A data-driven Koopman framework for programming the steady state of biological systems with parametric uncertainty

Aqib Hasnain, Nibodh Boddupalli, and Enoch Yeung

***Abstract*— A central challenge in industrial microbiological applications is engineering of genetic networks to achieve a target yield or steady state concentration. This is a particularly challenging problem when optimizing under novel reaction conditions where canonical models of metabolic pathways are no longer valid. In these scenarios, the biological systems are entirely represented by data for which no direct methods exist to optimize the reaction output. We introduce a data-driven model discovery approach that leverages Koopman operator theory and time-series expression measurements to discover models that predict untested reaction outcomes. These Koopman models allow us to design control strategies to achieve target steady state concentrations for products of interest. We develop a model of the trypotophan pathway and illustrate this steady state programming framework on this pathway. Tryptophan is an essential amino acid and is a valuable product in industrial microbiology. We show how Koopman operator theory can thus be used program the steady state of a genetic network in a data-driven context.**

I. Introduction

One of the aims of the field of synthetic biology is the programming of cells to exhibit specified functionality by engineering the cells to perform a series of logic computations i.e. a genetic circuit. The scale of interest is wide ranging from single-celled organisms such as bacteria to populations of cells and even organs [20]. Genetic circuits have recently been designed and built to tackle problems such as disease diagnosis and treatment [3], [29], detect and report toxic compounds in the environment [26], and conversion of biomass into biofuels [16].

Many industrial microbiological processes are concerned with achieving a target steady state yield or concentration [27]. As the field of synthetic biology continues to grow, genetic circuits are being designed to make these microbiological processes more efficient and productive. The role of uncertainty in biological systems is enormous and managing uncertainty is a pivotal ability in any engineered system [20]. Currently, microbiological processes are optimized using model-based or hypothesis-driven approaches which require iterative experimentation. With fine resolution time-series 'omics measurements being expensive and laborious to collect, it may not be economically or physically feasible to collect new data for perturbations in the system parameters. How can we leverage novel data-driven methods and learning algorithms to program cells and predict uncertain reaction outcomes?

A genetic program is formed when multiple genetic circuits are coupled with sensors and actuators [5]. The design of genetic programs is impeded by lack of robust genetic circuits that can be readily connected while functionally operating in a wide range of environmental conditions. Modeling may be helpful to identify parameter regimes and network topologies for which a specific function exists and is robust. However, models are often hypothesis driven, in that they rely on hypotheses or explicit knowledge of key biochemical processes. For example, biophysical models of compositional or host context reveal a link between circuit stability and genetic program design [7], [9], [30], [31]. Often these processes are difficult to observe or validate experimentally. Robust biological computing devices which can operate in varying environmental conditions have the potential to solve challenging problems for energy consumption, large-scale computing, and biological threat detection.

The above issues motivate the need for purely data-driven algorithms to accelerate the advancement of design of genetic circuits and genetic programs as well as to increase the productivity or efficiency of microbiological processes. The question we address in this paper is can the steady state concentration of an industrial microbiological process be optimized using data alone? Specifically, how do we utilize time-series measurements of a biological system to design optimal control inputs to achieve a target steady state output? This form of methodology can be especially useful for cases where system parameters are unknown or uncertain.

Spectral methods have been increasingly popular in data-driven analysis of nonlinear dynamical systems. Koopman operator theory has the potential to provide strategies for optimal control of biological systems and researchers working in this space have shown that it is possible to identify and learn the fundamental modes of a nonlinear dynamical system from data [14], [22]. The Koopman operator is an infinite-dimensional linear operator that represents nonlinear dynamics as a dynamically equivalent linear system. The development of dynamic mode decomposition (DMD) [23] has led to rapid growth in the use of Koopman spectral analysis of nonlinear dynamical systems in areas such as system identification [1], design of experiments [2], prediction and control [10], [11], [17], and sensor placement [8]. It was shown in [22] that the approximate Koopman operator obtained from DMD is closely related to a spectral analysis of the linear but infinite-dimensional Koopman operator. More recently, learning higher dimensional Koopman opera-

A. Hasnain and N. Boddupalli are with the Department of Mechanical Engineering, University of California, Santa Barbara aqib@ucsb.edu, nibodh@ucsb.edu

E. Yeung is with the Department of Mechanical Engineering, Center for Control, Dynamical Systems, and Computation, and Biomolecular Science and Engineering, University of California, Santa Barbara eyeung@ucsb.edu

Manuscript 1157 submitted to 2020 American Control Conference.
Received September 26, 2019.

tors from data has become computationally tractable, largely due to advances in integrating machine learning and deep learning to generate efficient representations of observable bases [13], [15], [25], [32].

In this paper we show that Koopman models facilitate the steady state programming of nonlinear systems with uncertain parameters through a convex optimization formulation made possible due to the linearity of the Koopman operator and the choice of basis functions. We first introduce Koopman operator theory in Section II. We then briefly show how to identify the Koopman model directly from data. In Section III, we introduce our novel method for programming the steady state of a nonlinear system and perform uncertainty analysis. Finally, in Section IV, we develop a model of the Tryptophan (Trp) Pathway and illustrate our steady state programming and find the optimal control strategy to maximize the production of Trp, an essential amino acid with commercial value in industrial microbiology [6].

II. Koopman Operator Formulation

Since our framework makes extensive use of Koopman operators, we first briefly introduce the theory. We focus on discrete-time dynamical systems to be consistent with the nature of data obtained from measurements.

Consider a discrete-time open-loop nonlinear system of the form

$$\begin{aligned} x_{t+1} &= f(x_t) \\ y_t &= h(x_t) \end{aligned} \tag{1}$$

with $f : \mathbb{R}^n \to \mathbb{R}^n$ is analytic and $h \in \mathbb{R}^p$. Then we know that there exists a countably infinite or finite dimensional Koopman operator [33] of (1), $\mathcal{K} : \mathcal{F} \to \mathcal{F}$ defined by

$$\mathcal{K}\psi(x_t) = \psi \circ f(x_t). \tag{2}$$

The function $\psi : \mathbb{R}^n \to \mathbb{R}$ is called an *observable* of the system and the set of all observables $\psi \triangleq \{\psi_i\}_{i=1}^{\infty}$ on a dynamical system form an infinite-dimensional vector space. Here $\mathcal{F}$ is the space of observable functions that is invariant under the action of $\mathcal{K}$.

The most important property of the Koopman operator that we utilize is the linearity of the operator, in other words,

$$\mathcal{K}(\alpha\psi_1 + \beta\psi_2) = \alpha\psi_1 \circ f + \beta\psi_2 \circ f = \alpha\mathcal{K}\psi_1 + \beta\mathcal{K}\psi_2$$

which follows from (2) since the composition operator is linear. Thus, we have that the Koopman operator of (1) is a linear operator that acts on observable functions $\psi(x_k)$ and propagates them forward in time. If we assume that $y_k = h(x_k) \in \mathcal{F}$ and that $h \in \text{span}\{\psi_1, \psi_2, ...\}$, the state-output equations may be expressed as

$$\begin{aligned} \psi(x_{t+1}) &= \mathcal{K}\psi(x_t) \\ y_t = h(x_t) &= W_h\psi(x_t) \end{aligned} \tag{3}$$

where the output matrix $W_h \in \mathbb{R}^{p \times n_L}$, $n_L \leq \infty$.

A. Finite dimensional approximations

Using data-driven approaches, commonly DMD [23] or extended DMD [28], an approximate Koopman operator, K, can be identified from time-series data of a dynamical system. The approach taken to compute an approximation to the Koopman operator in both DMD and extended DMD is to solve the following optimization problem

$$\min_K ||\Psi(X_f) - K\Psi(X_p)|| \tag{4}$$

where $X_f \equiv \begin{bmatrix} x_1 & \dots & x_{N-1} \end{bmatrix}$, $X_p \equiv \begin{bmatrix} x_2 & \dots & x_N \end{bmatrix}$ are snapshot matrices formed from the measurements of the discrete-time dynamical system (1) and $\Psi(X) \equiv \begin{bmatrix} \psi_1(x) & \dots & \psi_R(x) \end{bmatrix}$ is the mapping from physical space into the space of observables. DMD is a special case of extended DMD where $\psi(x) = x$.

B. Input-Koopman operator for systems with control

Koopman operator theory has been extended to controllable systems [10], [18], [19]. In [12], a method for learning input-Koopman operators from data by integrating deep learning and dynamic mode decomposition (deepDMD) [32] was introduced.

Consider the following discrete-time nonlinear system with analytic vector field $f(x_t, w_t)$ and control input $w_t \in \mathbb{R}^m$,

$$\begin{aligned} x_{t+1} &= f(x_t, w_t) \\ y_t &= h(x_t). \end{aligned} \tag{5}$$

It was shown in [18], [19] that an input-Koopman operator can be defined that satisfies

$$\psi(x_{t+1}, w_{t+1}) = K\psi(x_t, w_t) = \psi(f(x_t, w_t)). \tag{6}$$

For an exogenous memoryless input, [12] demonstrated that (6) can be modified by splitting $\psi(x, w)$ into components $\psi_x(x)$ and $\psi_u(x, u)$, where $\psi_x(x)$ is a stacked vector valued observable of all scalar valued observable functions from $\psi(x, w)$ that do not depend on w, and $\psi_u(x, u)$ is a stacked vector valued observable of all remaining terms. This results in the decomposed representation

$$\begin{aligned} \psi_x(x_{t+1}) &= K_x\psi_x(x_t) + K_u\psi_u(x_t, u_t) \\ y &= W_h\psi(x_t, u_t). \end{aligned} \tag{7}$$

Here $u_t = u(x_t, w_t)$ is a vector function consisting of univariate terms of w_t and multivariate polynomial terms consisting of x_t and w_t. This is a representation of the nonlinear system dynamics that is linear in the lifted state observable $\psi_x(x)$ and the lifted input-state mixture observable $\psi_u(u)$.

III. The steady state of nonlinear systems can be programmed using convex optimization

Consider the case where the system (5) is a biological network, sampled at uniform timesteps, that produces one or several products of interest. These products can be amino acids, proteins, waste byproducts, etc. The rate of production of these products of interest are controlled by the inputs of the system and canonical models may have been identified through rigorous and laborious experimental probing. Now assume that the system is operating in uncertain conditions such that the process of interest is impacted and the optimal

Manuscript 1157 submitted to 2020 American Control Conference.
Received September 26, 2019.

input is no longer known. We seek to program the steady state of products of interest using convex optimization, made possible through data-driven Koopman models. The nonlinear system (5) can be represented as (7) with linear dynamics through the proper lifting of the states and inputs. From measurements of the system, we use DMD to identify the Koopman operator and input-Koopman operator that models the system linearly.

Let's consider the case where we seek to maximize the steady state production of the n_L^{th} state of the lifted system (note that if using DMD, the n_L^{th} state is the same as the n^{th} state). At steady state $\psi_x(x_{k+1}) = \psi_x(x_k)$, and so the Koopman system becomes

$$\psi_x(x_{eq}) = (I - K_x)^{-1} K_u \psi_u(u_{eq}) \tag{8}$$

where the subscript 'eq' denotes the equilibrium or steady state values of the states and inputs. We can define a projection matrix, $P_x \in \mathbb{R}^{1\times n_L}$, as

$$P_x = \hat{e}_{n_L}^\top$$

where $\hat{e}_{n_L}$ is the unit column vector in the n_L^{th} direction so that P_x projects the states $\psi_x(x_{eq})$ onto the dimension which we wish to optimize. The optimization problem which we seek to solve is then

$$\begin{aligned} \min_{u^*} \sum_t & -P_x(I - K_x)^{-1} K_u \psi_u(u^*) \\ \text{s.t.} \quad & 0 \le u^* \le u_{max} \end{aligned} \tag{9}$$

where the lower and upper bound constraints on u^* arise from biophysical limitations. The argument of the optimization problem (9) is a convex function and can be solved using standard convex optimization routines. The convexity is a consequence of Koopman operators being linear operators and the choice of observable basis to be linear. Had we attempted to determine the optimal u^* from the original system or from a change of coordinates obtained from deepDMD, it is highly possible we would face the issue of the dynamics either being nonlinear or the steady state equation likely being nonconvex.

The scenario we pose of maximizing a single state in the system is an illustrative example and the framework can be applied in varying ways. For example, in biological systems it may be more useful to think about maximizing some product while minimizing another. The product to be minimized may be some waste product from an intermediate reaction in a metabolic pathway. One may also think about maximizing or minimizing multiple states simultaneously.

A. Programming the steady state of a perturbed system using an unperturbed model

It is often not economically or physically feasible to take time-series measurements of biological systems operating in novel conditions. In these cases, it may seem attractive to utilize unrefined models to program the steady state of the system. We would like to know what the error in this operation looks like.

The general steady state programming problem we want to solve is

$$\begin{aligned} \min_{u^*} \Big[\psi_x(u^*) = \sum_t & -(I - K_x)^{-1} K_u \psi_u(u^*)\Big] \\ \text{s.t.} \quad & 0 \le u^* \le u_{max} \end{aligned}$$

where we are dropping any projection matrices as it is not necessary for this analysis. K_x and K_u are state and input Koopman operators identified from the data ψ_x and ψ_u. For a perturbed system, the steady state programming formulation is written as

$$\begin{aligned} \min_{\hat{u}^*} \Big[\hat{\psi}_x(\hat{u}^*) = \sum_t & -(I - \hat{K}_x)^{-1} \hat{K}_u \hat{\psi}_u(\hat{u}^*)\Big] \\ \text{s.t.} \quad & 0 \le \hat{u}^* \le \hat{u}_{max} \end{aligned}$$

where the $\hat{}$ denotes data and Koopman operators of the perturbed system. We would like to quantify the error in the steady state programming solution if we were to use the Koopman model identified from the unperturbed system to program the steady state of the perturbed system, i.e.

$$\begin{aligned} \min_{\bar{u}^*} \Big[\hat{\psi}_x(\bar{u^*}) = \sum_t & -(I - K_x)^{-1} K_u \hat{\psi}_u(\bar{u}^*)\Big] \\ \text{s.t.} \quad & 0 \le \bar{u}^* \le \bar{u}_{max}. \end{aligned} \tag{10}$$

We define the error as

$$\begin{aligned} e &= ||\hat{\psi}_x(\bar{u^*}) - \hat{\psi}_x(\hat{u^*})|| \\ &= ||(I - K_x)^{-1} K_u \hat{\psi}_u(\bar{u}^*) - (I - \hat{K}_x)^{-1} \hat{K}_u \hat{\psi}_u(\hat{u}^*)||. \end{aligned}$$

We will define $\Delta_x \equiv \hat{K}_x - K_x$ so that we have

$$\begin{aligned} e = ||&(I + \Delta_x - \hat{K}_x)^{-1} K_u \hat{\psi}_u(\bar{u}^*) \\ &- (I - \hat{K}_x)^{-1} \hat{K}_u \hat{\psi}_u(\hat{u}^*)||, \end{aligned}$$

by the Woodbury matrix identity we have

$$\begin{aligned} e = ||&\Big[(I - \hat{K}_x)^{-1} \\ &- (I - \hat{K}_x)^{-1}\big(\Delta_x^{-1} + (I - \hat{K}_x)^{-1}\big)^{-1}(I - \hat{K}_x)^{-1}\big)\Big] \\ &\times K_u \hat{\psi}_u(\bar{u}^*) - (I - \hat{K}_x)^{-1} \hat{K}_u \hat{\psi}_u(\hat{u}^*)|| \\ = ||&- (I - \hat{K}_x)^{-1}\big(\Delta_x^{-1} + (I - \hat{K}_x)^{-1}\big)^{-1}(I - \hat{K}_x)^{-1}) \\ &\times K_u \hat{\psi}_u(\bar{u}^*) + (I - \hat{K}_x)^{-1}\big(K_u \hat{\psi}_u(\bar{u}^*) - \hat{K}_u \hat{\psi}_u(\hat{u}^*)\big)|| \\ \le ||&- (I - \hat{K}_x)^{-1}\big(\Delta_x^{-1} + (I - \hat{K}_x)^{-1}\big)^{-1}(I - \hat{K}_x)^{-1}) \\ &\times K_u \hat{\psi}_u(\bar{u}^*)|| + ||(I - \hat{K}_x)^{-1}\big(K_u \hat{\psi}_u(\bar{u}^*) - \hat{K}_u \hat{\psi}_u(\hat{u}^*)\big)||. \end{aligned}$$

This shows that the upper bound on the error in the steady state solution depends inversely on the difference between unperturbed state Koopman operator and the perturbed state Koopman operator. Similarly, the lower bound on the error is found as

$$\begin{aligned} e \ge ||&- (I - \hat{K}_x)^{-1}\big(\Delta_x^{-1} + (I - \hat{K}_x)^{-1}\big)^{-1}(I - \hat{K}_x)^{-1}) \\ &\times K_u \hat{\psi}_u(\bar{u}^*)|| - ||(I - \hat{K}_x)^{-1}\big(\hat{K}_u \hat{\psi}_u(\hat{u}^*) - K_u \hat{\psi}_u(\bar{u}^*)\big)|| \end{aligned}$$

which is the best case scenario for the error if one was unable to collect new time-series measurements after environmental conditions of the biological system have changed.

We note that for the special case where $\hat{K}_u = K_u$ and

$\hat{\psi}_u(\bar{u}^*) = \hat{\psi}_u(\hat{u}^*)$ we have

$$e = || - (I - \hat{K}_x)^{-1}\big(\Delta_x^{-1} + (I - \hat{K}_x)^{-1}\big)^{-1}(I - \hat{K}_x)^{-1}) \times K_u \hat{\psi}_u(\hat{u}^*)||$$

which is quadratic in $(I - \hat{K}_x)^{-1}$. As this term grows large, the error progressively reduces.

IV. Simulation Study: The Trp Pathway

Tryptophan (Trp) is an amino acid essential for survival [6]. *Escherichia coli* (*E. coli*) cells can synthesize their own Trp if it is not provided in their environment. There are multiple mechanisms of regulation present in the Trp pathway and it has been shown theoretically that the combinations of these mechanisms can lead to instability or nonlinear dynamical behavior [4]. There is no experimental evidence of the instability of the pathway, however instability is one of the leading hypothesis for inconsistent or inaccurate yield predictions in industrial process models [4]. It is also hypothesized that limitations in the yield of Trp may be due to unidentified regulatory architectures in the pathway [4]. Although the model that we derived of the Trp pathway does not incorporate all the known or unknown regulatory mechanisms that exist in the pathway, our steady state programming framework is still valid. This is due to the purely data-driven model discovery approach.

We seek to program the rate of production of Trp using the data-driven Koopman steady state programming method introduced in III. Specifically, we seek to maximize the production of Trp by controlling the *trp* operon. We model the Trp pathway, in *E. coli* cells, using Michaelis-Menten enzyme kinetics and control inputs. The reaction network is given in Figure 1. The biosynthesis pathway converts chorismate, a precursor for aromatic amino acids, into Trp through a series of six reactions. Our model incorporates the starting, intermediate, and final products (colored in red in Figure 1) as well as the enzymes produced by the *trp* operon genes (trpA, trpB, trpC, trpD, trpE, and trpR colored in green) and most of the byproducts (colored in blue), as states in the dynamical system. Lastly, dCas9 bound to gRNA act as control inputs (colored in purple) to the dynamical system. The full kinetic model is shown in Figure 3 and was derived from the chemical reaction network provided by the EcoCyc database (`https://biocyc.org/ECOLI/NEW-IMAGE?type=PATHWAY&object=TRPSYN-PWY&show-citations=NIL`).

For the purposes of modeling biological control of Trp production (to be experimentally implemented in future work), we model a CRISPRi gene knockdown system. The gRNA binds to dCas9 to form the gRNA:dCas9 molecule which can then bind to the enzymes produced by the genes in the *trp* operon. The gRNA:dCas9 complex is considered as the control input. Once the enzymes have been bound with the complex, they can no longer act as catalysts for the reactions in the Trp pathway. We model this mechanism of control as a decrease in the enzyme production with increasing concentration of gRNA:dCas9. Figure 2 shows the dynamics of the normalized concentration of system over a long time scale such that all the states reach a steady state.

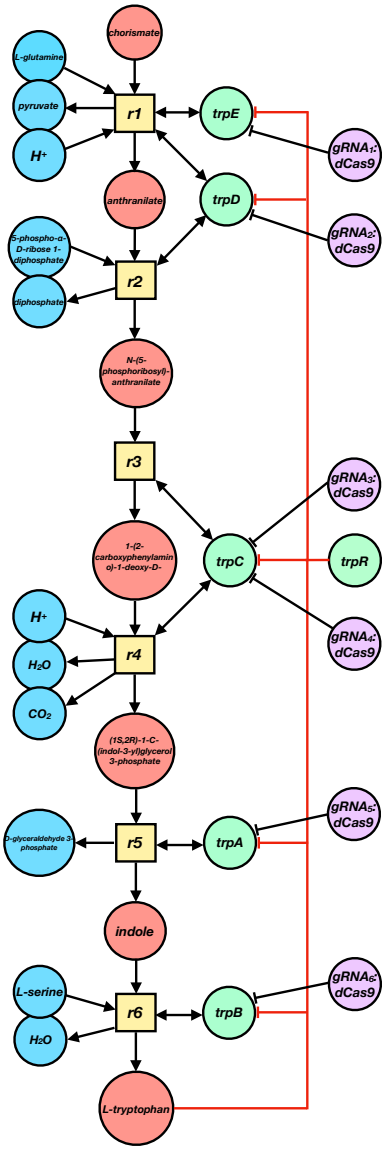

Fig. 1. Schematic of Trp biosynthesis in *Escherichia coli* K-12 substr. MG1655. The intermediate and final products are colored in red, the byproducts are colored in blue, the enzymes are colored in green, and the control inputs are colored in purple.

Many of the states approach zero concentration and a few states approach a concentration of one, including Trp.

We simulate uncertain conditions by randomly selecting the reaction rates, k_i's, in the reaction rate model from a Gaussian distribution. The Koopman operator and input-Koopman operator are identified from the simulated data using DMD. We define the projection operator in (9) as

$$P_x = \hat{e}_{10}^\top$$

since the Trp state is the tenth dimension in the model. We solve the minimization problem (9) using Operator Splitting Quadratic Programming Solver (OSQP) developed in [24].

The optimal u_* which maximizes the production of Trp is

Manuscript 1157 submitted to 2020 American Control Conference.
Received September 26, 2019.

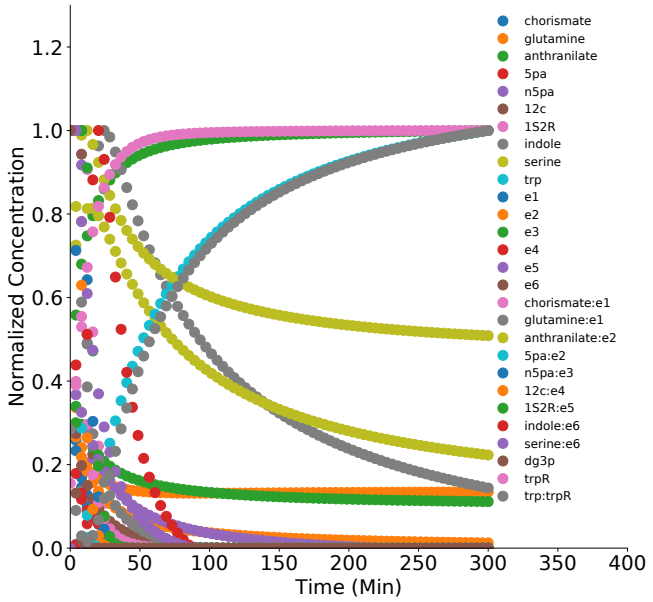

Fig. 2. Dynamics of the normalized concentration of the reactants and products in the Trp pathway. At short times, the dynamics evolve nonlinearly. At long times, the behavior is steady.

found to be

$$u^* = \begin{bmatrix} u_{max} & 0 & u_{max} & 0 & u_{max} & 0 \end{bmatrix}^\top$$

This corresponds to repressing enzymes 1, 3 and 5 or trpE, trpC, and trpB with maximal control while not repressing the other genes in the operon at all.

While using the control input as the dynamics of gRNA:dCas9, we achieved a steady state concentration of Trp to be 0.128 μM. The optimal control sequence u^* increases the steady state concentration of Trp to be 31.5 μM. This amounts to a 246 fold increase in the steady state production of Trp. It is important to note that the derived model of the Trp pathway is parameterized with random reaction rates, degradation rates, and activation rates. We therefore cannot comment on the biological accuracy of the steady state values, however we can show that the method allows us to increase the steady state production by a significant amount.

To understand how a perturbation in the parameters of the system would disrupt the effectiveness of Koopman models, we did the following benchmark experiment. We simulated the system with what we will call the original conditions i.e. biochemically accurate reaction rates, $\mathcal{N}(0.3, 0.01)$, as well as with perturbed reaction rates, $\mathcal{N}(1.0, 0.01)$. The corresponding Koopman models are written as

$$\psi_x(x_{t+1}) = K_x\psi_x(x_t) + K_u\psi_u(u_t)$$
$$\hat{\psi}_x(\hat{x}_{t+1}) = \hat{K}_x\hat{\psi}_x(\hat{x}_t) + \hat{K}_u\hat{\psi}_u(u_t).$$

We would like to know whether the original Koopman model K_x and K_u can be used to determine the optimal control strategy for maximizing the production of Trp at steady state. Therefore, we solve the steady state optimization problem (9) using the original Koopman model with the perturbed data i.e.

$$\min_{u^*} \sum_t -P_x(I - K_x)^{-1}K_u\hat{\psi}_u(u^*) \quad \text{s.t.} \quad 0 \le u^* \le u_{max}. \tag{11}$$

From the original Koopman model on the original data, we identified the optimal concentration of Trp at steady state to be 31.03 μM. From the perturbed Koopman model on the perturbed data, we identified the optimal concentration of Trp at steady state to be 65.99 μM. Finally, from the original Koopman model on the perturbed data, we identified the optimal concentration of Trp at steady state to be 34.82 μM, implying that attempting to use the Koopman model identified for a different set of system parameters can result in a suboptimal control strategy.

The analysis done here on the Trp pathway demonstrates that the steady state programming of cells can be made possible through the use of Koopman models. Using this framework, one can think about designing a series of logic computations that are performed at steady state by a cell or a population of cells.

V. Conclusion

In this work we introduced a data-driven method, leveraging Koopman operator theory, to program the steady state of nonlinear dynamical systems. The algorithm formulates the steady state programming problem as a convex optimization problem due to the linearity of Koopman operators and the observable basis chosen. To illustrate the algorithm on a high dimensional dataset, we developed a model of the Trp utilization pathway, an essential amino acid pathway, and demonstrated the capability by maximizing the production of Trp based on optimal control inputs. This method can be used for designing optimal control strategies directly from data which is specially useful in cases where complex model parameters are difficult to identify. Our approach is a step towards programming of cells, a critical component to the success of synthetic biology in solving challening problems.

Acknowledgements

The authors would like to thank Igor Mezic, Robert Egbert, Bassam Bamieh, Sai Pushpak, Sean Warnick, Umesh Vaidya, and Shara Balakrishnan for stimulating conversations. This work was supported partially by a Defense Advanced Research Projects Agency (DARPA) Grant No. DEAC0576RL01830 and an Institute of Collaborative Biotechnologies Grant. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Defense Advanced Research Projects Agency (DARPA), the Department of Defense, or the United States Government.

References

[1] Z. Bai, E. Kaiser, J. L. Proctor, J. N. Kutz, and S. L. Brunton. Dynamic mode decomposition for compressive system identification. *AIAA Journal*, pages 1–14, 2019.

$$
\begin{aligned}
\dot{[\text{chorismate}]} &= -k_0[\text{chorismate}][\text{glutamine}] - k_1[\text{chorismate}][\text{e}_{1,\text{trpE,trpD}}] + k_2[\text{chorismate:e}_{1,\text{trpE,trpD}}] - k_7[\text{chorismate}][\text{glutamine:e}_{1,\text{trpE,trpD}}] \\
\dot{[\text{glutamine}]} &= -k_0[\text{chorismate}][\text{glutamine}] - k_3[\text{glutamine}][\text{e}_{1,\text{trpE,trpD}}] + k_4[\text{glutamine:e}_{1,\text{trpE,trpD}}] - k_6[\text{chorismate:e}_{1,\text{trpE,trpD}}][\text{glutamine}] \\
\dot{[\text{anthranilate}]} &= k_0[\text{chorismate}][\text{glutamine}] + k_5[\text{chorismate:e}_{1,\text{trpE,trpD}}][\text{glutamine:e}_{1,\text{trpE,trpD}}] + k_6[\text{chorismate:e}_{1,\text{trpE,trpD}}][\text{glutamine}] \\
&\quad + k_7[\text{chorismate}][\text{ glutamine:e}_{1,\text{trpE,trpD}}] - k_8[\text{anthranilate}][5\text{p}\alpha] - k_9[\text{anthranilate}][\text{e}_{2,\text{trpD}}] + k_{10}[\text{anthranilate:e}_{2,\text{trpD}}] \\
&\quad - k_{15}[\text{anthranilate}][5\text{p}\alpha\text{:e}_{2,\text{trpD}}] \\
\dot{[5\text{p}\alpha]} &= -k_8[\text{anthranilate}][5\text{p}\alpha] - k_{11}[5\text{p}\alpha][\text{e}_{2,\text{trpD}}] + k_{12}[5\text{p}\alpha\text{:e}_{2,\text{trpD}}] \\
\dot{[\text{n5pa}]} &= k_8[\text{anthranilate}][5\text{p}\alpha] + k_{13}[\text{anthranilate:e}_{2,\text{trpD}}][5\text{p}\alpha\text{:e}_{2,\text{trpD}}] + k_{14}[\text{anthranilate:e}_{2,\text{trpD}}][5\text{p}\alpha] + k_{15}[\text{anthranilate}][5\text{p}\alpha\text{:e}_{2,\text{trpD}}] \\
&\quad - k_{16}[\text{n5pa}] - k_{17}[\text{n5pa}][\text{e}_{3,\text{trpC}}] \\
\dot{[12\text{c}]} &= k_{16}[\text{n5pa}] + k_{19}[\text{n5pa:e}_{3,\text{trpC}}] - k_{20}[12\text{c}][\text{H}^+] - k_{21}[12\text{c}][\text{e}_{4,\text{trpC}}] + k_{22}[12\text{c:e}_{4,\text{trpC}}] \\
\dot{[1\text{S2R}]} &= -k_{24}[1\text{S2R}] + k_{25}[\text{dg3p}][\text{indole}] - k_{26}[1\text{S2R}][\text{e}_{5,\text{trpA}}] + k_{27}[1\text{S2R:e}_{5,\text{trpA}}] + k_{20}[12\text{c}][\text{H}^+] + k_{23}[12\text{c:e}_{4,\text{trpC}}][\text{H}^+] \\
\dot{[\text{dg3p}]} &= k_{28}[1\text{S2R:e}_{5,\text{trpA}}] + k_{24}[1\text{S2R}] - k_{25}[\text{dg3p}][\text{indole}] \\
\dot{[\text{indole}]} &= k_{24}[1\text{S2R}] - k_{25}[\text{dg3p}][\text{indole}] + k_{28}[1\text{S2R:e}_{5,\text{trpA}}] - k_{29}[\text{e}_{5,\text{trpA}}][\text{dg3p}][\text{indole}] - k_{30}[\text{indole}][\text{serine}] \\
&\quad - k_{31}[\text{indole}][\text{e}_{6,\text{trpB}}] + k_{32}[\text{indole:e}_{6,\text{trpB}}] - k_{37}[\text{indole}][\text{serine:e}_{6,\text{trpB}}] \\
\dot{[\text{serine}]} &= -k_{30}[\text{indole}][\text{serine}] - k_{33}[\text{serine}][\text{e}_{6,\text{trpB}}] + k_{34}[\text{serine:e}_{6,\text{trpB}}] - k_{36}[\text{indole:e}_{6,\text{trpB}}][\text{serine}] \\
\dot{[\text{Trp}]} &= k_{30}[\text{indole}][\text{serine}] + k_{35}[\text{indole:e}_{6,\text{trpB}}][\text{serine:e}_{6,\text{trpB}}] + k_{36}[\text{indole:e}_{6,\text{trpB}}][\text{serine}] + k_{37}[\text{indole}][\text{serine:e}_{6,\text{trpB}}] \\
\dot{[\text{e}_{1,\text{trpE,trpD}}]} &= -k_1[\text{chorismate}][\text{e}_{1,\text{trpE,trpD}}] + k_2[\text{chorismate:e}_{1,\text{trpE,trpD}}] - k_3[\text{glutamine}][\text{e}_{1,\text{trpE,trpD}}] + k_4[\text{glutamine:e}_{1,\text{trpE,trpD}}] \\
&\quad + k_5[\text{chorismate:e}_{1,\text{trpE,trpD}}][\text{glutamine:e}_{1,\text{trpE,trpD}}] + k_6[\text{chorismate:e}_{1,\text{trpE,trpD}}][\text{glutamine}] + k_7[\text{chorismate}][\text{ glutamine:e}_{1,\text{trpE,trpD}}] \\
&\quad + \frac{1}{1 + [\text{gRNA}_1\text{:dCas9}]+[\text{Trp:trpR}]} - \frac{1}{1+\exp(\text{-}[\text{Trp}])[\text{Trp}]} - d_{\text{e}}[\text{e}_{1,\text{trpE,trpD}}] \\
\dot{[\text{e}_{2,\text{trpD}}]} &= -k_9[\text{anthranilate}][\text{e}_{2,\text{trpD}}] + k_{10}[\text{anthranilate:e}_{2,\text{trpD}}] - k_{11}[5\text{p}\alpha][\text{e}_{2,\text{trpD}}] + k_{12}[5\text{p}\alpha\text{:e}_{2,\text{trpD}}] + k_{13}[\text{anthranilate:e}_{2,\text{trpD}}][5\text{p}\alpha\text{:e}_{2,\text{trpD}}] \\
&\quad + k_{14}[\text{anthranilate:e}_{2,\text{trpD}}][5\text{p}\alpha] + k_{15}[\text{anthranilate}][5\text{p}\alpha\text{:e}_{2,\text{trpD}}] + \frac{1}{1 + [\text{gRNA}_2\text{:dCas9}]+[\text{Trp:trpR}]} - \frac{1}{1+\exp(\text{-}[\text{Trp}])[\text{Trp}]} - d_{\text{e}}[\text{e}_{2,\text{trpD}}] \\
\dot{[\text{e}_{3,\text{trpC}}]} &= -k_{17}[\text{n5pa}][\text{e}_{3,\text{trpC}}] + k_{18}[\text{n5pa:e}_{3,\text{trpC}}] + k_{19}[\text{n5pa:e}_{3,\text{trpC}}] + \frac{1}{1 + [\text{gRNA}_3\text{:dCas9}]+[\text{Trp:trpR}]} - \frac{1}{1+\exp(\text{-}[\text{Trp}])[\text{Trp}]} - d_{\text{e}}[\text{e}_{3,\text{trpC}}] \\
\dot{[\text{e}_{4,\text{trpC}}]} &= -k_{21}[12\text{c}][\text{e}_{4,\text{trpC}}] + k_{22}[12\text{c:e}_{4,\text{trpC}}] + k_{23}[12\text{c:e}_{4,\text{trpC}}][\text{H}^+] + \frac{1}{1 + [\text{gRNA}_4\text{:dCas9}]+[\text{Trp:trpR}]} - \frac{1}{1+\exp(\text{-}[\text{Trp}])[\text{Trp}]} - d_{\text{e}}[\text{e}_{4,\text{trpC}}] \\
\dot{[\text{e}_{5,\text{trpA}}]} &= -k_{26}[1\text{S2R}][\text{e}_{5,\text{trpA}}] + k_{27}[1\text{S2R:e}_{5,\text{trpA}}] + k_{28}[1\text{S2R:e}_{5,\text{trpA}}] + \frac{1}{1 + [\text{gRNA}_5\text{:dCas9}]+[\text{Trp:trpR}]} - \frac{1}{1+\exp(\text{-}[\text{Trp}])[\text{Trp}]} - d_{\text{e}}[\text{e}_{5,\text{trpA}}] \\
\dot{[\text{e}_{6,\text{trpB}}]} &= -k_{31}[\text{indole}][\text{e}_{6,\text{trpB}}] + k_{32}[\text{indole:e}_{6,\text{trpB}}] - k_{33}[\text{serine}][\text{e}_{6,\text{trpB}}] + k_{34}[\text{serine:e}_{6,\text{trpB}}] + k_{35}[\text{indole:e}_{6,\text{trpB}}][\text{serine:e}_{6,\text{trpB}}] \\
&\quad + k_{36}[\text{indole:e}_{6,\text{trpB}}][\text{serine}] + k_{37}[\text{indole}][\text{serine:e}_{6,\text{trpB}}] + \frac{1}{1 + [\text{gRNA}_6\text{:dCas9}]+[\text{Trp:trpR}]} - \frac{1}{1+\exp(\text{-}[\text{Trp}])[\text{Trp}]} - d_{\text{e}}[\text{e}_{6,\text{trpB}}] \\
\dot{[\text{chorismate:e}_{1,\text{trpE,trpD}}]} &= k_1[\text{chorismate}][\text{e}_{1,\text{trpE,trpD}}] - k_2[\text{chorismate:e}_{1,\text{trpE,trpD}}] - k_5[\text{chorismate:e}_{1,\text{trpE,trpD}}][\text{glutamine:e}_{1,\text{trpE,trpD}}] \\
&\quad - k_6[\text{chorismate:e}_{1,\text{trpE,trpD}}][\text{glutamine}] \\
\dot{[\text{glutamine:e}_{1,\text{trpE,trpD}}]} &= k_3[\text{glutamine}][\text{e}_{1,\text{trpE,trpD}}] - k_4[\text{glutamine:e}_{1,\text{trpE,trpD}}] - k_5[\text{chorismate:e}_{1,\text{trpE,trpD}}][\text{glutamine:e}_{1,\text{trpE,trpD}}] \\
&\quad - k_7[\text{chorismate}][\text{ glutamine:e}_{1,\text{trpE,trpD}}] \\
\dot{[\text{anthranilate:e}_{2,\text{trpD}}]} &= k_9[\text{anthranilate}][\text{e}_{2,\text{trpD}}] - k_{10}[\text{anthranilate:e}_{2,\text{trpD}}] - k_{13}[\text{anthranilate:e}_{2,\text{trpD}}][5\text{p}\alpha\text{:e}_{2,\text{trpD}}] - k_{14}[\text{anthranilate:e}_{2,\text{trpD}}][5\text{p}\alpha] \\
\dot{[5\text{p}\alpha\text{:e}_{2,\text{trpD}}]} &= k_{11}[5\text{p}\alpha][\text{e}_{2,\text{trpD}}] - k_{12}[5\text{p}\alpha\text{:e}_{2,\text{trpD}}] - k_{13}[\text{anthranilate:e}_{2,\text{trpD}}][5\text{p}\alpha\text{:e}_{2,\text{trpD}}] - k_{15}[\text{anthranilate}][5\text{p}\alpha\text{:e}_{2,\text{trpD}}] \\
\dot{[\text{n5pa:e}_{3,\text{trpC}}]} &= k_{17}[\text{n5pa}][\text{e}_{3,\text{trpC}}] - k_{18}[\text{n5pa:e}_{3,\text{trpC}}] - k_{19}[\text{n5pa:e}_{3,\text{trpC}}] \\
\dot{[12\text{c:e}_{4,\text{trpC}}]} &= k_{21}[12\text{c}][\text{e}_{4,\text{trpC}}] - k_{22}[12\text{c:e}_{4,\text{trpC}}] - k_{23}[12\text{c:e}_{4,\text{trpC}}][\text{H}^+] \\
\dot{[1\text{S2R:e}_{5,\text{trpA}}]} &= k_{26}[1\text{S2R}][\text{e}_{5,\text{trpA}}] - k_{27}[1\text{S2R:e}_{5,\text{trpA}}] - k_{28}[1\text{S2R:e}_{5,\text{trpA}}] + k_{29}[\text{e}_{5,\text{trpA}}][\text{dg3p}][\text{indole}] \\
\dot{[\text{indole:e}_{6,\text{trpB}}]} &= k_{31}[\text{indole}][\text{e}_{6,\text{trpB}}] - k_{32}[\text{indole:e}_{6,\text{trpB}}] - k_{35}[\text{indole:e}_{6,\text{trpB}}][\text{serine:e}_{6,\text{trpB}}] - k_{36}[\text{indole:e}_{6,\text{trpB}}][\text{serine}] \\
\dot{[\text{serine:e}_{6,\text{trpB}}]} &= k_{33}[\text{serine}][\text{e}_{6,\text{trpB}}] - k_{34}[\text{serine:e}_{6,\text{trpB}}] - k_{35}[\text{indole:e}_{6,\text{trpB}}][\text{serine:e}_{6,\text{trpB}}] - k_{37}[\text{indole}][\text{serine:e}_{6,\text{trpB}}] \\
\dot{[\text{trpR}]} &= -k_{38}[\text{trpR}][\text{trp}] + k_{39}[\text{trp:trpR}] + a_{\text{R}} - d_{\text{R}}[\text{trpR}] \\
\dot{[\text{trp:trpR}]} &= k_{38}[\text{trpR}][\text{trp}] - k_{39}[\text{trp:trpR}] \\
\dot{[\text{gRNA}_i]} &= -k_{40+2(i-1)}[\text{gRNA}_i][\text{dCas9}] + k_{41+2(i-1)}[\text{gRNA}_i\text{:dCas9}] + a_{\text{g}} - d_{\text{g}}[\text{gRNA}_i], i = 1, \ldots, 6 \\
\dot{[\text{gRNA}_i\text{:dCas9}]} &= k_{40+2(i-1)}[\text{gRNA}_i][\text{dCas9}] - k_{41+2(i-1)}[\text{gRNA}_i\text{:dCas9}], i = 1, \ldots, 6.
\end{aligned}
$$

Fig. 3. The full kinetics model of the Trp Pathway from *Escherichia coli*, modeled explicitly and derived from a chemical reaction network diagram from the EcoCyc database `https://biocyc.org/ECOLI/NEW-IMAGE?type=PATHWAY&object=TRPSYN-PWY&show-citations=NIL` [21].

[2] N. Boddupalli, A. Hasnain, and E. Yeung. Persistence of excitation for koopman operator represented dynamical systems. *arXiv preprint arXiv:1906.10274*, 2019.

[3] D. Chakravarti and W. W. Wong. Synthetic biology in cell-based cancer immunotherapy. *Trends in biotechnology*, 33(8):449–461, 2015.

[4] L. Chen, M. Chen, C. Ma, and A.-P. Zeng. Discovery of feed-forward regulation in l-tryptophan biosynthesis and its use in metabolic engineering of e. coli for efficient tryptophan bioproduction. *Metabolic engineering*, 47:434–444, 2018.

[5] K. Clancy and C. A. Voigt. Programming cells: towards an automated genetic compiler. *Current opinion in biotechnology*, 21(4):572–581, 2010.

[6] I. P. Crawford. Evolution of a biosynthetic pathway: the tryptophan paradigm. *Annual review of microbiology*, 43(1):567–600, 1989.

[7] A. Gyorgy, J. I. Jiménez, J. Yazbek, H.-H. Huang, H. Chung, R. Weiss, and D. Del Vecchio. Isocost lines describe the cellular economy of genetic circuits. *Biophysical journal*, 109(3):639–646, 2015.

[8] A. Hasnain, N. Boddupalli, and E. Yeung. Optimal reporter placement

in sparsely measured genetic networks using the koopman operator. *arXiv preprint arXiv:1906.00944*, 2019.

[9] A. Hasnain, S. Sinha, Y. Dorfan, A. E. Borujeni, Y. Park, P. Maschhoff, U. Saxena, J. Urrutia, N. Gaffney, D. Becker, et al. A data-driven method for quantifying the impact of a genetic circuit on its host. *arXiv preprint arXiv:1909.06455*, 2019.

[10] M. Korda and I. Mezic. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93:149–160, 2018.

[11] M. Korda and I. Mezic. Optimal construction of koopman eigenfunctions for prediction and control. *hal-02278835*, 2019.

[12] Z. Liu, S. Kundu, L. Chen, and E. Yeung. Decomposition of nonlinear dynamical systems using koopman gramians. In *2018 Annual American Control Conference (ACC)*, pages 4811–4818. IEEE, 2018.

[13] B. Lusch, J. N. Kutz, and S. L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):4950, 2018.

[14] I. Mezic. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41(1-3):309–325, 2005.

[15] S. E. Otto and C. W. Rowley. Linearly recurrent autoencoder networks for learning dynamics. *SIAM Journal on Applied Dynamical Systems*, 18(1):558–593, 2019.

[16] P. P. Peralta-Yahya, F. Zhang, S. B. Del Cardayre, and J. D. Keasling. Microbial engineering for the production of advanced biofuels. *Nature*, 488(7411):320, 2012.

[17] J. L. Proctor, S. L. Brunton, and J. N. Kutz. Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, 15(1):142161, 2016.

[18] J. L. Proctor, S. L. Brunton, and J. N. Kutz. Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, 15(1):142–161, 2016.

[19] J. L. Proctor, S. L. Brunton, and J. N. Kutz. Generalizing koopman theory to allow for inputs and control. *SIAM Journal on Applied Dynamical Systems*, 17(1):909–930, 2018.

[20] Y. Qian, C. McBride, and D. Del Vecchio. Programming cells to work for us. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:411–440, 2018.

[21] E. R. Radwanski and R. L. Last. Tryptophan biosynthesis and metabolism: biochemical and molecular genetics. *The Plant Cell*, 7(7):921, 1995.

[22] C. W. Rowley, I. Mezic, S. Bagheri, P. Schlatter, and D. S. Henningson. Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics*, 641:115, 2009.

[23] P. J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.

[24] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. Osqp: An operator splitting solver for quadratic programs. In *2018 UKACC 12th International Conference on Control (CONTROL)*, pages 339–339. IEEE, 2018.

[25] N. Takeishi, Y. Kawahara, and T. Yairi. Learning koopman invariant subspaces for dynamic mode decomposition. In *Advances in Neural Information Processing Systems*, pages 1130–1140, 2017.

[26] J. R. Van Der Meer and S. Belkin. Where microbiology meets microengineering: design and applications of reporter bacteria. *Nature Reviews Microbiology*, 8(7):511, 2010.

[27] M. J. Waites, N. L. Morgan, J. S. Rockey, and G. Higton. *Industrial microbiology: an introduction*. John Wiley & Sons, 2009.

[28] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley. A datadriven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):13071346, 2015.

[29] Z. Xie, L. Wroblewska, L. Prochazka, R. Weiss, and Y. Benenson. Multi-input rnai-based logic circuit for identification of specific cancer cells. *Science*, 333(6047):1307–1311, 2011.

[30] E. Yeung, A. J. Dy, K. B. Martin, A. H. Ng, D. Del Vecchio, J. L. Beck, J. J. Collins, and R. M. Murray. Biophysical constraints arising from compositional context in synthetic gene networks. *Cell systems*, 5(1):11–24, 2017.

[31] E. Yeung, J. Kim, Y. Yuan, J. Goncalves, and R. Murray. Quantifying crosstalk in biochemical systems. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 5528–5535. IEEE, 2012.

[32] E. Yeung, S. Kundu, and N. Hodas. Learning deep neural network representations for koopman operators of nonlinear dynamical systems. In *2019 American Control Conference (ACC)*, pages 4832–4839. IEEE, 2019.

[33] E. Yeung, Z. Liu, and N. O. Hodas. A koopman operator approach for computing and balancing gramians for discrete time nonlinear systems. *2018 Annual American Control Conference (ACC)*, 2018.