Computationally Efficient Hierarchical Model Predictive Control Via Koopman Operator

Stephen Boyle
Department of Mechanical Engineering,
Naval Surface Warfare Center Philadelphia Division,
Philadelphia, PA 19112

Stephanie Stockar[1]
Department of Mechanical and Aerospace Engineering,
The Ohio State University,
Columbus, OH 43212
e-mail: stockar.1@osu.edu

Combined powertrain and velocity optimization can achieve significant energy efficiency improvements. However, due to the multitime scales in the system, the optimization is performed hierarchically and by separating time scales. To enforce state constraints, iteration between controller is introduced, for example, using Lagrange multipliers as metric for constraint violation. In this paper, an extension of the Koopman operator theory is presented with to obtain a data-driven approximation of the multipliers' behavior hence eliminating the need for iterations. Because the evolution of the Lagrange multipliers is the result of a fast dynamics optimization problem, and not the response of a nonlinear dynamical system, a novel technique in which the Lagrange multipliers are interpreted as a dynamic system is presented here. The approximate Koopman linear system is then derived using extended dynamic mode decomposition and it is integrated with the slow dynamic optimization. Results show that the Koopman augmented controller, which is solved as one single optimization, meets state and input constraints and achieves similar energy savings compared to an iterative approach. [DOI: 10.1115/1.4056703]

1 Introduction

Class-8 trucks are responsible for approximately 20% of the total highway fuel consumption [1]. Hence, improvement in energy efficiency of commercial vehicle is crucial for addressing the sustainability challenges of the transportation sector. The repetitive nature of trucking routes lends itself to automation, potentially allowing semitrucks to take advantage of the recent developments in connected and autonomous vehicle (CAV) technology. Specifically, control strategies have been developed that utilize GPS, traffic, and road topography information available to CAVs to save energy, including vehicle speed optimization [2–5], eco-routing [6], platooning [7,8], gear selection optimization [9,10], optimal traffic signal arrival and departure [11,12], and engine and cabin thermal management [13,14]. When integrated, these strategies have been shown to lead to an average of 20% energy usage reduction on a variety of vehicle platforms [4,5].

One of the challenges of developing strategies that combine both, velocity optimization and powertrain control, is that the plant model is characterized by multiple time-scales. Singularly perturbed problems can be solved efficiently by partitioning the plant based on time-scale and solving the coupled subproblems independently, where, for a two-time-scales systems, the slower dynamics handled by the higher layer controller, and the faster dynamics handled by the lower layer controller. Hierarchical controllers can be categorized by the level of communication between layers, either no communication, one-way communication, and two-way communication. When coupling between the two time-scales is weak, a hierarchical controller with no communication between the layers can be used [15,16]. In most scenarios, coupling between time-scales is stronger, and requires some communication between the controllers. Most hierarchical controllers pass information in one direction from the higher to the lower layer where the higher layer controller optimizes a cost function over a longer horizon, while neglecting the fast dynamics, such as actuator dynamics. The output from the higher layer controller acts as a reference that the lower layer controller attempts to track. Other applications solve an optimization problem in both layers where the computation time is reduced by, for example, using different temporal discretizations in each layers [17] or using different fidelity models in each layers [18].

A challenge arises when the lower layer is unable to track the profile computed by the higher layer, which can often occur, for example, in heavy duty powertrain control where the engine is not capable to provide enough change in torque to meet the required acceleration. One approach consists in tightening the constraints on the higher layer such that the lower layer tracking problem remains feasible [19]. Other approaches that additional communication and iteration are based on treating the deviation of the lower layer from the desired setpoint as a disturbance to be rejected [20], on recomputing the solution of the higher level if the tracking is unfeasible [21], or on allowing deviation from the reference provided that some metrics on the slow-dynamics problem remain unchanged [22]. None of these approaches are systematic in the way the infeasibility is addressed, which will lead to a large optimality gap. Few examples exist where tracking failure is addressed by feedback from the lower layer. In Ref. [23], the higher layer provides a reference for the lower layer, and a constraint on the deviation from the reference. If the lower layer is unable to meet the relaxed tracking constraint, a binary variable is fed back to the higher layer, which is used to lower the performance and to trigger a recomputation of the profile. A similar approach is presented in [24], where if the lower layer violates this constraint, the penalty terms in the higher layer objective function are modified to favor a smoother profile, rather than an economic one. While simulation results are promising, the weight tuning process is entirely empirical and requires extensive calibration. In this context, a novel iterative hierarchical scheme for multitime scale systems has been presented by the authors in Ref. [25]. The strategy has the ability to guarantee constraint satisfaction across time-scales, while obtaining a near-optimal solution comparable to the centralized solution at a significantly reduced computation time. This is achieved by using the Lagrange multipliers of the fast dynamic as measure for constraints satisfaction in the slow dynamic optimization. Specifically, by using the properties of Lagrange multipliers that are nonzero if and only if the corresponding constraints are active, and their magnitude is a measure of the distance to the constraint. The main limitation of this approach is the need for an iterative process with unknown convergence rate, which in general will result in increased computational burden. This paper presents an extension of the iterative

[1]Corresponding author.

Contributed by the Dynamic Systems Division of ASME for publication in the Journal of Dynamic Systems, Measurement, and Control. Manuscript received July 12, 2022; final manuscript received January 12, 2023; published online February 8, 2023. Assoc. Editor: Mahdi Shahbahkti.

model predictive control (MPC) scheme presented in Ref. [25] based on the hypothesis that the fast dynamics Lagrange multipliers are described by an unknown nonlinear dynamic system that can be identified using Koopman operator theory. The resulting linear system is then integrated in the higher layer controller to predict the fast dynamics Lagrange multipliers without iteration. A simulation study is shown in this paper for the combined velocity and powertrain optimization in a heavy duty vehicle. The simulation results compare the iterative solution against the sequential method that uses the Koopman system. Results show that the Koopman-augmented MPC achieves computationally efficient near-optimal results. This paper is structured as follows. First, the optimal control problem is defined for the case of a general two-time scales system with coupled state constraints. Then an overview of Koopman theory is introduced and its application to the identification of the Lagrange is provided. Finally, the case study of the connected vehicle is introduced and the new method is compared against a traditional iterative approach.

2 Control Problem Definition

2.1 Summary of the Plant Model Equations. The vehicle considered in this paper is a Volvo VNL 300 class 8 semitruck with a 13 liter six cylinder turbocharged Diesel engine and a twelve speed automated manual transmission.

The relevant dynamics of the plant are the vehicle longitudinal dynamic, the air and the fuel dynamics, where the first is considerably slower than the other two. The engine transient response is modeled under the assumption that the air-handling and fuel dynamics, which determine the instantaneous fuel flowrate in transient conditions, are represented by first order systems [25,26]

$$\tau_i \frac{d\dot{m}_i(t)}{dt} + \dot{m}_i = \dot{m}_{i,\text{ss}} \tag{1}$$

where the subscript i is indicates either air or fuel. The time constants τ_i are function of the change in torque request ΔT_c and are calibrated to meet the transient responses collected on the engine dynamometer and lump the effect of air and fuel path components. The steady-state fuel flowrate $\dot{m}_{f,\text{ss}}$ is mapped as function of desired engine torque and engine speed, while the steady-state air mass flowrate is given by the speed density equation

$$\dot{m}_{\text{air,ss}} = \lambda_v \cdot \frac{N}{120} \cdot \frac{pV_d}{RT_{\text{im}}} \tag{2}$$

where V_d is engine displacement, N is the engine speed, R is the specific gas constant for air, p is the intake manifold pressure, T_{im} is the intake manifold temperature, which are mapped based on engine speed and torque request, and λ_v is volumetric efficiency [27]

$$\lambda_v = k_1(N) - \frac{k_2(N)T_{\text{im}}}{p} \tag{3}$$

where the coefficient k_1 and k_2 are function of engine speed [25]. In Diesel Engines, to prevent smoke and limit emission a saturation on the fuel flowrate that is implemented

$$\dot{m}_f = \begin{cases} \text{FAR}_{\text{lim}} \cdot \dot{m}_{\text{air}} & \text{if } \dfrac{\dot{m}_{\text{fuel}}}{\dot{m}_{\text{air}}} > \text{FAR}_{\text{lim}} \\ \text{fuel} & \text{otherwise} \end{cases} \tag{4}$$

where FAR_{lim} represents the maximum FAR and, in this paper, a constant value of 5.9×10^{-2} has been used [25,28]. The engine torque is calculated as

$$T_{\text{eng}} = \frac{V_d}{4\pi} \cdot (\text{IMEP} - \text{FMEP}) \tag{5}$$

where the indicated mean effective pressure (IMEP) and friction mean effective pressure (FMEP) are mapped as functions of fuel flow and engine speed. The wheel force F_t is calculated assuming that all shafts are rigid, the gear efficiency is constant and gear changes are instantaneous and the wheel speed is an algebraic function of the vehicle speed v. Finally, the road load equation is used to determine vehicle speed

$$m_{\text{eff}} \frac{dv(t)}{dt} = F_t(t) - \frac{1}{2}\rho v^2 C_d A_f + \\ -m \cdot g \cdot c_r \cdot \cos(\theta) - m \cdot g \cdot \sin(\theta) \tag{6}$$

where m_{eff} is the vehicle effective mass, g is the gravitational acceleration, θ is the road grade, c_r is the rolling resistance coefficient, ρ is the density of air, C_d is the drag coefficient, and A_f is the frontal area. A complete validation of the model against engine dynamometer and road data is shown in Ref. [25].

The equations governing the system are then written in the following discrete nonlinear state-space form

$$x_{k+1} = x_k + f(x_k, u_k) \tag{7}$$

where the states x and control input u are defined as

$$x_k = [\, d_k \quad v_k \quad \dot{m}_{f,k} \quad \dot{m}_{\text{air},k} \quad T_{c,k} \,]^T \tag{8}$$

$$u_k = \Delta T_{c,k} \tag{9}$$

where d is the distance, v is the vehicle speed, $\dot{m}_f$ and $\dot{m}_{\text{air}}$ are the fuel and air flowrate, respectively, T_c is the engine torque request, ΔT_c is the change in the control command and the subscript k is the time step. For the control design, the system is augmented to account for traveled distance and rate of change in the control input for drivability [25,29,30].

2.2 Centralized Problem. The objective of the problem is to find the optimal engine torque sequence to minimize the fuel required to travel a certain distance d_{ref} over a given time imposed by a supervisory controller. The optimal control sequence is obtained by solving the following finite horizon optimization problem

$$\min_{u_k} J = \sum_{k=0}^{N_p-1} (\gamma_1 \dot{m}_{f,k} + \gamma_2 u_k^2)\Delta t + \gamma_3 (d_{N_p} - d_{\text{ref}})^2 \tag{10}$$

subject to

$$x_{k+1} = x_k + f(x_k, u_k) \; \forall k = 1, \ldots, N_p - 1 \tag{11}$$

$$g(x_k, u_k) \leq 0 \; \forall k = 1, \ldots, N_p - 1 \tag{12}$$

where N_p is the length of the horizon, Δt is the discretization time, γ_1, γ_2, and γ_3 are weights penalizing fuel flowrate, change in torque command, and the distance traveled at the final time, respectively. The weights are selected though a calibration process to achieve a desired tradeoff, for example, optimal fuel economy with no changes in travel time [5]. The function g includes the state and input constraints, such as fuel-to-air constraint that limits the change in wheel torque due to availability of fuel and air as defined in Eq. (4). The dynamics in the centralized problems have two different time-scales. Namely, a slow dynamics associated with the vehicle longitudinal dynamic, and a fast dynamics associated with the engine transients.

2.3 Iterative Solution Via Communication of the Lagrange Multipliers. To address numerical issues and computation time, MPC design for systems characterized by multiple time-scales is often performed hierarchically, where the slow and fast dynamics

are considered as separate optimization problems with coupled dynamics [23,31]. For the problem considered in this paper

$$x^s_{k+1} = x^s_k + f_1(x^s_k, x^f_k, u^s_k) \tag{13}$$

$$x^f_{k+1} = x^f_k + f_2(x^s_k, x^f_k, u^f_k) \tag{14}$$

where the subscript s and f refer to the slow and fast dynamics, respectively. The states and control inputs for each time-scales are

$$x^s_k = [\, d_k \quad v_k \quad \dot{m}^{\text{ref}}_{f,k} \,]^T \tag{15}$$

$$x^f_k = [\, \dot{m}_{f,k} \quad \dot{m}_{\text{air},k} \quad T_{c,k} \,]^T \tag{16}$$

$$u^s_k = \Delta \dot{m}^{\text{ref}}_{f,k} \tag{17}$$

$$u^f_k = \Delta T_{c,k} \tag{18}$$

While each subsystem has a different control input, the two dynamics are coupled through the states. The objective of the slow dynamics optimization problem is then to find the optimal trajectory for the fuel flowrate, subject to the slow dynamics constraints

$$\min_{u^s_k} J = \sum_{k=0}^{N_s-1} [\gamma^s_1 \dot{m}^{\text{ref}}_{f,k} + \gamma^s_2 (u^s_k)^2] \Delta t^s + \gamma^s_3 (d_{N_s} - d_{\text{ref}})^2 \tag{19}$$

subject to

$$x^s_{k+1} = x^s_k + f_1(x^s_k, \bar{x}^f_k, u^s_k) \; \forall k = 1, \ldots, N_s - 1 \tag{20}$$

$$g^s(x^s_k, u^s_k) \leq 0 \; \forall k = 1, \ldots, N_s - 1 \tag{21}$$

where γ^s_1, γ^s_2, and γ^s_3 are the weights and N_s is the horizon for the slow dynamics optimization. The function g^s includes the state and input constraints and Δt^s is the discretization time used for the slow dynamic. Finally, $\bar{x}^f$ is defined as

$$\bar{x}^f_k = [\, \dot{m}_{f,\text{ss},k} \quad \dot{m}_{\text{air,ss},k} \,]^T \tag{22}$$

where the subscript ss indicates that the value of the fast states $\dot{m}_f$ and $\dot{m}_{\text{air}}$ are obtained under the assumption that the changes are instantaneous. The fuel flowrate is determined from the static fueling map, while the air flowrate is determined using the speed density equation [25]. Similarly, the fast dynamics optimization problem has the objective of tracking the reference fuel flow obtained by solving the slow dynamic problem, while satisfying constraints on air and fuel flow dynamics

$$\min_{u^f_k} J = \sum_{k=0}^{N_f-1} [\gamma^f_1 (\dot{m}_{f,k} - \dot{m}^{\text{ref}}_{f,k})^2 + \gamma^f_2 (u^f_k)^2] \Delta t \tag{23}$$

subject to

$$x^f_{k+1} = x^f_k + f_2(\bar{x}^s_k, x^f_k, u^f_k) \; \forall k = 1, \ldots, N_f - 1 \tag{24}$$

$$g^f(x^f_k, u^f_k) \leq 0 \; \forall k = 1, \ldots, N_f - 1 \tag{25}$$

where N_f is the length of horizon for the fast dynamic problem and $\bar{x}^s$ are the nominal trajectories of the slow states obtained from the slow dynamic optimization problem and are included as external inputs. The function g^f includes the state, input, and the FAR constraints. While it is clear that solving the problem sequentially will improve computation time and possibly lead to a real-time executable algorithm [23], this approach cannot guarantee that the optimal trajectories for $\dot{m}^{\text{ref}}_{f,k}$ are feasible due to the constraints on the fast dynamics [29]. Specific to the heavy duty control problem, an increased torque demand is met by increasing the amount of fuel to the engine. However, as shown in Eq. (4), this is limited by the amount of fresh air that can be induced in the cylinder and air and fuel are characterized by a different dynamic response leading to constraint violation. To address this issue, a novel iterative approach based on the Lagrange multipliers was presented in Ref. [25] and verified on selected maneuvers that require rapid changes in torque production. The optimal Lagrange multipliers are defined in vector form as

$$\lambda^* = \frac{\nabla J(u^*)}{\nabla G(u^*)} \tag{26}$$

where u^* is the optimal control sequence, G is the function of the constraints, and J is the objective function. The Lagrange multipliers λ represent the change of the objective function due to a perturbation of the constraints [32,33]. This concept is leveraged in the iterative method developed in Ref. [25] to find the minimum value of γ^s_2, such that $\sum \lambda^*$ is as small as possible. This corresponds to a solution of the slow time-scale optimization problem that is feasible for the fast dynamics, and associated with the smallest penalty on the actuation.

For the calculation of Lagrange multipliers, the two optimization problems are converted into standard quadratic programing problems through linearization. Augmenting the state vector to include the original states and tracking errors in the standard QP leads to the following:

$$\min_{u^s_k} J_s = \sum_{k=0}^{N_s-1} [||\bar{x}^s_k||^2_{\bar{Q}_s} + ||u^s_k||^2_{R_s}] + ||\bar{x}^s_{N_s}||^2_{\bar{P}_s} \tag{27}$$

subject to

$$\bar{x}^s_{k+1} = \bar{A}^s_k \bar{x}^s_k + \bar{B}^s u^s_k, \quad \forall k = 1, \ldots, N_s - 1 \tag{28}$$

The same is done for the fast dynamic

$$\min_{u^f_k} J_f = \sum_{k=0}^{N_f-1} [||\bar{x}^f_k||^2_{\bar{Q}_f} + ||u^s_k||^2_{R_f}] + ||\bar{x}^s_{N_f}||^2_{\bar{P}_f} \tag{29}$$

subject to

$$\bar{x}^f_{k+1} = \bar{A}^f_k \bar{x}^f_k + \bar{B}^f u^f_k, \quad \forall k = 1, \ldots, N_s - 1 \tag{30}$$

where $\bar{x}^f$ is the augmented state vector with the original states and tracking terms. Finally, the fast dynamic optimization is subject to the following inequality constraint

$$a^f_{\text{in}} x^f_k + b^f_{\text{in}} u^f_k - c^f_{\text{in}} \leq 0, \quad \forall k = 1, \ldots, N_f - 1 \tag{31}$$

which include state, input, and FAR constraints. The iterative approximation method for estimating the behavior of the optimal Lagrange multiplier λ^* as function of R_s presented is summarized in Algorithm 1 [25].

Algorithm 1 Iterative Selection of R_s

1: $q = 0$
2: **while** $\sum \lambda > \lambda_{max}$ &$q < q_{max}$ **do**
3: **if** $q < 2$ **then**
4: R_s is user defined
5: **else if** $q < q_{\text{exp}}$ **then**
6: Choose R_s s.t. $\sum \bar{\lambda}_l(R_s) = 0$, using a linear fit for $\sum \bar{\lambda}_l$
7: **else**
8: Choose R_s s.t. $\sum \bar{\lambda}_e(R_s) \leq \lambda_{max}$, using an exponential fit for $\sum \bar{\lambda}_e$
9: **end if**
10: $q++$
11: **end while**

While this approach has been shown to satisfy inequality constraints in multitime-scale optimal control problems without penalizing system performance with near real-time performance, the iteration required to approximate the Lagrange multipliers increases computation requirements compared to a sequential scheme [25]. One possible approach to eliminate the need for iteration is to perform a design of experiment (DOE), where all the possible combinations of output reference and operating conditions are explored and a static map for γ_2^s is stored. While conceptually simple, this calibration-based approach relies on a precomputed map, which might not perform well for combination of points that were not part of the original DOE. In addition, the evaluation of the DOE requires an extensive simulation campaign. An alternative approach to the mapping of γ_2^s consists in exploring data driven methods. In this paper, the Koopman operator theory is extended to the problem of predicting Lagrange multipliers.

3 Overview of Koopman Operator Theory and Dynamic Mode Decomposition

Finding a functional approximation for λ^* will allow to combine the benefit of an iterative scheme with the computational efficiency of a sequential approach. The simplest possible data-driven approach consists of training of a neural network [34,35]. However, using a neural network as the prediction function introduces nonlinearity and nonconvexity to the slow dynamics optimization problem, requiring more complicated and time-consuming algorithms to solve, such as dynamic programing [36]. Koopman operator theory offers a promising pathways for real-time implementation as it consists of a methodology to describe nonlinear dynamics using a higher dimensional linear model [37–41]. While this approach will increase the dimension of the state vector in the slow dynamic optimization problem, it retains the model convexity and hence allows for adopting computationally efficient quadratic programing solvers.

3.1 Koopman Operator Background. Koopman operator theory is based on the concept that finite dimensional, nonlinear differential equations can be represented as an infinite dimensional linear system [37]. Consider a nonlinear, differential equation given by

$$\dot{x} = f(x) \tag{32}$$

where $x \in \mathbb{R}^n$ is the state vector and the function $f : \mathbb{R}^n \to \mathbb{R}^n$ determines the time derivative of the state. Let us also define real-valued measurement functions $g : \mathbb{R}^n \to \mathbb{R}$, which are elements of an infinite-dimensional Hilbert space. The functions g are also known as observable functions. The Koopman operator $\mathcal{K}$ is an infinite dimensional linear operator that acts on the measurement function g such that

$$\dot{g}(x) = \mathcal{K}g(x) = g(f(x)) \tag{33}$$

This operation allows a nonlinear system to be represented by a Koopman linear system on the Hilbert space of observable functions. Despite being a linear operator, the Koopman operator can be complex and difficult to use due to its infinite dimensional nature. It is possible to obtain a finite-dimensional matrix representation of the Koopman operator by restricting it to an invariant subspace spanned by a finite number of observables [37–41]. A Koopman invariant subspace is defined as the span of a set of functions $\begin{bmatrix} g_1 & g_2 & \dots & g_p \end{bmatrix}$ if all functions in the subspace remain in the subspace after being acted on by the Koopman operator $\mathcal{K}$. Any finite set of eigenfunctions will span an invariant subspace, and will satisfy the condition given by

$$\mathcal{K}\varphi(x) = \lambda\varphi(x) \tag{34}$$

where $\varphi(x)$ is the eigenfunction and λ is the corresponding eigenvalue. The eigenfunctions can be used to construct the set of observable functions g, given by

$$g(x) = \begin{bmatrix} g_1(x) \\ g_2(x) \\ \vdots \\ g_p(x) \end{bmatrix} = \sum_{j=1}^{\infty} \varphi_j(x) v_j \tag{35}$$

where v_j is the Koopman mode given by

$$v_j = \begin{bmatrix} \langle \varphi_j, g_1 \rangle \\ \langle \varphi_j, g_2 \rangle \\ \vdots \\ \langle \varphi_j, g_p \rangle \end{bmatrix} \tag{36}$$

While the Koopman operator is a powerful mathematical tool, its original development was for autonomous systems. Within the last decade, there has been increased interest in using Koopman operator theory to develop control oriented linear models of nonlinear systems as it allows linear control techniques to be used for nonlinear systems [42–45]. This, however, requires an extension of the classical Koopman operator theory to include system inputs. Consider a nonlinear system given by

$$\dot{x} = f(x, u) \tag{37}$$

This system can be rewritten as an infinite dimensional Koopman linear system given by

$$\dot{g} = Ag + Bu \tag{38}$$

where A and B are the infinite dimensional Koopman linear system matrices and u is the original control input.

Whether the system is autonomous or not, in many scenarios no Koopman invariant subspace exists and an exact finite dimensional Koopman linear system cannot be found. Furthermore, for the application of Lagrange multiplier prediction, the governing nonlinear equations are unknown, making it impossible to postulate an analytical expression for the eigenfunctions. In this scenario, a data driven approach is the only viable solution to determine the Koopman eigenfunctions.

3.2 Data Driven Approximation of the Koopman Operator. Extended dynamic mode decomposition (EDMD) is used to approximate the leading eigenfunctions of a system using a dataset of snapshot pairs and a dictionary of candidate observable functions, without knowledge of explicit governing equations [46]. Because system measurements are sampled at discrete time-steps, the Koopman linear system is also transformed into discrete time form, given by

$$g_{k+1} = \tilde{A}g_k + \tilde{B}u_k \tag{39}$$

The sampled measurements are compiled in the snapshot matrices given by

$$X = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix} \tag{40}$$

$$X^+ = \begin{bmatrix} x_2 & x_3 & \dots & x_{n+1} \end{bmatrix} \tag{41}$$

$$U = \begin{bmatrix} u_1 & u_2 & \dots & u_n \end{bmatrix} \tag{42}$$

where $n+1$ is the total number of samples of x collected. The states in X and inputs U at each index represent the conditions at some instant k. The Koopman operator is calibrated such that the discrete Koopman linear system predicts the value in X^+ at the same index, corresponding to the time instant $k+1$. The candidate

observable functions are then applied to these snapshot states to find the snapshot observable functions, given by

$$G = \begin{bmatrix} g_1 & g_2 & \dots & g_n \end{bmatrix} \tag{43}$$

$$G^+ = \begin{bmatrix} g_2 & g_3 & \dots & g_{n+1} \end{bmatrix} \tag{44}$$

The snapshot data is then used to find the $\tilde{A}$ and $\tilde{B}$ matrices of the Koopman linear system by solving the least-squares regression problem given by

$$\arg\min_{\tilde{A},\tilde{B}} ||G^+ - \tilde{A}G - \tilde{B}U||_F \tag{45}$$

where $||\cdot||_F$ is the Frobenius norm.

4 Application to the Connected and Autonomous Vehicle Problem

While the popularity of Koopman operator theory for control has grown recently, its application has been restricted to identify linear models of nonlinear dynamical systems [38–44,46,47]. However, for the CAV problem presented in this paper and the issue associated with the identification of the Lagrange multipliers, a further extension of the method is needed. In this case, the evolution of the Lagrange multipliers is a result of the fast dynamics optimization problem, and not the response of a nonlinear dynamical system. Therefore, an novel technique in which the Lagrange multipliers are interpreted as a dynamic system is presented in this paper, and an approximate Koopman linear system is derived using EDMD.

4.1 Lagrange Multipliers as a Dynamic System. The fast dynamics Lagrange multipliers are determined by the optimization problem presented in Eq. (29), where the FAR constraint is applied at each time-step k. Therefore, there is an associated Lagrange multiplier at each fast dynamic time-step. In order to represent the Lagrange multipliers at the slow dynamics timescale, the value $\hat{\lambda}_k$ is defined as the sum of Langrange multipliers over one slow dynamics time-step. This concept is illustrated in Fig. 1. The red circles indicate the Lagrange multipliers sampled at the original fast dynamics time-step, while the green diamonds indicate the sum of the Lagrange multipliers over the interval between two slow dynamics sampling times. This procedure can also be expressed as

$$\begin{aligned} \hat{\lambda}_k &= \sum_{j=k\cdot\kappa}^{\kappa(k+1)} \lambda_j \\ \kappa &= \frac{\Delta t_s}{\Delta t_f} \end{aligned} \tag{46}$$

where k is the time index of the slow dynamic system, j is the time index of the fast dynamic system, and κ is the ratio of the slow time-step size Δt_s to the fast time-step size Δt_f.

Using the definition of $\hat{\lambda}_k$, it is possible to postulate that the $\hat{\lambda}_k$ can be modeled as a discrete nonlinear dynamic system given by

$$\hat{\lambda}_{k+1} = f\hat{\lambda}(\lambda_k, U_s) \tag{47}$$

and it is assumed the function $f\hat{\lambda}$ exists, it is both unknown and nonlinear. Hence, an exact finite set of eigenfunction cannot be determined analytically and a data driven approach is required.

4.2 Koopman Operator Model Fitting Using EDMD. The states and inputs for this application are

$$\begin{aligned} x &= \hat{\lambda} \\ u &= \delta\dot{m}_f^{\text{ref}} \end{aligned} \tag{48}$$

where $\delta\dot{m}_f^{\text{ref}}$ is the original control input of the slow dynamics optimization problem. The dataset of snapshot pairs was collected by solving the optimization problem given by Eq. (29) for a variety of $\delta\dot{m}_{f,k}^{\text{ref}}$ profiles. Specifically, 300 different step and ramp functions in $\delta\dot{m}_{f,k}^{\text{ref}}$, corresponding to both increase and decrease in torque demand were considered. The resulting Lagrange multipliers were then converted into $\hat{\lambda}_k$ using Eq. (46). Monomials and Gaussian radial basis functions have been chosen as candidate observable functions [46]

$$g = \begin{bmatrix} \hat{\lambda} & \dots & \hat{\lambda}^{p_m} & e^{-\epsilon\left(\hat{\lambda}-\hat{\lambda}_{c_1}\right)} & \dots & e^{-\epsilon\left(\hat{\lambda}-\hat{\lambda}_{c_{p_g}}\right)} \end{bmatrix} \tag{49}$$

where ϵ is a calibrated shaping parameter and $\hat{\lambda}_{ci}$ are the radial centers of the data clusters. The values of the hyperparameters p_m and p_g are calibrated through an exhaustive search, where the problem in Eq. (45) is solved using a subset of the snapshot matrices for various combinations of p_m and p_g. The remaining portion of the snapshot is used to validate the accuracy of the Koopman linear system compared to the original system. Following the calibration procedure, the values of p_m and p_g are 2 and 9, respectively. For the current set of observable functions, the calibration procedure required about one day on a desktop PC running an Intel® Xeon® CPU E5-1650 v2 at 3.5 GHz with 64 GB. However, this time can be reduced by using higher power computational resources, i.e., a supercomputer, or by evaluating each set of hyperparameters in parallel.

The comparison of the Koopman predicted value of the Lagrange multipliers against the true output is shown in Fig. 2. While the error is largely within ±10% and the prediction has an root mean square (RMS) of 10.9, there is an significant number of points that fall outside this range. This indicates that the chosen candidate observable functions are insufficient, and an expansion of function dictionary is required for improving the accuracy. In addition, there is a set of 50 points for which the predicted value is constant while the true value varies, forming a horizontal line in Fig. 2. Upon further investigation, all of the points forming the horizontal line are associated with $u = 0$. When u goes to zero after the fuel and air flow rates have reached steady-state, then $\hat{\lambda}$ will also go to zero and a feasible steady-state is reached. However, if u drops to zero before steady-state is reached, the transient conditions cause constraint violation, which results into a nonzero Lagrange multiplier. In its current implementation, the Koopman model cannot distinguish if steady-state has been reached, it cannot distinguish these two scenarios, and a group of erroneous predictions forms.

To address this issue and provide the Koopman linear model additional information on whether steady-state has been reached, the state x_k is augmented with past information. Hence the new state vector is defined by

$$x_k = \begin{bmatrix} \hat{\lambda}_k & \hat{\lambda}_{k-1} & \dots & \hat{\lambda}_{k-h} \end{bmatrix} \tag{50}$$

where h is the number of historical states included. The observables are therefore given by:

$$M = \begin{bmatrix} x(1)^{v_1} x(2)^{v_2} \dots x(h)^{v_h} \end{bmatrix}, \quad v = P_h^{\mathbf{p_m}} \tag{51}$$

$$R = e^{-\epsilon\left(x(i)-x(i)_{c_j}\right)} \quad \forall i \in \begin{bmatrix} 1 & h \end{bmatrix}, j \in \begin{bmatrix} 1 & p_g \end{bmatrix} \tag{52}$$

$$G = \begin{bmatrix} M \\ R \end{bmatrix} \tag{53}$$

The results for this new set of observable functions with $h = 1$ are summarized in Fig. 3, where the RMS error is lowered to 10.1, an improvement of about 7% compared to the previous set of basis functions. While the overall agreement is improved, a group of erroneous predictions is still present, as shown in Fig. 3.

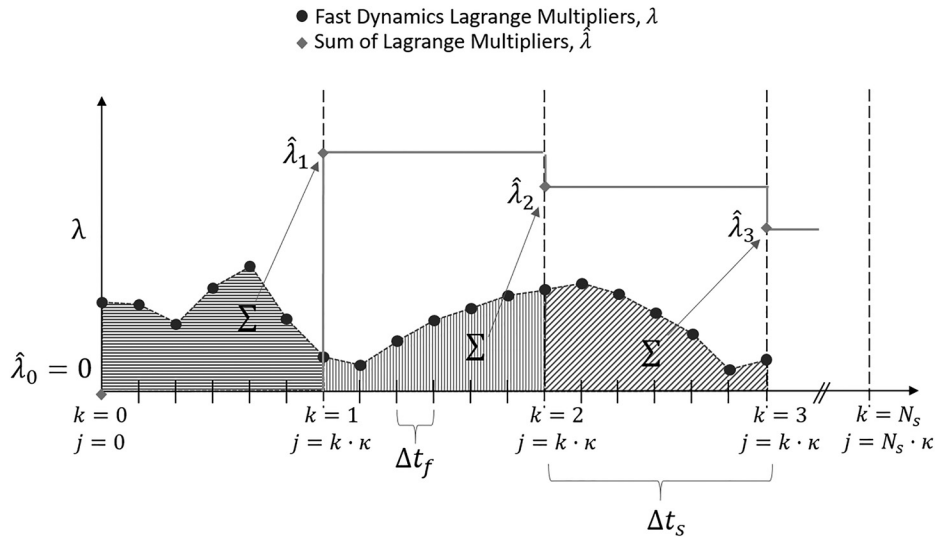

Fig. 1 Resampling of the Lagrange multipliers at the slow dynamics time scale

However, the number of points in the grouping is only 20, substantially fewer than the 50 points in the grouping when historical states are not used. The main drawback of including historical data in the state is the increased complexity of the least-squares problem solved to calibrate the functions. Specifically, Calibrating the hyperparameters for the case where $h = 1$ took five times longer on the same computer platform, and converged to 1 and 4 for p_m and p_g, respectively.

Results show that including a higher number of historical states improves accuracy and avoids *flat* conditions. An error distribution analysis is shown in Fig. 4 where the percentage error is calculated as

$$e = \frac{\lambda_k - \hat{\lambda}_k}{\bar{\lambda}} \cdot 100 \tag{54}$$

where $\bar{\lambda}$ is the average Lagrange multiplier determined in simulation. While the mean of the two distribution is similar, including the historical states reduces the standard deviation from 10.4 to 9.6 and decreases the average absolute error from 6.8% to 6.2%. However, due to the significant increase in computation time observed for the hyperparameters calibration when extending from $h = 0$ to $h = 1$, additional historical data would not be considered in this paper. Finally, both approaches have 10 instances where the predicted Lagrange multipliers are negative, which is not possible. However, no correction is used to bring those predictions to 0, which is another potential area of improvement.

4.3 Integration With Slow Dynamics Optimization. Once a data driven linear model of the Lagrange multipliers has been found, the model is integrated with the slow dynamics optimization. The slow dynamics are updated to include the observable functions as additional states, and the new slow dynamics optimization problem is given by

$$\min_{u_k^s} J_s = \sum_{k=0}^{N_s-1} \left[||\hat{x}_k^s||_{\hat{Q}_s}^2 + \hat{q}_s^T \hat{x}_k^s + ||u_k^s||_{R_s}^2 \right] + ||\hat{x}_{N_p}^s||_{\hat{P}_s}^2 \tag{55}$$

subject to

$$\hat{x}_k^s = \begin{bmatrix} [\bar{x}_k^s]^T & g_k^T \end{bmatrix}^T \tag{56}$$

$$\hat{x}_{k+1}^s = \begin{bmatrix} A_s & 0 \\ 0 & \tilde{A} \end{bmatrix} \hat{x}_k + \begin{bmatrix} B_s \\ \tilde{B} \end{bmatrix} u_k^s \tag{57}$$

$$\bar{g}_s(\bar{x}_k^s, u_k^s) \leq 0 \tag{58}$$

where the penalty terms are given by

$$Q_s = \text{diag}\left(\begin{bmatrix} 0 & 0 & 0 & 0 & \gamma_4^s & 0 & \cdots & 0 \end{bmatrix}\right) \tag{59}$$

$$q_s^T = \begin{bmatrix} 0 & 0 & \gamma_1^s & 0 & \cdots & 0 \end{bmatrix} \tag{60}$$

$$R_s = \gamma_2^s \tag{61}$$

$$P_s = \text{diag}\left(\begin{bmatrix} \gamma_3^s & 0 & \cdots & 0 \end{bmatrix}\right) \tag{62}$$

where γ_4^s is a penalty on the Lagrange multipliers to enforce $\hat{\lambda} = 0$ as a soft constraint. Because the effect of the fast dynamics constraint is included in the Koopman linear system, the hierarchical optimization problem no longer requires iteration.

5 Simulation Study

5.1 Case Studies. Two simulation cases are considered in this paper to evaluate the performance of the Koopman-

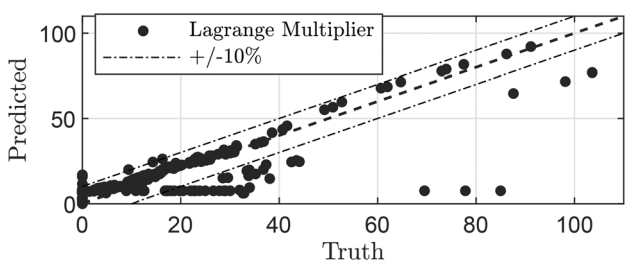

Fig. 2 Comparison of the predicted value of $\hat{\lambda}_k$ with the true value

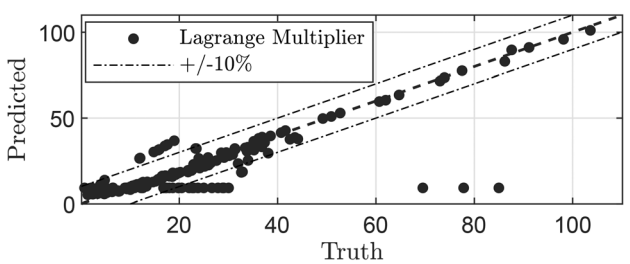

Fig. 3 Comparison of the predicted value of $\hat{\lambda}_k$ with the true value for the Koopman linear model with one historical state

Augmented MPC. Both test cases are characterized by a sharp acceleration, which causes a rapid increase in the fuel demand which will likely to lead to a constraint violation due to a limitation in the fast dynamic. The first case study represents a scenario where the ego-vehicle is following a slow lead vehicle while approaching a traffic light. Before approaching the intersection, the lead-vehicle changes lane allowing the ego-vehicle to pass while green, if an appropriate speed can be reached. The second case study examines the scenario in which the ego-vehicle is cruising at highway speed, when another vehicle enters the lane directly in front without maintaining a safety distance. The truck is forced to slow down to reestablish a safety distance and then accelerate back to highway speed in order to meet the desired overall driving distance provided by a supervisory controller.

5.2 Results. The slow dynamic MPC augmented with the Koopman model for the Lagrange multipliers is applied to both case studies and compared against the centralized and iterative MPC that were developed and presented in Ref. [25]. It is worth noting that, if perfect prediction is available, γ_4^s can be chosen arbitrarily large as it only affect unfeasible control sequences and it is zero elsewhere. However, as shown in Fig. 4, there are significant uncertainties associated with the prediction of $\hat{\lambda}$. For the simulations in this paper, $\gamma_4^s = 1 \times 10^{-8}$ is used.

The simulation results for the first case study is summarized in Fig. 5, where the control input (torque command), the actual engine torque, the vehicle speed and the constraint on the fuel-air-ration are shown for a centralized, the iterative and the Koopman augmented controller. At approximately 3 s, a sharp increase in the engine torques are observed in Figs. 5(*a*) and 5(*b*), which corresponds to a free flow condition and an update in the optimal distance traveled generated by the supervisors. The centralized MPC is able to increase the engine torque faster than the iterative and Koopman approaches and takes advantage of a pulse and glide strategy [48]. The smaller average engine torque commanded by the centralized controller result in a longer acceleration time compared to the two hierarchical strategies as shown in Fig. 5(*c*). At the same time, all strategies are guaranteeing the constraint satisfaction, as shown in Fig. 5(*d*). It is worth noting that, if the slow dynamic optimizer generates feasible trajectories, the hierarchical and the iterative schemes will provide the same solution with the same computation time, as the iteration would only take one step. Overall, the Koopman augmented approach shows the slowest response to a change in reference speed, which is attributed to the uncertainties in the prediction of $\hat{\lambda}_k$ and the selection of the weights in the optimization problem. However, the aggressiveness of the response can be tuned by calibrating γ_4^s in Eq. (60). A summary of the integral performance metrics are shown in Table 1, where fuel savings, trip time and traveled distance are compared. Because the terminal vehicle speed is different for each controller, an adjusted fuel consumption is used to account for the difference in kinetic energy

$$\bar{m}_f = m_f - \frac{\mathrm{KE} - \mathrm{KE}_c}{\mathrm{LHV}} \tag{63}$$

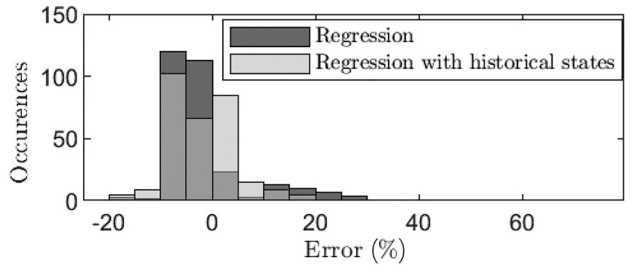

Fig. 4 Comparison of the error prediction distribution when including historical states

where m_f is the total fuel over the maneuver, KE is the kinetic energy at the end of the maneuver, the subscript c refers to the centralized controller and LHV is the lower heating value of the fuel. While the centralized controller achieves better overall performance, the Koopman-augmented controller approximates closely the control input profile and resulting state trajectories obtained by the iterative approach.

A similar comparison is performed for the second case study and the results are shown in Fig. 6.

In this case, the engine torque command drops to zero in response to the vehicle cut-in as shown in Figs. 6(*a*) and 6(*b*). Once a safe distance is established, the ego vehicle velocity has dropped from 25 m/s to 22 m/s, as shown in Fig. 6(*c*). The engine torque command increases to reach the desired speed. Similarly to the previous case, the centralized torque command has a chattering profile due to the different time scales and to benefit from a pulse and glide strategy. Moreover, the engine torque prescribed

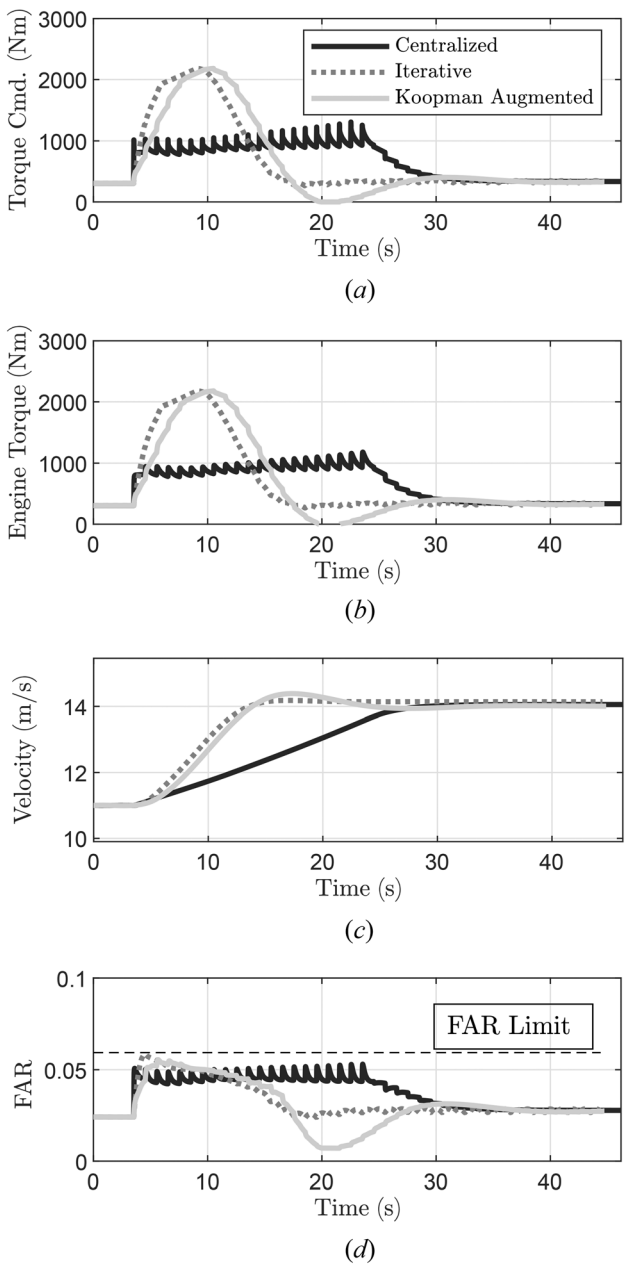

Fig. 5 Case Study 1: Comparison between iterative and Koopman-augmented MPC. (*a*) Engine torque command, (*b*) engine torque, (*c*) vehicle velocity, and (*d*) fuel-air ratio.

Table 1 Case study 1: summary of integral performance metrics

		Central.	Iter.	Koopman
Fuel	(g)	179.4	184.4	181.8
Savings	(%)	—	−2.79	−1.34
Adj. savings	(%)	—	−2.26	−1.30
Trip time	(s)	46.18	44.40	44.57
Savings	(%)	—	3.86	3.50
Distance	(m)	600.1	600.0	600.1

by the centralized controller is lower than the other two approaches resulting in a longer acceleration time, which results in a longer travel time as shown in Table 2. Again, all controllers meet the fuel air ratio (FAR) constraints as shown in Fig. 6(*d*).

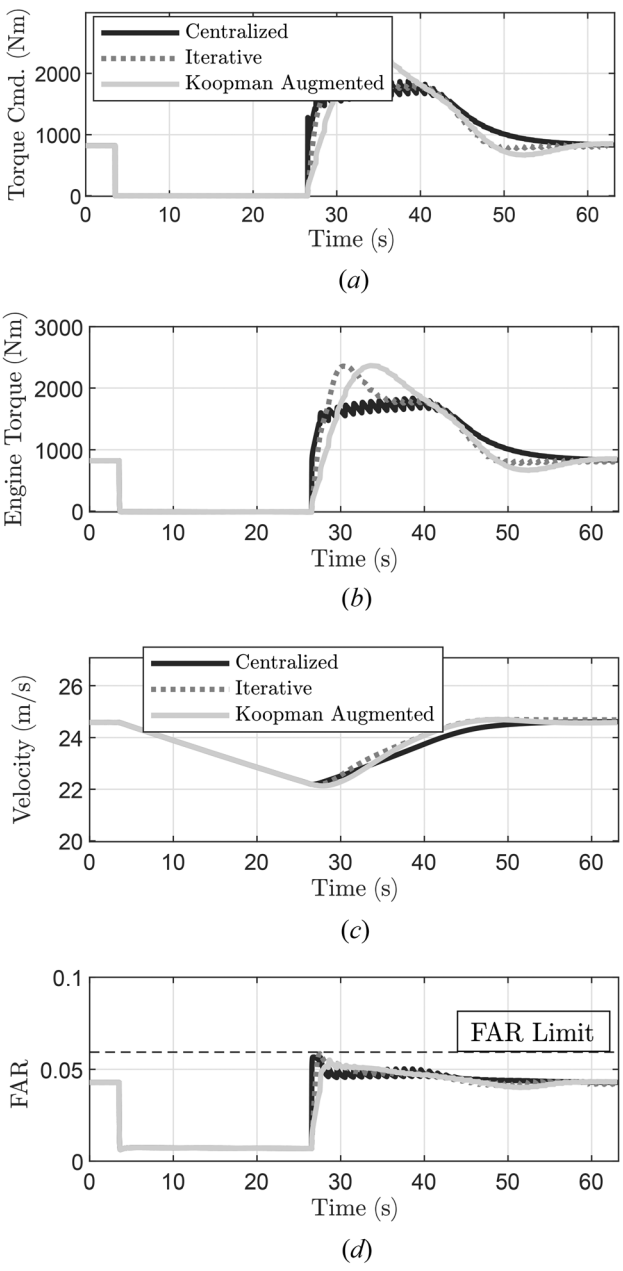

Fig. 6 Case Study 2: Comparison between iterative and Koopman-augmented MPC. (*a*) Engine torque command, (*b*) engine torque, (*c*) vehicle velocity, and (*d*) fuel-air ratio.

Table 2 Case study 2: summary of integral performance metrics

		Central.	Iter.	Koopman
Fuel	(g)	342.6	348.1	344.8
Savings	(%)	—	−1.62	−0.66
Adj. savings	(%)	—	−1.27	−0.60
Trip time	(s)	63.21	62.94	62.99
Savings	(%)	—	0.43	0.35
Distance	(m)	1500	1500	1500

6 Conclusions

This paper presented a data driven method for the prediction of the Lagrange multipliers associated with the fast dynamic optimization problem, which eliminates the requirement for iterations in a hierarchical iterative MPC for multitime-scale systems. The proposed method is based on an extension of the Koopman theory that interprets the Lagrange multipliers as an unknown nonlinear dynamical system. In this context, this paper presents the extension of the current Koopman method used in conjunction with EDMD to the prediction of the Lagrange multipliers and its integration with a slow dynamic optimization problem. Once the slow dynamic MPC was augmented the resulting Koopman linear model, the Koopman-based MPC was compared against centralized, hierarchical and iterative strategies. While the calibration process for identifying the Koopman linear system was computationally expensive and results show non-negligible uncertainties in the prediction, when integrated with the MPC, the new controller provided near optimal performance with significantly lower computational requirements compared to the initial iterative approach.

Funding Data

- This research was funded by the ARPA-E NEXTCAR Program, Project (Award No. DEAR0000801; Funder ID: 10.13039/100009224).

References

[1] Davis, S., and Boundy, R. G., 2022, "Transportation Energy Data Book," 40th ed., Oak Ridge National Labpratory, Oak Ridge, TN.

[2] Hellström, E., Ivarsson, M., Aslund, J., and Nielsen, L., 2009, "Look-Ahead Control for Heavy Trucks to Minimize Trip Time and Fuel Consumption," Control Eng. Pract., **17**(2), pp. 245–254.

[3] van Keulen, T., de Jager, B., Foster, D., and Steinbuch, M., 2010, "Velocity Trajectory Optimization in Hybrid Electric Trucks," Proceedings of the 2010 American Control Conference, IEEE, Baltimore, MD, June 30–July 2, pp. 5074–55079.

[4] Pelletier, E., Bai, W., Alvarez Tiburcio, M., Borek, J., Boyle, S., Earnhardt, C., Gao, L., et al., 2021, "In-Vehicle Validation of Heavy-Duty Vehicle Fuel Savings Via a Hierarchical Predictive Online Controller," SAE Paper No. 2021-01-0432.

[5] Deshpande, S. R., Gupta, S., Kibalama, D., Pivaro, N., Canova, M., Rizzoni, G., Aggoune, K., Olin, P., and Kirwan, J., 2021, "In-Vehicle Test Results for Advanced Propulsion and Vehicle System Controls Using Connected and Automated Vehicle Information," SAE Paper No. 2021-01-0430.

[6] Rengarajan, S., Hotz, S., Sarlashkar, J., Gankov, S., Bhagdikar, P., Gross, M. C., and Hirsch, C., 2020, "Energy Efficient Maneuvering of Connected and Automated Vehicles," SAE Paper No. 2020-01-0583.

[7] Vohra, V., Wahba, M., Akarslan, G., Ni, R., and Brennan, S., 2019, "An examination of Vehicle Spacing to Reduce Aerodynamic Drag in Truck Platoons," Proceedings of IEEE Vehicle Power and Propulsion Conference (VPPC), Institute of Electrical and Electronics Engineers Inc., Chicago, IL, Aug. 27–30.

[8] Smith, S. W., Kim, Y., Guanetti, J., Kurzhanskiy, A. A., Arcak, M., and Borrelli, F., 2019, "Balancing Safety and Traffic Throughput in Cooperative Vehicle Platooning," 18th European Control Conference (ECC), Naples, Italy, June 25–28, Institute of Electrical and Electronics Engineers Inc., pp. 2197–2202.

[9] Ngo, V. D., Colin Navarrete, J. A., Hofman, T., Steinbuch, M., and Serrarens, A., 2013, "Optimal Gear Shift Strategies for Fuel Economy and Driveability," Proc. Inst. Mech. Eng., Part D., **227**(10), pp. 1398–1413.

[10] Xu, C., Geyer, S., and Fathy, H. K., 2019, "Formulation and Comparison of Two Real-Time Predictive Gear Shift Algorithms for Connected/Automated Heavy-Duty Vehicles," IEEE Trans. Veh. Technol., **68**(8), pp. 7498–7510.

[11] Rodriguez, M., and Fathy, H., 2019, "Self-Synchronization of Connected Vehicles in Traffic Networks: What Happens When We Think of Vehicles as Waves?," 2019 American Control Conference (ACC), IEEE, Philadelphia, PA, July 10–12, pp. 2651–2657.
[12] Sun, C., Guanetti, J., Borrelli, F., and Moura, S. J., 2020, "Optimal Eco-Driving Control of Connected and Autonomous Vehicles Through Signalized Intersections," IEEE Internet Things J., **7**(5), pp. 3759–3773.
[13] Amini, M. R., Wang, H., Gong, X., Liao-Mcpherson, D., Kolmanovsky, I., and Sun, J., 2020, "Cabin and Battery Thermal Management of Connected and Automated Hevs for Improved Energy Efficiency Using Hierarchical Model Predictive Control," IEEE Trans. Control Syst. Technol., **28**(5), pp. 1711–1726.
[14] Block, B., Huynh, B., Boyle, S., Stockar, S., Geyer, S., Li, J., and Huber, J., 2019, "Analysis of the Effect of Vehicle Platooning on the Optimal Control of a Heavy Duty Engine Thermal System," SAE Paper No. 2019-01-1259.
[15] Chen, X., Heidarinejad, M., Liu, J., De La Peña, D. M., and Christofides, P. D., 2011, "Model Predictive Control of Nonlinear Singularly Perturbed Systems: Application to a Large-Scale Process Network," J. Process Control, **21**(9), pp. 1296–1305.
[16] Chen, X., Heidarinejad, M., Liu, J., and Christofides, P. D., 2012, "Composite Fast-Slow MPC Design for Nonlinear Singularly Perturbed Systems: Stability Analysis," Proceedings of the American Control Conference, Montreal, QC, Canada, June 27–29, pp. 4136–4141.
[17] Tica, A., Gueguen, H., Dumur, D., Faille, D., and Davelaar, F., 2012, "Hierarchical nonlinear Model Predictive Control for Combined Cycle Start-Up Optimization," Proceedings of the IEEE Conference on Decision and Control, Maui, HI, Dec. 10–13, pp. 2593–2598.
[18] Falcone, P., Borrelli, F., Tseng, H. E., Asgari, J., and Hrovat, D., 2008, "A Hierarchical Model Predictive Control Framework for Autonomous Ground Vehicles," 2008 American Control Conference, IEEE, Seattle, WA, June 11–13, pp. 3719–3724.
[19] Dang Doan, M., Keviczky, T., and De Schutter, B., 2011, "A Dual Decomposition-Based Optimization Method With Guaranteed Primal Feasibility for Hierarchical MPC Problems," IFAC Proc. Vol. 44(1), pp. 392–397.
[20] Picasso, B., De Vito, D., Scattolini, R., and Colaneri, P., 2010, "An MPC Approach to the Design of Two-Layer Hierarchical Control Systems," Automatica, **46**(5), pp. 823–831.
[21] Raimondo, D., Magni, L., and Scattolini, R., 2007, "Decentralized MPC of Nonlinear Systems: An Input-to-State Stability Approach," Int. J. Robust Nonlinear Control, **17**(17), pp. 1651–1667.
[22] Lefort, A., Bourdais, R., Ansanay-Alex, G., and Guéguen, H., 2013, "Hierarchical Control Method Applied to Energy Management of a Residential House," Energy and Buildings, **64**, pp. 53–61.
[23] Scattolini, R., and Colaneri, P., 2007, "Hierarchical model Predictive Control," Proceedings of the IEEE Conference on Decision and Control, New Orleans, LA, Dec. 12–14, pp. 4803–4808.
[24] Ulbig, A., Arnold, M., Chatzivasileiadis, S., and Andersson, G., 2011, "Framework for Multiple Time-Scale Cascaded MPC Application in Power Systems," IFAC Proc. Vol. 44(1), pp. 10472–10480.
[25] Boyle, S., and Stockar, S., 2022, "Multi time-Scale Engine and Powertrain Control for Autonomous Vehicles Via Lagrange Multipliers," ASME J. Dyn. Syst., Meas., Control, **144**(1), p. 011103.
[26] Chiara, F., Wang, J., Patil, C. B., Hsieh, M.-F., and Yan, F., 2011, "Development and Experimental Validation of a Control-Oriented Diesel Engine Model for Fuel Consumption and Brake Torque Predictions," Math. Comput. Modell. Dyn. Syst., **17**(3), pp. 261–277.
[27] Hendricks, E., Chevalier, A., Jensen, M., Sorenson, S. C., Trumpy, D., and Asik, J., 1996, "Modelling of the Intake Manifold Filling Dynamics," SAE Paper No. 960037.
[28] Heywood, J., 2018, *Internal Combustion Engine Fundamentals, Second Edition*, 2nd ed., McGraw-Hill Education, New York.
[29] Boyle, S., and Stockar, S., 2019, "Comparison of Input Shaping and Predictive Reference Generator Techniques for IC Engine Setpoints Commands," IFAC-PapersOnLine, **55**(5), pp. 279–284.
[30] Borrelli, F., Bemporad, A., and Morari, M., 2017, *Predictive Control for Linear and Hybrid Systems*, Cambridge University Press, Cambridge, UK.
[31] Brdys, M. A., Grochowski, M., Gminski, T., Konarczak, K., and Drewa, M., 2008, "Hierarchical Predictive Control of Integrated Wastewater Treatment Systems," Control Eng. Pract., **16**(6), pp. 751–767.
[32] Bertsekas, D., 2016, *Nonlinear Programming*, 3rd ed., Athena Scientific, Belmont, MA.
[33] Nocedal, J., and Wright, S., 1999, *Numerical Optimization*, 1st ed., Springer, New York.
[34] Gurney, K., 2018, *An Introduction to Neural Networks*, CRC Press, Boca Raton, FL.
[35] Qin, S., and Xue, X., 2015, "A two-Layer Recurrent Neural Network for Nonsmooth Convex Optimization Problems," IEEE Trans. Neural Networks Learn. Syst., **26**(6), pp. 1149–1160.
[36] Luo, X., Lv, Y., Zhou, M., Wang, W., and Zhao, W., 2016, "A laguerre Neural Network-Based ADP Learning Scheme With Its Application to Tracking Control in the Internet of Things," Pers. Ubiquitous Comput., **20**(3), pp. 361–372.
[37] Koopman, B., 1931, "Hamiltonian Systems and Transformations in Hilbert Space". Proc. Natl. Acad. Sci., **17**(5), pp. 315–318.
[38] Proctor, J. L., Brunton, S. L., and Kutz, J. N., 2016, "Generalizing Koopman Theory to Allow for Inputs and Control," SIAM J. Appl. Dyn. Syst., **17**(1), pp. 909–930.
[39] Cibulka, V., Haniš, T., and Hromčík, M., 2019, "Data-Driven Identification of Vehicle Dynamics Using Koopman Operator," 2019 22nd International Conference on Process Control (PC19), IEEE, Strbske Pleso, Slovakia, June 11–14, pp. 167–172.
[40] Korda, M., and Mezić, I., 2018, "Linear Predictors for Nonlinear Dynamical Systems: Koopman Operator Meets Model Predictive Control," Automatica, **93**(7), pp. 149–160.
[41] Mauroy, A., and Goncalves, J., 2020, "Koopman-Based Lifting Techniques for Nonlinear Systems Identification," IEEE Trans. Autom. Control, **65**(6), pp. 2550–2565.
[42] Arbabi, H., Korda, M., and Mezic, I., 2018, "A data-Driven Koopman Model Predictive Control Framework for Nonlinear Partial Differential Equations," IEEE Conference on Decision and Control, Miami Beach, FL, Dec. 17–19, pp. 6409–6414.
[43] Korda, M., and Mezić, I., 2020, "Optimal Construction of Koopman Eigenfunctions for Prediction and Control," IEEE Trans. Autom. Control, **65**(12), pp. 5114–5129.
[44] Bruder, D., Gillespie, B., Remy, C. D., and Vasudevan, R., 2019, "Modeling and Control of Soft Robots Using the Koopman Operator and Model Predictive Control," arXiv preprint arXiv:1902.02827.
[45] Igarashi, Y., Yamakita, M., Ng, J., and Asada, H. H., 2020, "MPC Performances for Nonlinear Systems Using Several Linearization Models," American Control Conference, Denver, CO, July 1–3, pp. 2426–2431.
[46] Williams, M. O., Kevrekidis, I. G., and Rowley, C. W., 2015, "A Data–Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition," J. Nonlinear Sci., **25**(6), pp. 1307–1346.
[47] Kaiser, E., Kutz, J. N., and Brunton, S. L., 2021, "Data-Driven Discovery of Koopman Eigenfunctions for Control," Mach. Learn.: Sci. Technol., **2**(3), p. 035023.
[48] Gilbert, E. G., 1976, "Vehicle Cruise: Improved Fuel Economy by Periodic Control," Automatica, **12**(2), pp. 159–166.