

Dynamic Modeling of Bucket-Soil Interactions Using Koopman-DFL Lifting Linearization for Model Predictive Contouring Control of Autonomous Excavators*

Filippos E. Sotiropoulos and H. Harry Asada, *Member, IEEE*

Abstract—A lifting-linearization method based on the Koopman operator and Dual Faceted Linearization is applied to the control of a robotic excavator. In excavation, a bucket interacts with the surrounding soil in a highly nonlinear and complex manner. Here, we propose to represent the nonlinear bucket-soil dynamics with a set of linear state equations in a higher-dimensional space. The space of independent state variables is augmented by adding variables associated with nonlinear elements involved in the bucket-soil dynamics. These include nonlinear resistive forces and moment acting on the bucket from the soil, and the effective inertia of the bucket that varies as the soil is captured into the bucket. Variables associated with these nonlinear resistive and inertia elements are treated as additional state variables, and their time evolution is represented as another set of linear differential equations. The lifted linear dynamic model is then applied to Model Predictive Contouring Control, where a cost functional is minimized as a convex optimization problem thanks to the linear dynamics in the lifted space. The lifted linear model is tuned based on a data-driven method by using a soil dynamics simulator. Simulation experiments verify the effectiveness of the proposed lifting linearization compared to its counterpart.

Index Terms—autonomous excavation, construction and mining robots, lifting linearization, Koopman operator, model predictive contouring control, dual faceted linearization

I. INTRODUCTION

In many earth-moving tasks such as trenching, foundation digging and bench forming, it is necessary to use an excavator to move soil so as to achieve a desired final shape of the site soil. There have been a multitude of attempts from both industry and academia to develop autonomous soil forming robots. Simple autonomous systems have already been commercially available for final grading, in which the very final layer of soil is removed to create a precise soil shape. This yields a precise final soil shape but does not deal with the most challenging issue of precision digging: at deeper digging depths, the excavator experiences large, nonlinear forces that arise from the dynamic bucket-soil interactions. This is often cited as one of the most prominent challenges [1] and various approaches have been proposed to tackle this.

Both model-free and model-based approaches have been employed. Singh [2] used the Fundamental Earthmoving Equation (FEE) and a learned model to constrain the action

*This material is based upon work supported by the National Science Foundation under Grant No. NSF-CMMI 2021625

F. E. Sotiropoulos (fes@mit.edu) and H. H. Asada (asada@mit.edu) are with the Department of Mechanical Engineering, Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, USA. (The corresponding author is F. E. Sotiropoulos).

space from which an optimal trajectory is planned. More recently, Yang et al. [3] have used an optimization approach with a simple analytical soil model to determine time varying minimum torque trajectories.

Limitations in available models have led to efforts to reduce the necessity for a model altogether. One common approach has been to use interaction control to regulate the forces between bucket and soil. An early effort employed by Bernold [4] was to use impedance control so as to control the relationship between the bucket trajectory error and the exerted force. In this vein, Richardson-Little and Damaren [5] utilized a rheological model in combination with a compliance controller. An alternate approach proposed by Jud et al. [6] is to use a desired force trajectory to maintain a desirable interaction with the soil.

While model free control methods can react to changing soil forces, they are limited in achieving precise path tracking. When precise tracking of a path is required, it is necessary to predict interaction forces that vary dynamically depending on nonlinear soil properties as well as on the changing soil profile. One must predict and compensate for the complex nonlinear forces. Model Predictive Control (MPC), a framework for realizing prediction and compensation, meets this goal. The challenge, however, is to construct a dynamic model to predict the highly nonlinear and complex bucket-soil interaction.

For the use of MPC, we seek a bucket-soil interaction model that meets several key requirements. Specifically, the model must:

- Capture the rich nonlinear bucket-soil dynamics;
- Be trainable from sampled trajectory data;
- Exploit sensing, including visual mapping;
- Be usable for real-time optimization; and
- Account for the effect of soil shape.

To fulfil these requirements, we aim to exploit an emergent modeling methodology, termed lifting linearization. This modeling methodology, underpinned by the theory of the Koopman operator [7] and grounded in physical modeling theory and Dual Faceted Linearization (DFL) [8], can capture complex nonlinear dynamics with a set of linear differential equations in a higher-dimensional space. The linear representation of the nonlinear dynamics can remove or alleviate the fundamental difficulty in the use of MPC for real-time control. The cost functional of MPC can be minimized as a convex optimization problem subject to linear dynamics and linear constraints.

In the following, Koopman operator and DFL will be briefly summarized for readability, and the nonlinear dynamics of bucket-soil interaction will be lifted and represented with higher-dimensional linear equations. MPC will then be applied to the lifted linear system. The path tracking problem will be formulated as a contouring control form of MPC [9], where the bucket can be controlled along a task defined spatial path while satisfying constraints relating to the task and actuation limits. Simulation experiments evaluate the modeling accuracy and demonstrate the contouring control performance.

II. LIFTING LINEARIZATION

Lifting linearization is rooted in the seminal work by Koopman [7]. His operator theory underpins more recent development of lifting linearization. The Koopman Operator, \mathcal{K} is a linear infinite dimensional operator which evolves observations $g(x)$ of the state $x \in \mathbb{R}^n$ of an autonomous nonlinear dynamical system such that:

$$\frac{d}{dt}g(x) = \mathcal{K}g(x) \quad (1)$$

or

$$g(x_{k+1}) = \mathcal{K}_d g(x_k) \quad (2)$$

for the continuous- and discrete-time systems with no control, respectively.

A. Approximating the Koopman Operator for systems with Control

Being infinite-dimensional and for purely autonomous systems means that the Koopman operator has limited utility in the context of practical nonlinear control systems. To remedy this, a finite-dimensional approximation of the Koopman operator may be obtained as well as an approximation of the closest linear effect of the exogenous inputs such that:

$$\xi_{k+1} = A\xi_k + Bu_k \quad (3)$$

where $u_k \in \mathbb{R}^{n_u}$ is the input, and $\xi_k = [x_k^T \ g_1(x_k) \ \dots \ g_{n_g}(x_k)]^T \in \mathbb{R}^{n+n_g}$ is the set of chosen observation functions including the state itself for convenience. The discrete transition matrices with compatible dimensions A and B can then be regressed directly from N_D data:

$$G^* = \Xi Y^\dagger \quad (4)$$

where $G^* = [A \ B]$, $\Xi = [\xi_1 \ \xi_2 \ \dots \ \xi_{N_D}]$ is the measured state and observables data matrix, $Y = [y_0 \ y_1 \ \dots \ y_{N_D-1}]$ is the regressor data matrix with $y_k = [\xi_k^T \ u_k^T]^T$ and Y^\dagger is the Moore-Penrose pseudo-inverse of Y .

This outline is only the briefest introduction to the concept of identifying linear representations of nonlinear control systems based on the Koopman Operator, but for a complete treatment the reader is referred to [10] as well as several examples of this idea being applied to robotic systems [11].

B. Observables for Lifting Linearization

One factor not mentioned thus far is what functions are to be used as the observables. This is a critical factor in applying lifting linearization, as a good selection of observables can not only improve the modeling accuracy but also do so with a lower order model. To this end, different approaches have been proposed, including identifying the best observables from large libraries of nonlinear observables [12], using generic observables such as radial basis functions [13], harnessing the kernel trick to efficiently represent the nonlinear observables [14], using an optimization framework to determine Koopman eigenfunctions [15], and even using a learning approach to learn the nonlinear dictionary terms [16]. A point that has been noted by multiple authors is that choosing observables that are either related to the form of the nonlinearity in the differential equation if known [17] or observables that are otherwise physically motivated [13] often leads to superior results.

In this vein, Dual-Faceted Linearization (DFL) [8], [18] is a method of lifting linearization which exploits principles from physical modeling theory, in particular, the structure of a lumped parameter model, to determine the observables. DFL determines a special set of measurements used as lifting variables (termed “auxiliary variables”) in a systematic manner based on the connectivity of physical elements depicted in a bond graph [19] or by inspecting the equations of motion of the system. By construction, the auxiliary variables all relate to physical quantities in the original nonlinear system and thus can often be *directly measured*. Thus, even without knowledge of the system’s nonlinear constitutive relations one can create the structure of the lifted linear system, and allow for regressing the linear model using the physical measurements of the states *and* the specified auxiliary variables.

C. Lifting Linearization based on DFL

In this section, we outline the essentials to generating DFL models. An in-depth treatment on this subject can be found in [8] but an overview is presented here for clarity.

The essence of the DFL method is defining the special lifting variables, termed auxiliary variables $\eta \in \mathbb{R}^{n_a}$, as the *output* of *all* nonlinear elements involved in the nonlinear dynamical system. This representation is motivated by the natural linearity arising from the connectivity of elements in physical systems. Connections of elements, be they linear or nonlinear, are governed by linear relations. For example, Kirchoff’s current law says that all currents going in and out of a point sum to zero. Similarly, in the mechanical domain, Newton’s second law dictates that all the forces acting on a body (including the inertial force) sum to zero.

Using this definition results in the system being modeled by the following set of linear differential equations:

$$\begin{bmatrix} \dot{x} \\ \dot{\eta} \end{bmatrix} = \underbrace{\begin{bmatrix} A_{c,x} & A_{c,\eta} \\ H_{c,x} & H_{c,\eta} \end{bmatrix}}_{A_c} \begin{bmatrix} x \\ \eta \end{bmatrix} + \underbrace{\begin{bmatrix} B_{c,x} \\ H_{c,u} \end{bmatrix}}_{B_c} u \quad (5)$$

where $A_{c,x} \in \mathbb{R}^{n \times n}$, $A_{c,\eta} \in \mathbb{R}^{n \times n_a}$ and $B_{c,x} \in \mathbb{R}^{n \times n_u}$ are exact parameter matrices determined by the system structure, whereas $H_{c,x} \in \mathbb{R}^{n_a \times n}$, $H_{c,\eta} \in \mathbb{R}^{n_a \times n_a}$, and $H_{c,u} \in \mathbb{R}^{n_a \times n_u}$ are approximate and to be determined from data.

The exactness of $A_{c,x}$, $A_{c,\eta}$ and $B_{c,x}$, which is proven in [8], is facilitated by the definition of the auxiliary variables which are defined as the outputs of all the nonlinear elements in a lumped-parameter system. A similar idea is presented in [20] for Lagrangian systems where the position variables have an exact differential equation.

The estimated matrices $H_{c,x}$, $H_{c,\eta}$, and $H_{c,u}$ can then be regressed from data using a least squares estimate:

$$H = \arg \min_H \mathbb{E} [\|\dot{\eta} - \dot{\eta}_m\|^2] \quad (6)$$

where $\dot{\eta}_m$ are measured data of the auxiliary variable time derivatives and $H = [H_{c,x} \ H_{c,\eta} \ H_{c,u}]$.

This methodology has been compared with more typical Koopman operator modeling and it was reported that choosing the observables based on DFL yielded performance comparable to a Koopman model utilizing a substantially larger vector of observables [21]. This makes DFL advantageous in applications where having a compact and interpretable model is necessary.

In a control setting we may be interested in utilizing a discrete representation of the system. This can be derived through using an ordinary system discretization: $A = e^{A_c \Delta t}$ and $B = A_c^{-1}(A - I)B_c$ from the continuous DFL model or directly regressing the system matrices exactly as in (4) where $\xi_k = [x_k^T \ \eta_k^T]^T$.

In this work, we will use the DFL method for selecting observable variables and constructing the lifting linearization. This is because it offers some specific features and benefits:

- The model is composed of physically meaningful variables. As such, the model is accountable and provides the control designer with physical intuitions.
- The lifting variables can be measured with sensors, providing richer data than independent state variables alone. It opens up an avenue for exploiting sensor technology for improved modeling and control.
- Choosing the observables is structured, reducing trial-and-error efforts for selecting lifting variables.

III. EXCAVATION MODEL WITH DFL

In this section, we present the modeling of the bucket-soil system using DFL. In the spirit of simplicity and to elucidate the capability of the lifted linearization to capture the soil dynamics, we purely model and control a bucket while decoupling the boom-arm structure and dynamics. This equates to controlling a robot in operational space [22]. Although challenging due to high levels of internal friction in excavation machinery, there have been several successful endeavours in controlling excavators in task space [6], [23], [24].

In addition, we consider the bucket moving purely in the vertical plane. This is a practical simplification since the forces necessary to move a bucket sideways generally mean these motions are either not possible or practical.

A. Bucket State Equations

Given the high level assumptions, the bucket is modeled to translate and rotate in the vertical plane (see Fig 1). As the soil ahead of the bucket fails through shearing, it exerts the forces $e_{T,x}$ and $e_{T,z}$ on the bucket. These nonlinear forces also generate a moment $e_{T,\phi}$ on the bucket. Physically, these forces and moment are generated through highly nonlinear bucket-soil interactions, which are nonlinear resistive elements. According to the DFL method, $e_{T,x}$, $e_{T,z}$, $e_{T,\phi}$ are auxiliary variables, since they are the outputs of nonlinear resistive elements.

These forces and moments in conjunction with the control inputs, which are the actuator forces and torque, $u = [u_x \ u_z \ u_\phi]^T$, represent the total forces and torque acting on the bucket. From Newton's Second Law, the total forces and moment are equal to the time rate of change to the momenta and angular momentum of the bucket and the soil that moves together with the bucket.

$$\begin{aligned} \dot{p}_x &= -e_{T,x} + u_x \\ \dot{p}_z &= -e_{T,z} + u_z \\ \dot{p}_\phi &= -e_{T,\phi} + u_\phi \end{aligned} \quad (7)$$

where p_x , p_z , p_ϕ are, respectively, momenta and angular momentum of the combined bucket and soil that move together. According to the physical modeling theory and the DFL modeling, the output of an inertial element is velocity. As the inertia of the soil captured by the bucket is not constant, the functional relationship between momenta and velocities of the combined bucket and soil is nonlinear. This implies that the velocities are auxiliary variables, according to the DFL formalism. Let x and z be the coordinates of the tip of the bucket and ϕ the orientation of the bucket, as shown in Fig. 1. Their time derivatives are:

$$\begin{aligned} \dot{x} &= v_x \\ \dot{z} &= v_z \\ \dot{\phi} &= \omega \end{aligned} \quad (8)$$

Equations (7) and (8) represent the state equations of the bucket dynamics, where the independent state variables are $x = [x \ z \ \phi \ p_x \ p_z \ p_\phi]^T$. Note that the state equations are linear equations of the auxiliary variables v_x , v_z , ω and $e_{T,x}$, $e_{T,z}$, $e_{T,\phi}$ as well as inputs. These correspond to the upper half of the lifted state equations (5).

B. Dynamics of auxiliary variables

In lifting linearization, nonlinear elements, such as the resistive element associated to the bucket-soil interactions and the inertial elements associated to the combined bucket-soil system, are linearized not merely by taking an algebraic approximation but by recasting the nonlinear dynamics in a higher dimensional space and approximating it as a higher-order linear differential equation. Auxiliary variables $e_{T,x}$, $e_{T,z}$, $e_{T,\phi}$ for example, are treated as state variables possessing linear state equations, as in the lower half of equation (5). As for the auxiliary variables, v_x , v_z , ω , associated to the nonlinear inertia of the combined bucket and soil, the

nonlinearity comes from the varying mass and moment of inertia.

$$\begin{aligned} v_x = p_x/m, \quad v_z = p_z/m & \quad \text{where } m = m_{\text{bucket}} + m_{\text{soil}} \\ \omega = p_\phi/I, & \quad \text{where } I = I_{\text{bucket}} + I_{\text{soil}}. \end{aligned} \quad (9)$$

Note that the bucket inertias, m_{bucket} and I_{bucket} , are constant and that soil inertia, m_{soil} and I_{soil} are variables that dynamically vary as the soil is collected into the bucket. We treat these variables as auxiliary variables and represent their dynamics as a linear differential equation of the motion of the bucket interacting with soil. As shown in Fig. 1, the transition of m_{soil} and I_{soil} depends on a) a bucket position, b) bucket velocity, c) soil surface profile, d) the amount and distribution of soil captured by the bucket and e) soil properties. Variables a) are involved in the bucket state x , b) is in the auxiliary variables η , d) is represented with the current soil mass and moment of inertia m_{soil} and I_{soil} which are now part of the auxiliary variables.

$$\eta = [v_x \ v_z \ \omega \ e_{T,x} \ e_{T,z} \ e_{T,\phi} \ m_{\text{soil}} \ I_{\text{soil}}]^T \in \mathbb{R}^8 \quad (10)$$

Parameters of soil properties are assumed constant. The remaining variables are related to the soil surface profile c).

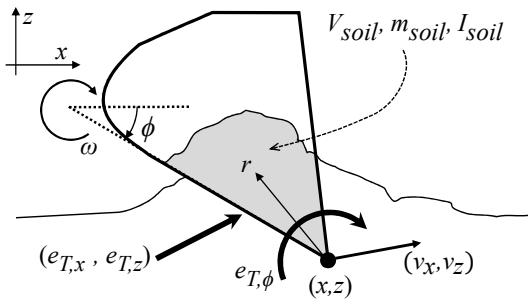


Fig. 1. Variables used as states and auxiliary variables in the planar excavation model.

C. Inclusion of Soil Shape

We represent the soil profile as a function of x : $s = s(x)$. See Fig. 2. The soil profile $s(x)$ influences the bucket dynamics. Variation in the profile results in changes in the resistive forces and moments, as well as in the soil captured by the bucket. Therefore, we treat the soil profile as an exogenous input or disturbance, and assume that the soil profile input drives the system linearly.

$$\xi_{k+1} = A\xi_k + Bu_k + B_ss_k \quad (11)$$

where $s_k = [s(x_k) \ s'(x_k)]^T \in \mathbb{R}^2$. Numerical analysis reveals that the soil height s and its gradient s' have significant influences upon the system dynamics, while the influence of its higher-order spatial derivatives are negligible. The auxiliary variable dynamics can be determined in the same ways as (4) where now $G^* \triangleq [A \ B \ B_s]$ and $y_k = [\xi_k^T \ u_k^T \ s_k^T]^T$.

IV. CONTROL

Based on the lifted dynamic model, this section aims to construct a controller for tracking a geometric path.

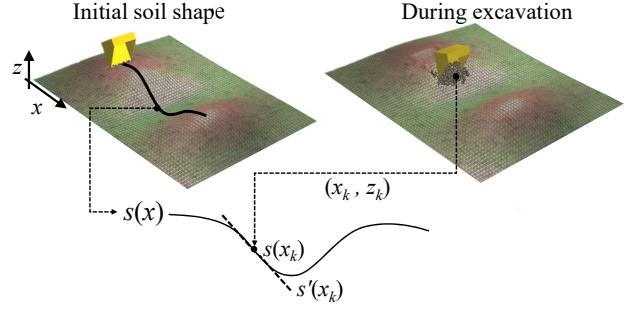


Fig. 2. The soil shape $s(x)$ is illustrated on AGX Dynamics simulation environment. s_k is determined by evaluating what the soil height and gradient of the initial soil shape was at the current bucket tip location x_k .

A. Model Predictive Contouring Control

In an excavation task, the high level objective is to have the bucket pass through a certain path relative to the surface of the soil so as to shape the soil in a specific manner. In this sense, the control objective is not to have the bucket follow a time trajectory of positions but rather to traverse the geometric and spatial path as closely as possible. Thus, the predictive control methodology used to determine the input u , is based on the model predictive contouring control (MPCC) algorithm presented by Lam et al. [9].

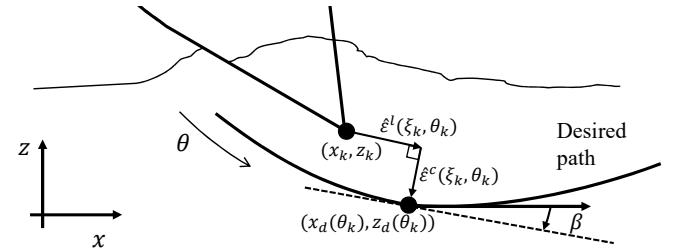


Fig. 3. The cost of the MPCC is composed of two error, $\hat{\epsilon}_k^c$ the contouring error and $\hat{\epsilon}_k^l$ the lag error between the bucket tip and a point on the path parametrized by θ_k .

The crux of the methodology is to augment the system dynamics with a virtual state θ , which represents the arc-length of a given path. As shown in Fig. 3, virtual state θ dictates the desired position on the path $(x_d(\theta), z_d(\theta))$ for each trajectory timestep. Instead of calculating the shortest distance between path and bucket tip, which is computationally expensive, the path error is represented based on two orthogonal components, the contouring and lag errors, as shown in Fig. 3.

$$\begin{aligned} \hat{\epsilon}_k^c(\xi_k, \theta_k) = & \sin \beta(\theta_k)(x_k - x_d(\theta_k)) \\ & - \cos \beta(\theta_k)(z_k - z_d(\theta_k)) \end{aligned} \quad (12)$$

$$\begin{aligned} \hat{\epsilon}_k^l(\xi_k, \theta_k) = & -\cos \beta(\theta_k)(x_k - x_d(\theta_k)) \\ & - \sin \beta(\theta_k)(z_k - z_d(\theta_k)) \end{aligned} \quad (13)$$

where,

$$\beta(\theta_k) = \arctan \left(\frac{\partial z_d / \partial \theta}{\partial x_d / \partial \theta} \Big|_{\theta=\theta_k} \right) \quad (14)$$

and the dynamics of the path variable θ are controlled by the virtual input v_k :

$$\theta_{k+1} = \theta_k + v_k, \quad v_k \in [0, v_{max}], \quad v_{max} > 0 \quad (15)$$

In the MPCC formulation, these contouring and lag errors are evaluated over a given time horizon, say N time-steps. While the dynamics of the system have been expressed linearly in the lifted space, the transformation of the contouring and lag errors to the state variables is nonlinear, as shown in eqs. (12) and (13). Furthermore, the soil profile $s(x_k)$ is a nonlinear function of x_k .

For both the errors, $\hat{\varepsilon}_k^c(\xi_k, \theta_k)$ and $\hat{\varepsilon}_k^l(\xi_k, \theta_k)$, and the soil surface shape s_k , a linearization is performed about the optimized trajectory at the previous time-steps: $\hat{\Xi}_k^* = \{\hat{\xi}_{k,k}^*, \dots, \hat{\xi}_{k+N-1,k}^*\}$, and $\hat{\Theta}_k^* = \{\hat{\theta}_{k,k}^*, \dots, \hat{\theta}_{k+N-1,k}^*\}$. This results in the linearized contouring and lag error:

$$\begin{aligned} \hat{\varepsilon}_{k+i,k}^{a,c} &= \hat{\varepsilon}^c(\hat{\xi}_{k+i,k}^*, \hat{\theta}_{k+i,k}^*) + \nabla \hat{\varepsilon}^c(\hat{\xi}_{k+i,k}^*, \hat{\theta}_{k+i,k}^*) \begin{bmatrix} \hat{\xi}_{k+1} \\ \theta_{k+1} \end{bmatrix} \\ \hat{\varepsilon}_{k+i,k}^{a,l} &= \hat{\varepsilon}^l(\hat{\xi}_{k+i,k}^*, \hat{\theta}_{k+i,k}^*) + \nabla \hat{\varepsilon}^l(\hat{\xi}_{k+i,k}^*, \hat{\theta}_{k+i,k}^*) \begin{bmatrix} \hat{\xi}_{k+1} \\ \theta_{k+1} \end{bmatrix} \end{aligned} \quad (16)$$

and linearized soil profile:

$$\begin{aligned} s_{k+i-1,k}^a &= \begin{bmatrix} s(\hat{x}_{k+i-1,k}^*) \\ s'(\hat{x}_{k+i-1,k}^*) \end{bmatrix} + \begin{bmatrix} s'(\hat{x}_{k+i-1,k}^*) \\ s''(\hat{x}_{k+i-1,k}^*) \end{bmatrix} (x_{k+i-1} - \hat{x}_{k+i-1,k}^*) \\ i &= 1, 2, \dots, N \end{aligned} \quad (17)$$

With these linearized errors and dynamics, the following convex QP can be defined to solve the MPCC problem:

$$\begin{aligned} \min_{\substack{u_{k+i-1}, v_{k+i-1} \\ \xi_{k+i}, \theta_{k+i} \\ i=1,2,\dots,N}} \quad & J_k \\ \text{s.t.} \quad & \xi_{k+i} = A\xi_{k+i-1} + Bu_{k+i-1} + Bs_{k+i-1,k}^a \\ & \theta_{k+i} = \theta_{k+i-1} + v_{k+i-1} \\ & u_{k+i-1} \in [u_{min}, u_{max}], \quad v_{k+i-1} \in [0, v_{max}] \\ & \xi_{k+i} \in [\xi_{min}, \xi_{max}], \quad \theta_{k+i} \in [\theta_s, 0] \end{aligned} \quad (18)$$

where the cost J_k at time k , is:

$$\begin{aligned} J_k = \sum_{i=1}^N & \left(\begin{bmatrix} \hat{\varepsilon}_{k+i,k}^{a,c} \\ \hat{\varepsilon}_{k+i,k}^{a,l} \end{bmatrix}^\top Q \begin{bmatrix} \hat{\varepsilon}_{k+i,k}^{a,c} \\ \hat{\varepsilon}_{k+i,k}^{a,l} \end{bmatrix} - q_\theta \theta_{k+1} \right. \\ & \left. + \begin{bmatrix} \Delta u_{k+1} \\ \Delta v_{k+1} \end{bmatrix}^\top R \begin{bmatrix} \Delta u_{k+1} \\ \Delta v_{k+1} \end{bmatrix} \right) \end{aligned}$$

where $Q \in \mathbb{R}^{2 \times 2}$ defines the contouring and lag cost, $q_\theta \in \mathbb{R}$ rewards progression along the path, $R \in \mathbb{R}^{4 \times 4}$ defines the cost of changes in the input and N is the prediction horizon.

One of the key benefits of using an optimization framework to control our system is that we can set constraints on the control inputs and states. Here for the majority of the states we set the limits so that the states fall within the 10-th and 90-th percentile of the training data used to determine the model. This is especially important since the linear model of the nonlinear system will be most valid in the region of the state space where it was trained.

Additionally we can set state constraints which may aid in the excavation task. Firstly, we constrain the horizontal velocity to be positive: $0 \leq v_{x,k}$. The vertical position of the bucket tip is constrained to be below the surface of the soil according to $s(x)$ and the previous optimized trajectory such that $z_{k+i} \leq s(\hat{x}_{k+i,k}^*)$.

Furthermore, DFL allows us to constrain the auxiliary variables. Unlike ordinary Koopman modeling where the observables do not possess any specific physical interpretation, in DFL the auxiliary variables (lifting variables) are physically meaningful and thus the system designer may seek to constrain them. In excavation this may be applied in several places. Firstly, the forces on the bucket can be directly constrained, for example, so as to reduce wear and damage to the bucket. Moreover, since the soil mass, m_{soil} , is part of the lifted space, one can constrain it to within desired amounts so as to collect a certain amount of soil.

V. NUMERICAL SYSTEM SIMULATION

This section aims to implement and test the proposed method in a simulation environment based on the *AGX dynamics*¹ physics simulator and the specialized module for the simulation of machine-soil interactions: *agxTerrain* [25]. This simulation environment allows for testing our proposed method on a system which embodies many of the characteristic nonlinearities present in an excavation system. The MPCC QP is solved using the OSQP solver² [26].

A. Digging environment

The soil shape is randomized by adding and subtracting multiple 2-D Gaussians (with randomly sampled height and standard deviation) from a height field with a randomized gradient. This yields soil surfaces with greatly varying shapes (See Fig. 2 for a sample site shape). The soil properties are implemented using the preset calibrated “gravel” material model.

B. Data Collection

In most examples in the lifting linearization literature, open loop random signals are applied to the system inputs to collect training data. However, this is not feasible in this application. Controlling the bucket without feedback readily results in either becoming deeply lodged in the soil and stalling, or leaving the soil altogether. Instead, we implement a simple PID control scheme whereby the translational d.o.f are speed controlled, and the angle of the bucket is position controlled:

$$\begin{aligned} u_x &= PID(v_x - \mathcal{U}(v_{x,min}, v_{x,max})) + \mathcal{U}(-w_x, w_x) \\ u_z &= PID(v_z - \mathcal{U}(v_{z,min}, v_{z,max})) + \mathcal{U}(-w_z, w_z) \\ u_\phi &= PID(\phi - \mathcal{U}(\phi_{min}, \phi_{max})) + \mathcal{U}(-w_\phi, w_\phi) \end{aligned} \quad (19)$$

The setpoints are drawn from a uniform distribution $(\mathcal{U}(a,b)$ between a and b) and additionally random noise, again from a uniform distribution, is added. This injection of noise is necessary so as to be able to correctly identify the system

¹<https://www.algoryx.se/agx-dynamics/>

²<https://osqp.org/>

dynamics despite utilizing feedback to collect the data [27]. The control and data sampling frequency is 30Hz.

C. Measurement of lifted state variables

Modeling and control based on DFL lifting linearization requires the access to all the states and auxiliary variables. This section discusses how these variables can be obtained in both simulation environment and experimental setting. The bucket positions (x, z, ϕ) and velocities (v_x, v_z, ω) are accessible in both simulation and experiment. The interactive bucket-soil forces $e_{T,x}$, $e_{T,z}$ and $e_{T,\phi}$ can be extracted from the internal variables in the simulation environment described above. The inertial variables, m_{soil} and I_{soil} , too, can be determined by counting the number and distribution of the grains of soil within the bucket.

In an experimental setting, additional instrumentation will be necessary to obtain these variables in real time. The bucket-soil interaction forces can be estimated based on piston pressure measurements [28], or be directly measured by instrumenting the bucket [29]. The inertial variables, m_{soil} and I_{soil} , can be estimated with use of a vision system. Assuming a constant soil density ρ_{soil} , the soil mass is given by $m_{soil} = \rho_{soil} V_{soil}$ where $V_{soil} = \int dV$ is the volume of all the soil grains in the bucket. Similarly $I_{soil} = \rho_{soil} \int r^2 dV$ where r is the distance from the tip to all the points within the bucket. These soil volume and distribution can be determined from images of the bucket and the surrounding soil. In recent years, camera systems used to monitor the bucket of excavation machines, mounted on the boom and arm, have been proposed and this technology can be used to estimate those quantities [30]. Given a known pose of the bucket and a depth camera attached properly, one can estimate the volume of soil within the bucket as well as evaluate the integral necessary for the inertia.

The soil surface shape function $s(x)$ is determined by matching a cubic spline to the initial soil surface and using that smooth functional approximation to evaluate the height, s and derivatives s' and s'' .

VI. RESULTS AND DISCUSSION

This section presents the simulation experiment results obtained from the simulation environment providing all the state and auxiliary variables.

A. Model Evaluation

First, we evaluate the model in terms of prediction accuracy compared to the ground truth agxTerrain simulation values. The Mean Squared Error (MSE) of the bucket position and orientation over an extended time horizon is shown in Fig. 4 in logarithmic scale. Each data-point on this plot is based on 30,000 test samples. From the data, 10 different data sets were created and the model was trained for each data set. Furthermore, each model was tested using 10 validation trajectories and forecasting forward from 300 randomly selected initial locations. The plots in Fig. 4 show the average of these test results. Prediction error is small in general. However, it increases as the time horizon increases.

For MPC application, this lifted linear model can be used as a valid model over a finite time horizon.

The proposed DFL-based lifted linear model was compared to a Koopman-based model, where a set of polynomial functions of state variables were used as observables. We can find that the DFL-based model outperforms the Koopman-based method, although the order of the DFL model (14th order) is much lower than that of the Koopman (27th order).

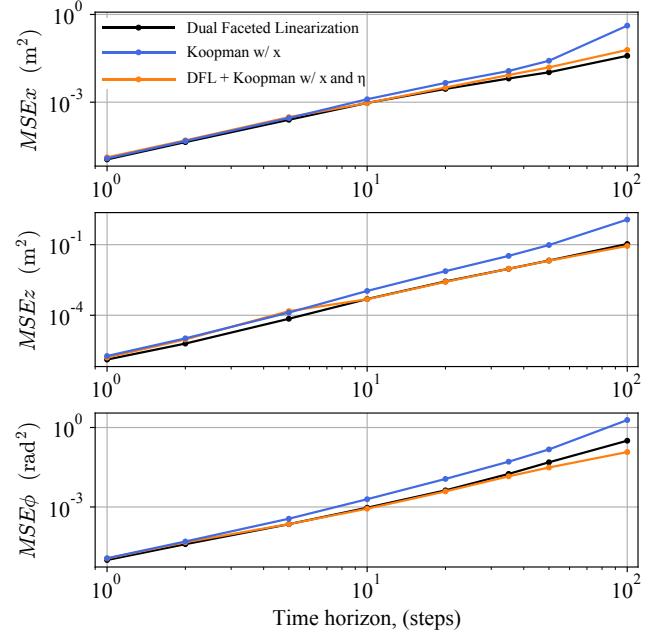


Fig. 4. The MSE error for each of the three position variables using DFL and Koopman. The black line is for the DFL model presented in this paper, blue is the conventional Koopman-operator method with polynomial observables of state variables, and orange is the Koopman-operator model utilizing both the independent states and auxiliary variables with polynomial observables. The DFL model (black) is 14th order, the Koopman with 2nd-order polynomial observables of state x (blue) is 27th order, and the combined Koopman + DFL with 2nd-order polynomial observables of x and η (orange) is 119th order.

Note that in DFL *measured* auxiliary variables are used. Even for a considerably smaller number of total observables, the DFL model generally outperforms the Koopman lifted model that utilizes only the state variables (black versus blue data). This highlights the importance of judicious choice of measurements with which to lift the dynamics. Physically meaningful measurements provide critical information for regressing higher accuracy models.

In Fig. 4, the DFL model was also compared to the Koopman-based model where both state and auxiliary variables were used for constructing observables. The resultant lifted system order of this Koopman model is 119th order. Despite the high order, the Koopman-model does not exhibit significant improvement in prediction accuracy compared to the DFL model.

In constructing these models, it was found that a model with fewer variables is more robust to training data-set selection and regression. We found that further increasing the polynomial order of the Koopman models incurred a

significant increase of validation test error. This was observed to be caused by the regressed model often being dynamically unstable. This is an issue which has been observed by prior works [31], [32]. The observables determined through DFL appear to be more robust to obtaining a model from data than higher order models when using a pure L2 regression. The effect of training dataset size shown in Fig. 5 revealed that approximately 2,000 sample pairs were sufficient to learn the model to the level where no more data significantly improve performance.



Fig. 5. Prediction MSE at a time horizon of 20 steps as a function of training dataset size. For all the variables it is observed that the model prediction improves with increased dataset size but saturates at approximately 2000-3000 sample pairs.

B. Trajectory Control Evaluation

We also evaluated the performance of the proposed model for contouring control of the bucket system along desired paths. The MPCC control was implemented for the DFL-based lifted linear model and tested in the agxTerrain simulation environment. Fig. 6 shows the simulation experiment results of the tip trajectories for several soil profiles. The bucket successfully follows the path, despite the initial transient error. Once the bucket tip arrived near the desired path, the MPCC controller allowed the bucket to track the desired path although the soil profile varied. The path tracking accuracy was improved as the model was trained with a larger training dataset, as shown in Fig. 7. With a high contouring error penalty, the mean path error decreased with increasing training dataset size. The performance appears to saturate at approximately the same dataset size as in the case of the model prediction performance.

Changing the relative magnitude of the weight matrices in the MPCC controller allows us to make the trade-off between speed progression along the path and precision of the digging task. As illustrated in Fig. 8, modulating q_θ alters the resulting trajectory and time to completion, for identical soil conditions and desired paths.

It is interesting to note that the proposed MPCC controller can also be utilized to incrementally find a multi-path excavation trajectory for digging a deep trench. See Fig. 9. As the profile becomes deep, it is impossible to excavate it in a single scooping cycle due to input or soil force constraints. Instead, the optimization subject to these constraints leads to incremental multi-cycle excavation. Using the same path and same control parameters the soil profile is excavated through consecutive cycles.

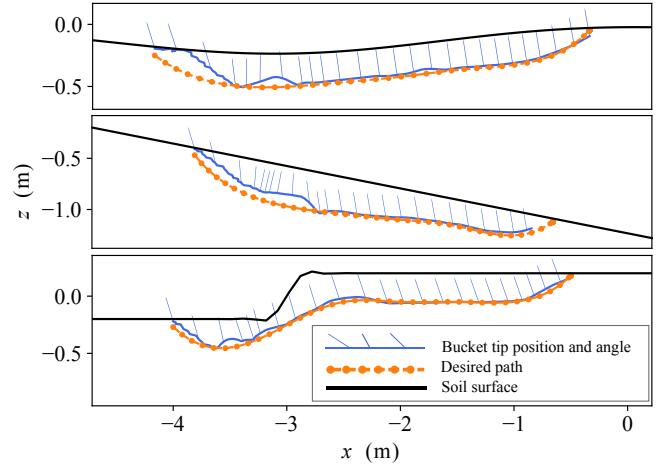


Fig. 6. Several sample results using the MPCC-DFL controller on soil with various shapes.



Fig. 7. The mean path error utilizing the proposed controller, with low q_θ such that a close following of the path is sought is plotted for varying dataset size used to train the DFL model. The path error is defined as the minimum distance between a (x, z) location and the path.

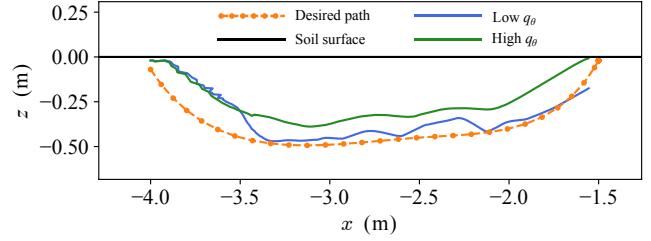


Fig. 8. Changing the path progression weight q_θ changes the balance of progressing rapidly versus accurately through the soil. For lower values of q_θ the path is followed more precisely but at a slower speed. For this trial the time to complete the excavation cycle was 4.2s for $q_\theta = 4$ (green) and 7.0s for $q_\theta = 1$ (blue).

VII. CONCLUSION AND FURTHER WORK

In this paper we have demonstrated the feasibility of using lifting linearization to model the dynamic interactions between a bucket and the soil it is excavating. The learned DFL model was compact, yet capable of capturing a significant portion of the nonlinear dynamics. Using this model we found that we were able to control a bucket to follow desired paths, while respecting state and input constraints, using a model predictive contouring controller with quadratic cost and linear constraints.

There are a few important issues to be addressed further. First, the proposed method must be integrated with a com-

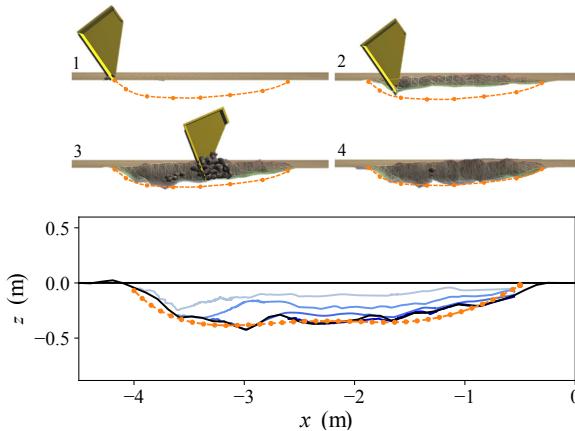


Fig. 9. Excavation of a desired soil shape is achieved through consecutive digging cycles utilizing the same path and control parameters. The black lines indicate the initial and final soil shape, the orange the desired shape, and the progressively darker blue lines the successive digging trajectories.

plete excavator controller. This paper focused on the bucket-soil dynamics and omitted the dynamics of the hydraulically-powered excavator dynamics. MPC control of hydraulic excavators has been addressed in multiple prior works [33], [34]. It is necessary to integrate the bucket-soil dynamic control with those prior works focusing on the excavator control with simple soil dynamics.

Another critical aspect which must be addressed is dealing with varying or changing soil types. Currently we rely on a static dataset which is trained on a single soil type. By leveraging techniques in active learning for lifted linearization [35] or selectively sampling from a large pre-existing dataset we may be able to expand the applicability to varying soil properties encountered by the excavator.

REFERENCES

- [1] S. Dadhich, U. Bodin, and U. Andersson, “Key challenges in automation of earth-moving machines,” *Autom. Constr.*, vol. 68, pp. 212 – 222, 2016.
- [2] S. Singh, “Synthesis of tactical plans for robotic excavation,” Ph.D. dissertation, Carnegie Mellon Univ., Pittsburgh, June 1995.
- [3] Y. Yang, J. Pan, P. Long, X. Song, and L. Zhang, “Time variable minimum torque trajectory optimization for autonomous excavator,” *ArXiv*, vol. abs/2006.00811, 2020.
- [4] L. E. Bernold, “Motion and path control for robotic excavation,” *J. Aerosp. Eng.*, vol. 6, no. 1, pp. 1–18, 1993.
- [5] W. Richardson-Little and C. J. Damaren, “Position accommodation and compliance control for robotic excavation,” in *Proceedings of 2005 IEEE Conference on Control Applications*, 2005, pp. 1194–1199.
- [6] D. Jud, G. Hottiger, P. Leemann, and M. Hutter, “Planning and control for autonomous excavation,” *IEEE Rob. Autom. Lett.*, vol. 2, no. 4, pp. 2151–2158, Oct 2017.
- [7] B. O. Koopman, “Hamiltonian systems and transformation in hilbert space,” *Proc. Natl. Acad. Sci. USA*, vol. 17, no. 5, pp. 315–318, 1931.
- [8] H. Harry Asada and F. E. Sotiropoulos, “Dual Faceted Linearization of Nonlinear Dynamical Systems Based on Physical Modeling Theory,” *J. Dyn. Syst. Meas. Control*, vol. 141, no. 2, 10 2018.
- [9] D. Lam, C. Manzie, and M. Good, “Model predictive contouring control,” in *49th IEEE Conference on Decision and Control (CDC)*, 2010, pp. 6137–6142.
- [10] M. Budišić, R. Mohr, and I. Mezić, “Applied koopmanism,” *Chaos*, vol. 22, no. 4, p. 047510, 2012.
- [11] I. Abraham, G. de la Torre, and T. Murphrey, “Model-based control using koopman operators,” *Robotics: Science and Systems XIII*, Jul 2017.
- [12] E. Kaiser, J. N. Kutz, and S. Brunton, “Data-driven discovery of koopman eigenfunctions for control,” *Mach. Learn.: Sci. Technol.*, March 2021.
- [13] M. Korda and I. Mezić, “Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control,” *Automatica*, vol. 93, pp. 149 – 160, 2018.
- [14] M. O. Williams, C. W. Rowley, and I. G. Kevrekidis, “A kernel-based method for data-driven koopman spectral analysis,” *J. Comput. Dyn.*, vol. 2, p. 247, 2015.
- [15] M. Korda and I. Mezić, “Optimal construction of koopman eigenfunctions for prediction and control,” *IEEE Trans. Autom. Contr.*, 2020.
- [16] Q. Li, F. Dietrich, E. M. Boltt, and I. G. Kevrekidis, “Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the koopman operator,” *Chaos*, vol. 27, no. 10, 2017.
- [17] B. Kramer and K. E. Willcox, “Nonlinear model order reduction via lifting transformations and proper orthogonal decomposition,” *AIAA Journal*, vol. 57, no. 6, pp. 2297–2307, 2019.
- [18] F. E. Sotiropoulos and H. H. Asada, “Causality in dual faceted linearization of nonlinear dynamical systems,” in *2018 Annual American Control Conference*, 2018, pp. 1230–1237.
- [19] D. C. Karnopp, D. L. Margolis, and R. C. Rosenberg, *System Dynamics: Modeling and Simulation of Mechatronic Systems*. New York: John Wiley and Sons, Inc., 2006.
- [20] C. Folkestad, D. Pastor, I. Mezić, R. Mohr, M. Fonoberova, and J. Burdick, “Extended dynamic mode decomposition with learned koopman eigenfunctions for prediction and control,” in *2020 American Control Conference (ACC)*, 2020, pp. 3906–3913.
- [21] Y. Igarashi, M. Yamakita, J. Ng, and H. H. Asada, “MPC performances for nonlinear systems using several linearization models,” in *2020 American Control Conference (ACC)*, 2020, pp. 2426–2431.
- [22] O. Khatib, “A unified approach for motion and force control of robot manipulators: The operational space formulation,” *IEEE J. Robot. Autom.*, vol. 3, no. 1, pp. 43–53, 1987.
- [23] G. J. Maeda, S. P. N. Singh, and D. C. Rye, “Improving operational space control of heavy manipulators via open-loop compensation,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 725–731.
- [24] G. J. Maeda, I. R. Manchester, and D. C. Rye, “Combined ILC and disturbance observer for the rejection of near-repetitive disturbances, with application to excavation,” *IEEE Trans. Contr. Syst. Technol.*, vol. 23, no. 5, pp. 1754–1769, Sept 2015.
- [25] M. Servin, T. Berglund, and S. Nystedt, “A multiscale model of terrain dynamics for real-time earthmoving simulation,” *Adv. Model. Simul. Eng. Sci.*, vol. 8, no. 1, p. 11, May 2021.
- [26] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, “OSQP: an operator splitting solver for quadratic programs,” *Math. Program. Comput.*, vol. 12, no. 4, pp. 637–672, 2020.
- [27] J. L. Proctor, S. L. Brunton, and J. N. Kutz, “Generalizing koopman theory to allow for inputs and control,” *SIAM J. Appl. Dyn. Syst.*, vol. 17, no. 1, pp. 909–930, 2018.
- [28] D. Jud, P. Leemann, S. Kerscher, and M. Hutter, “Autonomous free-form trenching using a walking excavator,” *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3208–3215, 2019.
- [29] K. Johnson, C. Creager, A. Izadnegahdar, S. Bauman, C. Gallo, and P. Abel, *Development of Field Excavator with Embedded Force Measurement*. ASCE, 2012, pp. 365–374.
- [30] H. H. Asada and F. E. Sotiropoulos, “Determining soil state and controlling equipment based on captured images,” 2020, US Patent 10,867,377.
- [31] G. Mamakoukas, I. Abraham, and T. Murphrey, “Learning data-driven stable koopman operators,” *ArXiv*, vol. abs/2005.04291, 2020.
- [32] V. Cibulka, T. Haniš, and M. Hromčík, “Data-driven identification of vehicle dynamics using koopman operator,” in *2019 22nd International Conference on Process Control (PC19)*, 2019, pp. 167–172.
- [33] F. A. Bender, S. Göltz, T. Bräunl, and O. Sawodny, “Modeling and offset-free model predictive control of a hydraulic mini excavator,” *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 4, pp. 1682–1694, 2017.
- [34] T. Tomatsu, K. Nonaka, K. Sekiguchi, and K. Suzuki, “Digging soil experiments for micro hydraulic excavators based on model predictive tracking control,” *J. Phys. Conf. Ser.*, vol. 744, p. 012067, sep 2016.
- [35] I. Abraham and T. D. Murphrey, “Active learning of dynamics for data-driven control using koopman operators,” *IEEE Transactions on Robotics*, vol. 35, no. 5, pp. 1071–1083, 2019.