2023 European Control Conference (ECC)
June 13-16, 2023. Bucharest, Romania

Data-enabled Predictive Control for Empty Vehicle Rebalancing*

Pengbo Zhu[1], Giancarlo Ferrari-Trecate[2], Nikolas Geroliminis[1]

Abstract— **A critical operational challenge in Mobility-on-demand systems is the problem of imbalance between vehicle supply and passenger demand. However, conventional model-based methods require accurate parametric system models with complex nonlinear dynamics that are non-trivial to build or identify. In this paper, we implement a novel data-enabled predictive control algorithm for empty vehicle rebalancing (DeePC-VR) to instruct the repositioning policy between regions. Constructed by collected historical data from the considered unknown system, a non-parametric representation is used to predict future behavior and obtain optimal control actions, circumventing the costly system modeling process. The effectiveness of the proposed method is verified by an agent-based simulator modeling the real road network of Shenzhen, China. The proposed methods can serve more passengers with less waiting time compared to other policies, improving system efficiency and quality of service.**

I. Introduction

Recent years have seen a rapid increase in the need for mobility due to the ongoing expansion of modern urban areas and the increasing resident population. Meeting these growing demands by privately owned vehicles, however, can result in heavy traffic congestion and a scarcity of parking and road space. Mobility-on-Demand (MoD) systems have emerged as promising transportation modes in urban regions, such as Uber, Lyft, and Didi Chuxing, which bridge mobility demands and ride providers. Compared to existing public transportation, MoD systems fill the gap by offering a more customer-satisfied, time-efficient, and point-to-point option.

However, the imbalance between passenger demand and vehicle supply can compromise the system efficiency. One of the key factors is the discrepancy between the origin and destination distribution of passenger trips. Without any operator intervention, vehicles will generally accumulate in the areas where passenger destinations are concentrated, rather than where the demand is located. Therefore, it is expected to improve the system performance by introducing efficient policies which govern the configuration of the fleet, more specifically, by repositioning empty vehicles to high-demand regions [1]. This vehicle rebalancing problem has attracted more attention over recent years, especially in the application of Autonomous MoD (AMoD) systems [2], where the employed vehicles are self-driving and coordinated with each other. Most studies have focused on a region-based setup. The road network in the urban area is partitioned into a finite number of regions (referred to as 'stations'), and the rebalancing policies are implemented by deploying vehicles from one station to another. In previous studies, a discrete-time fluid model for customers and vehicles was developed in [3], and a rebalancing policy was introduced to achieve equilibrium among all stations. To account for time delays, a Model Predictive Control (MPC) scheme was proposed in [4] to achieve a demand-aligned distribution of vehicles. In [5], a time-varying network flow model was devised that leveraged stochastic future demand prediction. Optimal taxi dispatching policies were proposed in [6] by describing the evolution of dynamic traffic conditions based on the Macroscopic Fundamental diagram to improve service quality and reduce network congestion. The rebalancing problem was cast into a coverage control problem in [7], where empty vehicles were guided to high-demand areas by maximizing the total weighted cover area of fleets.

In most of the previous literature, a dynamic model of the system is required for control design, which heavily relies on expert experience and system knowledge. However, deriving reliable models can be challenging, especially in varying scenarios. Nowadays, data-driven approaches are receiving more and more attention, especially when the system is too complex (for example, for human-in-the-loop applications) or too costly to model thoroughly and identify necessary parameters. Broadly speaking, data-driven methods provide a way to learn the control policies directly from data [8]. A model-free reinforcement learning (RL) approach was investigated in shared MoD systems [9]. However, RL may overreact to imbalance if vehicles are not coordinated during training process, e.g., sending more vehicles than the actual needs in target destination areas [10], and they typically require large amounts of data to perform well. Furthermore, most RL works do not consider system constraints. In contrast, motivated by learning system behavior directly from data to predict future trajectories [11], [12], a Data-enabled Predictive Control (DeePC) algorithm was first presented in [13]. Using real-time feedback, DeePC can drive the unknown system along a desired trajectory while satisfying system constraints. By considering the existence of measurable disturbance, DeePC has been implemented to mixed traffic flow and power systems[14], [15]. The robustness of regularized DeePC for the stochastic and nonlinear system was discussed in [16] and [17].

The AMoD systems are able to collect data from both passenger (e.g. request issuing time, origin and destination)

*Swiss National Science Foundation (SNSF) under National Centre of Competence in Research, Dependable Ubiquitous Automation (NCCR Automation), grant agreement 51NF40_180545

[1] Pengbo Zhu, Nikolas Geroliminis are with Urban Transport Systems Laboratory, École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland, pengbo.zhu@epfl.ch, nikolas.geroliminis@epfl.ch

[2] Giancarlo Ferrari-Trecate Dependable Control and Decision Group, 1015 Lausanne, Switzerland, giancarlo.ferraritrecate@epfl.ch

978-3-907144-08-4 ©2023 EUCA

and vehicle sides (e.g., vehicle coordinates), meanwhile can obtain real-time plant feedback (e.g., one request is answered or not). Inspired by the great potential of leveraging these accessible data without the need to model system dynamics, this paper proposes an empty fleet management scheme using the novel DeePC method to relocate vehicles to advantageous positions for efficient request answering. To the authors' best knowledge, no existing study has investigated the data-driven predictive control for vehicle rebalancing. The remaining part of the paper proceeds as follows: We firstly recall the data-enabled predictive control algorithm proposed by [13]. Then, we introduce a data-driven vehicle rebalancing scheme, called DeePC-VR, whose detailed implementation is presented in Section III. In Section IV, the proposed method is tested in a real city operating environment. This work is concluded in Section V, and some future steps are discussed.

II. Review of Data-enabled Predictive Control

In this section, we recall the main result of [11] on non-parametric system representation, and then we give a brief review of the DeePC algorithm proposed in [13].

A. Preliminaries

Consider a discrete-time Linear Time-Invariant (LTI) system as

$$\begin{cases} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k, \end{cases} \tag{1}$$

where $x_k \in \mathbb{R}^n$ is the state vector, $u_k \in \mathbb{R}^m$ is the control input vector, and $y_k \in \mathbb{R}^p$ is the output vector of the system at time $k \in \mathbb{Z}$. $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$, and $D \in \mathbb{R}^{p\times m}$. Instead of learning A, B, C and D via system identification to obtain a parametric representation of the system, we are interested in the case when A, B, C and D are unknown but the input and output data samples, i.e., $u = col(u_1, u_2, \dots)$ and $y = col(y_1, y_2, \dots)$ are available. Note here $col(a_1, a_2, \dots, a_i) := [a_1^T; a_2^T; \dots; a_i^T]^T$.

Let $L, T_d \in \mathbb{Z}$, the trajectory $u \in \mathbb{R}^{mT_d}$ is persistently exciting [13] of order L if the Hankel matrix

$$\mathcal{H}_L(u) := \begin{bmatrix} u_1 & u_2 & \dots & u_{T_d-L+1} \\ u_2 & u_3 & \dots & u_{T_d-L+2} \\ \vdots & \vdots & \ddots & \vdots \\ u_L & u_{L+1} & \dots & u_{T_d} \end{bmatrix} \tag{2}$$

has full row rank, which means the input trajectory u is sufficiently long and rich, therefore, $T_d \geq (m+1)L - 1$.

Theorem 1 ([11]). Let $L, T_d \in \mathbb{Z}$. Collected from unknown system (1), $u^d = col(u_1, u_2, \dots, u_{T_d})$ and $y^d = col(y_1, y_2, \dots, y_{T_d})$ are input and output trajectory of length T_d, such that u^d is persistently exciting of order $L+n$. Then (u, y) is a trajectory of system (1) if and only if there exists $g \in \mathbb{R}^{T_d-L+1}$ such that

$$\begin{bmatrix} \mathcal{H}_L(u^d) \\ \mathcal{H}_L(y^d) \end{bmatrix} g = \begin{bmatrix} u \\ y \end{bmatrix}. \tag{3}$$

Theorem 1 indicates that the subspace spanned by the columns of Hankel matrix $\begin{bmatrix} \mathcal{H}_L(u^d) \\ \mathcal{H}_L(y^d) \end{bmatrix}$ corresponds to the subspace of all possible trajectories of the system (1).

Furthermore, let $T_{ini}, N \in \mathbb{Z}$ be the lengths of time horizon for initial condition estimation and future prediction, respectively. The Hankel matrices constructed by u^d and y^d are partitioned into two parts, where the superscript 'p' stands for 'past data' and 'f' for 'future data' as

$$\begin{bmatrix} U^p \\ U^f \end{bmatrix} := \mathcal{H}_{T_{ini}+N}(u^d), \tag{4}$$

$$\begin{bmatrix} Y^p \\ Y^f \end{bmatrix} := \mathcal{H}_{T_{ini}+N}(y^d), \tag{5}$$

where $U^p \in \mathbb{R}^{mT_{ini}\times(T_d-T_{ini}-N+1)}$ consists of the first mT_{ini} block row of $\mathcal{H}_{T_{ini}+N}(u^d)$, and $U^f \in \mathbb{R}^{mN\times(T_d-T_{ini}-N+1)}$ consists of the last mN block row of $\mathcal{H}_{T_{ini}+N}(u^d)$. Similarly, $Y^p \in \mathbb{R}^{pT_{ini}\times(T_d-T_{ini}-N+1)}$, $Y^f \in \mathbb{R}^{pN\times(T_d-T_{ini}-N+1)}$.

According to [15], compared to system (1), a measurable external disturbance is considered additionally and this unknown LTI system can be expressed as

$$\begin{cases} x_{k+1} &= Ax_k + Bu_k + B_d w_k \\ y_k &= Cx_k + Du_k + D_d w_k, \end{cases} \tag{6}$$

where $w_k \in \mathbb{R}^q$ is the disturbance variable, and $B_d \in \mathbb{R}^{n\times q}$ and $D_d \in \mathbb{R}^{p\times q}$.

Considering w_k as a uncontrollable input, similar to Eq. (4) and Eq. (5), we can construct another Hankel matrix regarding the disturbance and further partition it into two parts as

$$\begin{bmatrix} W^p \\ W^f \end{bmatrix} := \mathcal{H}_{T_{ini}+N}(w^d), \tag{7}$$

where $W^p \in \mathbb{R}^{qT_{ini}\times(T_d-T_{ini}-N+1)}$, $W^f \in \mathbb{R}^{qN\times(T_d-T_{ini}-N+1)}$. Assume that a historical trajectory of external disturbance w^d of length T_d is measured such that $col(u^d, w^d)$ is persistently exciting of order $T_{ini} + N + n$ [15]. According to *Theorem 1*, $col(u, w, y)$ is a possible future trajectory of system (6) of initial condition $col(u_{ini}, w_{ini}, y_{ini})$, if and only if there exist $g \in R^{T_d-Tini-N+1}$ such that

$$\begin{bmatrix} U^p \\ W^p \\ Y^p \\ U^f \\ W^f \\ Y^f \end{bmatrix} g = \begin{bmatrix} u_{ini} \\ w_{ini} \\ y_{ini} \\ u \\ w \\ y \end{bmatrix}. \tag{8}$$

Therefore, the Hankel matrix constructed directly by raw data can serve as a non-parametric model representing the system behavior, bypassing the parameter identification of the system model. Next, we will implement optimal predictive control with the constructed Hankel matrices.

B. Review of Data-enabled Predictive Control (DeePC)

Viewing the collected data $col(u^d, w^d, y^d)$, the Hankel matrices can replace the system model to predict the future trajectory. DeePC [13] solves an optimization problem in a receding horizon manner, attempting to obtain optimal control actions meanwhile satisfying the input and output constraints. Considering the time horizon $N \in \mathbb{Z}$, we formulate the optimization problem as follows

$$
\begin{aligned}
&\min_{g,u,y,\sigma_y} f(u_{k+i}, y_{k+i}) + \lambda_g \|g\|_2^2 + \lambda_y \|\sigma_y\|_2^2 \\
&subject \quad to \begin{bmatrix} U^p \\ W^p \\ Y^p \\ U^f \\ W^f \\ Y^f \end{bmatrix} g = \begin{bmatrix} u_{ini} \\ w_{ini} \\ y_{ini} \\ u \\ w \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \sigma_y \\ 0 \\ 0 \\ 0 \end{bmatrix}, \qquad (9) \\
&u_{k+i} \in \mathcal{U}, \forall i \in 0, 1, \dots, N-1, \\
&y_{k+i} \in \mathcal{Y}, \forall i \in 0, 1, \dots, N-1,
\end{aligned}
$$

where $f(u_{k+i}, y_{k+i})$ is the control objective function. The input constraint is $\mathcal{U} \subseteq \mathbb{R}^{mN}$, output constraint is $\mathcal{Y} \subseteq \mathbb{R}^{pN}$. $\|\cdot\|_\alpha$ denotes α norm. $\lambda_g \in \mathbb{R}_{>0}$ is introduced as regularization parameter to avoid data overfitting. Compared to Eq. (8), $\sigma_y \in \mathbb{R}^{T_{ini}p}$ is added to the right side as an auxiliary slack variable, and a weight coefficient $\lambda_y \in \mathbb{R}_{>0}$is incorporated as well to ensure the feasibility of the constraints over all time [13]. More discussion on robust DeePC can be found in [16].

$col(u_{ini}, w_{ini}, y_{ini})$ represents the most recent input, disturbance and output measurement and is used as the initial condition to predict the future, thus it should be updated at each time step k. Note that u and y is not independent with g since by Eq. (8), $u = U^f g$ and $y = Y^f g$. Solving Eq. (9) gives an optimal control action sequence $u^* = col(u_k^*, u_{k+1}^*, \dots, u_{k+N-1}^*)$, and we only apply the first control input, i.e., u_k^*.

III. DeePC for Vehicle Rebalancing (DeePC-VR)

This section will present the problem formulation and detailed implementation of Data-enabled Predictive Control for Vehicle Rebalancing (DeePC-VR).

A. Motivation

A challenging problem that arises in Autonomous Mobility-on-Demand systems is the imbalance between vehicle supply and customer demand. For instance, when empty vehicles only cruise freely without cooperation, some districts can be oversupplied, whereas customers who need a ride in other regions cannot be served due to no available vehicles around. Therefore, we are interested in tackling this problem with an efficient rebalancing strategy that guides empty vehicles to move to another region/stay in the same region, in order to answer as many customer requests as possible meanwhile consider the rebalancing cost. Fig. 1 shows a schematic diagram of the rebalancing task in an urban area partitioned into K_n regions. The control input instructs how many vehicles should move from region I to region J, i.e., $u_k = col(u_k^1, u_k^2, \dots, u_k^{K_n})$, where u_k^I is a vector consisting of u_k^{IJ}, where $I, J = 1, 2, \dots, K_n$. Particularly, u_{II} stands for the number of vehicles that should stay in the current region.

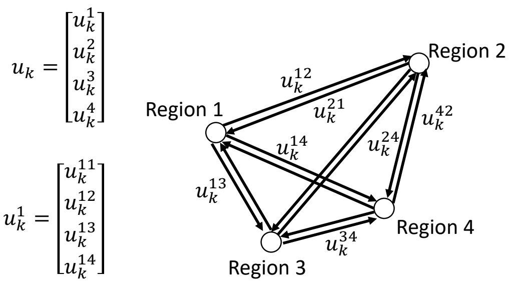

$u_k = \begin{bmatrix} u_k^1 \\ u_k^2 \\ u_k^3 \\ u_k^4 \end{bmatrix}$
$u_k^1 = \begin{bmatrix} u_k^{11} \\ u_k^{12} \\ u_k^{13} \\ u_k^{14} \end{bmatrix}$
Region 1
Region 2
Region 3
Region 4
u_k^{12}
u_k^{21}
u_k^{42}
u_k^{14}
u_k^{24}
u_k^{13}
u_k^{34}

Fig. 1: A schematic diagram of vehicle rebalancing. A four-region case is shown and a centralized controller gives region-level transfer guidance to empty vehicles in each region, e.g., u_k^{24} informs how many vehicles in Region 2 are asked to relocate to Region 4. Not all control inputs are listed in the diagram for clarity.

One conventional solution is to construct a parametric model and identify parameters from collected data. However, modeling large-scale Mobility-on-Demand systems is typically a complex problem, usually requiring complicated design efforts with expert knowledge. Meanwhile, the robustness is hard to guarantee because of many uncertainties, e.g., it is a human-in-the-loop system involving stochastic passenger demand. To bypass the complexity brought by system modeling, we turn to the data-driven control method, in particular, a data-enabled predictive control algorithm, which leverages a data Hankel matrix to replace the system model to provide optimal control policy.

B. Problem formulation

Taxi network companies, such as Uber and Lyft, can access and collect data from both the demand side (position and time information of passenger requests, etc.) and the supply side (vehicle positions and occupied status, etc.). The passenger demand is uncontrollable but measurable, therefore is referred to as the disturbance variable $w_k = col(w_k^O, w_k^D)$, where $w_k^O = col(w_k^{O,1}, w_k^{O,2}, \dots, w_k^{O,K_n})$, that $w_k^{O,I}$ and $w_k^{D,I}$ inform how many passenger requests starting from and ending in Region I, respectively. The output variable is $y_k = col(y_k^1, y_k^2, \dots, y_k^{K_n})$, where y_k^I states how many requests in Region I are successful answered.

Following the structure shown in Eq. (9), we define our objective as a linear cost that

$$f(u_{k+i}, y_{k+i}) = \sum_{i=0}^{N-1} (-||Q y_{k+i}||_1 + ||R u_{k+i}||_1), \qquad (10)$$

where $Q \in \mathbb{R}^{1 \times p}, R \in \mathbb{R}^{1 \times m}$. The first term with y_k is designed to encourage answering more requests. Moreover, the second term with u_k considers the rebalancing cost for repositioning vehicles from one region to another.

In the context of AMoD systems, not only we can collect the historical demand requests for W^p, W^f and timely measure the most recent w_{ini} during the control process, but also the future requests can be predicted accurately (for example, see [18] for demand forecasting using Long Short-Term Memory neural networks). For the purpose of simplicity, in this work, we assume that we have the perfect knowledge of demand information, therefore, utilize the true values for future demand $\tilde{w}$ to approximate the ideal, i.e., $w = \tilde{w}$. Thus, one additional equality constraint is considered as follows

$$W^f g = \tilde{w}. \tag{11}$$

n_k^I measures the number of empty vehicles in Region I at time step k. Because we can only operate the non-occupied vehicles for rebalancing, given the availability of empty vehicles n_k^I, the input constraint Eq. (12) limits the number of vehicles that can be relocated at the current time step.

$$\sum_{J=1}^{K_n} u_k^{IJ} = n_k^I, \forall I \in 0, 1, \ldots, K_n. \tag{12}$$

Additionally, constraints in Eq. (13) require positivities of input and output variables.

$$\begin{gathered} u_{k+i}^{IJ} \geq 0, \\ y_{k+i}^{I} \geq 0, \\ \forall i \in 0, 1, \ldots, N-1. \quad \forall I, J \in 1, 2, \ldots, K_n. \end{gathered} \tag{13}$$

C. Implementation of DeePC-VR actions

Solving the optimization problem Eq. (9) provides the optimal control sequence $u* = U^f g^*$, but the elements of u^* may have fractional but not integer values. In order to implement how many empty vehicles should move to another region for rebalancing, for $I \neq J$, $\lfloor u_k^{IJ} \rfloor$ empty vehicles in Region I should move to Region J, where $\lfloor \alpha \rfloor$ is the greatest integer less or equal to α; then the rest empty vehicles will stay in the current region. Moreover, let θ_k^{IJ} represent the empty vehicle flow transfer ratio from Region I to Region J as

$$\theta_k^{IJ} = \frac{u_k^{IJ}}{\sum_{J=1}^{K_n} u_k^{IJ}}. \tag{14}$$

ΔT is the control sampling time. After we apply the optimal control input u_k given by DeePC-VR at time step k, when $k\Delta T < t < (k+1)\Delta T$, once there is a vehicle that drops off its passenger in Region I and becomes empty, it follows the possibility of θ_k^{IJ} to be relocated to Region J. The procedure of implementing DeePC-VR is summarized in Algorithm 1.

Algorithm 1 DeePC for Vehicle Rebalancing (DeePC-VR)

Input: The Hankel matrices $U^p, U^f, W^p, W^f, Y^p, Y^f$ constructed by the historical data u^d, w^d and y^d. Most recent measured data u_{ini}, w_{ini} and y_{ini} of length T_{ini}, number of current empty vehicles in each region n_k.
Output: Optimal control input u_k, transfer ratio θ_k^{IJ}.

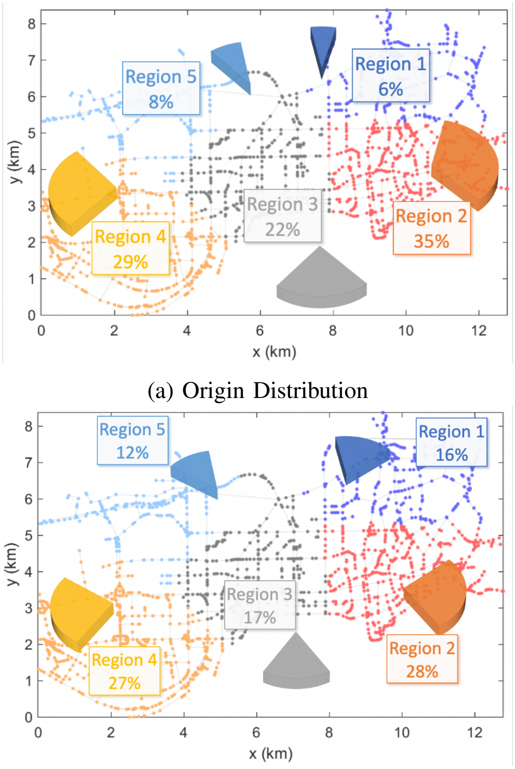
```
1) Obtain $g^*$ by solving the optimization problem (9),
2) Compute the optimal control input sequence $u^* = U^f g^*$, where $u^* = col(u_k^*, u_{k+1}^*, \ldots, u_{k+N-1}^*)$,
3) Apply the first element of the optimal control sequence, i.e., $u_k = u_k^*$, calculate $\theta_k^{IJ}$ by Eq. (14),
4) Set $k = k + 1$, update $u_{ini}, w_{ini}$ and $y_{ini}$ to the $T_{ini}$ most recent measurements,
5) return to Step 1).
```

IV. Case Study

The proposed method is tested on an AMoD simulator [19] replicating the urban road network of Luohu and Futian districts in Shenzhen, China. The network consists of 1858 intersections and 2013 road links.

A. City area partitioning

A scenario with imbalanced demand is considered (see [7]). The region of interest is divided into K_n regions using K-means clustering in accordance with the trip origin distribution. The urban area clustering/partitioning impacts the system performance, but a full discussion of different settings will be discussed in a future publication. Here, K_n was set to 5. The demand probability for these 5 regions is $\phi = [0.06, 0.35, 0.22, 0.29, 0.08]$ as it shows in Fig. 2a. More trips start from Regions 2, 3, and 4, while only a few are from Region 1 (6%) and Region 5 (8%). However, relatively more trips end in Region 1 (16%) and Region 5 (12%). Therefore, without any fleet management, empty vehicles will generally concentrate in low-demand regions, leading to an oversupply scenario in Regions 1 and 5.



(a) Origin Distribution

(b) Destination Distribution

Fig. 2: The city area is partitioned into 5 regions and colored accordingly.

B. Data collection and configuration for DeePC-VR

During the offline data collection process, the control actions were generated randomly meanwhile respecting the input constraints. And the issued time of the passenger requests is sampled from a constant rate Poisson distribution, which naturally satisfies the requirement for persistency of excitation in Eq. (8). At the same time, the number of answered requests (i.e., the output) and the requests information in each region (i.e., the external disturbance) were collected and further used to construct the Hankel matrices as in Eq. (4), Eq. (5) and Eq. (7). The length of collected historical data is $T_d = 3000$ with a sampling interval of $\Delta T = 10min$.

Meanwhile, we measured the rebalancing trip lengths between regions to form the penalty weights for control inputs in Eq. (10), in particular, $R = [0, 1.83, 2.07, 5.35, 3.94, 1.58, 0, 2.08, 4.98, 4.94, 2.56, 2.20, 0, 2.70, 2.15, 6.08, 4.77, 2.19, 0, 2.73, 4.23, 4.61, 1.70, 2.47, 0]$, where each element informs the average rebalancing trip length (unit: km) from one region to another, corresponding with the same order of control input in u_k. Particularly, the entries with 0 value indict that staying in one's current region costs no extra fuel. And the weight for output is chosen as $Q = 200 \cdot \phi$, where more weights are given to high-demand regions according to the demand probability ϕ. Besides, the following values of hyperparameters are used for the specific results shown in the following sections: $N = 30, T_{ini} = 35, \lambda_g = 1000, \lambda_y = 100$.

C. Simulation Results

Passengers cannot be kept waiting indefinitely for pickup. In this work, every issued request will remain in the matching pool for a period of $t_m = 1min$. If the nearest empty vehicle is close enough that is able to pick up the passenger within a constant threshold of $t_w = 4min$ (since the request has been issued), the passenger and the vehicle will be matched, then later cancellation of this request is not allowed in this study; otherwise, if no available vehicles are found within t_m, the passenger won't wait anymore and cancel this request.

A comparison is conducted from three perspectives: answer rate, which is the proportion of successfully answered requests over all requests; average waiting time, calculated as the time passengers spend waiting from the moment their requests are issued until they are picked up by vehicles, divided by the number of all answered requests; and rebalancing distance, which measures the total traveling distance caused by repositioning empty vehicles.

Our method is compared with another two methods: a baseline method called 'Do-nothing' policy, which means after dropping off the passenger, an empty vehicle will not move until it matches up with a new passenger; a distributed control method based on coverage control (denoted as 'CC' in Table I and Fig. 4). Assuming each vehicle can cover a circular area, this control policy steers empty vehicles to move toward high-demand regions by maximizing the total covered area regarding the demand density (see [7] for details).

One 3-hour experiment is carried out with a fleet size of 250, and around 4500 requests are issued in total. It can be seen from Table I that DeePC-VR achieves the best results over answer rate and average waiting time, compared with coverage control-based policy and Do-nothing policy. Despite the coverage control-based policy showing its advantage over Do-nothing policy in terms of answering more requests with less waiting time, it requires all empty vehicles to participate in rebalancing, which results in a large amount of energy cost. In contrast, DeePC-VR only dispatches a subset of all empty vehicles, and shows superior performance over answer rate and waiting time, meanwhile reduces 46.59% of the rebalancing distances compared to CC.

TABLE I: Performance matrices

	answer rate (%)	waiting time (s)	rebalancing distance (km)
DeePC-VR	90.15	131.99	3134.74
CC[7]	80.01	142.78	5869.27
Do-nothing	51.92	155.64	0.00

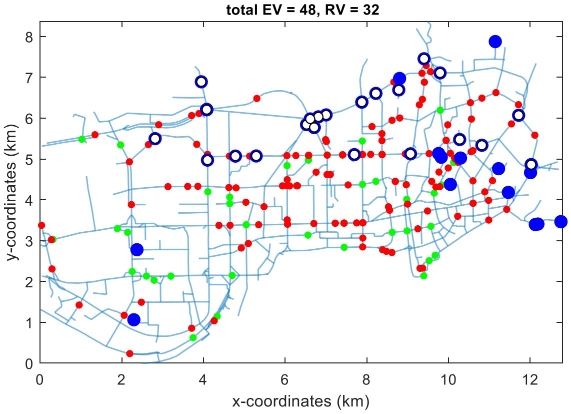


Fig. 3: A snapshot of the simulator. Empty vehicles staying in the current region and relocated to another region are shown as blue circles and blue dots, respectively. The vehicles traveling to pick up and drop off passengers are in green and red.

Fig. 3 shows a snapshot of the simulator when DeePC-VR is in action. Total empty vehicles (EVs) include the empty vehicles staying in the current region (i.e., blue dots) and rebalancing vehicles (RVs), which are being relocated to another region (i.e., blue circles). In terms of vehicle configuration, Regions 1 and 5 have many empty vehicles since they are oversupplied. In this case, DeePC-VR steers most of these EVs to move to high-demand regions, more blue circles are found indeed. On the other hand, empty vehicles in Regions 2 and 4 are commanded to stay since they are already located in high-demand areas.

The evolution of the total empty vehicles in each region is plotted in Fig. 4. For Do-nothing policy, without any intervention, more and more empty vehicles are accumulated in Regions 1 and 5 as time passes. These vehicles are not serving any passengers, meanwhile, many requests are canceled owing to this undesirable configuration of vehicle positions. By steering vehicles to move towards high-demand areas, the coverage control-based method [7] moderately alleviates this situation. One vehicle monitored by coverage control can obtain its own position guidance leveraging only local information, however, the control actions can also be myopic and may stuck in local optima due to such

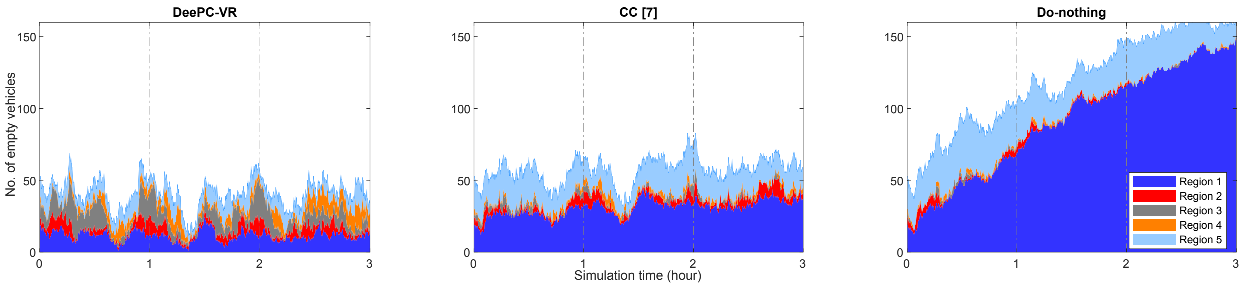


Fig. 4: Evolution of empty vehicles in each region.

distributed manner. In contrast, vehicles operated by DeePC-VR answer more requests and only a few are idle for the whole simulation. Particularly, it has more empty vehicles in high-demand regions that were relocated from low-demand ones, implying passengers will find more available vehicles nearby, thus spending less waiting time. These results suggest that DeePC-VR has a high potential in fleet management due to its ability to efficiently position vehicles in advantageous locations, leading to more requests being answered timely.

V. Conclusion

This paper presents a data-driven control algorithm (DeePC-VR) to solve the empty vehicle rebalancing problem in autonomous Mobility-on-Demand systems. Instead of modeling the complex system thoroughly, the proposed method can give regional position guidance leveraging Hankel matrices constructed from historical data. Our proposed algorithm is tested on a discrete city map using real road network geometry from Shenzhen. The proposed method outperforms other approaches in terms of answer rate and waiting time. Current results are based on perfect knowledge of regional demand information. Online learning of stochastic demands will be studied in future steps to improve robustness. Additionally, an interesting direction for future work is to develop a hierarchical structure that connects macroscopic and microscopic scopes. DeePC-VR can be used as a high-level controller for fleet transferring between regions, while a coverage control-based policy can serve as a low-level controller to provide detailed position instructions within each region. Communication and cooperation across different levels will also be investigated.

References

[1] K. Marczuk, H. Soh, C. L. Azevedo, D.-H. Lee, and E. Frazzoli, “Simulation framework for rebalancing of autonomous mobility on demand systems,” 2016.

[2] G. Zardini, N. Lanzetti, M. Pavone, and E. Frazzoli, “Analysis and control of autonomous mobility-on-demand systems,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, no. 1, p. null, 2022. [Online]. Available: https://doi.org/10.1146/annurev-control-042920-012811

[3] P. Marco, L. S. Stephen, F. Emilio, and R. Daniela, “Robotic load balancing for mobility-on-demand systems,” *The International Journal of Robotics Research*, vol. 31, no. 7, pp. 839–854, 2012. [Online]. Available: https://doi.org/10.1177/0278364912444766

[4] A. Carron, F. Seccamonte, C. Ruch, E. Frazzoli, and M. N. Zeilinger, “Scalable model predictive control for autonomous mobility-on-demand systems,” *IEEE Transactions on Control Systems Technology*, vol. 29, no. 2, pp. 635–644, 2021.

[5] M. Tsao, R. Iglesias, and M. Pavone, “Stochastic model predictive control for autonomous mobility on demand,” *CoRR*, vol. abs/1804.11074, 2018. [Online]. Available: http://arxiv.org/abs/1804.11074

[6] M. Ramezani and M. Nourinejad, “Dynamic modeling and control of taxi services in large-scale urban networks: A macroscopic approach,” *Transportation Research Part C: Emerging Technologies*, vol. 94, pp. 203–219, 2018, iSTTT22. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X17302152

[7] P. Zhu, I. Ilber Sirmatel, G. Ferrari-Trecate, and N. Geroliminis, “Idle-vehicle rebalancing coverage control for ride-sourcing systems,” *the European Control Conference 2022 (ECC22), Accepted*, 2022.

[8] F. Lamnabhi-Lagarrigue, A. Annaswamy, S. Engell, A. Isaksson, P. Khargonekar, R. M. Murray, H. Nijmeijer, T. Samad, D. Tilbury, and P. Van den Hof, “Systems & control for the future of humanity, research agenda: Current and future roles, impact and grand challenges,” *Annual Reviews in Control*, vol. 43, pp. 1–64, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1367578817300573

[9] M. Guériau and I. Dusparic, “Samod: Shared autonomous mobility-on-demand using decentralized reinforcement learning,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 1558–1563.

[10] J. Wen, J. Zhao, and P. Jaillet, “Rebalancing shared mobility-on-demand systems: A reinforcement learning approach,” in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 220–225.

[11] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. De Moor, “A note on persistency of excitation,” *Systems & Control Letters*, vol. 54, no. 4, pp. 325–329, 2005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167691104001434

[12] I. Markovsky, J. C. Willems, S. Van Huffel, and B. De Moor, *Exact and Approximate Modeling of Linear Systems*. Society for Industrial and Applied Mathematics, 2006. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9780898718263

[13] J. Coulson, J. Lygeros, and F. Dörfler, “Data-enabled predictive control: In the shallows of the deepc,” 2018. [Online]. Available: https://arxiv.org/abs/1811.05890

[14] J. Wang, Y. Zheng, K. Li, and Q. Xu, “Deep-lcc: Data-enabled predictive leading cruise control in mixed traffic flow,” 2022. [Online]. Available: https://arxiv.org/abs/2203.10639

[15] L. Huang, J. Coulson, J. Lygeros, and F. Dörfler, “Decentralized data-enabled predictive control for power system oscillation damping,” *IEEE Transactions on Control Systems Technology*, vol. 30, pp. 1065–1077, 2022.

[16] J. Coulson, J. Lygeros, and F. Dörfler, “Regularized and distributionally robust data-enabled predictive control,” in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 2696–2701.

[17] F. Dörfler, J. Coulson, and I. Markovsky, “Bridging direct & indirect data-driven control formulations via regularizations and relaxations,” 2021. [Online]. Available: https://arxiv.org/abs/2101.01273

[18] R. Iglesias, F. Rossi, K. Wang, D. Hallac, J. Leskovec, and M. Pavone, “Data-driven model predictive control of autonomous mobility-on-demand systems,” *CoRR*, vol. abs/1709.07032, 2017. [Online]. Available: http://arxiv.org/abs/1709.07032

[19] C. V. Beojone and N. Geroliminis, “On the inefficiency of ride-sourcing services towards urban congestion,” *Transportation Research Part C: Emerging Technologies*, vol. 124, p. 102890, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X20307907