Asian Engineering Review
Vol. 13, No. 1, 1-10, 2026
ISSN(E) 2409-6725/ ISSN(P) 2518-0150
DOI: 10.20448/aer.v13i1.8827

Artificial intelligence–driven stability-constrained optimal control of nonlinear artificial cell systems

Lakshmi N Sridhar

Chemical Engineering Department, University of Puerto Rico, Mayaguez, Puerto Rico.
Email: lakshmin.sridhar@upr.edu

Abstract

In nonlinear biochemical systems, complex dynamic behaviors, such as multistability and oscillations, are often induced by internal feedback mechanisms. These behaviors pose significant challenges for optimal control problems, as traditional optimization approaches that focus solely on performance criteria can lead to instability or oscillations. In this work, a novel stability-constrained optimal control approach is developed for a high-dimensional artificial cell system, leveraging bifurcation analysis, machine learning, and dynamic optimization. The mechanistic model accounts for the coupled dynamics of gene expression, metabolic reactions, energy balance, and cell growth. As a result, the mechanistic model is a nonlinear dynamic system with rich dynamics. Bifurcation analysis is used to identify regions of stability loss, including Hopf bifurcation points. A dataset is constructed from the states of the dynamic system and the corresponding dominant eigenvalue information. A neural network surrogate approximates the stability behavior based on the dynamic system's states. The neural network is integrated into a dynamic optimization problem using the Pyomo optimization library. As a result, stability information is included in the objective function of the optimization problem. A smooth penalty approach is used to define the objective function based on the dominant eigenvalue information. The smooth penalty approach is used to avoid non-differentiability in the objective function. The optimization problem is solved using a direct transcription approach with collocation. The results clearly indicate that the proposed framework can identify control trajectories that maximize product formation while maintaining system stability. In contrast, optimization problems that are not constrained for stability considerations lead to degenerate or physically unrealistic solutions. In addition, this study clearly indicates that proper model formulation, including consideration of metabolite decay, is important for achieving robust solutions.

Keywords: Artificial intelligence, Bifurcation analysis, Nonlinear dynamical systems, Optimal control, Stability-constrained optimization.

Citation | Sridhar, L. N. (2026). Artificial intelligence–driven stability-constrained optimal control of nonlinear artificial cell systems. *Asian Engineering Review*, *13*(1), 1–10. 10.20448/aer.v13i1.8827
History:
Received: 22 April 2026
Revised: 26 May 2026
Accepted: 5 June 2026
Published: 18 June 2026

Publisher: Asian Online Journal Publishing Group

Funding: This study received no specific financial support.
Institutional Review Board Statement: Not applicable.
Transparency: The author confirms that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.
Competing Interests: The author declares that there are no conflicts of interest regarding the publication of this paper.

Contents

1. Introduction **2**
2. Literature Review **2**
3. Results **6**
4. Discussion of Results **8**
5. Conclusions **9**
References **9**

Contribution of this paper to the literature
The main contribution of this paper is the integration of bifurcation analysis and optimal control for dynamic problems in artificial cells. The study systematically links nonlinear dynamical behavior, stability transitions, and control design, enabling the suppression of oscillatory dynamics while providing deeper insight into parameter-dependent instabilities and performance optimization.

1. Introduction

Nonlinear dynamic systems in biochemical and cellular networks often exhibit complex behaviors such as multistability, oscillations, and bifurcations. Among these behaviors, the Hopf bifurcation is significant, as it marks a transition from steady-state stable operation to sustained oscillatory operation. These oscillations in biochemical engineering and synthetic biology may reduce product yields and process stability. Therefore, it is crucial to directly incorporate stability into optimal control problems.

The dynamic system under consideration is based on synthetic gene regulatory networks that exhibit oscillatory behavior via cyclic repression mechanisms. This work is based on a seminal work by Elowitz and Leibler [1], which presented a foundational model of a repressilator, a coupled system of transcription and translation in which proteins act as repressors of gene expression cyclically.

The traditional method for solving stability-constrained optimization problems relies on repeated eigenvalue analysis of the system Jacobian, which is computationally expensive and difficult to implement in large-scale dynamic optimization problems. To overcome this difficulty, this work presents a novel artificial intelligence-based framework in which a neural network is trained on bifurcation data to mimic proximity to a Hopf instability, which is then symbolically embedded into a dynamic optimization problem.

The soft Hopf constraint is formulated as a smooth penalty term in the objective function via a differentiable approximation, compatible with gradient-based optimization algorithms. The optimization problem is discretized using orthogonal collocation, and a solution is obtained using a nonlinear programming solver that maximizes product formation while avoiding unstable oscillatory regimes, thereby providing a framework for stability-constrained optimal control of nonlinear biochemical systems.

2. Literature Review

The field of systems biology has been motivated by the need to understand complex biological systems through quantitative and computational approaches. Early work by Varma and Palsson [2] laid the foundation for the field by introducing flux balance analysis (FBA), a constraint-based modeling approach. Using this method, it is possible to predict flux distribution in complex cellular systems. Later, this work was extended to include the effects of gene expression and regulation, as illustrated by Covert, et al. [3] which established the link between metabolic and gene expression regulation.

The field of systems biology was formally introduced by Kitano [4], emphasizing the integration of biological processes across different scales. The mathematical approaches, as introduced by Murray [5] and Alon [6], formed the backbone of modeling complex biological processes. Experimental advances, including the introduction of synthetic gene oscillators by Elowitz and Leibler [1], were later improved by Stricker, et al. [7] and Danino, et al. [8] highlighting the importance of nonlinear effects such as oscillations in regulating complex biological processes.

The mid-2000s saw significant progress in integrating high-throughput data with computational models. For example, Qian and Beard [9] investigated the feasibility of conducting system-level analyses of metabolic and regulatory networks. Similarly, Antia and de Pillis [10] expanded the modeling framework to immunological systems. Further advanced genome-scale kinetic modeling by overcoming some drawbacks of stoichiometric modeling. During the same period, Min Lee, et al. [11] emphasized the need to integrate signaling, metabolic, and regulatory networks into unified dynamic frameworks.

The idea of "whole-cell modeling" gained prominence in the scientific community through the work of Karr, et al. [12] who proposed predicting phenotype from the genotype level. Their work built upon constraint-based modeling. O'Brien, et al. [13] expanded the modeling framework by incorporating gene expression, thereby improving prediction accuracy.

Recent years have focused on multi-scale integration and computational scalability. Tools such as COBRAme [14] and resource allocation modeling have enabled the integration of metabolic, transcriptional, and translational regulation within a single simulation framework. Data-driven integration methods [15] and proteome-level modeling have improved simulation accuracy. Attempts at integrating enzymatic constraints, as in GECKO 2.0 [16], represent significant steps toward more accurate metabolic modeling.

Parallel progress has been made in applying these modeling frameworks to various biomedical and synthetic biology problems. For example, Gonçalves, et al. [17] highlighted the importance of integrating signaling and metabolic pathways. More recently, Razzaq, et al. [18] and Dasgupta, et al. [19] proposed control-theoretic modeling in cancer metabolism. Presented a review of the expanding scope of genome-scale modeling in biotechnology and medicine. Additionally, Choi, et al. [20] explored dynamic modeling in cell-free systems, demonstrating the flexibility of these frameworks for novel synthetic biology applications.

Overall, the literature suggests an evolutionary trend from static constraint-based to dynamic, multi-scale, and data-integrated approaches. Modern systems biology focuses on integrating metabolic, gene regulatory, and signal transduction pathways, facilitated by advances in computational power and machine learning. This provides an excellent basis for current research in stability-constrained optimal control and the study of complex dynamic phenomena such as oscillations and bifurcations.

2.1. Model Equations

For the model equations, the state variables (m1-m5) represent the mRNA concentrations (mol/L), (p1-p5) represent the protein concentrations (mol/L), G is the glucose concentration (mol/L), and P, A, and N represent

the intermediate metabolites (mol/L). W is the waste metabolite (mol/L), and ATP represents the energy currency (mol/L). X is the biomass concentration (mol/L), and Prod is the product concentration (mol/L).

The control variable is G_{ext}, the external glucose concentration (mol/L).

The parameter values are

$$k_{tx1} = k_{tx2} = k_{tx3} = k_{tx4} = k_{tx5} = 1\ (1/time)$$

$$k_{tl1} = k_{tl2} = k_{tl3} = k_{tl4} = k_{tl5} = 2(1/time)$$

$$\delta_m = 0.5\ (1/time); \delta_p = 0.2\ (1/time)$$

$$K1 = K2 = K3 = K4 = K5 = 1\ (mol/L)$$

$$n1 = n2 = n3 = n4 = n5 = 2\ (dimensionless)$$

$$V1 = V2 = V3 = V4 = 2\ (mol/(L{\cdot}time))$$

$$Km1 = Km2 = Km3 = Km4 = 1\ (mol/L)$$

$$\alpha 1 = \alpha 2 = 1;$$

$$\beta 1 = \beta 2 = 1$$

$$\gamma = 0.5\ (1/time)$$

$$k_{in1} = 1\ (1/time); k_{out} = 0.5\ (1/time)$$

$$\mu max = 0.8\ (1/time); K_{ATP} = 1\ (mol/L);$$

$$K_P = 1\ (mol/L); k_{prod} = 0.5(1/time); k_{deg} = 0.1\ (1/time);$$

($k_{tx1} - k_{tx5}; k_{tl1} - k_{tl5}$) represent the transcription and translation rates, δ_m, δ_p mRNA and protein degradation times, K1-K5 represent the Hill constants, and n1-n5 the Hill coefficients. V1-V4 represent the maximum reaction rates, Km1–Km4 are the Michaelis constants. $\alpha 1, \alpha 2; \beta 1, \beta 2;$ are the ATP production and consumption coefficients.

γ is the ATP decay rate. k_{in1}, k_{out} are the glucose inflow rate and the waste outflow rate. μmax is the maximum growth rate. $K_{ATP}; K_P$ are the saturation constant. k_{prod}, k_{deg} are the product formation and product degradation rates. The model equations are

$$\begin{aligned}
f1 &= 1/(1+(p2/K1)^n 1)\\
f2 &= 1/(1+(p3/K2)^n 2)\\
f3 &= 1/(1+(p4/K3)^n 3)\\
f4 &= 1/(1+(p5/K4)^n 4)\\
f5 &= 1/(1+(p1/K5)^n 5)\\
v1 &= \frac{V1G}{(Km1+G)\,p1}\\
v2 &= \frac{V2P}{(Km2+P)\,p2}\\
v3 &= \frac{V3A}{(Km3+A)\,p3}\\
v4 &= \frac{V4N}{(Km4+N)\,p4}
\end{aligned} \tag{1}$$

$$
\begin{aligned}
&\frac{dm1}{dt} = k_{tx1} f1 - \delta_m m1 \\
&\frac{dm2}{dt} = k_{tx2} f2 - \delta_m m2 \\
&\frac{dm3}{dt} = k_{tx3} f3 - \delta_m m3 \\
&\frac{dm4}{dt} = k_{tx4} f4 - \delta_m m4 \\
&\frac{dm5}{dt} = k_{tx5} f5 - \delta_m m5 \\
&\frac{dp1}{dt} = k_{tl1} m1\left(\frac{ATP}{K_{ATP}+ATP}\right) - \delta_p p1 \\
&\frac{dp2}{dt} = k_{tl2} m2\left(\frac{ATP}{K_{ATP}+ATP}\right) - \delta_p p2 \\
&\frac{dp3}{dt} = k_{tl3} m3\left(\frac{ATP}{K_{ATP}+ATP}\right) - \delta_p p3 \\
&\frac{dp4}{dt} = k_{tl4} m4\left(\frac{ATP}{K_{ATP}+ATP}\right) - \delta_p p4 \\
&\frac{dp5}{dt} = k_{tl5} m5\left(\frac{ATP}{K_{ATP}+ATP}\right) - \delta_p p5 \\
&\frac{dG}{dt} = k_{in1}\left(G_{ext} - G\right) - v1 \\
&\frac{dP}{dt} = v1 - v2 - 0.05P \\
&\frac{dA}{dt} = v2 - v3 - 0.05A \\
&\frac{dN}{dt} = v3 - v4 - 0.05N \\
&\frac{dW}{dt} = v4 - k_{out} W \\
&\frac{dATP}{dt} = \alpha 1v1 + \alpha 2v2 - \beta 1v3 - \beta 2v4 - \gamma ATP \\
&mu = mu\max\left(\frac{ATP}{(K_{ATP}+ATP)}\right)\left(\frac{P}{K_P+P}\right) \\
&\frac{dX}{dt} = muX - 0.1X^2 \\
&\frac{dProd}{dt} = k_{prod} PX - k_{\deg} Prod
\end{aligned}
\qquad (2)
$$

2.2. Bifurcation Analysis and Optimal Control

2.2.1. Bifurcation Analysis

Bifurcation calculations are performed using the MATLAB software MATCONT. Bifurcation analysis explains the main causes of multiple steady states and limit cycles. Branch points and limit points cause multiple steady-state solutions, while limit cycles and oscillatory behavior are caused by Hopf bifurcation points. The MATLAB program that effectively locates limit points, branch points, and Hopf bifurcation points is MATCONT. This program was developed and improved by several researchers [21, 22]. It is very effective in identifying limit points (LP), branch points (BP), and Hopf bifurcation points (H) for a system of ordinary differential equations.

$$\frac{dx}{dt} = f(x, \alpha) \qquad (3)$$

$x \in R^n$ where the bifurcation parameter is α . The gradient vector is orthogonal to the tangent, and hence the tangent plane at any point $w = [w_1, w_2, w_3, w_4, \ldots . w_{n+1}]$ must satisfy.

$$Aw = 0 \qquad (4)$$

The matrix A is defined by,

$$A = [\partial f/\partial x \quad |\partial f/\partial \alpha] \qquad (5)$$

The sub-matrix $\partial f/\partial x$ is the Jacobian matrix. For both limit and branch points, the Jacobian matrix $J = (\partial f/\partial x)$ must have a determinant of 0.

At a limit point, the $n+1^{th}$ component of the tangent vector $w_{n+1} = 0$. For a branch point, the matrix $B = \begin{bmatrix} A \\ w^T \end{bmatrix}$ must be singular and have a determinant of 0.

At a Hopf bifurcation point,

$$det(2f_x(x, \alpha)@I_n) = 0 \qquad (6)$$

@ indicates the bialternate product, while I_n is the n-square identity matrix. Hopf bifurcations cause limit cycles and should be eliminated because they make optimization and control tasks very difficult. More details can be found in Kuznetsov [23], Kuznetsov [24] and Govaerts [25] respectively.

2.2.2. Optimal Control

Pyomo.dae, as referenced by Hart, et al. [26] is used for optimal control calculations. Pyomo.DAE extends the Pyomo optimization framework, making it suitable for solving dynamic systems of differential and algebraic equations. It provides a symbolic environment for addressing differential-algebraic equation systems within optimization problems. This capability is crucial in process systems engineering, chemical kinetics, and control systems, where understanding the dynamic response of systems is essential.

At its core, Pyomo.DAE enables users to define time-varying variables, derivatives, and constraints symbolically, which can be easily integrated into a Pyomo model. Users can define continuous sets for time or other variables, which are used to define derivatives over those sets. This symbolic approach allows users to discretize continuous differential-algebraic equation systems using finite difference, collocation, or orthogonal collocation methods, transforming continuous differential equations into algebraic equations solvable with standard solvers. The framework handles both initial-value problems and dynamic optimization problems. In dynamic optimization, Pyomo.DAE facilitates the formulation of time-dependent objective functions and constraints, making it particularly useful in optimal control, energy systems, and chemical process scheduling problems.

One of the major advantages of Pyomo.DAE is that it is compatible with the Pyomo ecosystem. This allows users to leverage existing solver interfaces, variable bounds, nonlinear constraints, and objective functions within a combined static and dynamic modeling framework. Furthermore, the symbolic framework makes it easier to perform model verification, automatic differentiation, and sensitivity analysis. Pyomo.DAE provides a flexible, extensible, and open-source environment for modeling, simulation, and optimization of dynamic systems. By integrating symbolic modeling of DAEs with powerful discretization and optimization capabilities, it offers a unique framework for solving complex time-dependent problems. Its tight integration with Pyomo enables the efficient solution of both simple and complex dynamic optimization problems, making it a cornerstone of modern computational modeling of dynamic systems. In Pyomo.DAE, the differential equations are converted to a nonlinear program (NLP) using the orthogonal collocation method. The NLP is solved using IPOPT [27].

2.3. Formation of Stability Dataset from MATCONT results

A stability dataset was developed based on the results from numerical continuation calculations carried out in MATCONT. The stability dataset consists of rows, each representing a continuation point from an equilibrium branch. The columns in each row consist of the state variables, the bifurcation parameter, and a stability measure. The stability measure is a numerical value derived from the Jacobian matrix. The Jacobian matrix is computed numerically at each equilibrium point. The eigenvalues are then computed automatically using MATLAB. The maximum value of the real part of these eigenvalues is then computed as a scalar stability measure.

The stability measure is computed using "eig_real_max = max(real(eigvals));" in MATLAB. The stability measure is a quantitative metric where negative values indicate locally asymptotically stable equilibria, positive values indicate instability, and a zero crossing indicates a Hopf bifurcation. The stability dataset is then saved as a CSV file. This dataset can be used in subsequent computational calculations to perform classification or regression to identify stability boundaries or approximate bifurcations.

2.4. Neural Network Surrogate for Stability Prediction

Direct embedding of eigenvalue calculations into IPOPT-based optimal control is impractical for several reasons: (i) computing eigenvalues at each time step is computationally expensive; (ii) the mapping from states to the maximum eigenvalue is non-smooth near eigenvalue crossings; and (iii) symbolic differentiation of eigenvalues is challenging.

Prior to neural network training, all input variables were standardized to improve numerical conditioning and training stability. Let x_{raw} denote the vector of state variables and bifurcation parameter obtained from the stability dataset. For each input variable j, the training mean μ_j and training standard deviation σ_j were computed over the training dataset as the arithmetic mean and standard deviation, respectively. The training mean for input variable j is defined as the arithmetic average over all training samples as $\mu_j = \frac{1}{N}\sum_{k=1}^{N} x_j^{(k)}$ and the training standard deviation for the input variable j is defined as: $\sigma_j = \frac{1}{N}(\sum_{k=1}^{N} x_j^{(k)} - \mu_j)^2$.

The standardized inputs were defined as:

$$x_j = \frac{x_{\text{raw},j} - \mu_j}{\sigma_j} \qquad (7)$$

This transformation ensures that each input variable has zero mean and unit variance over the training set, thereby improving neural network conditioning and gradient-based optimization performance.

The vectors μ, σ $\mu$and σ computed during training were stored and embedded identically within the Pyomo optimal control formulation to ensure consistency between neural network training and deployment.

To overcome these limitations, a feedforward neural network is trained to approximate the maximum eigenvalue as a smooth function of the system state and bifurcation parameter. A typical architecture employs the hyperbolic tangent (tanh) as a smooth activation function. If the input vector is denoted by, which are the scaled variables, then the network is defined as:

$$\begin{aligned} z1 &= tanh(W1\,x + b1) \\ z2 &= tanh(W2\,z1 + b2) \\ \lambda\_max\_NN &= W3\,z2 + b3 \end{aligned} \qquad (8)$$

Because tanh is infinitely differentiable, the network is fully smooth, guaranteeing the availability of first and second derivatives required by IPOPT. Here, W1, W2, and W3 are the weights that scale and combine inputs or hidden-layer features, while b1, b2, and b3 are the biases that allow neurons to shift their activation independently of inputs. Without biases, the network output would be constrained to pass through the origin, limiting flexibility.

The hidden-layer outputs z1, z2 represent nonlinear combinations of the inputs and previous-layer features, respectively. Each element of z_1 is a smoothed combination of the original inputs, while each element of z2 encodes more abstract patterns extracted from z1. The final output λ_max_NN provides a smooth approximation of the maximum real eigenvalue, enabling efficient and differentiable stability evaluation within the optimal control problem. Figure 1 shows a chart describing the computational strategy.

The integration into optimal control is done using a soft penalty formulation where we use a smooth max function that converts λ into a smooth, nonnegative penalty that only "activates" when the system is unstable.

$$s_{\max}(\lambda+\varepsilon_1) = smooth - \max(\lambda+\varepsilon_1) = \frac{(\lambda+\varepsilon_1)+\sqrt{(\lambda+\varepsilon_1)^2+\varepsilon_1}}{2} \qquad (9)$$

λ is the neural network's predicted maximum eigenvalue at the current state and parameters, while ε is a small positive safety margin to ensure differentiability. The soft penalty formulation involves the new objective function, where the original objective function $(\sum Prod(t))^2$ is modified to $\sum(Prod(t) + \alpha_{hopf}.s1^2{}_{max}$ α_{Hopf} controls how aggressively instability is penalized and ε_1 prevents numerical issues at exactly λ = 0, slightly shifting the stability boundary.

The latest numerical implementation includes a soft Hopf bifurcation constraint in a complex biochemical dynamic system using a Pyomo optimization problem. The soft Hopf is included using a symbolic neural network (NN) that estimates the local stability of the complex dynamic system for all possible states and control variables. The neural network is constructed with two hidden layers of 8 neurons each, with a hyperbolic tangent activation function with reduced slope (0.1) for solver stability during the calculation of derivatives. The neural network is fed with standardized variables from the biochemical dynamic system (m1-m5, p1-p5, G, P, A, N, W, ATP, X, Prod, Gext), with the output used for the calculation of the objective function. To make the soft Hopf work within the optimization problem, a smooth maximum function is used to make the output continuous. The soft Hopf is included in the objective function as a penalty subtracted from the total product with a small α_{Hopf} parameter. This allows the solver to maximize product formation while avoiding unstable oscillatory dynamics. The scaled tanh activation function, along with smooth_max, is used to ensure numerical stability for relatively large values of the penalty for the occurrence of the Hopf bifurcations, alpha2_hopf, which is set to a maximum of 0.4. The system is discretized using collocation (Lagrange Radau, 20 finite elements, 3 collocation points). This discretization transforms the differential-algebraic system into a nonlinear program that is solvable using IPOPT. The numerical soft Hopf bifurcations approach is a flexible and robust method for implementing stability constraints in the optimal control of nonlinear biochemical networks.

3. Results

The bifurcation analysis revealed a Hopf bifurcation at (m1-m5, p1-p5, G, P, A, N, W, ATP, X, Prod, G_{ext}) values of (0.874361, 0.874361, 0.874361, 0.874361, 0.874361, 1.134630, 1.134630, 1.134630, 1.134630, 1.134630, 0.404879, 0.388198, 0.372557, 0.357861, 1.196117, 0.149117, 0.290305, 0.563480, 1.058868). This is shown in Figures 1a and 1b.

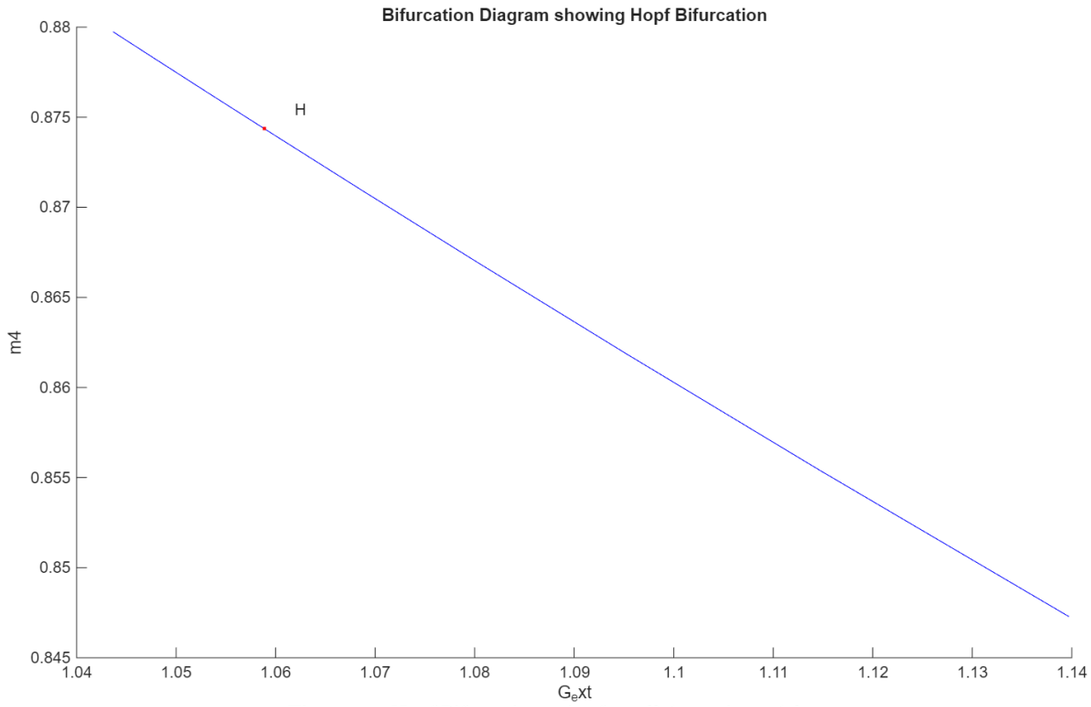

Figure 1a. Hopf Bifurcation point for cell dynamic model.

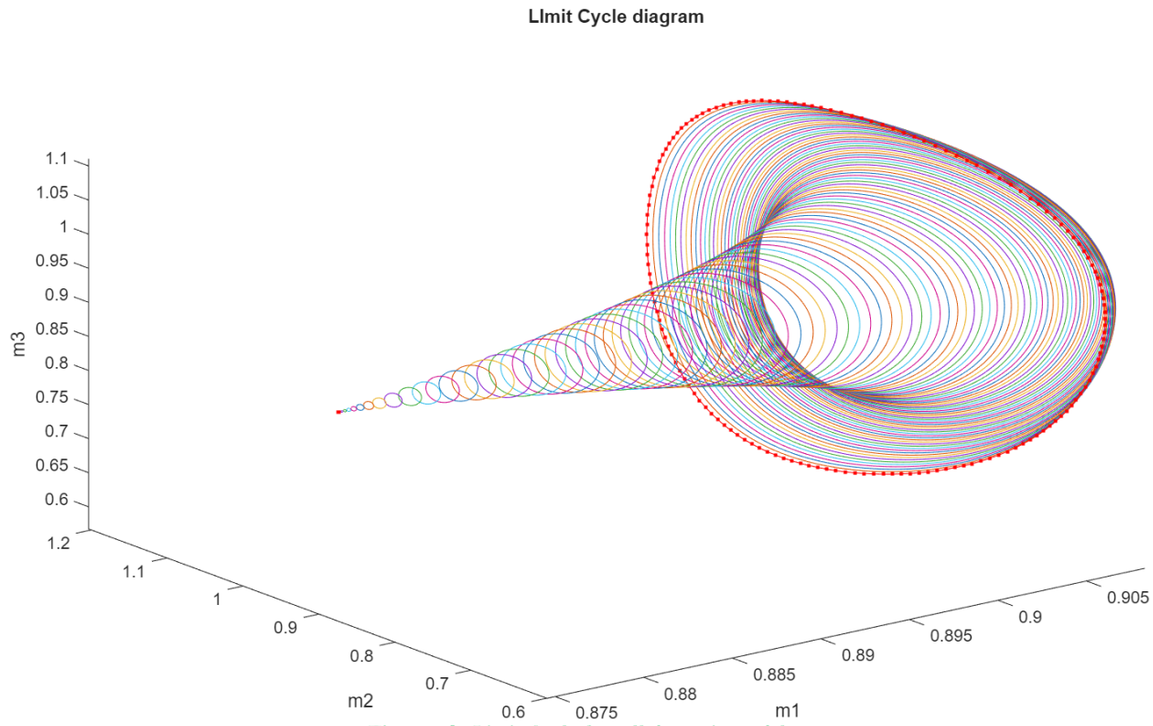

Figure 1b. Limit Cycle for cell dynamic model.

For the optimal control, when no measures are taken, $(\sum Pr\,o\,d(t))^2$ is maximized, and when the soft penalty formulation is integrated, $\sum(Pr\,o\,d(t) + \alpha_{hopf}.s1^2{}_{max}$ is maximized. When no measures were taken, the obtained value of $(\sum Pr\,o\,d(t))$ was 306.864, and when the soft penalty formulation is integrated, with $\alpha_{hopf} = 1$; the obtained value of $(\sum Pr\,o\,d(t))$ was 306.574.

This is a small price to pay to avoid the limit cycle causing Hopf bifurcations. Figures 2a, 2b, 2c, and 2d show the various Hopf bifurcation profiles.

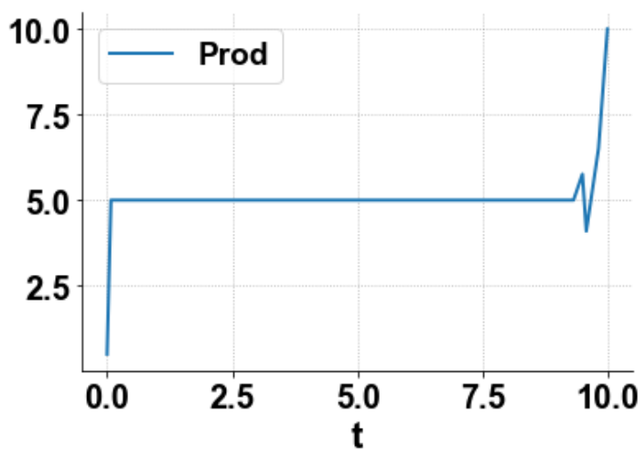

Figure 2a. Optimal control with no Hopf Constraint (Product Profile).

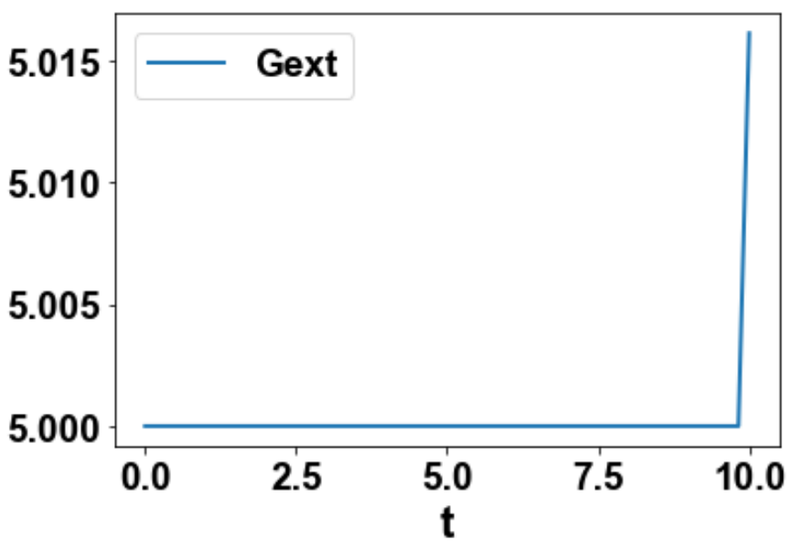

Figure 2b. Optimal control with no Hopf Constraint (Gext Profile).

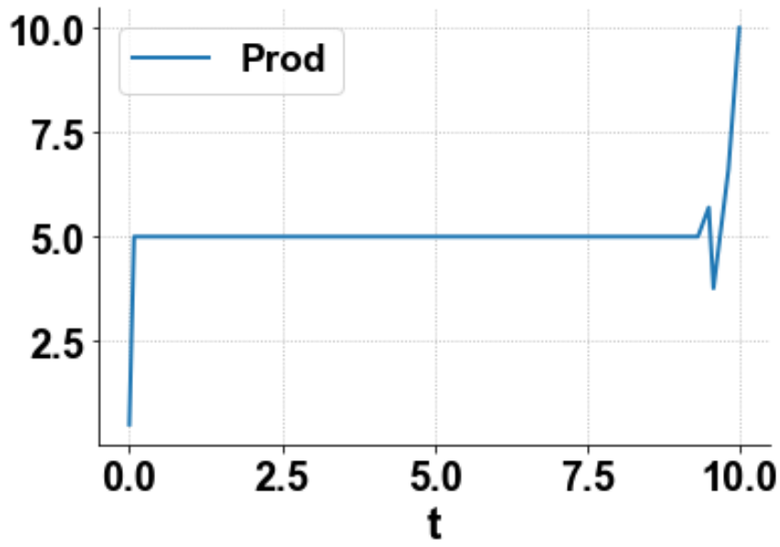

Figure 2c. Optimal control with Hopf Constraint (Product Profile).

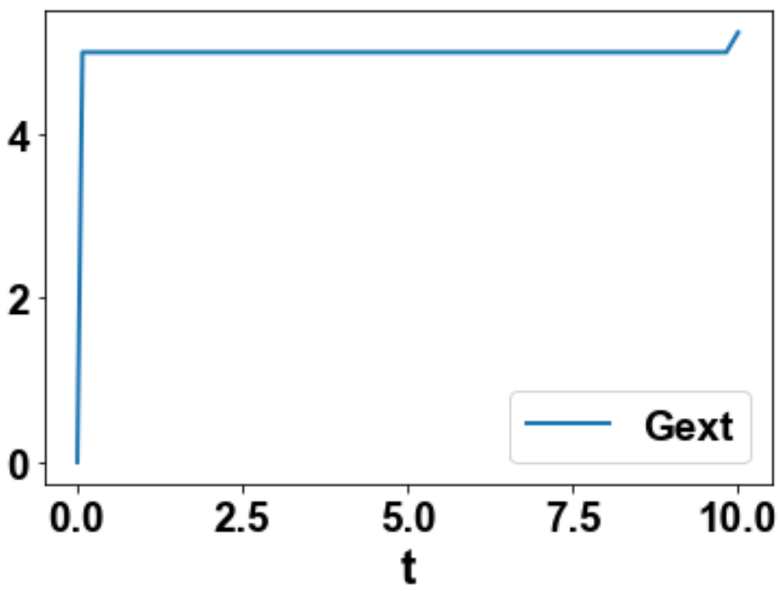

Figure 2d. Optimal control with Hopf Constraint (Gext Profile).

4. Discussion of Results

The results of this study clearly demonstrate the effectiveness of a unified approach combining bifurcation analysis, machine learning, and optimal control. In particular, the bifurcation analysis using continuation methods detects a Hopf bifurcation, indicating a transition from a stable steady state to an oscillatory solution. The detected bifurcation point indicates that the system is in a region close to a stability boundary under certain conditions. This is an important result, as small variations in external glucose concentration or the system's internal state can drive it to an oscillatory solution. In a biochemical production system, oscillatory solutions are undesirable because they can lead to yield variations.

The creation of the stability dataset based on continuation results offers an important connection between nonlinear dynamics and data-driven modeling. By calculating the Jacobian matrix at each equilibrium point and finding the maximum real part of its eigenvalues, a scalar stability measure is obtained. This measure is a continuous indicator of system stability, with negative values indicating stability and positive values indicating instability. Using a scalar measure of stability turns the problem into a regression task, which can be efficiently solved using machine learning. The dataset includes both stable and unstable operating conditions, including the transition at the Hopf bifurcation point.

The neural network surrogate model has approximated the relationship between the system states, the bifurcation parameter, and the stability metric using the provided dataset. The use of smooth activation functions, such as the hyperbolic tangent function, has enabled the surrogate model to be continuously differentiable. This is important for embedding the surrogate model into an optimization algorithm such as IPOPT. The direct computation of eigenvalues is known to be computationally intensive and is not easily differentiable. The neural network approach can provide a fast, continuously differentiable approximation for an optimal control problem. The results confirm the neural network's ability to approximate the stability boundary with sufficient accuracy.

The key contribution of this work is the introduction of a stability-constrained optimal control problem with a soft penalty constraint. A soft penalty constraint is a constraint that does not strictly prohibit a solution from entering a particular region. In this case, a smooth penalty constraint is added to the objective function. This constraint penalizes positive values of the predicted maximum eigenvalue. It prevents the system from entering unstable regions. The smooth constraint formulation ensures compatibility with gradient-based optimization algorithms. This prevents discontinuities that could hinder optimization.

The difference between the unconstrained and stability-constrained optimization results shows that the stability constraint plays a significant role in practical scenarios. When stability constraints are not considered, the optimization problem aims only to maximize product formation. However, it often happens that to achieve a higher product formation rate, the system must operate at the boundary of stability or even in unstable regions. Although this approach seems optimal from a practical perspective, as it allows for higher product formation rates, it is undesirable from a system perspective. This is because oscillations may occur, causing uncontrollable system behavior.

When the stability penalty is incorporated, the optimizer finds solutions that keep the system in a stable state. The control profiles are now smoother and more conservative, as a trade-off between maximum production and stability. What is more interesting, though, is that the decrease in the objective function value is not large. This implies that large improvements in stability can be achieved by sacrificing performance by a small amount. This trade-off is highly beneficial in real-world problems, where stability and robustness are much more important than small productivity gains. The control profile of the external glucose concentration further supports the idea of a trade-off between stability and performance. In the unconstrained case, the control profile varies aggressively. This could potentially drive the system to instability. In contrast, in the stability-constrained case, a smoother control profile is achieved. This indicates that the presented framework inherently produces stable control profiles. The overall findings validate the proposed approach as a potent, computationally efficient method for stability-aware optimal control of nonlinear systems. The method effectively overcomes key limitations of traditional approaches by combining bifurcation-informed data sets, neural network surrogates, and smooth optimization. Additionally, the method can be generalized to more complex biochemical systems or other types of nonlinear dynamical systems. The inclusion of stability as a key element of the optimization problem represents a significant improvement in the design and operation of dynamic biological systems.

5. Conclusions

In this study, a stability-constrained optimal control framework for a nonlinear artificial cell system is proposed by integrating bifurcation analysis, machine learning, and dynamic optimization. The proposed approach is particularly important for addressing one of the most significant limitations of conventional optimal control methods for solving highly nonlinear biochemical systems: the lack of stability consideration in the optimization process. The proposed mechanistic model describes the dynamics of gene expression, metabolic fluxes, energy balance, and growth, and is highly nonlinear, capable of multiple steady states and even oscillatory behavior. These dynamics are common in biochemical systems and can cause problems in optimization, as maximizing certain objective functions, such as product formation, can lead to unstable or oscillatory behavior. The study's findings showed that neglecting stability in optimization can lead to physically or dynamically undesirable solutions. To overcome this problem, bifurcation analysis was used for the first time to identify regions of stability loss, particularly around Hopf bifurcations. The collected data was used to train a neural network surrogate model capable of approximating the stability boundary of the original system. This is much more efficient than repeatedly performing eigenvalue analysis, which would be unfeasible in a dynamic optimization problem.

The neural network was incorporated into the optimal control problem using a smooth penalty approach. Here, trajectories leading to instability were penalized using a smooth function of the dominant eigenvalue. This approach is compatible with gradient-based nonlinear programming algorithms and prevents discontinuities that could interfere with the optimization process. The optimization problem was solved using a direct transcription approach implemented in the open-source library Pyomo. The numerical results provided in this study clearly demonstrate the effectiveness of the proposed framework. In particular, when incorporating the stability penalty term in the optimization problem, stable operation is maintained while substantial product formation is achieved. In contrast, when stability is not considered in the optimization problem, non-physical solutions arise, a critical takeaway from this study. Moreover, this study highlights the model's sensitivity to seemingly minor structural aspects, such as the inclusion of metabolite decay, which significantly affects the attainment of non-degenerate steady states and numerical convergence. A second important contribution of this study is the application of a hybrid modeling approach that couples mechanistic and data-driven modeling strategies. In particular, the successful application of a neural network surrogate within a dynamic optimization framework implemented in Pyomo is demonstrated, which highlights the potential benefits of combining different modeling approaches to improve prediction accuracy and computational efficiency. In particular, the use of smooth activation functions and appropriate scaling strategies is found to have a significant effect in ensuring solver convergence.

On a more general note, this study's proposed methodology applies to the stability-constrained optimization of nonlinear dynamical systems. The ability to leverage stability information via learned surrogates is particularly useful for addressing high-dimensional systems, where direct bifurcation analysis for optimization is computationally intractable. Such systems are common in biochemical engineering, synthetic biology, and process systems engineering, in which complex dynamics are commonplace. Possible future directions for this study's proposed methodology include developing the framework to quantify uncertainties, exploring different architectures for machine learning-based surrogates, such as physics-informed neural networks, and applying it to experimentally validated biological systems. In conclusion, this study's proposed methodology clearly demonstrates that combining bifurcation theory, machine learning, and dynamical optimization is a powerful approach to stability analysis in nonlinear systems. The proposed framework is a promising approach to more reliably and efficiently optimizing complex biochemical systems, which aligns with the general aim of developing data-enabled modeling and control in chemical engineering.

References

[1] M. B. Elowitz and S. Leibler, "A synthetic oscillatory network of transcriptional regulators," *Nature*, vol. 403, no. 6767, pp. 335-338, 2000. https://doi.org/10.1038/35002125

[2] A. Varma and B. O. Palsson, "Stoichiometric flux balance models quantitatively predict growth and metabolic by-product secretion in wild-type Escherichia coli W3110," *Applied and Environmental Microbiology*, vol. 60, no. 10, pp. 3724-3731, 1994. https://doi.org/10.1128/aem.60.10.3724-3731.1994

[3] M. W. Covert, E. M. Knight, J. L. Reed, M. J. Herrgard, and B. O. Palsson, "Integrating high-throughput and computational data elucidates bacterial networks," *Nature*, vol. 429, no. 6987, pp. 92-96, 2004. https://doi.org/10.1038/nature02456

[4] H. Kitano, "Systems biology: A brief overview," *Science*, vol. 295, no. 5560, pp. 1662-1664, 2002. https://doi.org/10.1126/science.1069492

[5] J. D. Murray, *Mathematical biology I: An introduction*, 3rd ed. New York, USA: Springer, 2002.

[6] U. Alon, *An introduction to systems biology: Design principles of biological circuits*. Boca Raton, FL, USA: CRC Press, 2006.

[7] J. Stricker, S. Cookson, M. R. Bennett, W. H. Mather, L. S. Tsimring, and J. Hasty, "A fast, robust and tunable synthetic gene oscillator," *Nature*, vol. 456, no. 7221, pp. 516-519, 2008. https://doi.org/10.1038/nature07389

[8] T. Danino, O. Mondragón-Palomino, L. Tsimring, and J. Hasty, "A synchronized quorum of genetic clocks," *Nature*, vol. 463, no. 7279, pp. 326-330, 2010. https://doi.org/10.1038/nature08753

[9] H. Qian and D. A. Beard, "Metabolic futile cycles and their functions: A systems analysis of energy and control," *IEE Proceedings-Systems Biology*, vol. 153, no. 4, pp. 192-200, 2006. https://doi.org/10.1049/ip-syb:20050086

[10] R. Antia and L. G. de Pillis, "A validated mathematical model of the cell-mediated immune response to tumor growth," *Journal of Theoretical Biology*, vol. 217, no. 3, pp. 367–376, 2005.

[11] J. Min Lee, E. P. Gianchandani, J. A. Eddy, and J. A. Papin, "Dynamic analysis of integrated signaling, metabolic, and regulatory networks," *PLoS Computational Biology*, vol. 4, no. 5, p. e1000086, 2008. https://doi.org/10.1371/journal.pcbi.1000086

[12] J. R. Karr *et al.*, "A whole-cell computational model predicts phenotype from genotype," *Cell*, vol. 150, no. 2, pp. 389-401, 2012. https://doi.org/10.1016/j.cell.2012.05.044

[13] E. J. O'Brien, J. A. Lerman, R. L. Chang, D. R. Hyduke, and B. Ø. Palsson, "Genome-scale models of metabolism and gene expression extend and refine growth phenotype prediction," *Molecular Systems Biology*, vol. 9, p. 693, 2013. https://doi.org/10.1038/msb.2013.52

[14] C. J. Lloyd *et al.*, "COBRAme: A computational framework for genome-scale models of metabolism and gene expression," *PLoS Computational Biology*, vol. 14, no. 7, p. e1006302, 2018. https://doi.org/10.1371/journal.pcbi.1006302

[15] S. Imam, S. Schäuble, A. N. Brooks, N. S. Baliga, and N. D. Price, "Data-driven integration of genome-scale regulatory and metabolic network models," *Frontiers in Microbiology*, vol. 6, p. 409, 2015. https://doi.org/10.3389/fmicb.2015.00409

[16] I. Domenzain *et al.*, "Reconstruction of a catalogue of genome-scale metabolic models with enzymatic constraints using GECKO 2.0," *Nature Communications*, vol. 13, no. 1, p. 3766, 2022. https://doi.org/10.1038/s41467-022-31421-1

[17] E. Gonçalves *et al.*, "Bridging the layers: Towards integration of signal transduction, regulation and metabolism into mathematical models," *Molecular BioSystems*, vol. 9, no. 7, pp. 1576-1583, 2013. https://doi.org/10.1039/C3MB25489E

[18] M. Razzaq, L. Chebouba, P. Le Jeune, H. Mhamdi, C. Guziolowski, and J. Bourdon, *Logic and linear programs to understand cancer response. In Automated Reasoning for Systems Biology and Medicine*. Cham: Springer International Publishing, 2019.

[19] A. Dasgupta, A. Bakshi, N. Chowdhury, and R. K. De, "A control theoretic three timescale model for analyzing energy management in mammalian cancer cells," *Computational and Structural Biotechnology Journal*, vol. 19, pp. 477-508, 2021. https://doi.org/10.1016/j.csbj.2020.12.019

[20] S. Choi, J. S. Song, and J. H. Lee, "Dynamic modeling of cell-free synthetic biology systems," *Current Opinion in Biotechnology*, vol. 58, pp. 84–91, 2019.

[21] A. Dhooge, W. Govaerts, and Y. A. Kuznetsov, "MATCONT: A MATLAB package for numerical bifurcation analysis of ODEs," *ACM Transactions on Mathematical Software*, vol. 29, no. 2, pp. 141-164, 2003. https://doi.org/10.1145/779359.779362

[22] A. Dhooge, W. Govaerts, Y. A. Kuznetsov, W. Mestrom, and A. M. Riet, "Cl_MatCont: A continuation toolbox in MATLAB," in *Proceedings of the 2003 ACM Symposium on Applied Computing*, 2003, pp. 161–166.

[23] Y. A. Kuznetsov, *Elements of applied bifurcation theory*. New York: Springer, 1998.

[24] Y. A. Kuznetsov, "Five lectures on numerical bifurcation analysis," *Utrecht University, NL*, 2009.

[25] W. J. F. Govaerts, *Numerical methods for bifurcations of dynamical equilibria*. Philadelphia, PA, USA: SIAM, 2000.

[26] W. E. Hart *et al.*, *Pyomo-optimization modeling in Python*. Berlin: Springer, 2017.

[27] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25-57, 2006. https://doi.org/10.1007/s10107-004-0559-y