Koopman Lyapunov-based model predictive control of nonlinear chemical process systems

Abhinav Narasingam | Joseph Sang-II Kwon

Abstract
In this work, we propose the integration of Koopman operator methodology with Lyapunov-based model predictive control (LMPC) for stabilization of nonlinear systems. The Koopman operator enables global linear representations of nonlinear dynamical systems. The basic idea is to transform the nonlinear dynamics into a higher dimensional space using a set of observable functions whose evolution is governed by the linear but infinite dimensional Koopman operator. In practice, it is numerically approximated and therefore the tightness of these linear representations cannot be guaranteed which may lead to unstable closed-loop designs. To address this issue, we integrate the Koopman linear predictors in an LMPC framework which guarantees controller feasibility and closed-loop stability. Moreover, the proposed design results in a standard convex optimization problem which is computationally attractive compared to a nonconvex problem encountered when the original nonlinear model is used. We illustrate the application of this methodology on a chemical process example.

KEYWORDS
dynamic mode decomposition, Koopman operator, linear predictors, Lyapunov-based model predictive control

1 | INTRODUCTION

A key step in the analysis and control of dynamical systems is to determine the evolution of system states. Fortunately, many chemical processes and fluid systems are characterized by models that describe their evolution dynamics to near-perfect accuracy. However, more often than not, complex nonlinear models and powerful tools from differential geometry are required to perform fully-resolved simulations which become computationally intractable. This limits the capability to perform parameter estimation or design feedback control systems, which require real-time computation of dynamic solutions. Nevertheless, in several problems, a much simpler, latent structure can be identified by using physical or mathematical arguments that could make the aforementioned tasks computationally tractable. In this article, we focus on a class of linear approximations for nonlinear controlled dynamical systems. In other words, we obtain a finite dimensional linear representation that can predict the future state (or output) of an underlying nonlinear system given the current state and applied inputs. Such a linear structure allows for the use of established linear control design methodologies for nonlinear dynamical systems.

The methodology discussed in this manuscript is based on the Koopman operator, which is first introduced in the seminal works of Koopman and Von Neumann.\textsuperscript{1,2} The Koopman formalism affords an operator-theoretic perspective to classical dynamical systems theory. In his early work, Koopman showed that nonlinear dynamical systems could be analyzed using an infinite dimensional linear operator on the space of observable scalar functions. In other words, the Koopman operator is a linear operator that describes the temporal evolution of scalar observables (which are essentially functions of system states)
which are driven by the underlying nonlinear dynamics. Therefore, one of the key features of such a formalism is the availability of linear models (albeit possibly high-dimensional) that allow future state prediction and scalable reconstruction of the underlying dynamics from measurement data.

In practice, however, this is attractive only when one can handle the infinite dimensional linear system, which is traded for an underlying nonlinear system. Fortunately, recent advances in numerical techniques and abundant availability of data have led researchers to develop finite dimensional approximations to the Koopman operator, which are useful to model the dynamics and synthesize controllers. For example, numerical implementations of generalized Fourier and Laplace analysis and most recently dynamic mode decomposition (DMD) are several algorithms that can numerically approximate the spectral decomposition of the Koopman operator. Due to the ease of implementation, DMD has found applications in a broad range of domains, including fluid dynamics, epidemiology, neuroscience, robotics, and video processing.

However, for a highly nonlinear system, the assumption of a finite linear relation might not work well, especially when the space of the scalar observables is very limited (as in the case of DMD where the observables are the states alone) to describe the rich dynamics observed in highly nonlinear systems. In order to tackle this, on one hand, our own previous work has focused on generating accurate approximations to highly nonlinear processes such as hydraulic fracturing by tailoring the basis functions to capture local dynamics of every portion of the system trajectory. On the other hand, recently, an extension to DMD was developed by Williams et al. called the extended dynamic mode decomposition (EDMD) which seeks to enrich the observable space by using nonlinear functions. The EDMD algorithm approximates the leading eigenfunctions and eigenvalues of the Koopman operator from time series data and a dictionary of observables that spans a subspace of the scalar observable functions. Specifically, EDMD identifies the "slow" subspace of the Koopman operator, which approximates the long term dynamics of observables by neglecting the fast transients. Once identified, they enable the reconstruction of system states as a linear combination of the Koopman eigenfunctions. For a comprehensive treatment of the EDMD method, we refer the readers to Williams et al. An alternative way, one that is more relevant to model identification and thus we follow in this manuscript, is to envision EDMD as a state-space transformation, so that the dynamics appear to be linear, to accurately compute a finite dimensional approximation of the controlled Koopman operator. Essentially, the EDMD procedure boils down to a nonlinear transformation of the data (lifting to the space of observables) followed by solving a regression problem to construct the required linear predictors. The effectiveness of obtaining a linear system that approximates the original nonlinear system in this way is well established where researchers have demonstrated its superior performance over several local linearization methods. Moreover, EDMD is a completely data-driven method and requires no knowledge of the original nonlinear model.

The attractiveness of this data-driven approach for approximating the Koopman operator spectrum has sparked increased research activity in the analysis and control of nonlinear dynamical systems. The ability to embed nonlinear dynamics in a linear model is particularly useful since the linear structure allows for the control of nonlinear systems using established techniques such as model predictive control (MPC) with the computational complexity comparable to that of linear systems (with the same number of inputs and states). Recently, several research works highlighted the use of the Koopman operator based data-driven methods for the predictive control of nonlinear systems. However, in these works, the tightness of the linear predictors is not established and the closed-loop guarantees on stability are not well studied. This is critical because, in practice, EDMD models are known to have closure issues as there is no guarantee that the selected observables form Koopman invariant subspaces. This is especially true for the systems with multiple steady states (as a linear system can only characterize a single steady state). This may lead to spurious dynamics, which may subsequently lead to unstable controllers. To address this issue, Huang et al. proposed a stabilizing feedback controller which relies on control Lyapunov function (CLF) and thus achieves stabilization in the truncated Koopman eigenfunction space. The authors comment on optimality of the controller using the principle of inverse optimality; however, it does not account for explicit state and input constraints. Moreover, the strong reliance on a CLF, while providing stability, can lead to suboptimal performance when the choice of the CLF is not judicious. Another approach seeks to represent the system using eigenfunctions of the Koopman operator directly, which are guaranteed to span an invariant subspace. However, in practice, it may be more difficult to obtain the Koopman eigenfunctions than solving the original nonlinear system.

Motivated by these considerations, this work proposes a systematic approach for the design of a stabilizing feedback predictive controller for nonlinear systems. To this end, we propose to integrate Koopman based linear predictors with Lyapunov-based model predictive control (LMPC) scheme. LMPC is known for its explicit characterization of stability properties and guaranteed closed-loop stabilization of nonlinear systems in the presence of state and input constraints. LMPC is a powerful tool that combines the complementary properties of the CLF (stability) and predictive control (optimality) approaches. Recently, researchers have also successfully designed LMPC that guarantees closed-loop stability even when (linear) empirical models are used in the design as long as a set of assumptions is satisfied. Motivated by this, in this work, we integrate Koopman linear predictors with LMPC for the stable control of nonlinear dynamical systems. Specifically, we perform a lifting (nonlinear transformation) of the system states to an observable space using nonlinear functions and obtain linear representations of the system in the observable space. We then design LMPC based controllers in the same observable space that stabilize the nonlinear system. In addition to the stability properties, the proposed method introduces a slight modification that exploits the construction of linear predictors and the structure of LMPC to yield a completely standard convex (quadratic) optimization problem (provided the original state and input constraints are linear) within the LMPC framework. Therefore, the large library of efficient solvers available for linear MPC can be readily used to solve the proposed Koopman based LMPC scheme for the control of nonlinear systems.
The remainder of this article is organized as follows. First, we provide a brief introduction to the Koopman formalism and present the data-driven identification of a nonlinear system as a linear model using the EDMD methodology. Then, a convex optimization-based LMPC formulation embedding the developed linear predictors for the feedback control of nonlinear systems is proposed in the next section. This is followed by an illustration of the proposed method for the control of a second-order continuously stirred tank reactor (CSTR) process which includes a series of numerical simulation results on the accuracy of the Koopman based linear predictors and the performance of the closed-loop system. Finally, we provide a few concluding remarks.

2 | DATA-DRIVEN IDENTIFICATION OF LINEAR PREDICTORS

2.1 | Background on Koopman operator

Let us consider a discrete-time nonlinear controlled dynamical system given by

\[ x_{k+1} = F(x_k, u_k) \]  

where \( x \in \mathcal{M} \subseteq \mathbb{R}^n \) is the vector of state variables sampled discretely in time so that \( x_k = x(\Delta t) \) with sampling time \( \Delta t \), \( u \in \mathcal{U} \subseteq \mathbb{R}^m \) is the vector of control inputs and \( F: \mathcal{M} \to \mathcal{M} \) is the evolution operator that represents the dynamics which map the system states forward in time.

In his seminal work, Koopman realized an alternative operator-theoretic perspective of Equation (1) in terms of the evolution of functions of state-space, also called observables, \( \phi(x) \) with \( \phi: \mathcal{M} \to \mathbb{R} \). He showed that there exists an infinite dimensional (because it acts on functions) linear operator \( \mathcal{K} \) that advances these observables forward in time. Although Koopman operator theory was initially developed for uncontrolled dynamical systems, to make it more relevant for the purposes of this manuscript, we adapt it for the controlled setting. To this end, we consider an augmented state-space given by the product of the original state-space and the space of all control inputs.

\[ x_{aug}^k = \begin{bmatrix} x^k \\ u^k \end{bmatrix}, \quad x_{aug}^k \in \mathbb{R}^n \times \mathcal{U} \]  

where \( \mathcal{U} \subseteq \mathbb{R}^m \) denotes the space of all input sequences \( \{u_i\}_{i=0}^\infty \) with \( u_i \in \mathcal{U} \). The definition of the Koopman operator in the controlled setting depends on the type of input.\(^{29}\) In this manuscript, we consider only closed-loop control and so the input is generated using a state-dependent control law, \( u = h(x) \). Therefore, the dynamics of the extended state is described by

\[ x_{aug}^{k+1} = F_{aug}(x_{aug}^{k}) = \begin{bmatrix} F(x_k, u_k) \\ h(F(x_k, u_k)) \end{bmatrix} \]  

Now, the action of the Koopman operator, \( \mathcal{K}: \mathcal{H} \to \mathcal{H} \), associated with the observables \( \phi \) is given by

\[ \mathcal{K}\phi = \phi \circ F_{aug} \]  

where \( \circ \) is the function composition operator. Therefore, the evolution of observables is governed by the Koopman operator as

\[ \mathcal{K}\phi(x_{aug}^{k+1}) = \phi(F_{aug}(x_{aug}^{k})) = \phi(F(x_k, u_k)) \]  

for each \( \phi \) belonging to infinite dimensional Hilbert space \( \mathcal{H} \). In other words, the Koopman operator for a given closed-loop system is equivalent to the Koopman operator for the following autonomous system

\[ \mathcal{K}\phi(x, h(x)) = \phi(x_{aug}) = \phi(x_{aug+1}) \]  

Simply put, the dynamics of the system defined by \( F \) in Equation (1) and the one defined by \( \mathcal{K} \) in Equation (5) are different parametrizations of the same underlying behavior. It is easy to observe that the Koopman operator is linear on the space of observables, that is, the below equation holds true.

\[ [\mathcal{K}(\alpha \phi_1 + \phi_2)](x) = \alpha [\mathcal{K}\phi_1](x) + [\mathcal{K}\phi_2](x) \]  

Defining the system using Equation (5) is particularly attractive because of the fact that in the observable space the evolution is linear even if the actual system is nonlinear. Therefore, it enables advanced prediction and control of the underlying nonlinear system using comprehensive theory available for linear systems. Additionally, in many processes it is not possible to obtain all the system states whereas an observable could, for instance, be a measurement or a sensor probe. Therefore, it is beneficial to use the Koopman formalism. However, since we are only trading a nonlinear system for an infinite dimensional linear one, it is not feasible unless we can practically determine finite dimensional approximations to the Koopman operator without a great loss in accuracy.

2.2 | Finite dimensional approximation

In the realm of Koopman theory, several data-driven methods exist to approximate the Koopman operator which include variational approach of conformation dynamics,\(^{30,31}\) and the more widely used DMD. For highly nonlinear systems, DMD may fail to capture the nonlinear transients because, by construction, DMD uses linear observable functions, \( \phi(x) = x \) (basically, the observable function is an identity operator on the system states). In order to improve the approximations, researchers have tried to enrich the model using nonlinear observable functions giving rise to EDMD. Specifically, in this improved algorithm, the Koopman operator is approximated as a linear map on the span of a finite observable function basis. EDMD was developed as a practical way to provide spectral decomposition of the Koopman operator, but it was shown that it can be used for time-domain prediction of the trajectories of controlled nonlinear dynamical systems. In this work, we use the EDMD algorithm for the approximation of \( \mathcal{K} \), thereby constructing linear predictors for the system described in Equation (1).
2.2.1 | EDMD for controlled systems

To construct a finite dimensional approximation to the Koopman operator for the controlled system in Equation (5), EDMD requires,

1. A data set of snapshot pairs satisfying the dynamical system in Equation (1) (and by extension Equation (5)) which can be organized in the following matrices

\[ X = [x_1, x_2, \ldots, x_k], \quad Y = [y_1, y_2, \ldots, y_k], \quad U = [u_1, u_2, \ldots, u_k] \quad (8) \]

Note that we use y instead of \( x_{k+1} \) here because the data above need not be temporally ordered as long as it satisfies \( y_k = F(x_k, u_k) \).

2. A finite library of nonlinear observable functions,

\[ \phi(x^{aug}) = \begin{bmatrix} \phi_1(x^{aug}), \phi_2(x^{aug}), \ldots, \phi_{N_{\phi}}(x^{aug}) \end{bmatrix} \quad (9) \]

where \( \phi_i \in \mathcal{O} \subset \mathbb{R}^n \times \ell(U) \rightarrow \mathbb{R}, i = 1, \ldots, N_{\phi} \) and \( N_{\phi} \) is the total number of observable functions in the dictionary.

The EDMD algorithm then seeks to solve a least-squares problem to obtain \( K^{aug} \in \mathbb{R}^{N_{\phi} \times N_{\phi}} \) which is the transpose of the finite dimensional approximation to the Koopman operator, \( K \):

\[ \min_{K^{aug}} \sum_{i=1}^{K} \|y^{aug}(\phi(x^{aug})) - K^{aug}\phi(x^{aug})\|_2^2 \quad (10) \]

where \( y^{aug} = [y \ T] \in \mathbb{R}^n \times \ell(U) \). One thing to note here is that since we are not interested in predicting the future values of inputs, without loss of generality we can assume that,

\[ \phi(x^{aug}) = \begin{bmatrix} \phi(x) \\ U \end{bmatrix}, \quad \phi(y^{aug}) = \begin{bmatrix} \phi(y) \\ U \end{bmatrix} \quad (11) \]

which is to say that the nonlinear observable functions are applied to the system states alone and not the inputs. We can then disregard the last \( m \) components of each of the terms in Equation (11) \( (N_{y} = N + m) \), and using the notation \( K = [A, B] \) to denote the first \( N \) rows of \( K^{aug} \) leads to the following minimization problem

\[ \min_{A, B} \sum_{i=1}^{N} \|y_i - A\phi(x_i) - Bu_i\|_2^2 \quad (12) \]

The practical solution to the above equation is obtained by using regularization via a truncated singular value decomposition and the value of \( K \) that minimizes Equation (12) is given by

\[ K = [A, B] = \phi_{xy}\phi_{xx}^+ \quad (13) \]

where \( ^+ \) denotes the pseudoinverse and the data matrices are given by

\[ \phi_{xx} = \begin{bmatrix} \phi_1 \ \\ U \end{bmatrix} \begin{bmatrix} \phi_1 \\ U \end{bmatrix}^T, \quad \phi_{xy} = \phi_y \begin{bmatrix} \phi_1 \ \\ U \end{bmatrix} \quad (14) \]

\[ \phi_x = [\phi(x_1), \ldots, \phi(x_K)], \quad \phi_y = [\phi(y_1), \ldots, \phi(y_K)] \quad (15) \]

Please note that any solution to Equation (13) is a solution to Equation (12), but the formulation of Equation (13) has an advantage of being independent of the number of data samples, \( K \). The predictors obtained using the above EDMD algorithm will be in the form of a controlled linear dynamical system given below

\[ z_{k+1} = Az_k + Bu_k, \quad z_0 = \phi(x_0) \quad (16) \]

where \( z \in \mathbb{R}^N \) is the lifted state in the observable space, \( \hat{x} \) is the prediction of \( x \) and \( A \in \mathbb{R}^{N \times N}, B \in \mathbb{R}^{N \times m} \) and \( C \in \mathbb{R}^{n \times N} \) are the matrices that describe the system dynamics in the lifted space. In practice, the nonlinear observable functions may also contain the state observable \( \phi(x) = x \) and therefore, typically the solution to matrix \( C \) is trivial and can be obtained by \( C = [I_n, 0] \).

The concept of linearizing dynamics is not new, but unlike local linearization around equilibrium points, the Koopman operator provides a global linearization of the original dynamics provided the set of (observable) basis functions is “rich enough.” So, the main source of error stems from our choice and the finite size of the set of basis functions used in the approximation of the Koopman operator. Nevertheless, the convergence of EDMD approximations has been recently established under the assumption that the dimension of the subspace \( N \) approaches infinity. \( ^{32} \) Therefore, this error can be made arbitrarily small by considering a large set of basis functions. However, making a shrewd choice of the basis functions for unknown systems is still an active research area and a few studies suggest that canonical choices like radial basis functions and Hermite polynomials are a good starting point. \( ^{14, 19} \) Several other approaches propose using deep learning methods to discover these dictionaries. \( ^{33, 34} \) Nevertheless, when the system model is known or a limited knowledge on the functional forms present in the model is available, it can be readily incorporated in the dictionary as will be shown with an illustrative example in the subsequent sections.

3 | FEEDBACK CONTROL DESIGN

The methodology presented in the last section allows us to construct accurate linear predictors of the dynamical system in the form of Equation (16). Here, we will utilize these predictors to design a model based control scheme to regulate the original nonlinear system. To this end, the LMPC is a powerful tool for the design of a stabilizing
feedback controller, which is also optimal with respect to the state and input constraints. This is particularly attractive in this context because the linear predictors obtained using regression methods for approximating the Koopman operator are known to have closure issues which may lead to unstable controllers. More specifically, the premise of EDMD is that if the set of observable functions is "rich enough" we can then reconstruct the original system states from these observables. However, since the dictionary depends on our choice of variables, in most applications, it is possible to miss some elements and therefore the Koopman operator cannot be fully represented by the chosen dictionary. As a result, the subspace spanned by the dictionary of observables is not guaranteed to be forward invariant which may lead to closure issues.\textsuperscript{13,14,23} Moreover, as described in the above section, we compute only a projection of the Koopman operator rather than the infinite dimensional Koopman operator, $K$, itself. Therefore, the tightness of these linear predictors is not guaranteed and this may lead to a closed-loop design that is unstable.

To address this issue, in this work, we use the Koopman based linear predictors in an LMPC framework, which allows for an explicit characterization of the stability region and provides guarantees on controller feasibility and closed-loop stability.

### 3.1 Lyapunov-based model predictive control

In this section, we briefly describe the LMPC design proposed by Mhaskar et al.\textsuperscript{26} We recall that LMPC is a control strategy that is designed based on an explicit stable (albeit not optimal) control law $h(x)$ and a Lyapunov constraint by virtue of which the controller is able to stabilize the closed-loop system. Before we go into the formulation, the LMPC framework requires the following assumption to be satisfied by the nonlinear dynamical system considered in Equation (1).

**Assumption** Stabilizability assumption. The nonlinear systems considered are restricted to a class of stabilizable systems which implies the existence of a feedback control law $u(t) = h(x)$ that satisfies input constraints for all $x(t)$ inside a given stability region and renders the origin of the closed-loop system asymptotically stable. This is equivalent to assuming that there exists a Lyapunov function for the nominal system (Lyapunov theorem).

Based on the above assumption, the predictive control of the system of Equation (1) under LMPC is formulated as follows:

$$\min_{u_k, \ldots, u_{k+N_p}} \sum_{j=1}^{N_p} \left| \mathbf{x}_{k+j} \right|^2 + \sum_{j=1}^{N_p} \left| u_{k+j-1} \right|^2 \quad \text{(17a)}$$

subject to:

$$\mathbf{x}_{k+j} = F_h(x_{k+j-1}, u_{k+j-1}), \quad j = 1, \ldots, N_p \quad \text{(17b)}$$

$$\mathbf{x}_k = \mathbf{x}_k \quad \text{(17c)}$$

$$V(\mathbf{x}_{k+j}) - V(\mathbf{x}_k) \leq V(F_h(x_{k}, h(x_k)))) - V(\mathbf{x}_k) \quad \text{(17d)}$$

where $x_{k+j}$ is the predicted state trajectory with initial (measured) state $x_k$, $u_{k+j}$ denote the calculated manipulated input variables $j$ time steps ahead and $N_p$. $N_u$ denotes the prediction and control horizons, respectively. The operator $\| \cdot \|_2^2$ denotes the weighted Euclidean norm defined for an arbitrary vector $x$ and weighting matrix $Q$ as $\|x\|_2^2 = x^T Q x$ and $W \in \mathbb{R}^{m \times m}$ denote the positive definite weighting matrices for the state and input vectors respectively. Furthermore, $V(x)$ is the Lyapunov function associated with the explicit control law $h(x)$. The manipulated input (optimal solution) of the above system under the LMPC control law is obtained as $u^* = [u_1^*, \ldots, u_{k+N_p-1}^*]$ and only the first value of $u_k^*$ is applied to the closed-loop system for the next sampling time period $[k, k+1)$ and the procedure is repeated until the end of operation.

In the LMPC formulation of Equations (17a–17c), Equation (17a) denotes a performance index that is to be minimized, Equation (17b) is the nominal model of the system of Equation (1) used to predict the future evolution of the states and Equation (17c) provides the initial state which is obtained as a measurement of the actual system state. In addition to these constraints, the LMPC formulation considers the Lyapunov constraints, Equations (17d and 17e). Equation (17d) ensures that the closed-loop system stays within the stability region $\Omega = \{x \in \mathbb{R}^n : V(x) \leq r\}$ and Equation (17e) guarantees that the rate of change of the Lyapunov function, $V(x)$, at time $k$ is smaller than or equal to that of the value obtained if the explicit control law $h(x)$ is applied to the closed-loop system in a sample-and-hold fashion. These constraints allow the LMPC controller to inherit the stability properties, that is, it possesses the same stability region $\Omega$ as the controller $h(x)$. This implies that the (equilibrium point of) closed-loop system is guaranteed to be stable for any initial state inside the region $\Omega$, provided the sampling time $\Delta$ is sufficiently small. Note that because of this property, the LMPC does not require the terminal constraint generally used in a traditional MPC setting. Additionally, the feasibility of LMPC is also guaranteed because $u = h(x)$ is always a feasible solution to the above optimization problem. Even though the above formulation does not explicitly consider the state and input constraints, they can be readily incorporated.

### 3.2 Integrating EDMD with LMPC

In this section, we show how the linear predictors based on EDMD can be embedded into the most general LMPC formulation described above. Additionally, we propose a simple modification, which will result in a standard convex quadratic optimization problem.

In order to use the Koopman based model in the LMPC framework, the problem must be formulated in the observable (lifted) space. Therefore, at each time step $t_k$ the predictions of the system trajectory are initialized from the lifted state $z_k = \phi(x_k)$. Similarly, the objective function and the state constraints are all transformed to the lifted space. In addition, we propose to include the Lyapunov function in
functions, we can effectively transform the nonlinear Lyapunov condition, the decrease of which dictates the stability of a nonlinear system. See that a Lyapunov function is indeed a particular observable function space to transform the nonlinear LMPC formulation to a standard convex quadratic optimization problem. Intuitively, we can write the Lyapunov constraint of Equation (17d) as follows:

\[ V(x_{k+1}) - V(x_k) \leq V(F(x_k, h(x_k))) - V(x_k) \]  

(18)

Substituting \( V(x_k) = Dz_k \) into the above equation, we get

\[ Dz_{k+1} - Dz_k \leq V(F(x_k, h(x_k))) - V(x_k) \]  

(19)

We exploit the linear structure of the nonlinear system in the observable space to transform the nonlinear LMPC formulation to a standard convex quadratic optimization problem. Intuitively, we can see that a Lyapunov function is indeed a particular observable function, the decrease of which dictates the stability of a nonlinear system. The motivation behind this comes from the fact that operator-theoretic approaches have been successfully applied (albeit implicitly) to nonlinear systems for global stability analysis and control. Particularly, Lyapunov’s stability criterion relies on the operator-theoretic framework which uses a (Lyapunov) function description in the infinite dimensional function space to analyze the stability of a system, rather than a pointwise description that traditionally studies stability with respect to equilibrium points in the finite dimensional state-space.

Based on the above modifications, the proposed Koopman-LMPC framework, solves the following optimization problem at each time step \( k \) of the closed-loop operation.

\[
\begin{align*}
\min_{a_{k},...a_{k-N_{p}+1}} \quad & \sum_{j=1}^{N_{p}} \left( (Cz_{j})^{T}W(Cz_{j}) + u_{j-1}^{T}Ru_{j-1} \right) \\
\text{s.t} \quad & z_{k+j} = Az_{k+j-1} + Bu_{k+j-1}, \quad j = 1,...,N_{p} \\
& z_{k} = \phi(x_{k}) \quad (20c) \\
& E_{k+j}z_{k+j} + H_{k+j-1}u_{k+j-1} \leq b_{k+j}, \quad j = 1,...,N_{p} \quad (20d) \\
& Dz_{k+1} \leq r, \quad j = 1,...,N_{p} \\
& Dz_{k+1} - Dz_{k} \leq V(F(x_{k}, h(x_{k}))) - V(x_{k}) \quad (20f)
\end{align*}
\]

In the above formulation, Equation (20a) represents the quadratic cost function which has been transformed to the observable state-space, Equation (20b) denotes the linear predictors based on EDMD that describe the evolution of the lifted states with matrices \( A \) and \( B \) constructed as described in the above section, Equation (20c) determines the initialization \( z_{0} \) using the observable functions mapping \( \phi \). Equation (20d) gives the state and input polyhedral constraints using the matrices \( E \in \mathbb{R}^{n\times N_{c}}, H \in \mathbb{R}^{n\times m} \) and the vector \( b \in \mathbb{R}^{n} \), and Equations (20e) and (20f) correspond to the Lyapunov constraints (Equations (17d and 17e)) which guarantee the closed-loop stability of the system. The above optimization problem Equations 20a - 20f is parametrized by the current state \( x_{k} \) of the nonlinear system (see Equations 20c and 20f). Please note that Equation (20f), although appears nonlinear, is still a linear constraint because it requires computation of the Lyapunov function derivative of the system under the nonlinear control law at the current state value alone, which is available from the measurement. The above optimization problem defines a feedback controller \( u_{k} = u_{k}^{*} \), where \( u_{k}^{*} \) denotes the first component of the optimal solution to problem Equations 20a - 20f parametrized by the current state \( x_{k} \). This problem is solved at each time step \( k \) of the closed-loop operation.

Several important features of the proposed method are summarized below:

1. The optimization problem Equation 20 is a convex quadratic programming problem even when the original dynamic model is nonlinear. Therefore, it avoids solving difficult nonconvex optimization problems and allows for a fast evaluation of the control input.

2. The evaluated controller possesses the same stability region, \( \Omega \), as that of the nonlinear control law \( h(x) \), thereby mitigating the closure issues that otherwise plague the linear predictors developed using the EDMD algorithm that solves a regression problem.

3. As previously shown by Korda and Mezić, if nonlinear objective function and/or state constraints are present in the system, they can be absorbed into the observable function dictionary \( \phi \). The case for nonlinear objective function is prevalent in economic MPC design where the goal is to optimize the cost using control. For example, consider the following nonlinear objective function and constraints,

\[
J = \sum_{i=1}^{N_{p}} J_{i}(x_{k}) \\
c_{i}(x_{k}) \leq 0, \quad i = 1,...,N_{p}
\]

In such cases, without loss of generality, one can consider the observable function dictionary as \( \phi = [x, J_{1}, J_{2},...,J_{N_{p}}} \] \( \in \mathbb{R}^{N} \).

The system matrices \( A,B \) are obtained as described in the EDMD algorithm. The remaining matrices for the objective function and the constraints can simply be determined as \( q_{i} = [0_{1\times n}, 0_{1\times N_{p}}, 0_{1\times N_{p}}] \) and \( E_{i} = [0_{1\times n}, 0_{1\times N_{p}}, 0_{1\times N_{p}}] \). Therefore, the nonlinear functions can be written in the lifted space as follows:

\[
J_{i} = \sum_{j=1}^{N_{p}} q_{i}^{T} z_{j} \\
E_{i}^{T} z_{j} \leq 0, \quad i = 1,...,N_{p}
\]

Please note that such an adaptation has already been shown in this manuscript where the nonlinear Lyapunov function has been
incorporated within the observable library and the resulting Lyapunov constraints have been recast as linear constraints in the lifted space transforming the nonlinear MPC problem into a convex optimization problem. Here, we would like to remark that such transformation would increase the dimension of the lifted space (to incorporate additional nonlinear functions) and may result in computational expense. This is a trade-off of the proposed framework where we swap low dimensional nonlinear optimization problem for a high-dimensional but linear (convex) optimization problem. In some special cases, where the objective function is simply a quadratic function it is not necessary to linearize it as there are many capable solvers for handling quadratic optimization problems. The example presented in this manuscript demonstrates this flexibility.

Remark 1 Please note that in order to use the proposed Koopman based LMPC formulation, a CLF for the nominal system must be available a priori. This is a limitation of the proposed approach; however, for simple systems such CLFs can be obtained by solving the Ricatti equation using the matrices of the associated linearized system. Furthermore, the stability region Ω used in the LMPC formulation should be characterized using extensive closed-loop simulations under the nonlinear explicit controller h(x). Under this control law the stability region is determined as a sufficiently large level set where the time-derivative of the Lyapunov function along the closed-loop state trajectories is negative.

4 | APPLICATION TO CHEMICAL PROCESS EXAMPLE

In this section, we illustrate the proposed method on a well-mixed, nonisothermal CSTR example. We consider an irreversible exothermic reaction, \( A \rightarrow B \), where the conversion of reactant A to product B is governed by second-order kinetics. The feed to the reactor is pure A at a constant flow rate \( F \), inlet temperature \( T_{in} \) and a molar concentration of \( C_{A0} \). Since the process is nonisothermal, heat is either removed or provided to the system through the reactor jacket. The dynamic model of the system can be easily obtained by applying mass and energy balances as given below:

\[
\frac{dC_A}{dt} = \frac{F}{V} (C_{A0} - C_A) - k_0 e^{\frac{E}{R T}} C_A^2 \\
\frac{dT_r}{dt} = \frac{F}{V} (T_{in} - T_r) - \frac{\Delta H}{\rho C_p} k_0 e^{\frac{E}{R T}} C_A^2 + \frac{Q}{\rho C_p V} 
\]

where \( C_A \) and \( T_r \) are the concentration of reactant A and the reactor temperature, respectively (system states), \( Q \) is the rate of heat input to the reactor (manipulated input), \( V \) is the volume of the reactor, \( k_0, E, \Delta H \) denote the pre-exponential factor, activation energy and the enthalpy of the reaction, respectively, and \( \rho, C_p \) denote the density of the fluid and its heat capacity, respectively.

The above system is characterized by three steady states (two asymptotically stable and one unstable) for \( Q_0 = 0 \) KJ/hr which is the steady-state value of the manipulated input. The control objective is to operate the CSTR in a compact state-space around the steady-state given by \( [C_{A0}, T_{in}] = [1.22 \text{ kmol/m}^3, 438.2 \text{ K}] \). The given steady-state is open-loop stable. The reason for choosing this steady-state for operation is that it has been shown to maximize the time-averaged production rate of product B. Throughout the rest of this manuscript, the dynamic model in Equation (21) is used in its deviation form that defines the state and input vectors as \( x = [C_A - C_{A0}, T_r - T_{in}]^T \) and \( u = [Q - Q_0] \), respectively. Therefore, the origin is a stable equilibrium point for the new system.

4.1 Model identification and validation

To demonstrate the application of EDMD, we assume that the nonlinear dynamic model for the CSTR, Equation 20, is not available and a linear time invariant state-space model (in the observable space) will be identified and validated using the algorithm described in the preceding section. In order to obtain the predictors, we first collect the data by solving the set of ODEs using the Matlab solver ode45 with an integration time step of \( h = 1 \times 10^{-4} \) hr. The values of the process parameters used in the simulations are presented in Table 1. Since the goal is to design a controller, we used closed-loop simulation data to build the linear predictors. The data required to construct these predictors using the EDMD algorithm is obtained from 250 simulated trajectories over an operating period of 1 hr. A total of 1,000 time samples per trajectory are used to populate the required data matrices. Each trajectory starts with an initial condition generated randomly over an interval around the operating steady-state, and is subjected to an input signal determined by a feedback controller applied to the process. The manipulated inputs are bounded as \( -5 \times 10^5 \leq Q \leq 5 \times 10^5 \) KJ/hr. We assume that the output is equal to the state (i.e., a state feedback problem). This results in the data matrices \( X \) and \( Y \) of size \( 2 \times 25 \cdot 10^4 \) and the matrix \( U \) of size \( 1 \times 25 \cdot 10^4 \).

For unknown systems, there is no established way to select the dictionary although some canonical choices like radial basis functions have been proposed as a good starting point. However, in this example, we wanted to show the flexibility of the method where any available knowledge of the system can be easily incorporated into the dictionary. Specifically, the observable functions \( \phi_i \) are chosen as the

<table>
<thead>
<tr>
<th>TABLE 1</th>
<th>Parameters of the continuously stirred tank reactor (CSTR) process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property</td>
<td>Unit</td>
</tr>
<tr>
<td>Inlet temperature, ( T_0 )</td>
<td>K</td>
</tr>
<tr>
<td>Volume, ( V )</td>
<td>m(^3)</td>
</tr>
<tr>
<td>Pre-exponential factor, ( k_0 )</td>
<td>m(^3)/hr/kmol</td>
</tr>
<tr>
<td>Pure fluid density, ( \rho )</td>
<td>kg/m(^3)</td>
</tr>
<tr>
<td>Heat capacity, ( C_p )</td>
<td>kJ/kg/K</td>
</tr>
<tr>
<td>Flow rate, ( F )</td>
<td>m(^3)/hr</td>
</tr>
<tr>
<td>Activation energy, ( E )</td>
<td>kJ/kmol</td>
</tr>
<tr>
<td>Enthalpy of reaction, ( \Delta H )</td>
<td>kJ/kmol</td>
</tr>
<tr>
<td>Gas constant, ( R )</td>
<td>kJ/kmol/K</td>
</tr>
</tbody>
</table>
state itself, quadratic and exponential functions of the state, and a quadratic Lyapunov function so the dimension of the observable state-space is \( N = 7 \).

\[
\phi = \begin{bmatrix} x_1 & x_2 & x_1^2 & x_2^2 & e^{1/4} & e^{3/2} & x^T P x \end{bmatrix}^T
\]  

where

\[
P = \begin{bmatrix} 1060 & 22 \\ 22 & 0.52 \end{bmatrix} \tag{23}
\]

The matrix \( P \) is obtained using the linearized matrices of the nominal system. The stability region \( \Omega = \{ x : V(x) \leq r \} \) is characterized as a sufficiently large level set where the time-derivative of the Lyapunov function along the closed-loop state trajectories is negative using extensive simulations under the explicit controller \( h(x) \). Please refer to Reference \(^{36}\) for more details. Using the collected data and the constructed library \( \phi \), we determined approximate linear predictors for the CSTR model described in Equation (21).

**Remark 2** The choice of this library is highly dependent on the specific process of interest. For example, in the case of hydraulic fracturing where only the proppant concentration at different spatial locations are important it was shown that considering monomials of the concentration terms yields accurate reduced-order models that can be used in the design of feedback controllers. \(^{37}\) Therefore, it is intuitive to use these monomials as the observable functions in the EDMD algorithm.

In order to evaluate the performance of the Koopman based linear predictors, we compared its prediction performance with two commonly used methods for obtaining linear models.

1. Linear model obtained using local linearization of the dynamics at the stable steady-state.
2. Linear model obtained using a subspace identification method, multivariable output error state-space (MOESP). \(^{38}\)

The identified matrices for the localized linear model of the CSTR in continuous-time are given below:

\[
A = \begin{bmatrix} -27.76 & -0.473 \\ 1130 & 18.1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0.0043 \end{bmatrix} \tag{24}
\]

where as to obtain the MOESP based model, we used the input/output data to regress a second-order linear model using \texttt{n4sid} function in Matlab.

Figures 1 and 2 show the output predictions of the above described linear models for two different initial conditions (randomly) chosen from the training data set. From the figure, it can be seen that a relatively good agreement between the true model and the identified linear models is achieved with respect to the training data. The associated input trajectories are depicted in Figure 3. To validate this model we used additional responses generated by applying a series of step, impulse and random input signals to the process initiated from a given initial condition. Specifically, (a) a step of magnitude \( Q = 5,000 \text{ kJ/hr} \) was given to the heat rate input starting at \( t = 0.5 \text{ hr} \), (b) an impulse was numerically simulated using a rectangular pulse of magnitude \( Q = -10,000 \text{ kJ/hr} \) for one sampling period starting at \( t = 0.5 \text{ hr} \) and (c) a random heat rate profile satisfying the input bounds was generated using \texttt{rand} function in Matlab and provided to the system throughout the operating period.

The results of the model validation are shown in Figures 4–6. As can be seen from the figures, the best prediction performance is achieved by Koopman based linear predictors followed by the local linearization model. Please note that the prediction quality of the MOESP based model can be enhanced by using a higher-order model. It is well known that for a CSTR process, the local linearization around the stable steady-state provides a very good approximation to the original system. This is clearly reflected in the above figures. Nevertheless, the Koopman based linear models perform even better by virtue of enriching the data with nonlinear observable functions. To assess this quantitatively, the relative root mean squared errors (RMSEs) of all the models starting from the same initial condition subjected to random input profiles are averaged over 100 simulations and presented in Table 2.

The average RMSE is computed as

\[
\text{RMSE} = \frac{\| \bar{x}_{\text{approx}} - \bar{x}_{\text{true}} \|_{\text{fro}}}{\| \bar{x}_{\text{true}} \|_{\text{fro}}} \quad \text{Average RMSE} = \frac{1}{100} \sum_{i=1}^{100} \text{RMSE}_i \tag{25}
\]

![FIGURE 1](https://wileyonlinelibrary.com)

**FIGURE 1** Prediction comparison of the continuously stirred tank reactor (CSTR) response with respect to the input profile shown in Figure 3a that was randomly chosen from the set of training data [Color figure can be viewed at wileyonlinelibrary.com]
where $\|\cdot\|_F$ is the Frobenius norm. From the table, we observe that the Koopman based linear model is superior to both the local and MOESP models in terms of the prediction accuracy. To test the closed-loop performance of these predictors, in the next section, we utilize them in the design a feedback controller.

4.2 Closed-loop simulation results

The formulation used in Equation 20 needs an explicit controller (see Equation 20e) to be constructed based on which the Lyapunov constraint and the stability region of the closed-loop operation are characterized. To this end, we design a Lyapunov-based controller developed initially by Sontag for a class of input affine nonlinear dynamical systems of the form:

$$ x = f(x) + g(x) \cdot u $$ \hspace{1cm} (26)

Utilizing the above notation the nonlinear controller for the rate of heat input is given as

$$ h(x) = \begin{cases} 
- \frac{L_V V + \sqrt{L_V^2 + L_g V^4}}{L_g V}, & \text{if } L_g V \neq 0 \\
0, & \text{if } L_g V = 0 
\end{cases} $$ \hspace{1cm} (27)

where $f(x)$ and $g(x)$ are the terms in the ODE corresponding to the temperature of the reactor in Equation 20, and $L_V$, $L_g V$ are the Lie derivatives of the Lyapunov function along the trajectories of the functions $f(x)$ and $g(x)$ respectively (for example, $L_V = \frac{\partial}{\partial x} f(x)$). The quadratic Lyapunov function considered in this example is given in the
above section. Following Alanqar et al.,\textsuperscript{28} the stability region \( \Omega_r = \{ x \in \mathbb{R}^n : V(x) \leq r \} \) which will be used in the LMPC formulation is taken as \( r = 64.3 \).

Using the nonlinear control law shown above, we designed the proposed LMPC formulation in the form of Equation 20 using the Koopman based linear predictors and the Lyapunov constraint. The sampling time and the prediction horizon of the optimization problem are taken as \( \Delta = 0.01 \) hr and \( N_p = 10 \) (i.e., 0.1 hr), respectively. The weighting matrices are taken as \( W = \text{diag}(10^2,1) \) and \( R = 10^{-6} \) based on the magnitudes of \( x_1, x_2 \) and \( u \). In this specific example, we did not consider any explicit state constraints so that \( E, H, b = 0 \). At the beginning of each sampling interval, the state measurements are assumed to be available, which are used to initialize the linear model, as shown in Equation (20c), and perform future state predictions. Additionally, the state measurements are used to calculate \( h(x) \) which enforces the Lyapunov constraint of Equation (20e) numerically by imposing it at the beginning of each sampling period. Please note that the input \( h(x) \) was applied with input saturation at the bounds and setting it to zero at the steady-state according to Equation (27). The predicted state values are then used in the optimization problem to compute the control inputs and the corresponding process behavior that minimizes the squared deviation from the origin. At any time step, \( k \), the optimization is performed over the prediction horizon length of \( N_p \) and the first step, \( u_0^* \), of the optimal input profile, \( \{ u_0^*, \ldots, u_{N_p-1}^* \} \), is applied to the process in a sample-and-hold fashion, and this procedure is repeated at every sampling time over the entire operating period of 1 hr.

### Table 2

<table>
<thead>
<tr>
<th>Predictor</th>
<th>Average RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Koopman</td>
<td>0.5549</td>
</tr>
<tr>
<td>Local linear</td>
<td>0.6323</td>
</tr>
<tr>
<td>MOESP</td>
<td>0.6331</td>
</tr>
</tbody>
</table>

Koopman based linear predictors and the Lyapunov constraint. The sampling time and the prediction horizon of the optimization problem are taken as \( \Delta = 0.01 \) hr and \( N_p = 10 \) (i.e., 0.1 hr), respectively. The weighting matrices are taken as \( W = \text{diag}(10^2,1) \) and \( R = 10^{-6} \) based on the magnitudes of \( x_1, x_2 \) and \( u \). In this specific example, we did not consider any explicit state constraints so that \( E, H, b = 0 \). At the beginning of each sampling interval, the state measurements are assumed to be available, which are used to initialize the linear model, as shown in Equation (20c), and perform future state predictions. Additionally, the state measurements are used to calculate \( h(x) \) which enforces the Lyapunov constraint of Equation (20e) numerically by imposing it at the beginning of each sampling period. Please note that the input \( h(x) \) was applied with input saturation at the bounds and setting it to zero at the steady-state according to Equation (27). The predicted state values are then used in the optimization problem to compute the control inputs and the corresponding process behavior that minimizes the squared deviation from the origin. At any time step, \( k \), the optimization is performed over the prediction horizon length of \( N_p \) and the first step, \( u_0^* \), of the optimal input profile, \( \{ u_0^*, \ldots, u_{N_p-1}^* \} \), is applied to the process in a sample-and-hold fashion, and this procedure is repeated at every sampling time over the entire operating period of 1 hr.
The constrained optimization problem was solved using the interior point solver within the fmincon function in MATLAB R2016b. The default settings of the solver were used with the steady-state value of the input chosen as the initial guess at every sampling time. The CSTR was initialized from \([x_1, x_2] = [0.5, -18]\) and the simulations were performed using an Intel(R) Core(TM) i7-4,790 CPU at 3.60 GHz with a 16 GB RAM and an x64-based processor running Windows 8.1 Enterprise.

The closed-loop trajectories of the CSTR under the proposed linear LMPC scheme are shown in Figure 7. As can be seen from the figure, starting from the initial condition, successful stabilization of the closed-loop system was achieved and the controller was able to drive the system to its steady-state (dotted line). The proposed controller was able to accomplish this due to accurate predictions of the Koopman-based linear model over the prediction horizon of the optimization problem. The associated input profile determined by solving the optimization problem is shown in Figure 8. Note that, in addition to good closed-loop performance, the proposed control scheme has the benefit of being completely data-driven requiring only state (output) measurements. Also, the average computation time required to evaluate the control input from the proposed method is low because it solves a convex quadratic programming problem (when compared to using nonlinear models within the LMPC framework).

5 | CONCLUSIONS

In this manuscript, we presented a data-driven method for the design of stabilizing controllers for nonlinear dynamical systems by integrating Koopman linear predictors within the LMPC framework. The key idea is to compute finite dimensional approximations to the Koopman operator, which yields linear models that are valid on the entire state-space or at least a larger subset of it compared to a small neighborhood around an equilibrium point. These linear models can be used to predict future state evolution in the design of feedback controllers. Specifically, we proposed the use of these predictors in combination with LMPC to obtain closed-loop stability of the time-varying nonlinear operation. Moreover, we showed that the integration of Koopman linear predictors within the LMPC framework leads to a convex (quadratic) optimization problem, which can be solved using any of the extensive array of solvers suitable for linear MPC. We demonstrated the application of the proposed method on a nonlinear chemical process example and extensive simulation results were presented. The average RMSE taken for 100 simulations subjected to random input profiles showed the superior performance of Koopman linear predictors compared to local linearization and subspace identification methods. The resultant linear model was then used in the design of a feedback controller using the LMPC approach. In the simulations, the LMPC formulation with the Koopman linear predictors successfully achieved closed-loop stability.

ACKNOWLEDGMENTS

The authors gratefully acknowledge financial support from the National Science Foundation (CBET-1804407), the Artie McFerrin Department of Chemical Engineering, and the Texas A&M Energy Institute.

ORCID

Abhinav Narasingam https://orcid.org/0000-0002-1723-1708
Joseph Sang-Il Kwon https://orcid.org/0000-0002-7903-5681

REFERENCES


24. Huang B, Ma X, Vaidya M. Feedback stabilization using koopman operator. 2018 IEEE Conference on Decision and Control (CDC); 2018; Miami Beach, FL; pp. 6434-6439.


How to cite this article: Narasingam A, Kwon JS-I. Koopman Lyapunov-based model predictive control of nonlinear chemical process systems. AIChE J. 2019;65:e16743. https://doi.org/10.1002/aic.16743