Designing System Level Synthesis Controllers for Nonlinear Systems with Stability Guarantees

Lauren Conger LCONGER@CALTECH.EDU
Sydney Vernon SVERNON@CALTECH.EDU
Eric Mazumdar MAZUMDAR@CALTECH.EDU
1200 E. California Blvd., Pasadena, CA 91125

Editors: N. Matni, M. Morari, G. J. Pappas

Abstract

We introduce a method for controlling systems with nonlinear dynamics and full actuation by approximating the dynamics with polynomials and applying a system level synthesis controller. We show how to optimize over this class of controllers using a neural network while maintaining stability guarantees, without requiring a Lyapunov function. We give bounds for the domain over which the use of the class of controllers preserves stability and gives bounds on the control costs incurred by optimized controllers. We then numerically validate our approach and show improved performance compared with feedback linearization— suggesting that the SLS controllers are able to take advantage of nonlinearities in the dynamics while guaranteeing stability.

Keywords: nonlinear control, polynomial dynamics, system level synthesis

1. Introduction

Many physical systems systems are modeled by nonlinear dynamics, and these systems need to be controlled. One existing option is approximating nonlinear systems systems with polynomials and using sum of squares to construct controllers (Strässer et al., 2021). Other successful techniques include model predictive control (MPC), learning dynamics or controllers with neural networks (Shi et al., 2019, 2022; Furieri et al., 2022), and control Lyapunov functions (CLFs)(Ames et al., 2014). Tools for nonlinear control are less developed than those for linear control because of the breadth and complexity of various nonlinear systems, so locally linearizing the system along a trajectory or lifting to linear dynamics (Lovchakov and Sibyakin, 2020) allows engineers to utilize linear system theory. However, sometimes the accuracy of the model must be better than is feasible with linearized models, as in the case of modeling braking of wheeled vehicles (Vataeva et al., 2019), motivating us to pursue a controller in the nonlinear setting.

We choose a polynomial form of the dynamics because many nonlinear systems can be lifted into this form (Qian et al., 2020; Mauroy and Goncalves, 2016) and polynomial dynamics and state feedback controller can be used to ensure stable periodic orbits (Yuno and Ohtsuka, 2014). Additionally, polynomials have been successfully used as a basis for identifying nonlinear Hammerstein systems (Hammar et al., 2015). We offer a method for discrete-time control for systems with nonlinear dynamics, where the system is fully actuated, with a polynomial approximation of the dynamics and a nonlinear system level synthesis (SLS) controller resulting in input-to-state stable dynamics with respect to external disturbances. The majority of work uses SLS, the basis of which is described in (Anderson et al., 2019), and (Ho, 2020) describes conditions that transfer functions must satisfy in the nonlinear SLS setting.

1.1. Contributions

We propose a new method for controlling nonlinear systems in which we approximate nonlinear dynamics with polynomial terms, and design a system level synthesis (SLS) controller for the polynomial approximation. We show that using the controller on the original system results in stable closed-loop dynamics with respect to disturbances. Previous work in SLS focused on controlling polynomial systems (Conger et al., 2022), and we show how to extend this technique to general nonlinear systems, and how such a controller can be represented with function approximators like neural networks—while preserving stability— and then optimized to minimize a control objective. Furthermore, we give bounds on the control cost incurred by the optimized controllers in terms of the degree of the polynomial approximation and the effective domain of approximation. We empirically observe that the optimized nonlinear controllers have improved performance when compared with feedback linearization in a numerical example.

Organization: The remainder of the paper is organized as follows. We give a brief overview of related work and in Section 2 we give details about the class of nonlinear systems that we consider and background for the polynomial SLS controller. In Section 3, we derive a class of SLS controllers that are based on a polynomial approximation of the nonlinear dynamics that still have stability guarantees for the true nonlinear system. In Section 4 we discuss how to parameterize the controller by a neural network and optimize the controller. We give a bound on the control cost incurred by a properly optimized controller as a function of the disturbance size and approximation order. Lastly, in Section 5, we illustrate our method on a point mass system that requires the neural network to parameterize and optimize 14 controller functions, comparing our results with an unoptimized controller and feedback linearization. We conclude with a brief discussion of future work.

1.2. Related Work

Previous work (Conger et al., 2022) proved stability of the nonlinear SLS controller (3) for polynomial systems, and we extend this work by giving a performance bound on the control cost for k-times differentiable functions, a much broader class of nonlinear dynamics than just polynomial functions. The work by (Furieri et al., 2022) offers a solution for learning stabilizing controllers, similar to our work, but requires a stabilizing controller initially; our work provides that stabilizing controller. The criteria for general operator stability and robustness to model errors is developed in Ho (2020) which inspired our ideas for contributing specific operators that satisfy that criteria.

2. Preliminaries

We begin by describing the class of nonlinear systems which we can estimate with polynomial dynamics. We focus on nonlinear dynamics of the form:

$$x_{t+1} = f(x_t) + u_t + w_t, \tag{1}$$

where $w_t \in \mathbb{R}^n$ is a disturbance, $x_t \in \mathbb{R}^n$ is the state, and $u_t \in \mathbb{R}^n$ is the input. In this work we focus on the case where the system has an equilibrium point at x^* and the system is being perturbed by noise near x^*. We remark that the equilibrium does not need to be stable. Given this setup, we make the following assumption on the dynamics.

Assumption 1 *Assume that the nonlinear function $f \in C^\infty$ has bounded derivatives up to order k:*

$$\partial^p f(x_t) \leq M,$$

for all $|p| \leq k$. Further assume that the noise is bounded so that $\|w_t\|_\infty \leq W$.

The assumption on bounded noise allows us to treat the case with adversarial disturbances, while the assumption on bounded partial derivatives of up to order k is necessary to give our stability and boundedness guarantees for the SLS controllers. This is a generalization of the common Lipchitzness assumptions in nonlinear control, where in essence we show that one can leverage higher order smoothness to design stabilizing controllers.

2.1. Polynomial SLS Controller

Key to our controller design is a polynomial approximation of the nonlinear dynamics (1) and previous work on designing SLS controllers for polynomial systems (Conger et al., 2022). Indeed, for a polynomial system of the form:

$$x_{t+1} = \sum_{j=1}^{n} A_j x_t^{\otimes j} + u_t + w_t, \tag{2}$$

where $A_j \in \mathbb{R}^{n^j \times n^j}$ and $\otimes$ denotes the column-wise Kronecker product of x_t with itself $j-1$ times, a stabilizing finite impulse response (FIR) controller with time horizon T is given by:

$$u_t = -\sum_{m=0}^{T-1} \sum_{j=1}^{c_m} \alpha_j^{(m)} G_j^{(m)}(w_{t:t-m}) - \sum_{j=1}^{c_T} G_j^{(T)}(w_{t:t-T}), \tag{3}$$

where $w_{t:t-m}$ denotes disturbances from time $t-m$ to time t, where $w \in \mathbb{R}^n$ (Conger et al., 2022). The functions $G_j^{(m)}$ are monomials of entries in the disturbance vectors, that is, $G_j^{(m)}(w_{t:t-m}) = d_j^{(m)} w_q(i) w_p(j) \dots w_s(k)$ where each $w_q(i)$ denotes the ith entry of qth w vector raised to some power that is less than or equal to m, and $d_j^{(m)}$ is a coefficient from the dynamics. The c_m constant denotes the number of monomials that depend on the w_m terms multiplied with any more recent disturbances, and we say $c = \max_{m \in (0,T-1)} c_m$, which is the maximum number of terms in the inner sum. Lastly, each $\alpha_j^{(m)} \in [0,1]$ weights each monomial term.

This controller results in the state x_t as a function of $w_{t-1:t-T}$

$$x_t = \sum_{k=0}^{T-1} \sum_{j=1}^{c_m} (1-\alpha_j^{(m)}) G_j^{(m)}(w_{t-1:t-1-k}) + w_{t-1}. \tag{4}$$

which in turn implies that the state is uniformly bounded across time. This is a crucial step in our later analysis.

3. Polynomial Approximation of Nonlinear Dynamics

Given the setup described above, we now show that nonlinear systems of the form (1) can be stabilized by SLS controllers designed based on polynomial approximations to the dynamics.

To begin, we assume that we have access to a polynomial approximation to the nonlinear dynamics around the (potentially unstable) equilibrium. We assume that the approximation is derived using a Taylor expansion, rather than a Chebyshev approximation, because we are stabilizing to a single point and would like the highest accuracy of the estimation at that point. This allows us to rewrite the nonlinear dynamics as:

$$x_{t+1} = \sum_{j=1}^{k} H_j x_t^{\otimes j} + u_t + e(x_t) + w_t \tag{5}$$

where $e(x_t)$ is the error from the Taylor series approximation, and $e(0) = 0$. Given this form, we proceed by designing a SLS controller tailored to this polynomial approximation and show that it still has stability guarantees for the true system. Indeed, in the next section we illustrate the order-of-magnitude tradeoffs between the order of the polynomial approximation, domain over which a bounded control is required for control, and state dimension of the system. Intuitively, we expect a higher order of approximation to result in a larger region over which we can stabilize with bounded control, and that a larger state dimension will shrink this region, which we show via a specific relationship.

3.1. Stability Guarantee Under Polynomial Approximation

We now analyze the stability of the SLS controller (3) based off of the polynomial approximation to the nonlinear system. In particular, we prove that the controller is stabilizing over a certain domain around the equilibrium. To do so, we use some definitions that facilitate the proof. We denote $\alpha = \min_{j,m} \alpha_j^{(m)}$, and note that $G_j^{(m)}(w_{t:t-m}) < l \left\| w_{t:t-m} \right\|_\infty$ for some $l > 0$. Note that the linear bound on $G_j^{(m)}$ is possible because w_t is bounded for all t. Lastly, define input-to-state stability (ISS) with respect to the disturbance as a guarantee that $\|x_t\| \leq \gamma(\|W\|)$ for the initial condition $\|x_0\| = 0$, where γ is some class $\mathcal{K}$ function.

Theorem 2 *The system is ISS using approximate dynamics with the SLS controller* (3) *for k large enough and all $\alpha > 0$ such that*

$$\frac{MW^k}{(k+1)!} + lc(1-\alpha) < 1.$$

This gives a continuum of α over which to optimize while guaranteeing stability. Considering the case where $\alpha_j^{(m)} = 1$, which means disturbances are canceled as quickly as possible, we are left with the condition that $\frac{MW^k}{(k+1)!} \leq 1$ which illustrates how k must be sufficiently large. Note that this bound is not tight, and larger values of α than the ones satisfying the lemma result in stable behavior in simulation.

Proof We use the Lagrange form of the remainder for the Taylor expansion to bound the error term $e(W) \leq \frac{MW^{k+1}}{(k+1)!}$. To show stability we will show that the state remains bounded given bounded noise and $x_0 = 0$. We use a recursion relation to show how the state evolves and bound the final state. Starting with the first state, we have that $\|x_1\| \leq W$ since the first state was zero and was hit with a disturbance of size at most W. The second state, computed from (5), will also have the error from the approximate dynamics and the partially cancelled terms left by the controller

$$\|x_2\| \leq (1-\alpha)lcW + \left(\frac{MW^k}{(k+1)!} + 1\right) W.$$

Computing this again for the next state, x_3, we have

$$\|x_3\| \leq W\left[(1-\alpha)^2(lc)^2 + (1-\alpha)lc\left(2\frac{MW^k}{(k+1)!} + 1\right) + \left(\left(\frac{MW^{k+1}}{(k+1)!}\right)^2 + \frac{MW^k}{(k+1)!} + 1\right)\right]$$

and we can continue computing an upper bound on $\|x_t\|$ as

$$\|x_t\| \leq W \sum_{\tau=0}^{t-1} \sum_{i=0}^{\tau} \binom{\tau}{i} \left(\frac{MW^k}{(k+1)!}\right)^i (1-\alpha)^{\tau-i}(lc)^{\tau-i}.$$

Simplifying the above summation yields $\|x_t\| \leq W\frac{b^t-1}{b-1}$ where $b = \frac{MW^k}{(k+1)!} + lc(1-\alpha)$. From here we deduce that in order for the sum to be finite, the system must satisfy

$$\frac{MW^k}{(k+1)!} + lc(1-\alpha) < 1.$$

■

4. Optimizing the Controller

Given the set of stabilizing controllers in the previous section, we now give a method for parametrizing the controllers and optimizing their performance on a control objective. In particular we allow the coefficients $\alpha_j^{(k)}$ to be functions of the disturbances w and use a function approximator (e.g., a neural network) to learn the coefficients $\alpha_j^{(k)}$ as a function of the past disturbances. Crucially, we show that an optimized controller results in a bounded control effort.

We remark that previous work treated $\alpha_j^{(k)}$ as constant coefficients. However, allowing them to be functions broadens the class of controllers that (3) parameterizes, leading to better performance. Let $H_\theta : \mathbb{R}^{nk} \to (\alpha, 1)^k$ represent the function parameterized by a neural network with weights θ. We assume without loss of generality that H_θ is a class of functions such that there exists a $\theta \in \Theta$ such that $H_\theta \equiv 1$. We minimize the quadratic control cost

$$\begin{aligned} \min_\theta \sum_{t=1}^{N_T} u_t^\top R u_t & \qquad (6) \\ \alpha_j^{(m)} &= H_\theta(w_{t-1:t-1-m}) \\ u_{t+1} &= -\sum_{m=0}^{T-1}\sum_{j=1}^{c_m} \alpha_j^{(m)} G_j^{(m)}(w_{t:t-m}) - \sum_{j=1}^{c_T} G_j^{(T)}(w_{t:t-T}) \\ x_{t+1} &= f(x_t) + u_t + w_t \\ x_0 &= 0 \end{aligned}$$

for $R \succ 0$ and N_T the number of time steps for which we run the training disturbance series. Note that because the controller forces the system to have a finite impulse response, a cost on the state is not necessary for stabilization. The NN trains on random noise sequences that are longer than the FIR horizon and is penalized by the quadratic cost function, with the states and control inputs computed in the loss function. Note that the true dynamics f should be used to generate the next state, and the approximated dynamics are used to compute the controller. The controller is illustrated in 5.1 for a FIR horizon of $T = 3$.

4.1. Tradeoffs

Once the controller has been optimized, the $\alpha_j^{(m)}$ coefficients will be tuned so disturbances that decay are propagating through the dynamics and canceled at the end of the FIR horizon, and disturbances that are not decaying are cancelled immediately. This will have a lower cost than a controller that cancels all of the disturbances immediately, because cancelling a decayed disturbance is less costly than cancelling the larger disturbance. We upper bound the cost of the optimized controller by the control cost of setting all $\alpha_j^{(m)} = 1$. We show that cost is bounded by a constant, and illustrate tradeoffs for the size of the domain over which the approximation is valid, the order (power) of the polynomial approximation, and the input size required to stabilize the system. We denote the desired domain over which stability is guaranteed is h so that $\|x_t\|_1 \leq h$ which a design decision that can be satisfied by choosing the number of polynomial terms k in accordance with Theorem 2.

Theorem 3 *The controller u_t^1 with all $\alpha_j^{(m)} = 1$ subject to disturbances $\{w_i\}_{i=0}^T$ results in:*

$$\|u_t^1\|_2 \leq M\left(\frac{Wd(W^k d^k - 1)}{Wd - 1} + \frac{W(1 - W^k)}{W - 1}\right) = U^1 \quad \forall t,$$

where $d = \frac{M(nh)^k}{W(k+1)!} + 1$.

Expanding this upper bound for u_t^1 shows that $\left\|u_t^1\right\|_2 \leq M\left(\frac{(Wd)^{k+1}}{Wd-1} - \frac{W^{k+1}}{W-1} - \frac{Wd}{Wd-1} + \frac{W}{W-1}\right)$. Ignoring the last two terms, which are constant as W becomes large, we notice that $\frac{(Wd)^{k+1}}{Wd-1} - \frac{W^{k+1}}{W-1}$ has the same behavior as $(Wd)^k - W^k$. Examining how $(Wd)^k - W^k$ evolves for increasing k, we see that $(Wd)^k - W^k = \left(\frac{M(nh)^{k+1}}{(k+1)!} + W\right)^k - W^k$ goes to zero as k goes to infinity, since $\frac{M(nh)^{k+1}}{(k+1)!}$ goes to zero. We see that a higher order approximation, k, makes the error term smaller for sufficiently large k, while increasing the domain over which we expect the approximation bound to hold, h, or the state dimension, n, makes the error term larger. The following corollary follows directly from the theorem because we include in the class of functions H_θ the possibility that $H_\theta \equiv 1$, which gives a cost greater than or equal to the optimal solution to (6).

Corollary 4 *The optimized controller with a polynomial approximation of order k, bounded disturbances, and desired stability radius h therefore has a control cost at most*

$$L(H_\theta, \{w_i\}_{i=0}^T) \leq rTU^1$$

where $r = \sigma_{max}(R)$ and L is the cost of the optimization of (6).

The corollary shows that the optimal control cost is finite, decreases for sufficiently large k, and increases for increasing h.

Proof [Theorem 3] We approximate f using a Taylor expansion around the point to which we stabilize the system, x^*, and we assume without loss of generality $x^* = 0$. The expansion takes the form $p_k(x_t) = \sum_{j=1}^{k} H_j x_t^{\otimes j}$. Note that we do not include the $j = 0$ term because it corresponds to $f(x^*)$ which equals zero. The approximation error from the Taylor expansion is given by $e(x) = f(x) - p_k(x)$ where $p_k(x)$ is a polynomial approximation of order k. We can rewrite the dynamics as

$$x_{t+1} = p_k(x_t) + u_t + w_t + e(x_t). \tag{7}$$

The error can be treated as a disturbance term. In the worst (performance) case, we do not allow the error to propagate and feedback linearize immediately. The total disturbance term from time t will be determined at time step $t+1$, and can be counteracted at time step $t+1$, so $u_{t+1} = -p_k(x_{t+1})$ which can be expanded as

$$\begin{aligned} u_{t+1} &= -p_k(w_t + e(x_t)) \\ &= -\sum_{j=1}^{k} H_j (w_t + e(x_t))^{\otimes j} \\ &= -\sum_{j=1}^{k} H_j \sum_{i=1}^{j} C_{ji} \left(w_t^{\otimes j-i} \otimes e(x_t)^{\otimes i} \right) \end{aligned}$$

where the C_{ij} matrices contain the binomial expansion coefficients. We take the 2-norm of u_t to upper bound the control input, and then use the property that the 2-norm of the Kronecker product can be upper bounded by the product of the individual norms (Cochrane, 2020; Kong et al., 2018). The norm of the control input is given by

$$\|u_t\|_2 \leq \sum_{j=1}^{k} \|H_j\|_2 \sum_{i=1}^{j} \|C_{ji}\|_2 \, \|w\|_2^{j-i} \, \|e(x_t)\|_2^{i}.$$

We use an upper bound on the binomial matrices $\|C_{ij}\|_2 \leq j!\binom{j}{i}$, and the worse-case noise magnitude W to further bound the expression for $\|u_t\|_2$. We also denote $\|e(x_t)\|_2 = e_x$ for ease of notation

$$u_t \leq \sum_{j=1}^{k} \|H_j\|_2 \sum_{i=1}^{j} \frac{j!^2}{(j-i)!i!} W^{j-i} e_x^i.$$

Evaluating the inner sum in a symbolic solver gives the relation

$$\sum_{i=1}^{j} \frac{j!}{(j-i)!i!} \left(\frac{e_x}{W}\right)^i = \left(\frac{e_x}{W} + 1\right)^j - 1$$

which we can use to simplify the bound u_t to a single sum

$$u_t \leq \sum_{j=1}^{k} \|H_j\|_2 \, j! \, W^j \left(\left(\frac{e_x}{W} + 1\right)^j - 1 \right).$$

Now we use the definition of $\|H_j\|_2$ as the matrix coefficients for the Taylor expansion of the function, which means each norm is upper-bounded by the bound of the $k+1$-th derivatives divided by a factorial, that is, $\|H_j\|_2 \leq M/j!$. This leaves us with the sum

$$u_t \leq M \sum_{j=1}^{k} W^j \left(\left(\frac{e_x}{W} + 1 \right)^j - 1 \right).$$

Lastly, we can again evaluate this sum in a symbolic solver. Let $d = \frac{e_x}{W} + 1$ for ease of notation, and then we have that $\|u_t\|_2 \leq M \left(\frac{Wd(W^k d^k - 1)}{Wd - 1} + \frac{W(1 - W^k)}{W - 1} \right)$. ■

5. Example

We illustrate the polynomial approximation on an example system. We approximate the system dynamics and compute the controllers. Then we parameterize the $\alpha_j^{(m)}(w_{t:t-m})$ using a NN and implement a quadratic cost function. After training the NN, we compare the total quadratic cost with feedback linearization under simulation scenarios in which the system is near its unstable equilibrium and continually battered by disturbances. We trained the network on Google Colab's freely available Intel(R) Xeon(R) CPU @ 2.20GHz using stochastic gradient descent.

5.1. Neural Network Architecture

The neural network accepts as input a list of the previous T disturbances $w_{t:t-T}$. The disturbances pass through a NN with linear layers, ReLU activation functions, and dropout layers. The NN outputs the values assigned to each value of $\alpha_j^{(m)}$ corresponding to the most recent disturbance. Causality is enforced by computing each set of $\alpha_j^{(m)}$ at time t using inputs $w_{t-1:t-1-m}$, so that disturbances are not from current or future time steps. The diagram in Figure 5.1 illustrates how disturbances are used to compute coefficient weights.

5.2. Point Mass

In this example we model a hovering point mass subject to disturbance. We seek to stabilize the point to a fixed vertical height, and minimize a linear quadratic cost function of the state and input. The discretized dynamics are given by

$$\ddot{z} = -g + \frac{1}{m} f_a(z, \dot{z}) + u + w \tag{8}$$

where $\ddot{z}$ denotes the acceleration, g is the gravitational constant, m is the mass of the point, and f_a is a NN model for the acceleration dynamics. The model is taken from (Shi et al., 2019, Eq 8), which demonstrates a method for learning the dynamics of a system while flying and providing stability guarantees via Lipschitz constraints on the network layers.

First, we fit a polynomial to the nonlinear function f_a above, using order up to $k = 3$ for the polynomial. We then generate the controller using a symbolic python package for an FIR horizon of $T = 4$, which means that the effects of a disturbance arriving at time t will be completely neutralized by time $t + 4$. This results in 14 different functions of $\alpha_j^{(m)}$ to be computed at each time step. The

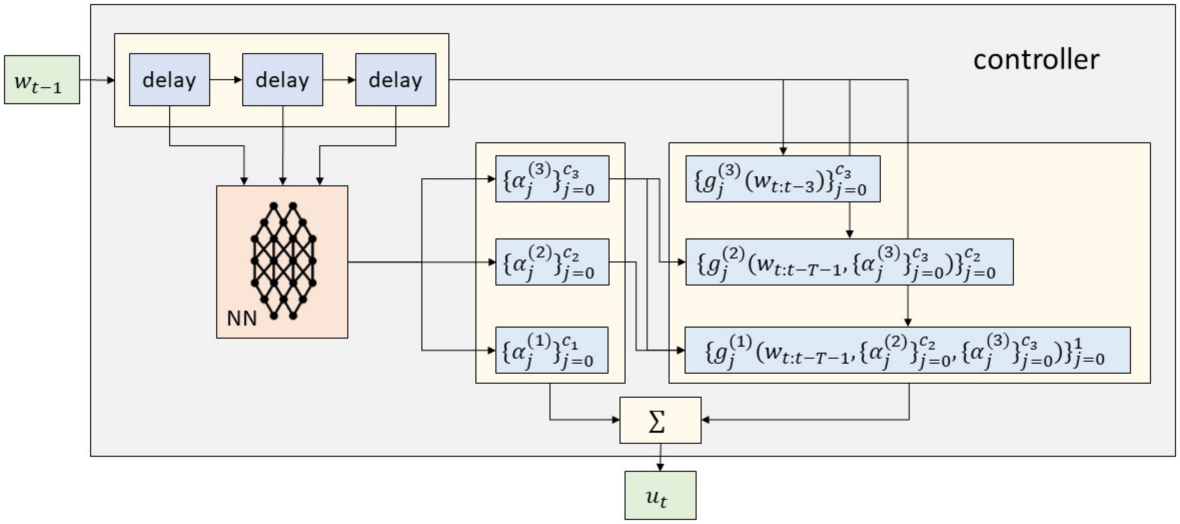
controller
w_{t-1}
delay
delay
delay
NN
$\{\alpha_j^{(3)}\}_{j=0}^{c_3}$
$\{\alpha_j^{(2)}\}_{j=0}^{c_2}$
$\{\alpha_j^{(1)}\}_{j=0}^{c_1}$
$\{g_j^{(3)}(w_{t:t-3})\}_{j=0}^{c_3}$
$\{g_j^{(2)}(w_{t:t-T-1}, \{\alpha_j^{(3)}\}_{j=0}^{c_3})\}_{j=0}^{c_2}$
$\{g_j^{(1)}(w_{t:t-T-1}, \{\alpha_j^{(2)}\}_{j=0}^{c_2}, \{\alpha_j^{(3)}\}_{j=0}^{c_3})\}_{j=0}^{1}$
Σ
u_t

Figure 1: The controller diagram shows a FIR horizon of $T = 3$. The NN generates values for each α coefficient at each time step, which are coefficients to the $G_j^{(m)}$ and used to compute the $G_j^{(m)}$ functions of more recent disturbances.

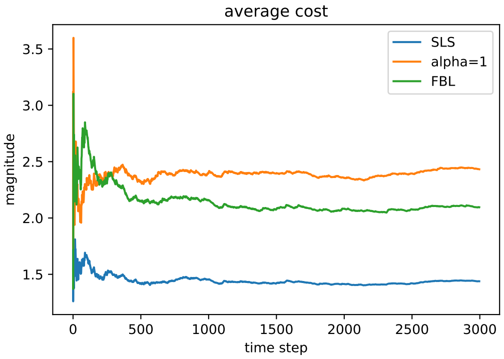

Figure 2: The time-averaged cost of the SLS controller is lower than the cost resulting from a traditional feedback linearization controller.

NN has 8 linear layers of size 2048 each with ReLU and dropout activation functions between each layer. The stochastic gradient descent has a learning rate of 1e-6 and the disturbance bound is 1. The time-averaged cost of our controller is shown in Figure 5.2. The feedback linearizing (FBL) controller is given by $u = g - \frac{1}{m} f_a(z_t, \dot{z}_t) + v_t$, where v_t is selected to minimize the quadratic cost, as determined by the discrete Ricatti equation with $Q = I$ and $R = I$.

We see that our controller has a lower average cost compared with the FBL controller by about 33%. In the case where we set all $\alpha_j^{(k)}$ to one, which we test as a baseline for an untrained NN, we can see that the untrained SLS controller performs worse than the FBL controller. A key difference between the untrained SLS and FBL controllers is that the FBL controller makes the state exponentially decay, while the untrained SLS controller cancels disturbances in finite time, which are inherently different behaviors. We do not necessarily expect the untrained SLS controller to outperform the FBL controller because the FBL controller is optimal for the quadratic cost without disturbances.

6. Conclusion and Outlook

We showed that using a polynomial approximation of nonlinear dynamics and applying the SLS controller leads to stable closed-loop dynamics, and that the control cost can be decreased by parameterizing the coefficients as NN functions. The stability guarantees hold regardless of the how well the NN is optimized, and when optimized well, we illustrate the tradeoffs among the stability size h, polynomial order k and the disturbance size W. Future work will investigate the explicit relationship between the coefficients and the disturbances, formalize the controller paramterization for dynamics of the form $x_{t+1} = f(x_t) + g(x_t)u_t$, and implement this controller on a physical system.

Acknowledgments

Thanks to Guanya Shi, Eitan Levin, Chris Yeh, and Steven L. Brunton for helpful technical discussions around machine learning and polynomial systems. LC is supported by a NDSEG Graduate Fellowship from the Air Force Office of Scientific Research.

References

Aaron D. Ames, Kevin Galloway, Koushil Sreenath, and Jessy W. Grizzle. Rapidly Exponentially Stabilizing Control Lyapunov Functions and Hybrid Zero Dynamics. *IEEE Transactions on Automatic Control*, 59(4):876–891, April 2014. ISSN 1558-2523. doi: 10.1109/TAC.2014.2299335.

James Anderson, John C. Doyle, Steven H. Low, and Nikolai Matni. System level synthesis. *Annual Reviews in Control*, 47:364–393, 2019. ISSN 13675788. doi: 10.1016/j.arcontrol.2019.03.006. URL https://linkinghub.elsevier.com/retrieve/pii/S1367578819300215.

Robert Cochrane. Nuclear Norm Under Tensor Kronecker Products, February 2020. URL http://arxiv.org/abs/2001.07202. arXiv:2001.07202 [cs, math].

Lauren Conger, Jing Shuang Li, Eric Mazumdar, and Steven Brunton. Nonlinear System Level Synthesis for Polynomial Dynamical Systems. July 2022.

Luca Furieri, Clara Lucia Galimberti, and Giancarlo Ferrari-Trecate. Neural System Level Synthesis: Learning over All Stabilizing Policies for Nonlinear Systems, March 2022. URL http://arxiv.org/abs/2203.11812. arXiv:2203.11812 [cs, eess].

Karima Hammar, Tounsia Djamah, and Maamar Bettayeb. Fractional Hammerstein system identification using polynomial non-linear state space model. In *2015 3rd International Conference on Control, Engineering & Information Technology (CEIT)*, pages 1–6, May 2015. doi: 10.1109/CEIT.2015.7232994.

Dimitar Ho. A System Level Approach to Discrete-Time Nonlinear Systems. In *2020 American Control Conference (ACC)*, pages 1625–1630, July 2020. doi: 10.23919/ACC45564.2020.9147857. ISSN: 2378-5861.

Xu Kong, Jicheng Li, and Xiaolong Wang. New estimations on the upper bounds for the nuclear norm of a tensor. *Journal of Inequalities and Applications*, 2018(1):282, 2018. ISSN 1025-5834. doi: 10.1186/s13660-018-1861-1.

V.I. Lovchakov and O.A. Sibyakin. Multidimensional Linearization of Control Objects with Polynomial Nonlinearities. In *2020 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon)*, pages 1–4, October 2020. doi: 10.1109/FarEastCon50210.2020.9271438.

Alexandre Mauroy and Jorge Goncalves. Linear identification of nonlinear systems: A lifting technique based on the Koopman operator. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 6500–6505, December 2016. doi: 10.1109/CDC.2016.7799269.

Elizabeth Qian, Boris Kramer, Benjamin Peherstorfer, and Karen Willcox. Lift & Learn: Physics-informed machine learning for large-scale nonlinear dynamical systems. *Physica. D, Nonlinear Phenomena*, 406(C), February 2020. ISSN 0167-2789. doi: 10.1016/j.physd.2020.132401. URL https://www.osti.gov/pages/biblio/1803677.

Guanya Shi, Xichen Shi, Michael O'Connell, Rose Yu, Kamyar Azizzadenesheli, Animashree Anandkumar, Yisong Yue, and Soon-Jo Chung. Neural Lander: Stable Drone Landing Control Using Learned Dynamics. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9784–9790, May 2019. doi: 10.1109/ICRA.2019.8794351. ISSN: 2577-087X.

Guanya Shi, Wolfgang Hönig, Xichen Shi, Yisong Yue, and Soon-Jo Chung. Neural-Swarm2: Planning and Control of Heterogeneous Multirotor Swarms Using Learned Interactions. *IEEE Transactions on Robotics*, 38(2):1063–1079, April 2022. ISSN 1941-0468. doi: 10.1109/TRO.2021.3098436.

Robin Strässer, Julian Berberich, and Frank Allgöwer. Data-Driven Control of Nonlinear Systems: Beyond Polynomial Dynamics. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 4344–4351, December 2021. doi: 10.1109/CDC45484.2021.9683211. ISSN: 2576-2370.

E. Yu. Vataeva, V. F. Shishlakov, D. V. Shishlakov, and N. V. Reshetnikova. Parametric Synthesis of Nonlinear Automatic Control Systems with Polynomial Approximation. In *2019 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF)*, pages 1–5, June 2019. doi: 10.1109/WECONF.2019.8840123.

Tsuyoshi Yuno and Toshiyuki Ohtsuka. Realization of a vector field via state feedback for polynomial dynamical systems. In *2014 European Control Conference (ECC)*, pages 2454–2459, June 2014. doi: 10.1109/ECC.2014.6862184.