXiv:2001.04540 (2020).

15. M. Jones, Y. Nie, and M. M. Peet, "Model Predictive Bang-Bang Controller Synthesis via Approximate Value Functions," arXiv Preprint arXiv:2402.08148 (2024).

16. N. Wen, Z. Liu, W. Wang, and S. Wang, "Feedback Linearization Control for Uncertain Nonlinear Systems via Generative Adversarial Networks," *ISA Transactions* 146 (2024): 555–566.

17. J. Deng, V. M. Becerra, and R. Stobart, "Input Constraints Handling in an Mpc/Feedback Linearization Scheme," *International Journal of Applied Mathematics and Computer Science* 19, no. 2 (2009): 219–232.

18. Y. S. Chou and W. Wu, "Robust Controller Design for Uncertain Nonlinear Systems via Feedback Linearization," *Chemical Engineering Science* 50, no. 9 (1995): 1429–1439.

19. J. Gonzalez-Trejo, J. A. Ramirez, and G. Fernandez, "Robust Control With Uncertainty Estimation for Feedback Linearizable Systems: Application to Control of Distillation Columns," *Journal of Process Control* 9, no. 3 (1999): 221–231.

20. K. Shojaei, A. M. Shahri, and A. Tarakameh, "Adaptive Feedback Linearizing Control of Nonholonomic Wheeled Mobile Robots in Presence of Parametric and Nonparametric Uncertainties," *Robotics and Computer-Integrated Manufacturing* 27, no. 1 (2011): 194–204.

21. D. Simon, J. Löfberg, and T. Glad, "Nonlinear Model Predictive Control Using Feedback Linearization and Local Inner Convex Constraint Approximations," in *2013 European Control Conference (ECC)*, 2056–2061. (IEEE, 2013).

22. G. C. Konstantopoulos and C. Bechlioulis, "Resilient Integral Control for Regulating Systems With Convex Input Constraints," in *2023 62nd IEEE Conference on Decision and Control (CDC)*, 2072–2077. (IEEE, 2023).

23. E. N. Johnson and A. J. Calise, "Pseudo-Control Hedging: A New Method for Adaptive Control," in *Advances in Navigation Guidance and Control Technology Workshop*, Alabama, USA, 1–2, (2000).

24. E. Enenakpogbe, J. F. Whidborne, and L. Lu, "Control Allocation Problem Transformation Approaches for Over-Actuated Vectored Thrust Vtols," *Aerospace Science and Technology* 161 (2025): 110145.

25. Q. Gong, W. Kang, and I. M. Ross, "A Pseudospectral Method for the Optimal Control of Constrained Feedback Linearizable Systems," *IEEE Transactions on Automatic Control* 51, no. 7 (2006): 1115–1129.

26. E. van Oort, Q. Chu, and J. Mulder, "Robust Model Predictive Control of A Feedback Linearized f-16/Matv Aircraft Model," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 6318, (2006).

27. Y. V. Pant, H. Abbas, and R. Mangharam, "Robust Model Predictive Control for Non-Linear Systems With Input and State Constraints via Feedback Linearization," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, 5694–5699. (IEEE, 2016).

28. A. Ruiz, D. Rotondo, and B. Morcego, "Design of Shifting State-Feedback Controllers for Constrained Feedback Linearized Systems: Application to Quadrotor Attitude Control," *International Journal of Robust and Nonlinear Control* 34, no. 4 (2024): 2614–2638.

29. F. Schnelle and P. Eberhard, "Constraint Mapping in a Feedback Linearization/Mpc Scheme for Trajectory Tracking of Underactuated Multibody Systems," *IFAC-PapersOnLine* 48, no. 23 (2015): 446–451.

30. G. C. Konstantopoulos and P. R. Baldivieso-Monasterios, "State-Limiting Pid Controller for a Class of Nonlinear Systems With Constant Uncertainties," *International Journal of Robust and Nonlinear Control* 30, no. 5 (2020): 1770–1787.

31. D. Jacobson and M. Lele, "A Transformation Technique for Optimal Control Problems With a State Variable Inequality Constraint," *IEEE Transactions on Automatic Control* 14, no. 5 (1969): 457–464.

32. C. Tomlin and S. S. Sastry, "Switching Through Singularities," *Systems & Control Letters* 35, no. 3 (1998): 145–154.

33. W. H. Chen and D. J. Ballance, "On a Switching Control Scheme for Nonlinear Systems With Ill-Defined Relative Degree," *Systems & Control Letters* 47, no. 2 (2002): 159–166.

34. M. Jones and M. M. Peet, "Solving Dynamic Programming With Supremum Terms in the Objective and Application to Optimal Battery Scheduling for Electricity Consumers Subject to Demand Charges," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 1323–1329. (IEEE, 2017).

35. M. Jones and M. M. Peet, "Extensions of the Dynamic Programming Framework: Battery Scheduling, Demand Charges, and Renewable Integration," *IEEE Transactions on Automatic Control* 66, no. 4 (2020): 1602–1617.

36. S. Palanki and C. Kravaris, "Controller Synthesis for Time-Varying Systems by Input/Output Linearization," *Computers & Chemical Engineering* 21, no. 8 (1997): 891–903.

37. A. Isidori, *Nonlinear Control Systems: An Introduction* (Springer, 1985).

38. I. Ha and E. Gilbert, "Robust Tracking in Nonlinear Systems," *IEEE Transactions on Automatic Control* 32, no. 9 (1987): 763–771.

39. A. Isidori, "The Zero Dynamics of a Nonlinear System: From the Origin to the Latest Progresses of a Long Successful Story," *European Journal of Control* 19, no. 5 (2013): 369–378.

40. P. J. Olver, *Applications of Lie Groups to Differential Equations*, vol. 107 (Springer Science & Business Media, 1993).

41. G. B. Folland, *Real Analysis: Modern Techniques and Their Applications* (John Wiley & Sons, 1999).

42. R. L. Williams and D. A. Lawrence, *Linear State-Space Control Systems* (John Wiley & Sons, 2007).

Appendix A

This appendix provides the technical analysis supporting the discussion in Section 4.2.

Ill-Defined Relative Degree in Augmented Systems Along Constraint Boundaries: For the augmented dynamics, Σ_A, defined by Equation (14), with state space $x_A := \begin{bmatrix} x \\ \tilde{z} \end{bmatrix} \in \mathbb{R}^{n_A}$ from Equation (12), we show in the following proposition that the matrix $\Gamma_{g_A}(t, x_A)$ is not invertible whenever $z_i = 0$ for some $i \in \{1, \dots r\}$, recalling z_i is a constraint slack variable defined in Equation (8). This result can be interpreted as $\Gamma_{g_A}(t, x_A)$ is not invertible whenever the original (non-augmented) system's trajectory enters the boundary of the state constraint, that is $Z(t) := -\phi(t, x(t)) = 0$. This, in turn, shows that the definition of Relative Degree (Definition 1) is ill-defined for the augmented dynamics Σ_A, that is $L_{g_A} L_{f_A}^{\sigma_k - 1} h_k(t, x_A)$ may be non-zero in some regions of the state space but zero in others.

Proposition 1 (FBL is Incompatible with Augmentation). *Consider the augmented system Σ_A defined in Equation (14) with state space $x_A := \begin{bmatrix} x \\ \tilde{z} \end{bmatrix} \in \mathbb{R}^{n_A}$ from Equation (12). Suppose there exists $\sigma \in \mathbb{N}^m$ such that*

$$L_{g_A} L_{f_A}^{i-1} h_k(t_0, x_A(t_0)) \equiv 0 \text{ for all } 1 \le k \le m \text{ and } 1 \le i \le \sigma_k - 1.$$
$$L_{g_A} L_{f_A}^{\sigma_k - 1} h_k(t_0, x_A(t_0)) \neq 0 \text{ for all } 1 \le k \le m. \tag{A1}$$

Then the matrix $\Gamma_{g_A}(t, x_A) \in \mathbb{R}^{m \times m}$ defined as

$$\Gamma_{g_A}(t, x_A) := [L_{g_A} L_{f_A}^{\sigma_1 - 1} h_1(t, x_A)^\top, \dots, L_{g_A} L_{f_A}^{\sigma_m - 1} h_m(t, x_A)^\top]^\top$$

does not have full column rank when $z_i = 0$ for any $i \in \{1, \dots r\}$.

Proof. For any $l \in \{1, \dots, m\}$ we first prove by induction that $L_{f_A}^i h_l(t, x_A)$ is independent of the augmentation/slack variables $\tilde{z} := [z_1, \dots, z_1^{(\rho_1 - 1)}, \dots, z_r, \dots, z_r^{(\rho_r - 1)}]^\top$ for $i \in \{0, \dots \sigma_l - 1\}$.

For the case $i = 0$ we have $L_{f_A}^0 h_l(t, x_A) = h_l(t, x)$ and therefore is trivially independent of the augmentation variables. Assuming that this holds for $i - 1$ we next prove that it holds for i. For any $j \in \{1, \dots, r\}$ and $k \in \{0, 1, \dots, \rho_l - 1\}$ it follows:

$$\begin{aligned}
\frac{\partial L_{f_A}^i h_l(t, x_A)}{\partial z_j^{(k)}} &= \frac{\partial}{\partial z_j^{(k)}} \left(\frac{\partial L_{f_A}^{i-1} h_l(t, x_A)}{\partial x_A} f_A(t, x_A) + \frac{\partial L_{f_A}^{i-1} h_l(t, x_A)}{\partial t} \right) \\
&= \frac{\partial}{\partial z_j^{(k)}} \left(\begin{bmatrix} \frac{\partial L_{f_A}^{i-1} h_l(t, x_A)}{\partial x}, & 0_{1 \times \sum_{k=1}^r \rho_k} \end{bmatrix} f_A(t, x_A) \right) \\
&= \frac{\partial}{\partial z_j^{(k)}} \left(\frac{\partial L_{f_A}^{i-1} h_l(t, x_A)}{\partial x} (f(t, x) - g(t, x) \cdot \Omega_g^{-1}(t, x) \Omega_f(t, x)) \right) = 0,
\end{aligned}$$

where the second equality holds in the above equation since $L_{f_A}^{i-1} h_l(t, x_A)$ is independent of the augmented variables, $\tilde{z}$, by the induction hypothesis. The third equality follows by the definition of f_A given in Equation (15) and the fourth and final equality follows since all terms inside of the partial derivative are independent of $z_j^{(k)}$ for $j \in \{1, \dots, r\}$ and $k \in \{0, 1, \dots, \rho_l - 1\}$.

Now, using the definition of the Lie derivative given in Equation (2) and the fact that $L_{f_A}^i h_l(t, x_A)$ is independent of the augmentation variables, we get that,

$$L_{g_A} L_{f_A}^{\sigma_l-1} h_l(t,x) = \frac{\partial L_{f_A}^{\sigma_l-1} h_l(t,x_A)}{\partial x_A} g_A(t,x_A)$$
$$= \left[\frac{\partial L_{f_A}^{\sigma_l-1} h_l(t,x)}{\partial x},\ 0_{1\times\sum_{k=1}^{r}\rho_k} \right] g_A(t,x_A)$$
$$= -\frac{\partial L_{f_A}^{\sigma_l-1} h_l(t,x)}{\partial x_A} g(t,x)\Omega_g^{-1}(t,x)D(z) \in \mathbb{R}^{1\times r}, \quad \text{(A2)}$$

recalling $D(z) := \mathrm{diag}(z_1, \ldots, z_r) \in \mathbb{R}^{r\times r}$.

If $z_k = 0$ for some $k \in \{1, \ldots r\}$ then $D(z)$ contains a column and row of zeros. From basic linear algebra, if A and B are any matrices and A has a column of zeros, then BA also has a column of zeros. Hence, it follows $L_{g_A} L_{f_A}^{\sigma_l-1} h_l(t,x_A)$ is a row vector with a zero element at component $k \in \mathbb{N}$. Hence, $\Gamma_{g_A}(t,x_A) := [L_{g_A} L_{f_A}^{\sigma_1-1} h_1(t,x_A)^\top, \ldots, L_{g_A} L_{f_A}^{\sigma_m-1} h_m(t,x_A)^\top]^\top$ has a column of zeros and therefore cannot have full column rank. □

In the next corollary, we extend Proposition 1 by proving that the relative degree of the integral augmented system Σ_I in Equation (21) is one greater than that of the augmented system Σ_A in Equation (14), whenever the relative degree exists. Furthermore, we demonstrate that Assumption 2 fails along the boundary of the state constraint for the integral augmented system, mirroring the result established for the augmented system in Proposition 1.

Corollary 1. *Suppose there exists $\sigma \in \mathbb{N}^m$ such that Equation (A1) holds. Then,*

$$L_{g_I} L_{f_I}^{i-1} h_k(t_0, x_I(t_0)) \equiv 0 \text{ for all } 1 \le k \le m \text{ and } 1 \le i \le \sigma_k.$$
$$L_{g_I} L_{f_I}^{\sigma_k} h_k(t_0, x_I(t_0)) \neq 0 \text{ for all } 1 \le k \le m, \quad \text{(A3)}$$

where f_I and g_I are defined in the integral augmented system given in Equation (21). Moreover, the matrix $\Gamma_{g_I}(t,x_I) := [L_{g_I} L_{f_I}^{\sigma_1} h_1(t,x_I)^\top, \ldots, L_{g_I} L_{f_I}^{\sigma_m} h_m(t,x_I)^\top]^\top \in \mathbb{R}^{m\times m}$ does not have full column rank when $z_i = 0$ for any $i \in \{1, \ldots r\}$.

Proof. We first show by induction that for all $i \in \{0, \ldots, \sigma_l - 1\}$ and $l \in \{1, \ldots m\}$:

$$L_{f_I}^i h_l(t,x_I) \equiv L_{f_A}^i h_l(t,x_A) \quad \text{(A4)}$$

For the case $i = 0$ we have that $L_{f_I}^0 h_l(t,x_I) = h_l(t,x) = L_{f_A}^0 h_l(t,x_A)$ and hence Equation (A4) is clearly true for $i = 0$. Now, assuming Equation (A4) to be true for $i-1$ we show it to be true for $1 \le i \le \sigma_l - 1$ in the following:

$$L_{f_I}^i h_l(t,x_I) = \frac{\partial L_{f_I}^{i-1} h_l(t,x_I)}{\partial x_I} f_I(t,x_I) + \frac{\partial L_{f_I}^{i-1} h_l(t,x_I)}{\partial t}$$
$$= \left[\frac{\partial L_{f_I}^{i-1} h_l(t,x_I)}{\partial x_A},\ \frac{\partial L_{f_I}^{i-1} h_l(t,x_I)}{\partial \xi} \right] \begin{bmatrix} f_A(t,x_A) + g_A(t,x_A)s_\beta(\xi) \\ 0_{r\times 1} \end{bmatrix} + \frac{\partial L_{f_I}^{i-1} h_l(t,x_I)}{\partial t}$$
$$= \frac{\partial L_{f_A}^{i-1} h_l(t,x_A)}{\partial x_A} f_A(t,x_A) + \frac{\partial L_{f_A}^{i-1} h_l(t,x_A)}{\partial x_A} g_A(t,x_A)s_\beta(\xi) + \frac{\partial L_{f_A}^{i-1} h_l(t,x_A)}{\partial t}$$
$$= L_{f_A}^i h_l(t,x_A) + L_{g_A} L_{f_A}^{i-1} h_l(t,x_A)s_\beta(\xi) = L_{f_A}^i h_l(t,x_A). \quad \text{(A5)}$$

where the first equality of Equation (A5) follows from the definition of the Lie derivative given in Equation (2). The second equality follows from the definition of x_I in Equation (20) and f_I in Equation (21). The third equality follows from the induction hypothesis that $L_{f_I}^{i-1} h_l(t,x_I) \equiv L_{f_A}^{i-1} h_l(t,x_A)$. The fourth equality follows from the definition of the Lie derivative given in Equation (2). Finally, the fifth equality follows from Equation (A1), that is $L_{g_A} L_{f_A}^{i-1} h_l(t,x_A) = 0$. Hence, Equation (A4) follows by induction.

Moreover, by a similar argument to Equation (A5), it follows that

$$L_{f_I}^{\sigma_l} h_l(t,x_I) = L_{f_A}^{\sigma_l-1} h_l(t,x_A) + L_{g_A} L_{f_A}^{\sigma_l-1} h_l(t,x_A)s_\beta(\xi). \quad \text{(A6)}$$

Now,

$$L_{g_I} L_{f_I}^{\sigma_l} h_l(t,x_I) = \frac{\partial L_{f_I}^{\sigma_l} h_l(t,x_I)}{\partial x_I} g_I(t,x_I)$$
$$= \frac{\partial}{\partial x_I}\left(L_{f_A}^{\sigma_l} h_l(t,x_A) + L_{g_A} L_{f_A}^{\sigma_l-1} h_l(t,x_A)s_\beta(\xi) \right) \begin{bmatrix} 0_{n_A\times m} \\ 1_{m\times m} \end{bmatrix} = L_{g_A} L_{f_A}^{\sigma_l-1} h_l(t,x_A)\mathrm{diag}\left(\frac{\partial}{\partial \xi} s_\beta(\xi) \right)$$
$$= -\frac{\partial L_{f_A}^{\sigma_l-1} h_l(t,x)}{\partial x_A} g(t,x)\Omega_g^{-1}(t,x) D(z)\mathrm{diag}\left(\frac{\partial}{\partial \xi} s_\beta(\xi) \right). \quad \text{(A7)}$$

where the first equality of Equation (A7) follows from the definition of the Lie derivative given in Equation (2). The second equality follows by applying Equation (A6) and the definition of g_I given in Equation (21). The third equality follows from the definition of x_I given in Equation (20). The fourth equality follows from Equation (A2) of Proposition 1.

Clearly, by Equation (A7) we have that $L_{g_I} L_{f_I}^{\sigma_l} h_l(t,x_I) = L_{g_A} L_{f_A}^{\sigma_l-1} h_l(t,x_A)\mathrm{diag}\left(\frac{\partial}{\partial \xi} s_\beta(\xi) \right) \neq 0$ through application of Equation (A1) and the fact that the derivative of the sigmoid function, s_β, is strictly positive everywhere. This shows Equation (A3).

Finally, it is clear that Γ_{g_I} does not have full column rank whenever $z_i = 0$ by the same argument used in Proposition 1 due to the $D(z)$ term given in Equation (A7) □

Ill-Defined Relative Degree in Integral Control Structure Dynamics: As established in Corollary 1, $\Gamma_{g_I}(t,x_I)$ is non-invertible along the state constraint boundary. The following lemma further shows that its invertibility also depends on the integral parameter β, becoming singular as $\beta \to 0$.

Lemma 2. *Consider the integral augmented system Σ_I defined in Equation (21). Suppose there exists $\sigma \in \mathbb{N}^m$ such that Equation (A1) holds. Then, $\Gamma_{g_I}(t,x_I)$ becomes singular as $\beta \to 0$.*

Proof. For system Σ_I (21), Equation (A7) shows that each row of Γ_{g_I}, specifically $L_{g_I} L_{f_I}^{\sigma_l} h_l(t,x_I)$ for $1 \le l \le m$, contains the term $\mathrm{diag}\left(\frac{\partial}{\partial \xi} s_\beta(\xi) \right)$, where s_β is defined in Equation (19). Since

$$\frac{\partial}{\partial \xi} s_{\beta,i}(\xi) = \frac{2\beta e^{-\xi}}{(e^{-\xi}+1)^2},$$

each derivative carries a factor of $\beta > 0$, implying

$$\Gamma_{g_I}(t,x_I) = \beta\tilde{\Gamma}_{g_I}(t,x_I),$$

where $\tilde{\Gamma}_{g_I}$ is independent of β. Thus, as $\beta \to 0$, $\Gamma_{g_I}(t,x_I)$ becomes singular. □