Model free control policy search for liftable systems

Alessandra French, Jack Umenberger, Paul Goulart

Department of Engineering Science, University of Oxford

UNIVERSITY OF OXFORD

Introduction and Aims

A central concern of control theory is to design an optimal controller for a system with partially unknown dynamic properties.

Consider modelling the dynamics of a physical system with the equation $\frac{dx}{dt} = Ax + Bu$. If the dynamics are linear, one could derive an optimal feedback controller of the form $u = Kx$ by minimising an infinite horizon quadratic cost function, known as the linear quadratic regulator (LQR). This optimal controller balances how fast you stabilise the state and how much control energy you expend to do so.

$$l(t) = \int_0^\infty x(t)^T Q x(t) + u(t)^T R u(t)\, dx$$

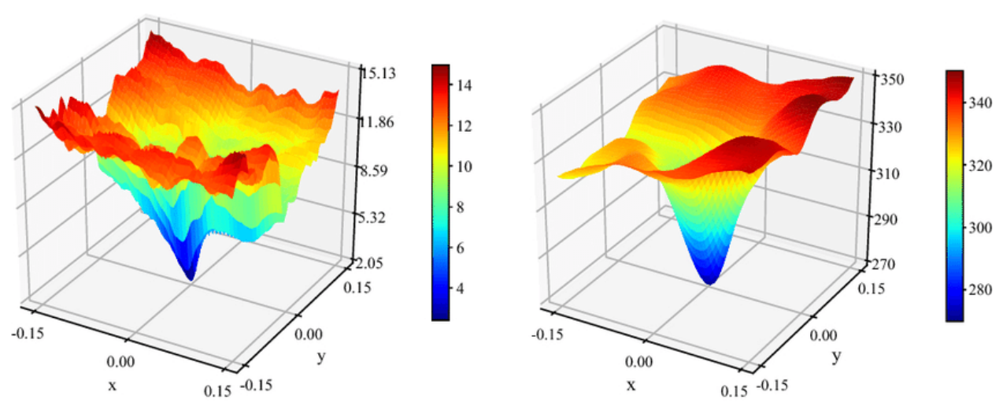


Figure 1. LQR cost function.

However, when dealing with a system with a nonlinear, monomial squared, term in the system dynamics, leading to a nonlinear term in the optimal controller, one could consider 'lifting' this system to a higher dimensional state space representation by assigning a new variable to the squared monomial, and adjusting the time derivative equation accordingly, such that the new linear system is equivalent in its dynamic behaviour to the nonlinear system.

The optimal LQR controller can be computed by solving the algebraic Riccati equation. If you have a fairly simple system with known system dynamics, one can exploit that knowledge and solve the Riccati equation. However, for very large-scale problems, it may be too expensive to compute the optimal solution exactly, so it is necessary to settle for a suboptimal solution that we can compute in "reasonable time". In these settings, first order methods like gradient descent can be very attractive. It is well-known that gradient descent in the space of control parameters K converges to the global optimum for the LQR problem. Given the equivalence explained above, gradient descent on the quadratic cost function with respect to the parameters of the nonlinear control policy, with states generated by the nonlinear dynamics, should converge to the optimal policy.

The first key research question can then be stated as follows:

Can we confirm via numerical simulations that gradient descent on the nonlinear quadratic regulator problem, characterised by nonlinear dynamics and nonlinear control, does indeed converge to the global optimum, as expected?

In most practical problems the system dynamics are not known precisely. My second research question focuses on exploring the case where you have a potential additional nonlinear monomial term in the system dynamics, leading to an additional nonlinear term in the controller. In this case, it is not at all obvious how gradient descent will behave, and what effect this will have on the behaviour of the closed-loop system, as the closed-loop is no longer obviously equivalent to a lifted, linear system. The effect of additional monomials can also be investigated.

Background: Dynamic Systems and the Koopman Operator

For this project, I will use a discrete time version of the system. For this, the system dynamics can be modelled by the equation

$$x_{k+1} = F_t(x_k, u_k)$$

The Koopman Operator

The Koopman operator K is an infinite dimensional linear operator which advances the set of observable functions of the state of a dynamical system.

$$Kg = g(F_t(x_k, u_k)) = g(x_{k+1})$$

The set of observable functions on the state form an infinite dimensional Hilbert space. We can consider a finite subspace of these functions. If operating on this subspace with the Koopman operator produces a result which remains in the subspace, we call this a Koopman-invariant subspace. It is then possible to restrict the Koopman operator to a finite dimensional linear operator which acts on the subspace, and in doing so represent a system with nonlinear dynamics as an equivalent system with linear dynamics.

Methodology

The system I chose to test on was a modified version of one from Brunton et al. [5] Although it is not controllable in all directions, it is stable in the uncontrollable direction.

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{k+1} = \begin{bmatrix} \mu & 0 \\ 0 & \lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_k + \begin{bmatrix} 0 \\ (1-\lambda)x_1^2 \end{bmatrix}_k + \begin{bmatrix} 0 \\ u \end{bmatrix}_k$$

Using Koopman operator theory, it is possible to transform this to the equivalent higher dimensional linear system,

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}_{k+1} = \begin{bmatrix} \mu & 0 & 0 \\ 0 & \lambda & (1-\lambda) \\ 0 & 0 & \mu^2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}_k + \begin{bmatrix} 0 \\ u \\ 0 \end{bmatrix}_k$$

where

$$y_1 = x_1, y_2 = x_2, y_3 = x_1^2$$

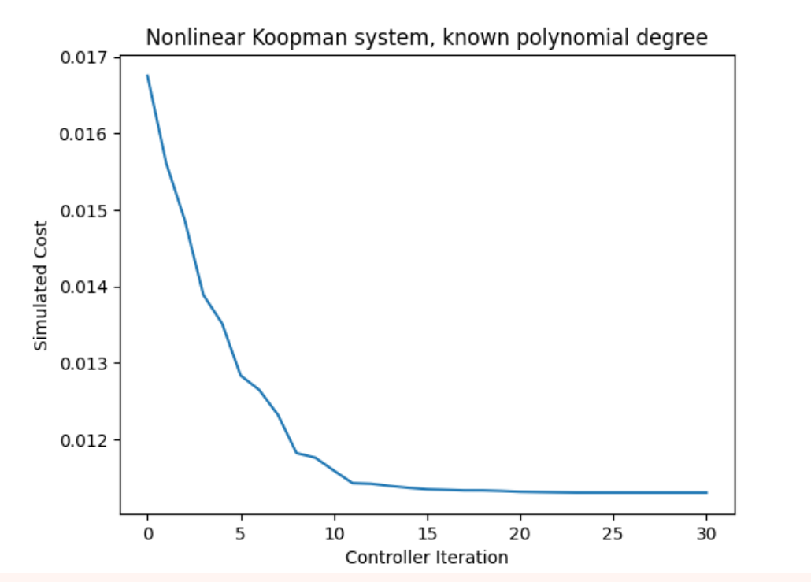
The basic algorithm for computing the controller which minimises the quadratic cost operates as follows:

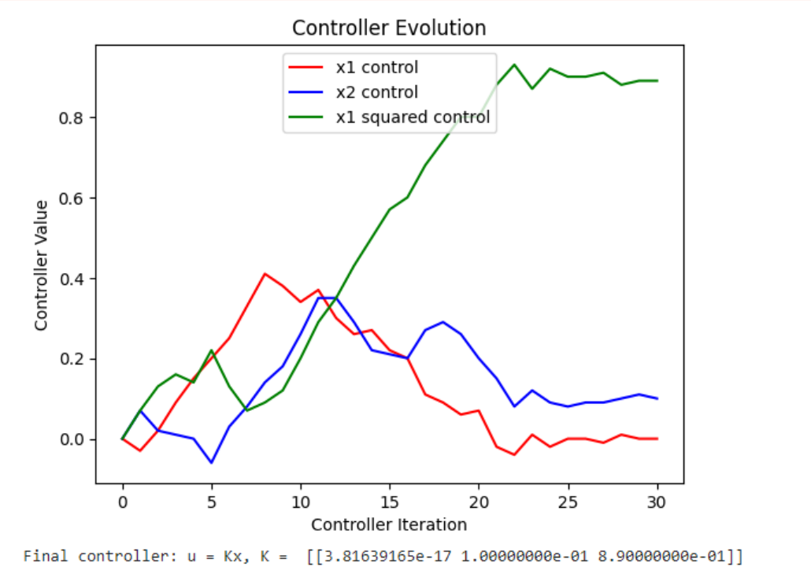
- **Initialisation:** I initialise the starting state vector, and choose my desired state to be a vector of zeros for simplicity and consistency. I initialise my control to be a vector of zeros, and calculate the initial cost of my system (explained more below).
- **Trial move:** A random move in the controller parameter space is chosen, and is added to the current controller.
- **Simulation:** I approximate the infinite horizon cost function with a finite horizon cost function, which inputs the system and current (trial) controller and calculates the cost for a set number of time steps. For the small stable systems I was using, I found that 100 time steps is enough to not affect the solution.
- **Selection:** If the cost with the trial controller is lower than the previous lowest cost, the move is accepted and the controller is updated to be the trial controller. Otherwise, the trial controller is rejected.
- **Iteration:** This process is repeated until convergence.

Results and Findings

Shown below are graphs showing the evolution of LQR cost and controller parameters of the system as gradient descent is applied.

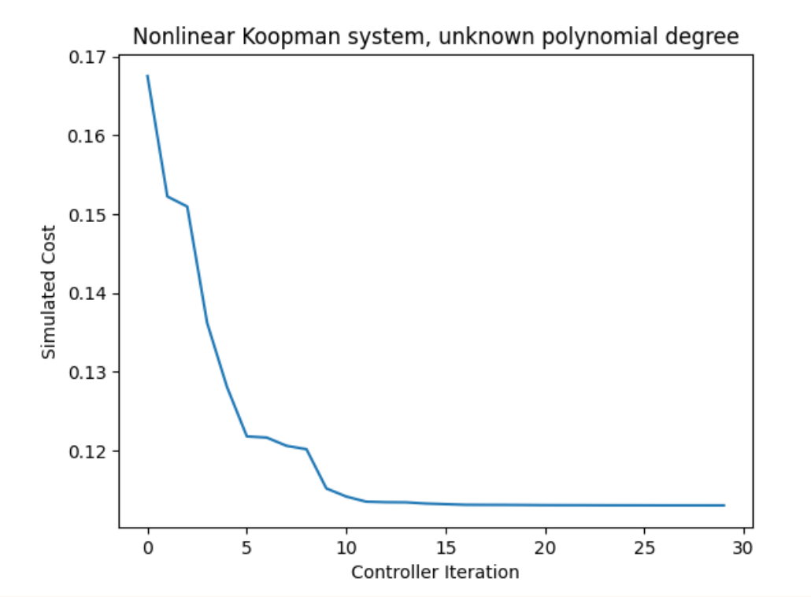
Known Polynomial Degree

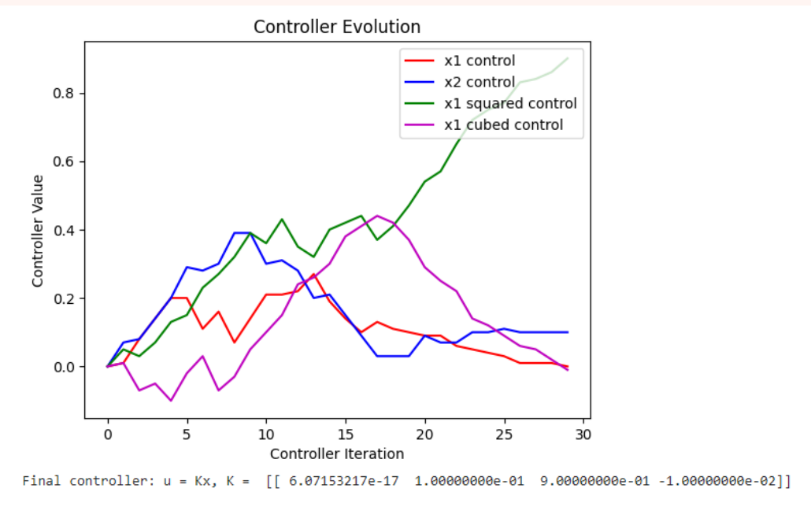




```
Final controller: u = Kx, K =  [[3.81639165e-17 1.00000000e-01 8.90000000e-01]]
```

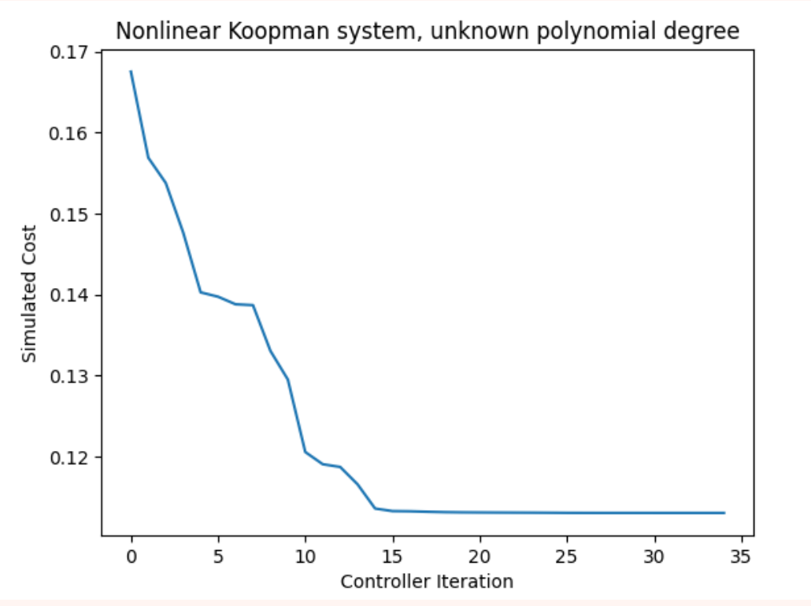
Unknown Polynomial Degree - Up to Cubic

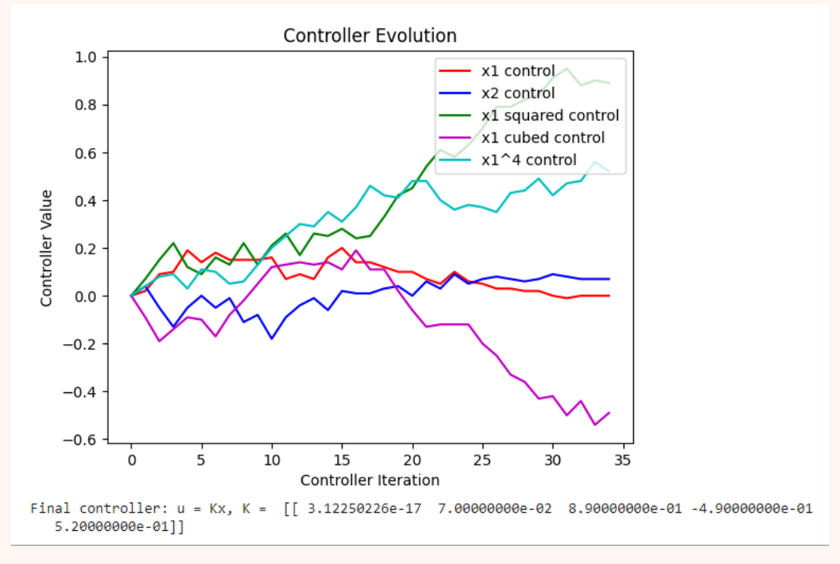




```
Final controller: u = Kx, K =  [[ 6.07153217e-17  1.00000000e-01  9.00000000e-01 -1.00000000e-02]]
```

Unknown Polynomial Degree - Up to Quartic



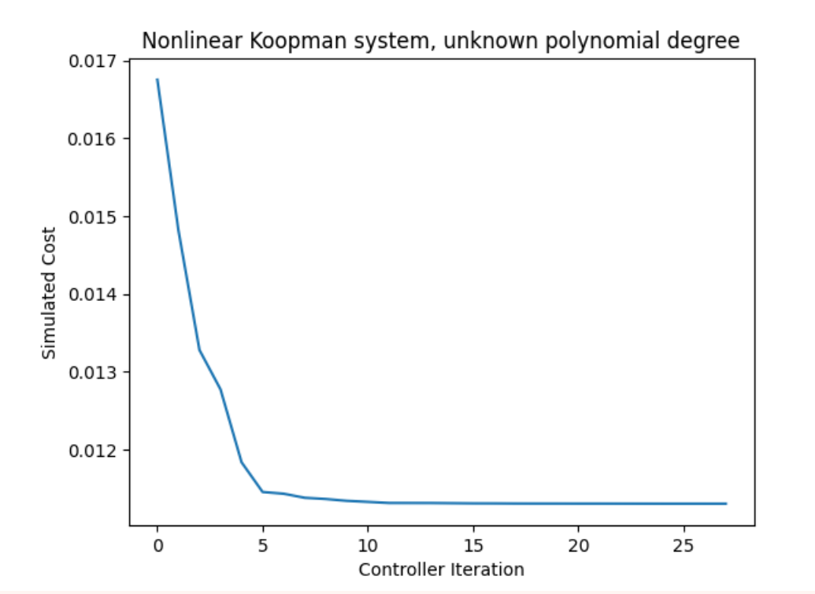


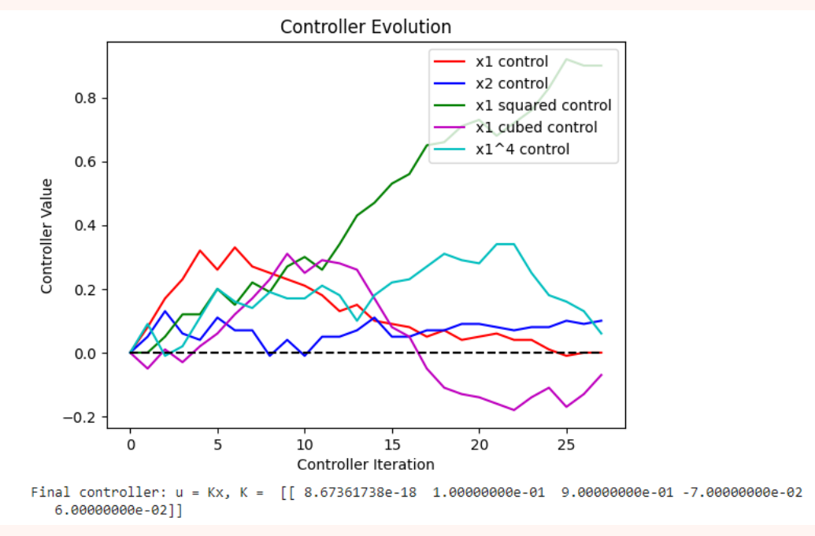
```
Final controller: u = Kx, K =  [[ 3.12250226e-17  7.00000000e-02  8.90000000e-01 -4.90000000e-01
   5.20000000e-01]]
```

Results and Findings cont.

The system with additional monomials up to quartic degree returned a very different controller to the one I was expecting. I had run it with 1,000,000 gradient descent iterations, and 100 simulation steps per iteration. I decided to try running the code again, this time with 1,000 simulation steps and 100,000 gradient descent iterations.

Quartic degree, iteration 2

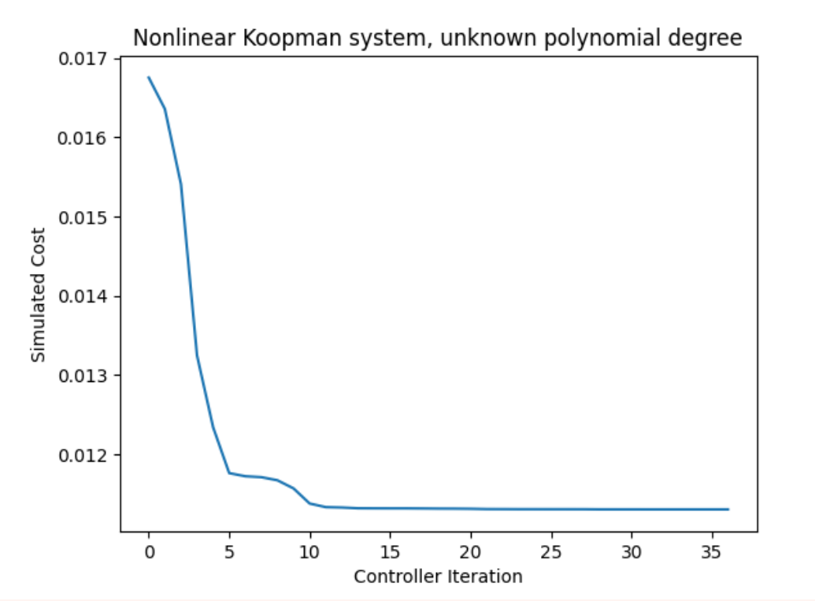


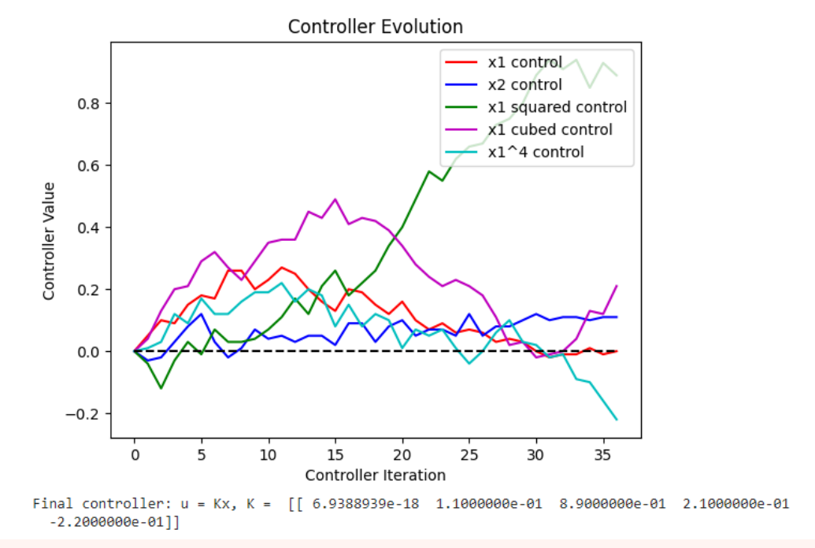


```
Final controller: u = Kx, K =  [[ 8.67361738e-18  1.00000000e-01  9.00000000e-01 -7.00000000e-02
   6.00000000e-02]]
```

This now looked like the controller parameters were close to converging, so I ran it one final time, still with 1,000 steps per iteration, but this time with 300,000 gradient descent iterations.

Quartic degree, iteration 3





```
Final controller: u = Kx, K =  [[ 6.9388939e-18  1.1000000e-01  8.9000000e-01  2.1000000e-01
  -2.2000000e-01]]
```

Discussion and Conclusion

For the system I chose, the exact solution obtained by directly solving the Riccati equation returns the controller $u = 0.099x_2 + 0.8919x_1^2$, corresponding a proportional controller on the lifted solution with parameters $K = [0\,0.09\,0.8919]$.

With known polynomial degree, my code returns a final controller $u = 0.1x_2 + 0.89x_1^2$, which is almost identical to the optimal controller, as expected.

Including an additional cubic monomial to the possible dynamics changes the final controller to $u = 0.1x_2 + 0.9x_1^2 - 0.01x_1^3$, however this is again almost identical to the optimal controller. In this case, the addition of a potential extra nonlinear monomial does not prevent convergence to the optimal controller.

However, the addition of a further (quartic) nonlinear term does change the results. It seems that having so many degrees of freedom makes the optimization less reliably convergent, and more prone to errors caused by approximating the infinite horizon cost function with a finite approximation. In the final iteration of controller optimisation with a quartic degree polynomial, the system briefly hits the true optimal solution at iteration 30, but then quickly diverges. I think that an avenue for further research would be to investigate whether the optimization could be made more robust but adding a small regularization term to the cost function.

References

[1] Bassam Bamieh.
Myths, misconceptions and misuses of nonlinearity.
Department of Mechanical Engineering, University of California, Santa Barbara.

[2] Juan C Perdomo Kaiqing Zhang Jack Umenberger, Max Simchowitz and Russ Tedrake.
Globally convergent policy search for output estimation.
Advances in Neural Information Processing Systems, 2022.

[3] Sham Kakade Maryam Fazel, Rong Ge and Mehran Mesbahi.
Global convergence of policy gradient methods for the linear quadratic regulator, journal = Proceedings of Machine Learning Research, year = 2018,.

[4] Claude E. Shannon.
A mathematical theory of communication.
Bell System Technical Journal, 27(3):379–423, 1948.

[5] Joshua L. Proctor J. Nathan Kutz Steven L. Brunton, Bingni W. Brunton.
Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control.
PLOS ONE, 2016.

alessandra.french@some.ox.ac.uk