2017 IEEE 56th Annual Conference on Decision and Control (CDC)
December 12-15, 2017, Melbourne, Australia

Improving Model Predictive Controller Turnaround Time Using Restricted Lagrangians

Anson Maitland and John McPhee

Abstract— **We present a new application of proper orthogonal decomposition (POD) to optimal control. By restricting the Lagrangian of an optimal control problem to a suitable affine subspace, we can achieve a reduction in computational cost leading to faster turnaround times with minimal degradation in controller performance. An explicit algorithm for nonlinear model predictive control (NMPC) reduction using POD is presented along with some initial error analysis. To the best of our knowledge, this is the first time such an approach has been presented. We applied this approach to the control of a vehicle during a double lane change maneuver using NMPC and achieved 2 times faster turnaround times with excellent controller performance. This reduction approach for the development of real-time optimal controls is very promising and introduces some new research directions.**

I. Introduction

With advances in modelling software and computation we can now build and simulate highly complex systems. Subsequently, there is a strong desire within industry to use these new models with advanced control strategies, such as model predictive control (MPC), to push their technologies to greater levels of efficiency and autonomy. These models, however, often suffer from one major drawback: they are expensive to compute. Historically, this has meant that these models have been limited to purely research settings or control applications with slow sample times. In an effort to expand the application of these models, significant research has been devoted to model reduction. In model reduction the goal is to capture a model's desired behaviour by a computationally simpler one. These reduced models can then act as a surrogate within the controller leading to fast advanced controllers.

One of the most successful and widely utilized model reduction strategies is the proper orthogonal decomposition-Galerkin (POD-Galerkin) projection method which we refer to simply as POD. It is a reduction method that stems from linear theory but has nonetheless been successfully applied to a wide variety of nonlinear models. Applications include modeling and control of in-cylinder flow [1], the heat diffusion equation [2], [3], fluid channel flow [4], distributed reactor systems [5] and vehicle dynamics [6], to list a few. POD has been applied to control problems by reducing the order of the original plant models. Interestingly, POD when applied to a nonlinear model does not reduce its computational burden for fixed-step integrators typically used in controls applications. However, due to the reduction of model order achieved by this method, the dimension of the resulting optimization problem is reduced and a speedup in controller computation has been observed, *e.g.* [7].

In this paper we present a strategy to reduce the computational burden of MPC by a novel application of POD. We are able to reduce the dimension of the finite horizon optimal control problem (FHOCP) found within each timestep of an MPC. This is done by applying POD to the Lagrangian of the FHOCP - not the plant model. We extract an affine subspace of reduced dimension that best captures the trajectory of the FHOCP solutions over the controller timesteps. Then we restrict the Lagrangian to that affine subspace. This is a new strategy that to the authors knowledge has not been presented elsewhere.

This paper is organized as follows. In Section II we review the fundamentals of POD and its application to model reduction. In Section III we briefly review the role of Newton's method in MPC and in Section IV we describe the algorithm to reduce an MPC via POD and provide an initial error analysis of the method. In Section V we apply the POD-reduced NMPC to the control of a simple car model during a double lane change maneuver and analyze the results. Lastly, in Section VI we summarize our findings and provide future directions for investigation.

II. POD and Model Reduction

Suppose we are given a model represented by a system of explicit ordinary differential equations

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)), \quad \mathbf{x}(0) = \mathbf{x}_0$$

where $\mathbf{x}(t) \in \mathbb{R}^n$. The POD-reduced model can be found by the following algorithm:

1) Construct a zero-mean snapshot matrix $\mathbf{X} = \begin{pmatrix}\mathbf{x}(t_1) - \hat{\mathbf{x}} & \cdots & \mathbf{x}(t_N) - \hat{\mathbf{x}}\end{pmatrix} \in \mathbb{R}^{n \times N}$ made up of $N > n$ observations of the model over a collection of simulations where $\hat{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}(t_i)$.
2) Compute the singular value decomposition (SVD) of the snapshot matrix $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^T$ and then truncate $\mathbf{U}$ to its first $r < n$ columns to form $\mathbf{U}_r \in \mathbb{R}^{n \times r}$.
3) Construct the reduced order model using a Galerkin projection

$$\dot{\mathbf{y}}(t) = \mathbf{U}_r^T \mathbf{f}(\mathbf{U}_r \mathbf{y}(t) + \hat{\mathbf{x}}), \quad \mathbf{y}(0) = \mathbf{U}_r^T(\mathbf{x}_0 - \hat{\mathbf{x}}) \quad (1)$$

where $\mathbf{y}(t) \in \mathbb{R}^r$ are coordinates of the reduced space. The original states can be approximated via $\mathbf{x}_r(t) = \mathbf{U}_r \mathbf{y}(t) + \hat{\mathbf{x}} \approx \mathbf{x}(t)$.

This work was supported by Natural Sciences and Engineering Research Council of Canada (NSERC), the Toyota Motor Corporation and Maplesoft™.

The authors are with the Department of Systems Design Engineering, University of Waterloo, 200 University Ave W, Waterloo, Canada. Corresponding author: agmaitla@uwaterloo.ca

Underlying POD is a data compression problem. POD captures the system's output data in as few dimensions as possible by solving the following constrained optimization problem for a fixed $r < n$,

$$\min_{\mathbf{U}_r \in \mathbb{R}^{n\times r}} \quad \sum_{i=1}^{N} \left\| (\mathbf{I} - \mathbf{U}_r \mathbf{U}_r^T)(\mathbf{x}(t_i) - \hat{\mathbf{x}}) \right\|^2$$
$$\text{subject to} \quad \mathbf{U}_r^T \mathbf{U}_r = \mathbf{I}$$

where $\|\cdot\|$ denotes the Euclidean norm. The truncated $\mathbf{U}_r$ found in step 2) of the algorithm presented above is the optimal solution to this constrained optimization problem [8]. Typically r is chosen such that $\frac{\sum_{k=r+1}^{n} \sigma_k^2}{\sum_{k=1}^{n} \sigma_k^2} < \epsilon$ where $\sigma_k \in \sigma(\mathbf{X})$ and $\sigma(\mathbf{X}) = [\sigma_1, \sigma_2, \ldots, \sigma_n]$ denotes the list of singular values of $\mathbf{X}$ in descending order and $0 < \epsilon < 1$ is a threshold close to 0.

We note that as $\mathbf{U}$ is an orthogonal matrix so that $\mathbf{P}_r = \mathbf{U}_r \mathbf{U}_r^T \in \mathbb{R}^{n\times n}$ is an orthogonal projection matrix onto $\langle \mathbf{U}_r \rangle$ which denotes the column space of $\mathbf{U}_r$. From this we can see that (1) is equivalent to

$$\dot{\mathbf{x}}_r(t) = \mathbf{P}_r \mathbf{f}(\mathbf{x}_r(t)), \quad \mathbf{x}_r(0) = \mathbf{P}_r \mathbf{x}_0 + \mathbf{P}_r^{\perp} \hat{\mathbf{x}}$$

which is the projection of the restriction of $\mathbf{f}$ to the affine space $\langle \mathbf{U}_r \rangle + \hat{\mathbf{x}}$ where $\mathbf{P}_r^{\perp} = \mathbf{I} - \mathbf{P}_r$ is the projection matrix orthogonal to $\mathbf{P}_r$ and $\mathbf{x}_r \in \langle \mathbf{U}_r \rangle + \hat{\mathbf{x}}$.

It is important to note that model reduction via POD requires an offline stage where the snapshot matrix is computed using simulations of the original model. Different choices of snapshot matrix will lead to different error statistics of the reduced model and for nonlinear models choosing which simulations are best to use in constructing your snapshot matrix remains a difficult problem, see [9].

III. Newton's Method in MPC

MPC is a closed-loop control strategy that solves a FHOCP at each timestep. Broadly speaking there are two approaches to solving the FHOCP: direct and indirect methods [10]. In the following, we focus on the direct approach where the FHOCP is transcribed directly to a constrained optimization problem. To solve this problem we need to find the minima of a suitably formulated unconstrained Lagrangian $\mathcal{L}(\mathbf{z}) : \mathbb{R}^l \to \mathbb{R}$ where $\mathbf{z} \in \mathbb{R}^l$ contains all states and controls over the horizon, any slack variables and Lagrange multipliers. The optimum is found by solving for the root of the gradient of the Lagrangian, *i.e.* the $\mathbf{z}^*$ satisfying $\nabla \mathcal{L}(\mathbf{z}^*) = 0$. This is most often done using an optimization routine that utilizes Newton's (or a quasi-Newton) method since Newton's method converges to the solution very quickly (quadratically) given a good enough initial guess. Beginning with an initial guess $\mathbf{z}^{(0)}$, approximations to the solution are computed by iteratively updating the latest approximation via $\mathbf{z}^{(k+1)} = \mathbf{z}^{(k)} + \Delta \mathbf{z}_{NS}^{(k)}$ where $\Delta \mathbf{z}_{NS}^{(k)}$ is the Newton step which is found by solving the following linear system of l equations

$$\mathbf{H}_{\mathcal{L}}(\mathbf{z}^{(k)}) \Delta \mathbf{z}_{NS}^{(k)} = -\nabla \mathcal{L}(\mathbf{z}^{(k)}) \tag{2}$$

where $\mathbf{H}_{\mathcal{L}}$ is the Hessian of $\mathcal{L}$.

The computational burden of Newton's method in MPC applications is dependent on the solution of (2). Taking advantage of the symmetry of the Hessian, the cost of solving (2) is $O(\frac{1}{3} l^3)$ operations, in the worst case, using $\mathrm{LDL}^{\mathrm{T}}$ factorization [11]. Further speedup can be had if the Hessian is sparse, which is the case for some direct transcription methods, *e.g.* direct collocation. In the next section we demonstrate how one can use POD to reduce the cost of this step by reducing the dimension of the linear system to be solved.

IV. POD-Reduced MPC

We can reduce the computational cost of computing the Newton step if we restrict the Lagrangian $\mathcal{L}$ of the FHOCP to a reduced dimensional space. In particular, if we restrict $\mathcal{L}$ to an affine subspace containing its minimum $\mathbf{z}^*$, then a minimum of the restricted function when embedded in the full space will be exactly $\mathbf{z}^*$. Thus, given a good enough initial guess $\mathbf{z}^{(0)}$, Newton's method applied to $\mathcal{L}$ restricted to an affine subspace containing $\{\mathbf{z}^{(0)}, \mathbf{z}^*\}$ will converge to $\mathbf{z}^*$. For a real world application of MPC we can make use of this observation only in an approximate sense as the solutions $\mathbf{z}^*$ are not known.

A. Algorithm

In MPC we solve a sequence of optimization problems that are connected to one another through the history of the variables. Typically one problem is not too different from its neighbours which means we can take advantage of using the optimum at one timestep to determine the initial guess at the next timestep. Due to the interconnectedness of the sequence of optimization problems we might expect that there is some common direction shared in the progression of Newton steps across solutions. We can collect these directions in a snapshot matrix and use POD to find the r dimensional subspace that best captures these directions. Then at each timestep t_i we can approximate the affine subspace (which contains $\{\mathbf{z}^{(0)}(t_i), \mathbf{z}^*(t_i)\}$) and solve for the minimum of $\mathcal{L}$ restricted to that space to get an approximation of $\mathbf{z}^*(t_i)$. We expect that the quality of our approximated minimum is dependent on the choice of r and the snapshot.

The above provides us with an algorithm to apply POD to MPC:

1) Construct a snapshot matrix $\mathbf{Z} = \begin{pmatrix} \mathbf{z}^{(0)}(t_1) - \mathbf{z}^*(t_1) & \cdots & \mathbf{z}^{(0)}(t_N) - \mathbf{z}^*(t_N) \end{pmatrix} \in \mathbb{R}^{l\times N}$ by running the MPC over a representative control scenario with $N > l$ timesteps.
2) Compute the SVD of the snapshot matrix $\mathbf{Z} = \mathbf{U}\Sigma\mathbf{V}^T$ and then truncate $\mathbf{U}$ to its first $r < l$ columns to form $\mathbf{U}_r \in \mathbb{R}^{l\times r}$.
3) Construct the reduced order Lagrangian

$$\bar{\mathcal{L}}_{[\mathbf{z}']}(\mathbf{y}) = \mathcal{L}(\mathbf{U}_r \mathbf{y} + \mathbf{z}')$$

where $\mathbf{z}' \in \mathbb{R}^l$ and $\mathbf{y} \in \mathbb{R}^r$ are coordinates of the affine space $\langle \mathbf{U}_r \rangle + \mathbf{z}'$.

4) At timestep t_i within the MPC apply Newton's method to $\bar{\mathcal{L}}_{[\mathbf{z}^{(0)}]}$ to compute its minimum $\mathbf{y}^*$ which can be used to approximate the original minimum by $\mathbf{z}_r^* = \mathbf{U}_r\mathbf{y}^* + \mathbf{z}^{(0)} \approx \mathbf{z}^*$.

The computational savings of this method comes from using the reduced order Lagrangian to find an approximation of the minimum. Since

$$\nabla\bar{\mathcal{L}}_{[\mathbf{z}^{(0)}]}(\mathbf{y}) = \mathbf{U}_r^T\nabla\mathcal{L}(\mathbf{U}_r\mathbf{y} + \mathbf{z}^{(0)}),$$
$$\mathbf{H}_{\bar{\mathcal{L}}_{[\mathbf{z}^{(0)}]}}(\mathbf{y}) = \mathbf{U}_r^T\mathbf{H}_{\mathcal{L}}(\mathbf{U}_r\mathbf{y} + \mathbf{z}^{(0)})\mathbf{U}_r$$

the reduced Newton's step $\Delta\mathbf{y}_{NS}^{(k)}$ can be computed by solving the following system of r linear equations

$$\mathbf{U}_r^T\mathbf{H}_{\mathcal{L}}(\mathbf{U}_r\mathbf{y}^{(k)}+\mathbf{z}^{(0)})\mathbf{U}_r\Delta\mathbf{y}_{NS}^{(k)} = -\mathbf{U}_r^T\nabla\mathcal{L}(\mathbf{U}_r\mathbf{y}^{(k)}+\mathbf{z}^{(0)}) \tag{3}$$

or equivalently,

$$\mathbf{H}_{\mathcal{L}1}^{\mathbf{U}}(\mathbf{U}_r\mathbf{y}^{(k)} + \mathbf{z}^{(0)})\Delta\mathbf{y}_{NS}^{(k)} = -\nabla\mathcal{L}_1^{\mathbf{U}}(\mathbf{U}_r\mathbf{y}^{(k)} + \mathbf{z}^{(0)}) \tag{4}$$

where $\mathbf{U}^T\mathbf{H}_{\mathcal{L}}\mathbf{U} = \begin{pmatrix} \mathbf{H}_{\mathcal{L}1}^{\mathbf{U}} & \mathbf{H}_{\mathcal{L}2}^{\mathbf{U}} \\ \mathbf{H}_{\mathcal{L}3}^{\mathbf{U}} & \mathbf{H}_{\mathcal{L}4}^{\mathbf{U}} \end{pmatrix}$ with $\mathbf{H}_{\mathcal{L}1}^{\mathbf{U}} \in \mathbb{R}^{r\times r}$ and $\nabla\mathcal{L}_1^{\mathbf{U}} \in \mathbb{R}^r$ is a column vector made up of the first r entries of $\mathbf{U}^T\nabla\mathcal{L}$. We call an NMPC that uses (3) or (4) and the transformation $\mathbf{z}_r^* = \mathbf{U}_r\mathbf{y}^* + \mathbf{z}^{(0)}$ to compute its Newton steps a 'POD-reduced NMPC' or POD-rNMPC, for short. From (4) we can see that if $\mathbf{H}_{\mathcal{L}}$ is positive definite then so to is $\mathbf{H}_{\mathcal{L}1}^{\mathbf{U}}$ by applying Sylvester's criterion and noting that $\mathbf{U}$ is orthogonal. This means if the original Newton step $\Delta\mathbf{z}_{NS}^{(k)}$ can be computed so too can the reduced Newton step $\Delta\mathbf{y}_{NS}^{(k)}$. We also note that $\mathbf{H}_{\mathcal{L}1}^{\mathbf{U}}$ inherits the symmetry of $\mathbf{H}_{\mathcal{L}}$ but not its sparsity.

The updates of the reduced problem proceed as usual $\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} + \Delta\mathbf{y}_{NS}^{(k)}$ until convergence. In these reduced coordinates the initial guess is always $\mathbf{y}^{(0)} = 0$ since the initial guess $\mathbf{z}^{(0)}$ gets updated between timesteps. Since the reduced problem is nothing but an unconstrained optimization problem we expect the same rate of quadratic convergence to the reduced optimum $\mathbf{y}^*$.

There is an equivalence of the reduced problem to the following linearly constrained optimization problem

$$\begin{aligned} \min \quad & \mathcal{L}(\mathbf{z}) \\ \text{subject to} \quad & (\mathbf{U}_r^{\perp})^T\mathbf{z} = (\mathbf{U}_r^{\perp})^T\mathbf{z}^{(0)} \end{aligned}$$

where $\mathbf{U}_r^{\perp} \in \mathbb{R}^{l\times l-r}$ is the matrix made up of the last $l-r$ columns of $\mathbf{U}$. Essentially what we have done is add linear constraints to our original problem and then we eliminated those constraints by restricting the Lagrangian to a constraint submanifold. This approach turns out to be very similar to the idea for mechanical model reduction found in [12]. There they propose a model reduction method for mechanical systems by first imposing constraints to restrict the Lagrangian to a constraint submanifold and then deriving the equations of motion via the Euler-Lagrange equations. In this way the reduced dynamics will still satisfy many truly mechanical properties like preservation of symmetries and energy. As minima correspond to stationary orbits we are trying to accomplish the same task as in [12] but for a different purpose. Further, we have proposed an explicit algorithm to do so using POD.

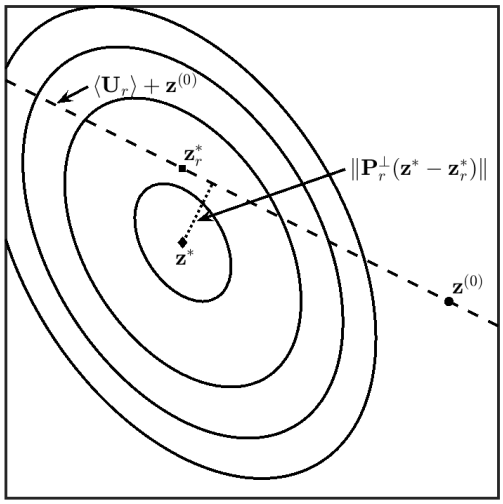

Fig. 1. A depiction of optimization (function represented by the contours) when restricted to an affine subspace. We can see that if $\|\mathbf{P}_r^{\perp}(\mathbf{z}^* - \mathbf{z}_r^*)\| = 0$ then $\mathbf{z}^* = \mathbf{z}_r^*$ and no error is incurred by approximating the original minimum with that of the restricted minimum.

It should be noted that the snapshot only captures the controller outputs and thus the algorithm should, in theory, be able to accommodate a wide variety of controller formulations.

B. Error Analysis

Applying POD in this manner yields good performance because the error in approximation of the minimum is related to the truncated POD modes. In Fig. 1 we present a depiction of the original and reduced optimization problems. In this subsection we analyse the error incurred by approximating the minimum using the reduced minimum for a single optimization problem. We assume that this exact optimization problem is captured in the snapshot matrix.

Recall that $\mathbf{U}_r$ is an optimal solution to

$$\begin{aligned} \min_{\mathbf{U}_r\in\mathbb{R}^{l\times r}} \quad & \sum_{i=1}^{N}\left\|\mathbf{P}_r^{\perp}(\mathbf{z}^*(t_i) - \mathbf{z}^{(0)}(t_i))\right\|^2 \\ \text{subject to} \quad & \mathbf{U}_r^T\mathbf{U}_r = \mathbf{I} \end{aligned}$$

for fixed $r < l$. Given the optimal $\mathbf{U}_r$ we have the relation

$$\sum_{i=1}^{N}\left\|\mathbf{P}_r^{\perp}(\mathbf{z}^*(t_i) - \mathbf{z}^{(0)}(t_i))\right\|^2 = \sum_{k=r+1}^{l}\sigma_k^2$$

where $\sigma_k \in \sigma(\mathbf{Z})$. Thus, at our particular timestep the error between the actual and approximate minimum is

$$\begin{aligned} \|\mathbf{z}^* - \mathbf{z}_r^*\|^2 &= \|\mathbf{P}_r(\mathbf{z}^* - \mathbf{z}_r^*)\|^2 + \|\mathbf{P}_r^{\perp}(\mathbf{z}^* - \mathbf{z}_r^*)\|^2 \\ &\leq \|\mathbf{P}_r(\mathbf{z}^* - \mathbf{z}_r^*)\|^2 + \sum_{k=r+1}^{l}\sigma_k^2. \end{aligned}$$

The quantity $\|\mathbf{P}_r(\mathbf{z}^* - \mathbf{z}_r^*)\|$ is dependent on $\mathcal{L}$ and the initial guess $\mathbf{z}^{(0)}$ for which no simple bound has been found. In the

following we provide a way to estimate this quantity using a quadratic approximation of $\mathcal{L}$ about $\mathbf{z}^*$.

In the case $\mathcal{L} = \frac{1}{2}\mathbf{z}^T\mathbf{Q}\mathbf{z}+\mathbf{c}^T\mathbf{z}+\mathbf{b}$ is a convex quadratic we can compute an exact bound for $\|\mathbf{P}_r(\mathbf{z}^* - \mathbf{z}_r^*)\|$. We know $\mathbf{z}^* = -\mathbf{Q}^{-1}\mathbf{c}$ and given an initial guess $\mathbf{z}^{(0)}$ then $\mathbf{z}_r^* = \mathbf{U}_r\mathbf{y}^* + \mathbf{z}^{(0)}$ where $\mathbf{y}^* = -[\mathbf{U}_r^T\mathbf{Q}\mathbf{U}_r]^{-1}\mathbf{U}_r^T(\mathbf{c} + \mathbf{Q}\mathbf{z}^{(0)})$. Thus

$$\|\mathbf{P}_r(\mathbf{z}^* - \mathbf{z}_r^*)\| = \|\mathbf{U}_r[\mathbf{U}_r^T\mathbf{Q}\mathbf{U}_r]^{-1}\mathbf{U}_r^T(\mathbf{c} + \mathbf{Q}\mathbf{z}^{(0)}) + \mathbf{P}_r(\mathbf{z}^* - \mathbf{z}^{(0)})\|$$

and with some manipulation we can find a bound

$$\begin{aligned}\|\mathbf{P}_r(\mathbf{z}^* - \mathbf{z}_r^*)\| &\leq \frac{\max\sigma(\mathbf{Q}_2^{\mathbf{U}})}{\min\sigma(\mathbf{Q}_1^{\mathbf{U}})}\|\mathbf{z}^* - \mathbf{z}^{(0)}\| \\ &\leq \kappa(\mathbf{Q})\|\mathbf{z}^* - \mathbf{z}^{(0)}\|\end{aligned}$$

where we have written $\mathbf{U}^T\mathbf{Q}\mathbf{U} = \begin{pmatrix}\mathbf{Q}_1^{\mathbf{U}} & \mathbf{Q}_2^{\mathbf{U}} \\ \mathbf{Q}_3^{\mathbf{U}} & \mathbf{Q}_4^{\mathbf{U}}\end{pmatrix}$ in block form with $\mathbf{Q}_1^{\mathbf{U}} \in \mathbb{R}^{r\times r}$ and $\kappa(\mathbf{Q}) = \frac{\max\sigma(\mathbf{Q})}{\min\sigma(\mathbf{Q})}$ is the condition number of $\mathbf{Q}$. The last inequality follows from the Cauchy interlacing theorem and Theorem 1 of [13]. This bound can be used to better estimate the error in the general case by replacing $\mathbf{Q}$ with $\mathbf{H}_{\mathcal{L}}(\mathbf{z}^*)$ to get the following approximate bound

$$\|\mathbf{z}^* - \mathbf{z}_r^*\|^2 \lesssim \kappa(\mathbf{H}_{\mathcal{L}}(\mathbf{z}^*))^2\|\mathbf{z}^* - \mathbf{z}^{(0)}\|^2 + \sum_{k=r+1}^{l}\sigma_k^2.$$

Thus we can say, with some confidence, that provided the Hessian is well-conditioned we can control the error of the approximation by selecting r large enough and making a good initial guess $\mathbf{z}^{(0)}$. It should be noted that since the POD-reduced NMPC provides different solutions at each timestep we expect that the initial guesses for increasing timesteps over the NMPC simulation will diverge and so the error may accumulate.

V. Numerical Study

A. Car Model and Test Scenario

The vehicle model used is a common single-track nonlinear car model with Pacejka tires. The model has 7 states and 3 inputs. A detailed description of the model including the parameters and equations of motion can be found in [14], [15]. The states of most interest to our investigation are the position (x, y), speed v and yaw angle ψ of the vehicle. The values x, y, v are taken at the center of gravity of the vehicle. The control inputs are the steering angle speed $w_\delta \in [-0.5, 0.5]$ (rad/s), the total braking force $F_B \in [0, 15000]$ (N) and the throttle position $\phi \in [0, 1]$. We denote the state and control vectors by $\mathbf{x} = [x, y, v, \beta, \psi, w_z, \delta]^T$ and $\mathbf{u} = [w_\delta, F_B, \phi]^T$, respectively, where the hitherto unmentioned states are side slip angle β, rate of change of yaw w_z and steering angle δ. We note that the equations of motion of this model are given by ordinary differential equations and so we can express the model by $\dot{\mathbf{x}} = \mathbf{c}(\mathbf{x}, \mathbf{u})$ where $\mathbf{c}$ is the nonlinear model. The only change made by the authors was to fix the gear input at $\mu = 2$ to yield a continuous model.

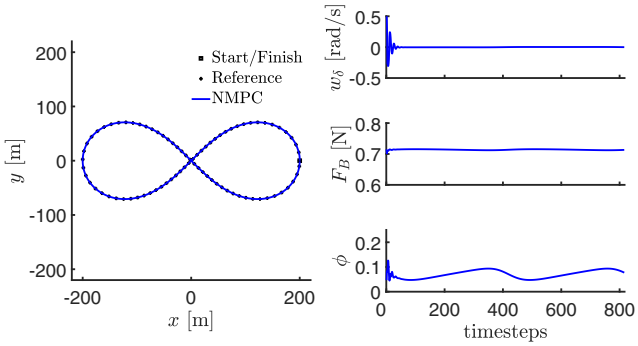

Fig. 2. The reference NMPC simulation from which the snapshot matrix was generated for the POD-rNMPC. On the left is the reference trajectory and controlled vehicle position and on the right is the obtained controls. Note: the reference trajectory excites slow dynamics leading to very smooth controls over the simulation apart from the initial acceleration from rest.

Control of the car using POD-rNMPC was tested using a double lane change maneuver with constant forward speed of 12 m/s as the reference trajectory. The simulated maneuver is set on a straight road 128 m in length with a lane offset of 3.2 m. The reference trajectory was generated using a piecewise linear interpolation of the maneuver waypoints.

To demonstrate the flexibility of the POD reduction approach, the snapshot matrix for the POD-rNMPC was generated using a different reference trajectory, see Fig. 2.

B. NMPC Formulation

The method of direct collocation was used to formulate the FHOCP. A horizon length of $H = 10$ was selected along with a constant timestep of $\Delta t = 100$ ms. We denote the states (controls) at timestep t_i over the prediction horizon by $\bar{\mathbf{x}}(t_i) = [\mathbf{x}(t_i), \mathbf{x}(t_{i+1}), \ldots, \mathbf{x}(t_{i+H-1})]$ ($\bar{\mathbf{u}}(t_i) = [\mathbf{u}(t_i), \mathbf{u}(t_{i+1}), \ldots, \mathbf{u}(t_{i+H-1})]$). The cost function was chosen to minimize controller effort and position tracking error. We included penalty functions in the cost to enforce the inequality constraints on the control inputs. The cost function is given by

$$\begin{aligned}J(\bar{\mathbf{x}}(t_i), \bar{\mathbf{u}}(t_i)) = &\sum_{k=0}^{H-1}\Delta\mathbf{x}(t_{i+k})^T\mathbf{Q}\Delta\mathbf{x}(t_{i+k}) \\ &+ \mathbf{u}(t_{i+k})^T\mathbf{R}\mathbf{u}(t_{i+k}) + \sum_j \rho_p(g_j(\mathbf{u}(t_{i+k})))\end{aligned}$$

where $\Delta\mathbf{x} = \mathbf{x} - \mathbf{x}_{ref}$ and $\mathbf{x}_{ref}$ is a reference trajectory, $\mathbf{Q} = \text{diag}(100, 100, 0, 0, 0, 0, 0)$, $\mathbf{R} = \text{diag}(10, 0.001, 10)$, $\mathbf{g}(\mathbf{u}) \leq 0$ is the vector of inequality constraints and the penalty function ρ_p is given by

$$\rho_p(z) = \begin{cases}0 & z \leq -1 \\ (z+1)^p & z > -1\end{cases}.$$

This penalty function was chosen to enforce constraints of the form $z \leq 0$ due to its behaviour in the limit $p \to \infty$. In our study we fixed $p = 8$.

The dynamics of the model are discretized using the explicit Euler method so that at each timestep t_i we minimize

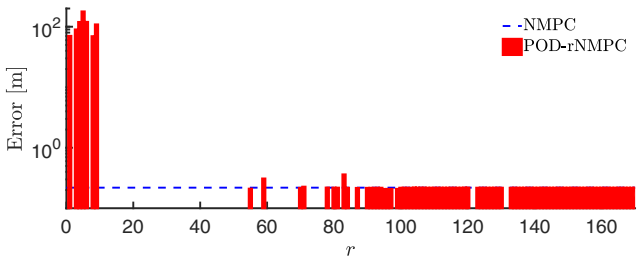

Fig. 3. The maximum position error of a POD-rNMPC over the double lane change maneuver with r variables. We provide the maximum error of the original NMPC as a reference. No red bar indicates that the simulation with that number of reduced variables failed to complete.

J subject to the following vector of equality constraints

$$\mathbf{h}(\bar{\mathbf{x}}(t_i), \bar{\mathbf{u}}(t_i)) = \begin{pmatrix} \mathbf{x}(t_i) - \mathbf{x}_0 \\ \mathbf{x}(t_{i+1}) - \mathbf{x}(t_i) - \mathbf{c}(\mathbf{x}(t_i), \mathbf{u}(t_i))\Delta t \\ \vdots \\ \mathbf{x}(t_{i+H-1}) - \mathbf{x}(t_{i+H-2}) - \mathbf{c}(\mathbf{x}(t_{i+H-2}), \mathbf{u}(t_{i+H-2}))\Delta t \end{pmatrix}$$

where $\mathbf{x}_0$ is the initial condition of the FHOCP which is derived from the solution of the previous timestep. The Lagrangian of the FHOCP is then $\mathcal{L}(\bar{\mathbf{x}}, \bar{\mathbf{u}}, \boldsymbol{\lambda}) = J(\bar{\mathbf{x}}, \bar{\mathbf{u}}) + \boldsymbol{\lambda}^T \mathbf{h}(\bar{\mathbf{x}}, \bar{\mathbf{u}})$ where $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers. The dimension of the full FHOCP is 170.

C. Implementation Details

All simulations were run using MATLAB 2016a on a desktop PC with an Intel i7-4790 CPU. Simulations were begun with a warm start computed using `fmincon` to find the initial values of $\bar{\mathbf{x}}$ and $\bar{\mathbf{u}}$. The initial Lagrange multipliers were then computed using least-squares [16]. The gradient $\nabla\mathcal{L}$ and Hessian $\mathbf{H}_{\mathcal{L}}$ were computed symbolically using MAPLE-generated optimized code which was then converted to MEX files. Our optimization method employed full Newton steps with only a single iteration per timestep. All matrices were declared `sparse` where appropriate to take advantage of MATLAB's sparse solvers and timing data was found using `tic` and `toc`. In our implementation of POD-rNMPC the full gradient and Hessian are computed and then transformed within each timestep following (3).

D. Results

By applying the snapshot matrix generated from the figure-eight trajectory to the double lane change maneuver we found that the optimal number of reduced variables was $r = 55$. In Fig. 3 we can see the effect that choice of r has on the quality of the POD-rNMPC's performance. We note that for many choices of r the POD-rNMPC failed to operate over the entirety of the simulation due to the transformed Hessian becoming ill-conditioned at some timestep. In the subsequent discussion we focus on the results of the POD-rNMPC with $r = 55$.

The POD-rNMPC resulted in turnaround times on average 2 times faster than the original NMPC. In Table I we present

TABLE I
MEAN COMPUTATION TIME OF 100 SIMULATIONS

	Linear System Solution Time	Turnaround Time
NMPC	0.48 ms	0.59 ms
POD-rNMPC (r=55)	0.083 ms	0.30 ms
POD-rNMPC (r=80)	0.14 ms	0.39 ms
POD-rNMPC (r=115)	0.25 ms	0.57 ms

measured computation times averaged over all timesteps over 100 simulations. Despite the fact that the reduced linear system is dense and the original system sparse, the dimensional reduction resulted in a 5.8 times faster solution of the linear system by going from 170 to 55 equations.

As can be seen in Figs. 4 and 5 the performance of the POD-rNMPC is remarkably similar to the original NMPC. Surprisingly, in terms of maximal position error and lateral position error the POD-rNMPC marginally improved the performance of the controller reducing the error from 22 cm to 21 cm and from 6.8% to 6.6% of the lane offset, respectively. Greatest error was seen in the speed where the POD-rNMPC deviated by at most 3.0% from the reference values compared to the NMPC which deviated at most 1.8%. It is expected this could be improved by including a speed tracking term in the cost function. In Fig. 6 we can see that the control inputs of the POD-rNMPC are of the same magnitude as the controls of the original NMPC. We do notice that the controls obtained by the POD-rNMPC exhibit greater rates of change than the NMPC controls. This is a direct result of the dimensional reduction. On average the POD-rNMPC and NMPC inputs for steering angle speed, braking force and throttle position differed by $4.6\%, 4.3\times10^{-7}\%$ and 3.0% of the input range, respectively.

VI. CONCLUSIONS AND FUTURE WORK

A. Conclusions

We introduced a novel application of POD to NMPC. Instead of using POD to reduce the order of the plant model, we instead applied POD to the Lagrangian of the optimization problem within NMPC. This allows for a greater reduction of the dimension of the optimization problem leading to faster controller turnaround times. In our study of the control of a nonlinear car model during a double

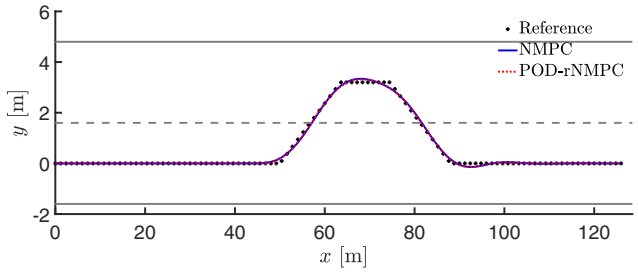

Fig. 4. Vehicle position during the controlled double lane change maneuver. Note that the axes are not scaled equally. The waypoints defining the maneuver are: $\{(0,0),(0,50),(3.2,63.5),(3.2,74.5),(0,88),(0,128)\}$.

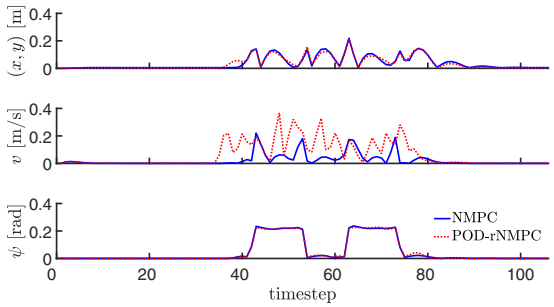

Fig. 5. The absolute error of position, speed and yaw relative to the reference trajectory over the double lane change maneuver.

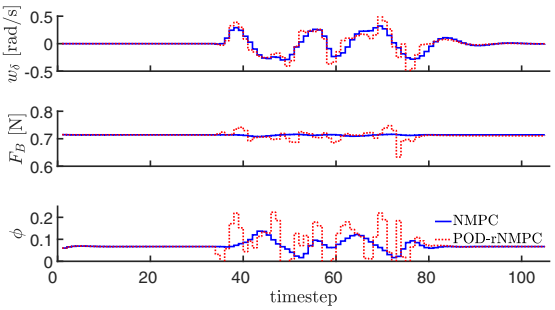

Fig. 6. The obtained control inputs over the double lane change maneuver.

lane change maneuver we halved the turnaround time of an NMPC while introducing minimal degradation in controller tracking performance.

B. Future Work

The introduction of this new approach brings up many questions to be answered. Some are fundamental, such as, does POD-reduced MPC preserve stability or controllability of an MPC? Others are more numerics-oriented, like, how does the integration of POD-reduced Newton steps affect the behaviour of more complex nonlinear programming methods like interior-point or sequential quadratic programming? Beyond these important questions our results bring up a few immediate points to be investigated. How sensitive is POD-rNMPC to the choice of snapshot? Can we get past the ill-conditioning of some reduced Hessians to achieve even greater dimensional reduction? What are the exact computational savings of POD-rNMPC in an HIL simulation? In future work we will address these questions along with new applications.

References

[1] H. Chen, D. L. Reuss, D. L. Hung, and V. Sick, "A practical guide for using proper orthogonal decomposition in engine research," *International Journal of Engine Research*, vol. 14, no. 4, pp. 307–319, 2013.

[2] J. A. Atwell and B. B. King, "Reduced order controllers for spatially distributed systems via proper orthogonal decomposition," *SIAM Journal on Scientific Computing*, vol. 26, no. 1, pp. 128–151, 2004.

[3] P. Prabhat, S. Balakrishnan, D. C. Look, and R. Padhi, "Proper orthogonal decomposition based modeling and experimental implementation of a neurocontroller for a heat diffusion system," in *American Control Conference, 2003. Proceedings of the 2003*, IEEE, vol. 3, 2003, pp. 2652–2657.

[4] C. W. Rowley, "Model reduction for fluids, using balanced proper orthogonal decomposition," *International Journal of Bifurcation and Chaos*, vol. 15, no. 03, pp. 997–1013, 2005.

[5] A. Marquez, J. J. Oviedo, D. Odloak, *et al.*, "Model reduction using proper orthogonal decomposition and predictive control of distributed reactor system," *Journal of Control Science and Engineering*, vol. 2013, p. 3, 2013.

[6] R. Masoudi, T. Uchida, and J. McPhee, "Reduction of multibody dynamic models in automotive systems using the proper orthogonal decomposition," *Journal of Computational and Nonlinear Dynamics*, vol. 10, no. 3, p. 031 007, 2015.

[7] A. Alla and S. Volkwein, "Asymptotic stability of pod based model predictive control for a semilinear parabolic pde," *Advances in Computational Mathematics*, vol. 41, no. 5, pp. 1073–1102, 2015.

[8] S Volkwein, "Proper orthogonal decomposition: applications in optimization and control," *CEA-EDFINRIA Numerical Analysis Summer School*, 2007.

[9] K. Kunisch and S. Volkwein, "Optimal snapshot location for computing pod basis functions," *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 44, no. 3, pp. 509–529, 2010.

[10] A. V. Rao, "A survey of numerical methods for optimal control," *Advances in the Astronautical Sciences*, vol. 135, no. 1, pp. 497–528, 2009.

[11] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[12] S. Lall, P. Krysl, and J. E. Marsden, "Structure-preserving model reduction for mechanical systems," *Physica D: Nonlinear Phenomena*, vol. 184, no. 1, pp. 304–318, 2003.

[13] R. C. Thompson, "Principal submatrices ix: interlacing inequalities for singular values of submatrices," *Linear Algebra and its Applications*, vol. 5, no. 1, pp. 1–12, 1972.

[14] M. Gerdts, "Solving mixed-integer optimal control problems by branch&bound: a case study from automobile test-driving with gear shift," *Optimal Control Applications and Methods*, vol. 26, no. 1, pp. 1–18, 2005.

[15] ——, *Optimal control of ODEs and DAEs*. Walter de Gruyter, 2012.

[16] R. H. Byrd, M. E. Hribar, and J. Nocedal, "An interior point algorithm for large-scale nonlinear programming," *SIAM Journal on Optimization*, vol. 9, no. 4, pp. 877–900, 1999.