Solving for Koopman Operators in Series to Approximate Ideal Feedback-oriented Control Strategies

Michael Napoli, Roberto Tron

[This document was submitted on May 5th, 2023 as an independent study project within my Master's program. It has not been extensively peer-reviewed and may contain faults, but I feel is a good representation of my technical writing capabilities outside my more thorough published work. **Please refer to my resume for official publications.**]

***Abstract*—Koopman operator theory has recently been given significant attention in the modeling and controls community because of its ability to describe system dynamics in terms of a data-driven, linear operator without sacrificing accuracy of the original system. To date, the infinitely-dimensioned operators have been primarily used to transform nonlinear models to linear domains as a means to study the system's behavior. The finite form of the operators can also be used to supplement fast model-based control strategies with some degree of approximation error.**

In this paper, the process of using multiple Koopman operators in series is investigated as a method for replicating feedback control in a system whose measurements are a nonlinear function of the state space. The solution to which is found via a recursive form of the extended dynamic mode decomposition algorithm. It is shown that, for a system which receives measurement terms at each time-step, a control-oriented Koopman operator can be found separately from the model-oriented operator and combined using shift functions to replicate the full system feedback control structure. The method for finding these operator series, titled cascade extended dynamic mode decomposition, proves to be generalizable to a broad range of applications - making for an interesting evolution of the dynamic mode decomposition formula.

I. Introduction

While nonlinear dynamical systems are ubiquitous to the robotics community, they remain among the most difficult systems to control reliably. Current methods for modeling these systems are based on exact analytical expressions, which can be difficult, even impossible to derive, or complicated approximation methods which incorporate a degree of error and can be tedious to compute. That said, the first step to system design and implementation is developing a comprehensive model which incorporates control inputs, disturbances and other user-defined parameters. Not to mention, the growth of these systems with the intent of accuracy inevitably correlates with increases in nonlinearities, making it difficult to implement comprehensive full-model control.

This paper explores the application of Koopman operators in series to modeling nonlinear feedback control systems. The Koopman operator is an infinitely-dimensioned, linear operator which propagates scalar-valued functions acting on the system of interest forward in time. This behavior is powerful as it serves to lift otherwise difficult to understand nonlinear systems into linear domains where dynamics are more easily evaluated.

The serialized approach to Koopman operator theory is presented as a method to describe more complex feedback-oriented control architectures. A recursive algorithm for solving for series of operators from data will be discussed as the process is non-trivial and requires an understanding of the shifts required to linearly combine multiple operators. Then, demonstrations will be given on a system whose dynamics are linear but whose state is measured by nonlinear scalar-valued functions which are ambiguously coupled with the system dynamics and control policies.

II. Literature Review

The application of Koopman operators as a data-driven approach to modeling has gained popularity as developments towards making them more susceptible to control injections have advanced. The typical approach to utilizing Koopman operators is to define a system using standardized analytical approaches before lifting them into the observation space. Once lifted, these control algorithms can be studied using easily understood linear system checks for stability. In this section, a brief summary of the development of Koopman operator theory will be given in order to build a stronger foundation in preexisting methods.

In modeling theory, the Koopman operator has become a useful tool for observing and categorizing the behavior of nonlinear dynamical systems. They work by linearly propagating systems of scalar-valued functions, called observables, which represent the underlying dynamics of the physical system. Furthermore, the Koopman operator has become a popular technique in the realm of data-driven modeling as methods such as dynamic mode decomposition (DMD) and extended DMD (EDMD) have been developed [1, 2, 3].

In these methods, series of data snapshots of the system under the appropriate perturbations are saved and used to perform least-squares regression over a predefined selection of observation functions. In most applications the observation functions which correlate most heavily to the model propagation are prioritized through the single-value decomposition algorithm. In systems where the state space is understood and can be approximated, this process is sufficient for finding the operator necessary to propagate the function space. As the primary method used in this paper, the EDMD method will be discussed in more detail in Section III-C.

Fundamental Koopman theory initially considered only the modeling of systems without control [4, 5]. Recently, the theory has been extended to incorporate control injections as a preliminary step to implementing more complex, model-based control techniques [6, 7, 8, 9]. In these scenarios, inputs are

commonly implemented using control-affine functions of the state and inputs leading to the ability to perform linear model-based control techniques.

That being said, there are many systems where strictly linear observables do not operator to the degree of accuracy desired by the user. In response to this, recent investigations have shown increases in policy accuracy and performance when using bilinear observables [10, 11]. The authors here claim that the bilinearly structured observation space outperforms the linearly structured space when applied to optimal control strategies by nearly every metric evaluated: accuracy, reliability and computation speed. While the bilinear form will not be explicitly implemented here, it is a noteworthy considerations for future applications.

III. Preliminaries

This section will serve to present the general theory behind the Koopman operator (Section III-A), its form under control injections (Section III-B) and its derivation using the data-driven learning approach, EDMD (Section III-C).

It should be noted that the methods described here were primarily extrapolated from [6] with various notational changes. They are rewritten here for completeness.

A. Koopman Operator Theory

Ordinarily, a model function for a given dynamic system is calculated in discrete-time using a transition function, F. The state and its propagation forward in time will be denoted as x and x^+, respectively.

$$x^+ = F(x). \tag{1}$$

Where $x, x^+ \in \mathbb{M} \subset \mathbb{R}^n$ is a collumn vector of a state of size n. F is a discrete model function which maps $F : \mathbb{M} \to \mathbb{M}$. It is common for F to be nonlinear with respect to the states; leading to increases in difficulty when studying the model's behaviors.

The Koopman operator presents a data-driven alternative to analytical derivations of F in terms of a linear operator. To utilize the Koopman operator, an observation function list, g, which communicates the state space into the infinitely-dimensioned function space, $\mathbb{C}$, is used.

$$z = g(x) \tag{2}$$

Where z is composed of the observable functions built from the state, x. In other words, z represents the current state translated to the observation space such that $z \in \mathbb{C} \subset \mathbb{R}^\infty$ and $g \in \mathbb{G} : \mathbb{M} \to \mathbb{C}$. If the function space is defined as a Hilbert space, a similar function to (1) can thus be written in terms of its own propagation operator.

$$z^+ = \mathcal{K}_\infty z = \mathcal{K}_\infty g(x). \tag{3}$$

Where $\mathcal{K}_\infty \in \mathbb{R}^{\infty \times \infty}$ is the ideal-form of the Koopman operator and is used to propagate the observation functions forward in time. Putting (2) and (3) together, the operator can be better represented without the intermediate term, z.

$$g(x^+) = \mathcal{K}_\infty g(x) = (g \circ F)(x). \tag{4}$$

Where $\mathcal{K}_\infty : \mathbb{G} \to \mathbb{G}$ and maintains the time-step defined by the discrete model, F, and $\circ$ represents the function composition operator. In summary, the Koopman operator, $\mathcal{K}_\infty$, is linear when $\mathbb{G}$ is a vector space, even when $\mathbb{M}$ is a nonlinear domain.

B. Koopman Operator With Control

Incorporating control into the Koopman operator is a relatively trivial process. Let a nonlinear model with control be represented in a similar manner as (1).

$$x^+ = F(x, u) \tag{5}$$

Where $u \in \mathbb{R}^m$ represents the control injection. To communicate the control term to an appropriate Koopman operator, the observation function must also be restated in terms of both the state and control term.

$$z = g(x, u) \tag{6}$$

In this way, the structure of the propagation equation from (4) remains the same, but includes observation functions of the input.

$$g(x^+, u^+) = \mathcal{K}_\infty g(x, u) \tag{7}$$

The propagation of $u \to u^+$ is not necessarily a trivial process. This is because input terms are commonly dependent on complex control policies which may not have an easily selected set of observable functions. For this purpose, the input term will be assumed to be held constant over the propagation of g.

$$g(x^+, u) = \mathcal{K}_\infty g(x, u) \tag{8}$$

The next section will discuss the principal method for calculating the Koopman operator from data; extended dynamic mode decomposition. For demonstration purposes the observation functions discussed in Section III-C will not include inputs. That said, the strategy discussed does not change when inputs are incorporated.

C. Extended Dynamic Mode Decomposition

In order to solve for the Koopman operator, the dimensionality of the function space must first be addressed. Being an infinitely-dimensioned space is not realistic when put in terms of real world applications so (2) is redefined as a subset of the original space.

$$z = \Psi(x) \in g(x) \tag{9}$$

Where $\Psi \in \mathbb{G}^N \subset \mathbb{G}$ is some observation function with dimension N which approximates its infinitely dimensioned counterpart and is dependent on the accuracy desired by the user. More specifically, Ψ can be written as the list of chosen observation functions.

$$\Psi(x) = \{\psi_i(x) : \forall i \leq N, i \in \mathbb{N}\}. \tag{10}$$

Where $\mathbb{N}$ is the set of natural numbers and each ψ_i term is a scalar-valued function of the state variable. Combining this approach with (4) yields the following.

$$\Psi(x^+) = \mathcal{K}_N \Psi(x) + r(x, x^+) \tag{11}$$

Where $r(x, x^+)$ is the residual error moving from x to x^+ through the truncated function space, Ψ. Likewise, the Koopman operator is now represented as $\mathcal{K}_N \in \mathbb{R}^{N \times N} \subset \mathbb{R}^{\infty \times \infty}$. Next, a series of data (found through data collection or simulation) is defined for use in finding $\mathcal{K}_N$.

$$X = \{x_i : x_i \in \mathbb{M},\ \forall i < P,\ i \in \mathbb{N}\} \tag{12}$$

Where X is a tuple of (P-1)-evenly spaced snapshots of the state. The forward snap shot of X will also be defined so as to make notation clear.

$$X^+ = \{x_{i+1} : x_{i+1} = F(x_i),\ \forall x_i \in X\} \tag{13}$$

Using this data, the solution for the Koopman operator can be defined as the minimization of the residual error over the entire data set.

$$J = \frac{1}{2}||r(X, X^+)||^2 \tag{14}$$

Which can be restated in terms of the Koopman operator approximation from (11).

$$J = \frac{1}{2}||\Psi(X^+) - \mathcal{K}_N \Psi(X)||^2 \tag{15}$$

In this form, the solution for J is a simple least-squares regression such that $\mathcal{K}_N = G^\dagger A$ where $G^\dagger$ is the pseudo-inverse of G. More specifically, the matrices G and A are composed of the observation functions for the current state and its propagation forward in time.

$$G = \frac{1}{P}\Psi(X)\Psi^\intercal(X), \tag{16}$$

$$A = \frac{1}{P}\Psi(X)\Psi^\intercal(X^+). \tag{17}$$

It is important to note that not all observation functions may be necessary in the final representation of the model. For this reason, single value decomposition (SVD) will be used to prioritize the higher impact terms when computing $G^\dagger$. The single value decomposition equation is restated here for completeness.

$$G = USV^\intercal \tag{18}$$

Where $U, V \in \mathbb{R}^{N \times s}$ and $S \in \mathbb{R}^{s \times s}$ such that the matrices represent the s-most prominent terms in the observation space. The pseudo-inverse can be found by exploiting the nature of the SVD results.

$$G^\dagger = VS^{-1}U^\intercal \tag{19}$$

Note that for use in pseudocode, the calculation for $G^\dagger$ will be referred to by a call to the SVD(G) function. This form can then be used to calculate a minimum-error Koopman operator.

$$\mathcal{K}_N = \left(VS^{-1}U^\intercal\right) A \tag{20}$$

Where $\mathcal{K}_N$ is a learned approximation of the Koopman operator from data. For use in the algorithm descriptions in Section IV the EDMD solution strategy is also written in the form of pseudocode for when the data X has already been communicated to the observation space.

Algorithm 1 Extended Dynamic Mode Decomposition

```
function EDMD(Ψ̂, Ψ̂⁺)
    G ← (1/P) Ψ̂ Ψ̂ᵀ
    A ← (1/P) Ψ̂ Ψ̂ᵀ
    G† ← SVD(G)
    K ← G†A return K
end function
```

Where $\hat{\Psi}$ and $\hat{\Psi}^+$ represent the shift of the entire data sets X and X^+ to the observation space, respectively.

IV. Solving for Koopman Operators in Series

As is discussed throughout Section III, the Koopman operator is primarily considered in the form of a *single* solution. That said, there are scenarios when systems perform operations over decoupled subsets of the larger space description and can be strung together in *series* to achieve some desired behavior described by the interaction of these spaces.

For instance, a model might have a series of inputs which represent the available controls. Once the control terms are calculated, they in some way modify the performance of the model, but the calculation of the controls and the propagation of the model are two separate processes.

Here, a recursive method for identifying these operations separately is identified before calculating the cumulative behavior. To generalize notation to any system, the architecture will be described in terms of a series of N_K operators and their corresponding shift functions, $\mathcal{T}_i(\mathcal{K}_i)$.

$$\begin{aligned} \mathcal{K}_{1 \to N_K} &= \{\mathcal{K}_i : \forall i \leq N_K, i \in \mathbb{N}\} \\ \mathcal{T}_{1 \to N_K} &= \{\mathcal{T}_i : \forall i \leq N_K, i \in \mathbb{N}\} \end{aligned} \tag{21}$$

Where $\mathcal{T}_i$ represents the shift function required to communicate operators between function spaces. The full propagation is thus a series of transformations through the list of operators and their respective shift functions.

$$\Psi(x^+) = \left(\prod_{i=1}^{N_K} \mathcal{T}_i(K_i)\right) \Psi(x). \tag{22}$$

For evaluating Ψ with more clarity (specifically the matrices product direction), (22) can also be represented by an N_K-series of function compositions.

$$\Psi(x^+) = (\mathcal{T}_1(\mathcal{K}_1) \circ \dots \circ \mathcal{T}_{N_K-1}(\mathcal{K}_{N_K-1}) \circ \mathcal{T}_{N_K}(\mathcal{K}_{N_K}) \circ \Psi)(x) \quad (23)$$

Each operator maintains the property of mapping within its own observation space $\mathcal{K}_i : \mathbb{G}_i \to \mathbb{G}_i$, but the shift function maps observation spaces to observation spaces. In the context of (23) we have $\mathcal{T}_i : \mathbb{G}_i \to \mathbb{G}_{i+1}$.

Using the notation expressed in (22) the objective function for calculating the series of operators through EDMD is written as

$$J = \frac{1}{2} \left\| \Psi(X^+) - \left(\prod_{i=1}^{N_K} \mathcal{T}_i(K_i) \right) \Psi(X) \right\|^2. \quad (24)$$

In the following sections the solution to (24) will be investigated through the cascade EDMD algorithm, before being demonstrated in Section VII.

A. Cascade EDMD

The cascade EDMD strategy exploits the sequential structure of (23) to recursively solve for each of the operators. Under the assumption that each of the operators observation lists are known at initialization, the algorithm navigates from the N_K-th operator to the 1-st operator in order; at each step performing EDMD. This process is more explicitly described in algorithm 2.

Algorithm 2 The Cascade EDMD Strategy

function CASCADEEDMD($\mathcal{T}_{1\to N}, \Psi_N, X, X^+$)
  $\hat{\Psi} \leftarrow \Psi_N(X)$ ▷ Lift X-data
  if N is equal to 1 **then**
    $\hat{\Psi}^+ \leftarrow \Psi_N(X^+)$ ▷ Lift X^+-data (1)
    $\mathcal{K}_1 \leftarrow \text{EDMD}(\hat{\Psi}, \hat{\Psi}^+)$ **return** $\mathcal{K}_1$
  end if
  $\mathcal{K}_{2\to N} \leftarrow$ CASCADEEDMD($\Psi_{2\to N}, X, X^+$)
  $\hat{\Psi}^+ \leftarrow (\Pi_{i=2}^{N} \mathcal{T}_i(\mathcal{K}_i))\hat{\Psi}$ ▷ List X^+-data (2)
  $\mathcal{K}_1 \leftarrow \text{EDMD}(\hat{\Psi}, \hat{\Psi}^+)$ **return** $\mathcal{K}_{1\to N}$
end function

The cascade algorithm is a reliable method when the observation functions, and their relationship to one another, are well understood. Pitfalls of the method have not been definitively verified but are expected when the observables and their corresponding operators are not by definition decoupled. In other words, in cases where an operator, $\mathcal{K}_{i-j}$ with $j \neq 0$, has some dependence on a higher order operator $\mathcal{K}_i$; in which case the propagation of the observation may become ambiguous w.r.t the real system.

V. The Koopman-Feedback Operator

Initially, the fundamental theory of Koopman operators considered only the modeling of autonomous systems without input; later, research was completed to extend the framework to systems with inputs. In the latter case, the modeling is only a means to the end of controlling the system; as such, the system is first modeled (approximated) with a Koopman operator and the appropriate observables, before a feedback controller is designed based on the linearized model. This approach can work, but, for non-trivial cases, has shortcomings. On the one hand, the Koopman model can be restricted to linear terms w.r.t. the inputs, making the controller design a simple exercise in linear control theory, but heavily limiting the class of systems that can be represented (essentially, it is impossible to accurately model state-varying control fields). On the other end, the model can be affine-linear or nonlinear in the inputs; this allows to model most practical systems, but makes the feedback controller design much more complicated.

Here, the concept of Koopman operators in series is extended to the design and implementation of a predetermined feedback controller based on an imperfect state description through measurement terms. The cumulative operator, hereto referred to as the Koopman-feedback operator (KFO), is implemented in parts; its definition and observation selection will be discussed in more detail in the following sections.

A. Derivation of the KFO

The goal for this section of research will be to establish a standardized method of formulating sets of observable functions: $\Psi_x(x)$, $\Psi_u(x)$ and $h(x)$ such that a desired model and control policy are accurately approximated by the cumulative operator, $\mathcal{K}_{xu}$. As stated previously, the propagation of x can be modeled using some nonlinear system of equations and a control injection as in (5), or an acceptable Koopman operator.

$$x^+ \in \Psi(x^+) = \mathcal{K}_x \begin{bmatrix} \Psi_x(x) \\ \Psi_u(x) \otimes u \end{bmatrix} \quad (25)$$

Where $\Psi_x(x) \in \mathbb{R}^p$ and $\Psi_u(x) \in \mathbb{R}^q$ represent the function space which describe the desired model and are cumulatively represented by $\Psi(x) \in \mathbb{R}^{p+qm}$. The Kronecker product, represented by $\otimes$, is used to bilinearly expand the control variables over the input-specific function space. This leads to $\mathcal{K}_x$ being of dimensions $\mathbb{R}^{p+qm\times p+qm}$. The objective of this section will be to identify some secondary Koopman operator, denoted $\mathcal{K}_u \in \mathbb{R}^{b\times b}$, which approximates the necessary control policy, C, for a given state.

$$u = C(x) \in \mathcal{K}_u h(x) \quad (26)$$

Where $h(x) \in \mathbb{R}^b$ is the list of control-specific observable functions that are dependent solely on x. In practice, this vector contains functions of the actual measurements taken by the system. It eventually plays a role similar to the vector of observations Ψ_x, with the difference being that its value is not predicted and propagated by the Koopman operator, but is given by actual online measurements.

Here, it is understood that the size of $\mathcal{K}_x$ grows with the observation list being propagated. Using this concept and in combination with (26), (25) can be reformulated as

$$F(x, C(x)) \in \Psi(x^+) = \mathcal{K}_x \begin{bmatrix} \Psi_x(x) \\ \Psi_u(x) \otimes \mathcal{K}_u h(x) \end{bmatrix}. \quad (27)$$

Ideally, there exists some cumulative operator which fits the form described in (22),

$$\mathcal{K}_{xu} = \mathcal{T}_x(\mathcal{K}_x)\mathcal{T}_u(\mathcal{K}_u). \tag{28}$$

Where $\mathcal{K}_{xu}$ propagates both the state and inputs simultaneously and $\mathcal{T}_x$ and $\mathcal{T}_u$ describe the transition function necessary to combine the operators and the appropriate observables list into a cumulative operation. The objective now becomes to prove the algebraic existence of such an operator. Establishing that $\Psi_u(x) \otimes \mathcal{K}_u h(x) \in \mathbb{R}^{qb}$ is a vector, $\mathcal{K}_u$ can be brought outside using the mixed-product property of the Kronecker multiplier.

$$\Psi_u(x) \otimes \mathcal{K}_u h(x) = (\mathbf{I}_q \otimes \mathcal{K}_u)(\Psi_u(x) \otimes h(x)) \tag{29}$$

Where $\mathbf{I}_q \in \mathbb{R}^{q\times q}$ is an identity matrix. The input-specific Koopman operator, $\mathcal{K}_u$ can then be brought outside of the observation space from (27).

$$\Psi(x^+) = \mathcal{K}_x \begin{bmatrix} \mathbf{I}_p & \mathbf{0}_{p\times qb} \\ \mathbf{0}_{qb\times p} & \mathbf{I}_q \otimes \mathcal{K}_u \end{bmatrix} \begin{bmatrix} \Psi_x(x) \\ \Psi_u(x) \otimes h(x) \end{bmatrix} \tag{30}$$

Where $\mathbf{0}_{i\times j}$ represents a matrix of all zeros with i-rows and j-columns. In this arrangement, the two Koopman operators, $\mathcal{K}_x$ and $\mathcal{K}_u$, can be grouped together in terms of their shift functions, forming the cumulative operator $\mathcal{K}_{xu} \in \mathbb{R}^{p+qb\times p+qb}$ as shown in (28).

$$\begin{aligned} \mathcal{T}_x(\mathcal{K}_x) &= \mathcal{K}_x \\ \mathcal{T}_u(\mathcal{K}_u) &= \begin{bmatrix} \mathbf{I}_p & \mathbf{0}_{p\times qb} \\ \mathbf{0}_{qb\times p} & \mathbf{I}_q \otimes \mathcal{K}_u \end{bmatrix} \end{aligned} \tag{31}$$

Replacing (27) with a more general-form expression;

$$\Psi(x^+) = \mathcal{K}_{xu} \begin{bmatrix} \Psi_x(x) \\ \Psi_u(x) \otimes h(x) \end{bmatrix}. \tag{32}$$

Interestingly, the input-specific operator, $\mathcal{K}_u$, can be truncated to only propagate the elements which represent the control injection. This way the observation functions, h, can be composed of terms which are not ideal to propagate forward in time. This behavior is beneficial when h is dependent on measured data which may not be an easily understood linear combination of Ψ_x and Ψ_u, or is charaterized by runtime measurements, and would lead to inaccurate propagation by the cumulative operator.

To account for this, the measured terms are dropped and only the components Ψ_x and Ψ_u are propagated forward. This implies that the measured data must be manually updated at every time step which will be discussed in more detail in Section VII.

B. The KFO Objective Function

The objective function necessary to solve for (30) now becomes an optimization over two Koopman operators. Allowing (15) to be restated as

$$J = \frac{1}{2}\|\Psi(X^+) - \mathcal{K}_x\mathcal{T}_u(\mathcal{K}_u)\Psi(X)\|^2, \tag{33}$$

giving a more nuanced optimization problem than the objective described in Section III-C. That said, the KFO clearly fits the description of the methods from Section IV and can thus be found via the cascade EDMD framework.

VI. Problem Definition

In this section the environment that the cascade EDMD and KFO formulations were tested in is presented. The *anchor* system will first be defined before addressing the closed-loop system equation derivation.

A. The Anchor System

The anchor system is defined as a discrete holonomic 2-D vehicle moving through an unbounded environment. The system is therefore represented by the equation

$$x^+ = x + (\Delta t)u. \tag{34}$$

Where $x \in \mathbb{R}^2$ and $u \in \mathbb{R}^2$ describe the state and control, respectively, and Δt is the difference in time between each step. In the environment there exists N_a-anchors which are volume-less: i.e. there is no obstruction to the vehicle's motion. At every point in time the anchors give the vehicle the L_2-norm distance between itself and the vehicle.

$$d(x) = \{\sqrt{(x-a_i)^\intercal(x-a_i)} : \forall i \leq N_a, i \in \mathbb{N}\} \tag{35}$$

Where a_i is the center point of the i-th anchor. At initialization, the vehicle is given an approximation of its position, $x_0 = x(0)+\delta$, but is otherwise only given the anchor distances at each time-step. The problem then becomes creating a closed-loop control strategy which navigates the vehicle to some desired goal position; replacing (34) with

$$\begin{aligned} u &= \mathcal{K}_u h(d(x)) \\ x^+ &= x + (\Delta t)\mathcal{K}_u h(d(x)). \end{aligned} \tag{36}$$

Where $\mathcal{K}_u : \mathbb{R}^{N_a} \to \mathbb{R}^2$ is the anchor-based control policy as acted over functions, h, of the measurement terms, and represented by the control-oriented Koopman operator. In the implementation shown here the desired position will be at the origin, but it is understood that the observation space could be constructed for any arbitrary goal.

In the following section it will be demonstrated that the squared L_2-norm is a better choice for accurate anchor following. This understanding will then be utilized to construct a sufficient closed-loop operator from the measurement terms.

B. Closed-loop Anchor System Equations

In the closed-loop system the intent is to implement a form of feedback control which utilizes the measurement terms to navigate the vehicle to the origin. In this scenario, a controller which moves the vehicle has already been designed, but is dependent on the state terms. The goal is to utilize a two-step Koopman operator to replicate this controller using only the distance terms discussed in Section VI-A.

To start, it will be demonstrated that the $d^2(x)$ terms are a better choice for observation selection and will be instrumental in performing the closed-loop control circuit discussed. The distance function and its propagation forward is found to be

$$\begin{aligned} d_i(x) &= \sqrt{(x-a_i)^\intercal(x-a_i)} \\ \Rightarrow d_i(x^+) &= \sqrt{(x^+-a_i)^\intercal(x^+-a_i)} \\ &= \sqrt{(x+(\Delta t)u-a_i)^\intercal(x+(\Delta t)u-a_i)} \end{aligned} \tag{37}$$

From this equation it is difficult to select observation functions which will always appropriately propagate the anchor distances through the linear Koopman operator. In order to get terms outside of the $\sqrt{\ }$ operation, both sides will be squared and the inside will be evaluated term-by-term.

$$\begin{aligned} d_i^2(x^+) &= (x^+-a_i)^\intercal(x^+-a_i) \\ &= (x+(\Delta t)u-a_i)^\intercal(x+(\Delta t)u-a_i) \\ &= x^\intercal x+(\Delta t)^2u^\intercal u+a_i^\intercal a_i+2(\Delta t)x^\intercal u-2x^\intercal a_i-2(\Delta t)u^\intercal a_i \end{aligned} \tag{38}$$

It is clear from (38) that the squared distance function leads to a series of summations of nonlinear functions of the state and input. For this reason, $d^2(x)$ is a better choice for constructing the KFO as it better aligns with the Koopman operator form. The next step is to identify the observable functions necessary to communicate the anchor distances to the control policy.

Because EDMD is a data-driven approach to modeling, the desired control behavior must be defined *before* the start of the method. For use here we will describe the desired control in terms of a linear, position based operation.

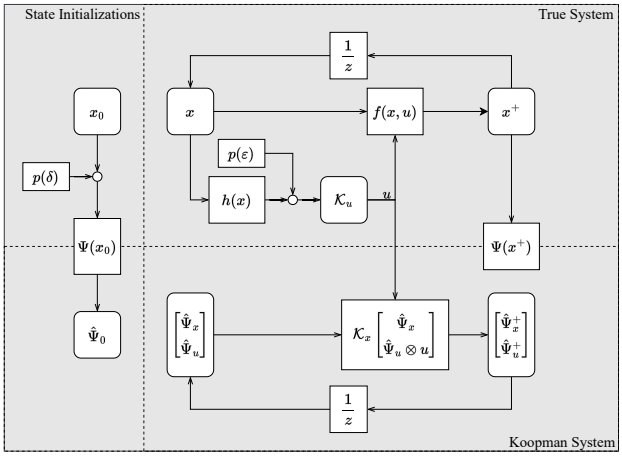


Fig. 1: Closed-loop System with KFO Implementation

$$\begin{aligned} u &= Cx \\ \text{with } C &= -\mathbf{I}_2 \\ \text{s.t. } u &= -x \end{aligned} \tag{39}$$

This controller is understood to move the vehicle to the origin independent of initial position. It should be noted that any controller could have been chosen, but the identity matrix, $\mathbf{I}_2$, makes for easy algebraic manipulation.

The next step is to again identify the observation functions necessary to give zero residual error w.r.t the control. In general the only values available will be the anchor distances (feedback term) and the approximation of the state as given by preceding Koopman operator calculations. To start, the equations for a single anchor (denoted 1) and substituting in (39) will be evaluated.

$$\begin{aligned} d_1^2(x) &= (x-a_1)^\intercal(x-a_1) \\ &= (-u-a_1)^\intercal(-u-a_1) \\ &= u^\intercal u+a_1^\intercal a_1+2u^\intercal a_1 \end{aligned} \tag{40}$$

Now, subtracting the equivalent equation but for a second anchor (denoted 2).

$$\begin{aligned} d_1^2(x)-d_2^2(x) = \;& u^\intercal u-u^\intercal u+a_1^\intercal a_1-a_2^\intercal a_2 \\ & +2u^\intercal a_1-2u^\intercal a_2 \\ = \;& a_1^\intercal a_1-a_2^\intercal a_2+2u^\intercal a_1-2u^\intercal a_2 \end{aligned} \tag{41}$$

This turns the control equation as defined in (39) into a composition of the distance squared and constant position functions.

$$2u^\intercal(a_1-a_2)=d_1^2(x)-d_2^2(x)-a_1^\intercal a_1+a_2^\intercal a_2 \tag{42}$$

Using (42) constraints on the positioning of the anchors can be made. That is, for the system to be capable of implementing zero-error feedback control there must be at least three anchors with center points at $[a_1^x,a_1^y]^\intercal=[-a_2^x,a_2^y]^\intercal=[a_3^x,-a_3^y]^\intercal$. Resulting in the zero-error equation

$$\begin{bmatrix}u_1\\u_2\end{bmatrix}=\frac{1}{2}\begin{bmatrix}(a_1^x-a_2^x)^{-1}(d_1^2(x)-d_2^2(x))\\(a_1^y-a_3^y)^{-1}(d_1^2(x)-d_3^2(x))\end{bmatrix}, \tag{43}$$

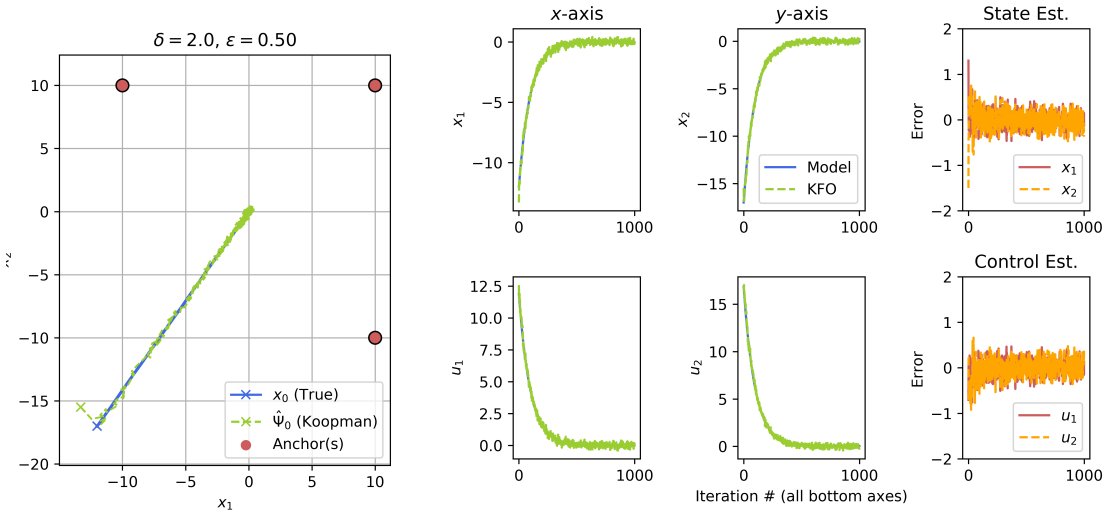
and leading to a simple selection of the observation list.

$$\begin{aligned} \Psi_x^{CL} &= x \\ \Psi_u^{CL} &= 1 \\ h^{CL} &= d^2(x) \quad \text{with } N_a=3 \end{aligned} \tag{44}$$

In this case, the control operator, found using the cascade EDMD methods discussed in Section IV, is rectangular w.r.t the control terms.

$$u=\mathcal{K}_u h(x)=\mathcal{K}_u d^2(x) \tag{45}$$

Giving the full system propagation the final form



(a) Environment with Anchors and Path (b) State and Input Responses with Estimation Error

Fig. 2: Simulation Results for a Single Arbitrary Initial Position

$$\begin{bmatrix} x^+ \\ u \end{bmatrix} = \mathcal{K}_x \begin{bmatrix} \Psi_x^{CL} \\ \Psi_u^{CL} \otimes \mathcal{K}_u h^{CL} \end{bmatrix} = \mathcal{K}_x \begin{bmatrix} x \\ 1 \otimes \mathcal{K}_u d^2(x) \end{bmatrix} = \mathcal{K}_{xu} \begin{bmatrix} x \\ 1 \otimes d^2(x) \end{bmatrix} \tag{46}$$

By intuition it is also understood that if the anchors did not follow the aforementioned constraints, the error which culminates would be negated by incorporating a 1 into the observation space $h^{CL}(x)$.

VII. Results

Here, the methods discussed in Sections IV and V will be applied to the anchor system as defined in Section VI-A. On top of solving for the control and dynamics of the anchor system to a high degree of accuracy, it is also shown that the KFO is a sufficient method for filtering out large levels of noise when introduced into the measurement terms. The full simulation structure can be seen in figure 1.

In the diagram, there are two clearly defined decision processes. The *true system* (shown in the top right) represents the system dynamics as defined by the real model implementation, and are technically unknown to the user. The *Koopman system* represents the approximation being followed by the KFO. In this case, the state terms held in the observation space are only updated through the predefined operators and thus represent estimations of the true dynamics. For readability, variables with the ˆ-symbols are estimated and all other variables are predetermined or unknown to the Koopman system.

The left-most column illustrates the process initialization. In this stage an initial state is given to the Koopman system with some level of randomly generated noise inside the known range space $p(\delta) \in [-\delta, \delta]$. Moreover, at each time-step noise is also incorporated through the measurement terms with a range space of $p(\varepsilon) \in [-\varepsilon, \varepsilon]$.

A. Closed-loop Controller Results

As previously mentioned, the closed-loop controller was implemented with a predetermined level of zero-mean noise. For demonstration purposes the noise was set to an appreciable value so that it was noticeable during simulation. As such, the initialization noise, δ, was set to $2[m]$ while the measurement noise, ε, was set to $0.5[m]$.

The environment, vehicle path and vehicle for a single run can be seen in figure 2 along with the position and control estimations. As can be seen, the vehicle is capable of navigating to the origin in an environment that incorporates a relatively large degree of noise. The path followed by the estimated vehicle through the Koopman operator is generally off by a small degree, but is able to maintain its approximation of the true model through the feedback measurement functions, $h(x)$.

Moreover, although the path constructed by the KFO is chaotic w.r.t the vehicle estimation, the inputs given to the true vehicle maintain a smooth trajectory towards the origin. This is largely in part to the use of $d^2(x)$ as the principle observation function being used for control. In the process of squaring the distance data, the error is also squared, giving it a smaller significance on the performance of the controller.

To demonstrate this more explicitly, figure 3 shows the performance of 15 randomly initialized tests with varying magnitudes for ε. As can be seen, the smoothness of the path does not change considerably as the magnitude of noise increases through the set $\varepsilon \in [0, 0.5, 1, 2]$. When $\varepsilon = 2$, the path begins to see a subtle deviation from desirable behavior, but the vehicle is still able to approach the origin of the environment. As previously mentioned, this is likely due to the use of $d^2(x)$ as the principal observation function for controller calculations. Interestingly, these results show that Koopman operators in series can be leveraged as a method

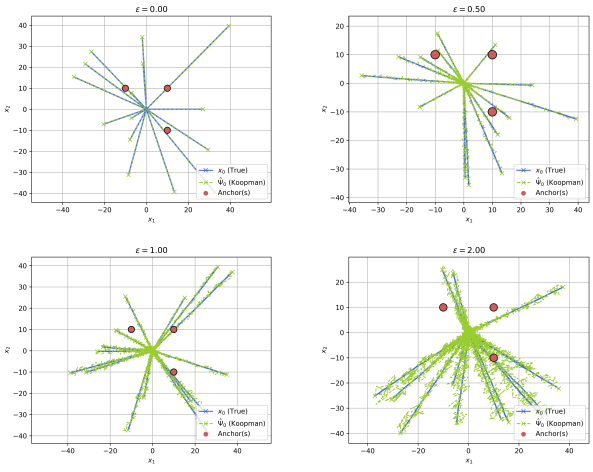


Fig. 3: Randomly Generated Paths with Varying ε

for organizing observation functions such that they filter out the impact from poorly measured data.

VIII. Conclusion

Through the research performed it is demonstrated that Koopman operators organized and solved for through the cascade EDMD algorithm are capable of mimicking the policy of an arbitrary linear controller solely from measurement data. By selecting a desired control which is linear, it is guaranteed that the behaviors learned by the KFO satisfy equivalent stability behaviors as the principal controllers.

It is also shown that the KFO system indirectly accomplishes the task of filtering out measurement noise. While this was initially an unintended side affect from formatting the observation functions, $h(x)$, in terms of squared measurement terms, it can be extrapolated that, when designed properly, the KFO structure can be used to remove noise from control injections without the need for complicated filtering techniques.

It is important to note that while in the context of the anchor system $\Psi_u^{CL}(x) = 1$, it is very common that inputs take the form of a nonlinear control-affine function

$$x^+ = \xi(x) + \zeta(x)u,$$

or something similar. In these scenarios it is clear that the observation list would become $\xi(x) \in \Psi_x(x)$, $\zeta(x) \in \Psi_u(x)$ and $u \in h(x)$ as well as the other assorted functions required to propagate the observation space accurately. In which case, the function list $\Psi_u(x)$ is instrumental to the accurate propagation of x.

In the future, the KFO form and the process of cascade EDMD will be extended to systems with more complex, largely nonlinear dynamics. The hope is to show that the application of shift functions to communicate Koopman operators between domains is a systematic method for breaking large, complex systems into decoupled sub-problems.

References

[1] Peter J. Schmid. "Dynamic mode decomposition of numerical and experimental data". In: *Journal of Fluid Mechanics* 656 (Aug. 2010). Publisher: Cambridge University Press, pp. 5–28. ISSN: 1469-7645, 0022-1120. DOI: 10.1017/S0022112010001217. URL: https://www.cambridge.org/core/journals/journal-of-fluid-mechanics/article/dynamic-mode-decomposition-of-numerical-and-experimental-data/AA4C763B525515AD4521A6CC5E10DBD4 (visited on 10/05/2022).

[2] Matthew O. Williams, Ioannis G. Kevrekidis, and Clarence W. Rowley. "A Data–Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition". In: *Journal of Nonlinear Science* 25.6 (Dec. 1, 2015), pp. 1307–1346. ISSN: 1432-1467. DOI: 10.1007/s00332-015-9258-5. URL: https://doi.org/10.1007/s00332-015-9258-5 (visited on 04/28/2023).

[3] Gregory Snyder and Zhuoyuan Song. *Koopman Operator Theory for Nonlinear Dynamic Modeling using Dynamic Mode Decomposition*. Oct. 15, 2021. DOI: 10.48550/arXiv.2110.08442. arXiv: 2110.08442[cs,math]. URL: http://arxiv.org/abs/2110.08442 (visited on 09/20/2022).

[4] B. O. Koopman. "Hamiltonian Systems and Transformation in Hilbert Space". In: *Proceedings of the National Academy of Sciences* 17.5 (May 1931). Publisher: Proceedings of the National Academy of Sciences, pp. 315–318. DOI: 10.1073/pnas.17.5.315. URL: https://www.pnas.org/doi/abs/10.1073/pnas.17.5.315 (visited on 11/16/2022).

[5] Marko Budišić, Ryan Mohr, and Igor Mezić. "Applied Koopmanism". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 22.4 (Dec. 21, 2012), p. 047510. ISSN: 1054-1500. DOI: 10.1063/1.4772195. URL: https://doi.org/10.1063/1.4772195 (visited on 04/21/2023).

[6] Ian Abraham, Gerardo De La Torre, and Todd D. Murphey. "Model-Based Control Using Koopman Operators". In: *Robotics: Science and Systems XIII*. July 12, 2017. DOI: 10.15607/RSS.2017.XIII.052. arXiv: 1709.01568[cs,math]. URL: http://arxiv.org/abs/1709.01568 (visited on 10/05/2022).

[7] Joshua L. Proctor, Steven L. Brunton, and J. Nathan Kutz. "Generalizing Koopman Theory to Allow for Inputs and Control". In: *SIAM Journal on Applied Dynamical Systems* 17.1 (Jan. 2018). Publisher: Society for Industrial and Applied Mathematics, pp. 909–930. DOI: 10.1137/16M1062296. URL: https://epubs.siam.org/doi/10.1137/16M1062296 (visited on 10/20/2022).

[8] Xu Ma, Bowen Huang, and Umesh Vaidya. "Optimal Quadratic Regulation of Nonlinear System Using Koopman Operator". In: *2019 American Control Conference (ACC)*. 2019 American Control Conference (ACC). ISSN: 2378-5861. July 2019, pp. 4911–4916. DOI: 10.23919/ACC.2019.8814903.

[9] Yunzhu Li et al. "LEARNING COMPOSITIONAL KOOPMAN OPERATORS FOR MODEL-BASED CONTROL". In: (2020).

[10] Debdipta Goswami and Derek A. Paley. "Global Bilinearization and Reachability Analysis of Control-Affine Nonlinear Systems". In: *The Koopman Operator in Systems and Control: Concepts, Methodologies, and Applications*. Ed. by Alexandre Mauroy, Igor Mezić, and Yoshihiko Susuki. Lecture Notes in Control and Information Sciences. Cham: Springer International Publishing, 2020, pp. 81–98. ISBN: 978-3-030-35713-9. DOI: 10.1007/978-3-030-35713-9_4. URL: https://doi.org/10.1007/978-3-030-35713-9_4 (visited on 10/12/2022).

[11] Daniel Bruder, Xun Fu, and Ram Vasudevan. "Advantages of Bilinear Koopman Realizations for the Modeling and Control of Systems With Unknown Dynamics". In: *IEEE Robotics and Automation Letters* 6.3 (July 2021). Conference Name: IEEE Robotics and Automation Letters, pp. 4369–4376. ISSN: 2377-3766. DOI: 10.1109/LRA.2021.3068117.