

Data-driven Feedback Control of Lattice Structures with Localized Actuation and Sensing

Dominik Fischer, Loi Do, Miana Smith, and Jiří Zemánek

Abstract—Assembling lattices from discrete building blocks enables the composition of large, heterogeneous, and easily reconfigurable objects with desirable mass-to-stiffness ratios. This type of building system may also be referred to as a digital material, as it is constituted from discrete, error-correcting components. Researchers have demonstrated various active structures and even robotic systems that take advantage of the reconfigurable, mass-efficient properties of discrete lattice structures. However, the existing literature has predominantly used open-loop control strategies, limiting the performance of the presented systems. In this paper, we present a novel approach to feedback control of digital lattice structures, leveraging real-time measurements of the system dynamics. We introduce an actuated voxel which constitutes a novel means for actuation of lattice structures. Our control method is based on the Extended Dynamical Mode Decomposition algorithm in conjunction with the Linear Quadratic Regulator and the Koopman Model Predictive Control. The key advantage of our approach lies in its purely data-driven nature, without the need for any prior knowledge of a system's structure. We illustrate the developed method via real experiments with custom-built flexible lattice beam, showing its ability to accomplish various tasks even with minimal sensing and actuation resources. In particular, we address two problems: stabilization together with disturbance attenuation, and reference tracking.

Index Terms—lattice structures, digital materials, extended dynamic mode decomposition, Koopman model predictive control

I. INTRODUCTION

Lattices are periodic structures that aim to achieve high stiffness and strength to weight ratios. Natural systems have demonstrated many examples of lattice structures, such as in honeycomb, leaf structure, or bone tissue. Because of their desirable mechanical properties, researchers across multiple domains, such as robotics, civil engineering, and architecture, have been studying these materials. However, manufacturing large objects with complex internal structures remains challenging. The decomposition of a period lattice into repeated discrete blocks, or voxels (volumetric pixel), can help address this as it enables the assembly of arbitrarily sized structures from the voxel feedstock. Such structures are also referred to as *digital* structures, as they are composed of discrete elements instead of continuous matter. Arranging

building elements in repeating patterns to achieve unique and unconventional properties is also typical for metamaterials.

A digital material system is inherently modular, enabling easier reconfigurability and replicability as compared to a continuously fabricated structure. Additionally, the voxelized system simplifies the processes of both making changes and repairs to the system since voxels may easily be switched out of the structure or added in. This substantially improves the reparability of the voxel system, as a failure in one part of the structure only requires the affected voxel to be removed, as opposed to replacing the entire structure, or adding a patch which might impact structure performance. By combining voxels with different mechanical structures, the resulting structure can have very specific bulk mechanical properties, including pre-programmed anisotropies, such as relative compliance in only one axis. This type of heterogeneous structure is difficult to achieve using conventional approaches to fabrication, in which either a combination of material types or significant design experience would be required to achieve this level of mechanical property control. Additionally, unlike standard approaches to fabrication, the size of the final structure is not constrained by the size of the machine used to make it. Instead, the digital material enables the assembly of almost arbitrarily sized structures. The discrete structure also allows for automated (dis)assembly of these materials.

Digital structures have been predominantly studied in connection to their structural design [1], [2], [3], [4] and static behavior [5]. Active control of lattice structures has been previously explored for space applications, primarily for active damping, such as in [6] and comprehensive description of feed-forward and integral force feedback control of lattice-structured satellites was given in [7]. However, these works are not aimed at discrete lattice structures, limiting their extendibility. In recent years the focus has been shifting also towards studying and controlling the dynamic properties of metamaterials for larger deflections. In [8], the authors presented two structures composed by assembling voxel elements into a lattice: a morphing aircraft wing and an underwater snake-like robot. Active control was introduced into these structures allowing the wing to change shape and the robot to achieve swimming, nature-inspired motion, representing a modular approach to approximately continuum-style robotics. Among other fields where digital materials find relevance are soft robotics, production of wearable devices, prosthetics [9], as well as optimized electronic devices [10]. Possible applications of digital metamaterials extend even to the construction of space structures [11], [12],

This work was supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS22/166/OHK3/3T/13, and co-funded by the European Union under the project ROBOPROX (reg. No. CZ.02.01.01/00/22_008/0004590)

Dominik Fischer, Loi Do, and Jiří Zemánek are with Faculty of Electrical Engineering, Czech Technical University in Prague {fischdom, dolo, jiri.zemaneck}@fel.cvut.cz. Miana Smith is with Center for Bits and Atoms, Massachusetts Institute of Technology, Cambridge, MA, USA

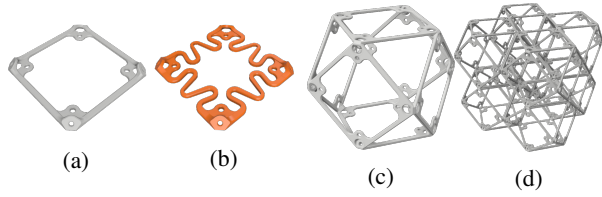


Fig. 1: (a) The rigid face, (b) the compliant face, an assembled voxel (c), and larger structure (d)

where autonomous structures with active shape control are of significant interest.

Our primary focus is on robotic structures akin to those presented in [8], where a voxel structure with specific anisotropies is actuated. These structures have so far been subjected to precomputed input sequences, leaving space for more advanced control algorithms to be explored and applied. In this paper, we present a novel mechanism to actuate digital lattice structures and develop a systematic method for feedback control synthesis. Controlling digital structures presents a challenge, primarily because it is difficult to obtain a dynamical model of a structure with many distinct building blocks. Furthermore, flexible structures naturally exhibit nonlinear dynamics. In light of that, we opted for a data-driven modeling approach based on the Extended dynamic mode decomposition (EDMD) algorithm [13]. Using the EDMD, we were able to obtain a linear predictor, allowing us to use standard linear control synthesis methods. We use Linear Quadratic Regulator (LQR) and Koopman Model Predictive Control (KMPC) to address the tasks of stabilization disturbance attenuation, and reference tracking. The process and results presented in this paper enable the solution of more complex tasks and expand the potential applications of digital lattice-based structures.

II. SYSTEM DESCRIPTION AND PROBLEM DEFINITION

A. Cuboctahedron as Building Block

We use a face-connected cuboctahedron lattice, i.e., a tiled polyhedron with eight triangular and six square faces. In particular, we adopt a construction design developed by the *Center for Bits and Atoms* at MIT published in [14]. A cuboctahedron voxel is assembled using six square faces connected at the corners. The faces are 3D-printed using PETG plastic on a commercial 3D printer, and are connected using rivets. Voxels are assembled together face-to-face with rivets.

By changing the geometry of each face, we can compose cuboctahedra with different mechanical properties. In Fig. 1, we show an example of two different face types, the assembled cuboctahedron, and a structure from multiple cuboctahedra. Because the voxels are made from plastic, the assembled structures have the potential to be highly flexible with many degrees of freedom.

B. Actuated Voxel

We developed a novel mechanism for inducing controlled motion in the overall structure. Compared to other actua-

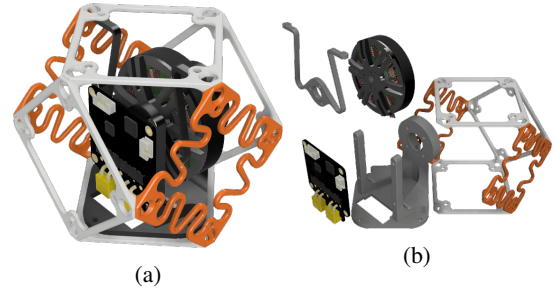


Fig. 2: Render of the actuated voxel: (a) assembled; (b) individual parts

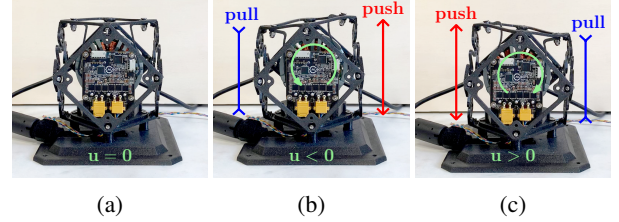


Fig. 3: Demonstration of the real actuated voxel (a) at rest; (b) tilted to the left; (c) tilted to the right

tion mechanisms for lattice structures (see Remark 1), our mechanism is contained within a voxel, making it fully compatible with the overall voxel assembly system. The actuation is based on deforming a voxel with an internally housed motor. The actuated voxel is displayed in Fig. 2a with its individual components in Fig. 2b. The housing voxel is created from two compliant faces and four rigid faces in a configuration that allows significant deformation along one axis and relative stiffness in the others. The deformation is induced by a motor that generates a torque which is then converted into a push-pull motion by a compliant transmitter that is connected to the top face of the voxel. This allows to directly deform specific places in the structure. The motor is a *mj5208* brushless DC motor controlled by the Moteus r4.11 driver by *mjbots*, which allows desired torque to be set directly. We illustrate the motion principle in Fig. 3, where we show three snapshots of the physical prototype.

Remark 1: One option to actuate voxel-lattice structures introduced in [8], [14] is using *tendons*, i.e., anchored wires, that span multiple voxels. The structure's motion is then induced by suitably shortening the tendons' lengths. Another option, frequently utilized in modular robotics, is to use a separate distinct module, an actuated *joint*, such as in [15], [16]. In contrast, the actuation mechanism presented in this paper is fully contained in a single voxel. Additionally, unlike a specific component such as actuated joint, the housing voxel does not differ from other building blocks, which simplifies the construction of the overall structure. See Fig. 4 for an illustration of differences between the mechanisms.

C. Sensing Voxel

For measuring the system's motion, we use an inertial measurement unit (IMU) inserted into a voxel. See Fig. 5 for

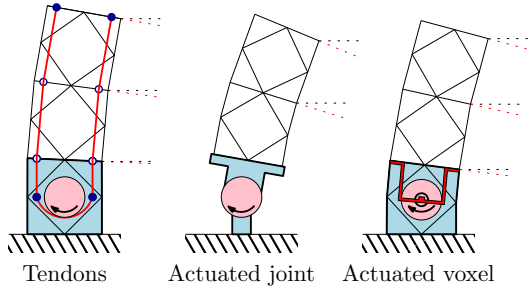


Fig. 4: Comparison of actuator types

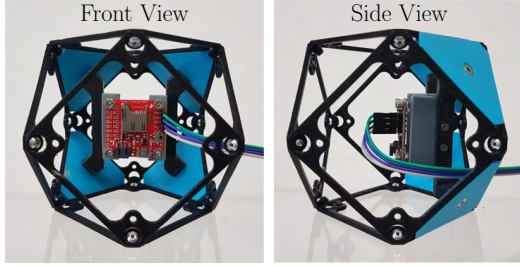


Fig. 5: Sensing Voxel

reference. Similar to the actuated voxel, the sensing is also self-contained within a single voxel. We created the housing voxel using only rigid faces, but different options could also be possible with minor changes to the construction. We used an MPU9250 IMU, which allows measurement of acceleration, angular speed, and estimated orientation along all three axes.

D. Problem Definition

Consider a general (arbitrary) voxel-lattice structure with a fixed arrangement of empty, actuated, and sensing voxels. We assume that the arrangement of the structure allows the input u to control the output y .

Let the structures' dynamics be in a form of a discrete-time nonlinear system

$$x_{k+1} = f(x_k, u_k), \quad (1a)$$

$$y_k = h(x_k), \quad (1b)$$

where $x \in \mathbb{R}^n$ is the system's state, $u \in \mathbb{R}^m$ is the control input, and $y_k \in \mathbb{R}^q$ is the system's measured output. Both $f(\cdot)$ and $h(\cdot)$ are unknown, generally nonlinear functions. The task is to design a feedback control for the system (1) to get a desirable behavior of the system's output y , e.g., disturbance attenuation or reference tracking.

III. LINEAR PREDICTOR BASED ON KOOPMAN OPERATOR THEORY

To address the defined problem, we employ a data-driven approach. We start by identifying a *predictor* for the dynamics (1). By predictor, we mean an artificial dynamical system designed to predict the original system's dynamics (or output) based on the initial state and the input sequence. We restrict the predictor to have a linear system structure, so it can be then used for control synthesis through linear

controller design methodologies. The data-driven approach is particularly well suited for digital structures as it provides a systematic way to identify the system's dynamics, regardless of the system's structure and placement of controllers or sensors. The identification is based only on supplying a suitable input signal into the system and measuring the outputs.

To obtain accurate predictions of the nonlinear dynamics using only a linear predictor, the main idea is to *lift* the nonlinear dynamics into a higher-dimensional state-space. With the original state $x \in \mathbb{R}^n$ of the nonlinear system, we consider the predictor in a form of a discrete-time linear system

$$\begin{aligned} z_{k+1} &= Az_k + Bu_k, \\ \hat{y}_k &= Cz_k, \end{aligned} \quad (2)$$

with \hat{y}_k being the prediction of the nonlinear system's output y_k , and $z_k \in \mathbb{R}^N$, $N \gg n$ being the *lifted* state. The concept of getting better prediction using lifting is supported by the *Koopman operator theory* which we briefly explain in the next section.

However, from the practical point-of-view, the Koopman operator theory only provides theoretical foundations for the state-space *lifting*. The main challenge is finding the matrices A, B and C that represent the nonlinear dynamics well, for which we use the Extended Dynamic Mode Decomposition (EDMD) algorithm.

A. Koopman Operator

We provide only a brief introduction to Koopman operator theory, as our main goal is to establish the context for the terminology used. For more rigorous exposition of Koopman operator theory, we refer the reader to [17] or [18]. The foundations of the Koopman operator for analyzing dynamical systems were established by the works of Koopman and von Neumann in the 1930s [19], [20]. We restrict ourselves to the case of autonomous systems. The extension to systems with external inputs can be found in [21].

Consider an autonomous discrete dynamical system

$$x_{k+1} = \mathcal{T}(x_k), \quad x_k \in \mathcal{M}, \quad (3)$$

where $\mathcal{T}(x_k)$ is a nonlinear transition mapping and \mathcal{M} is a state space. The core idea is to shift the focus from the mapping \mathcal{T} to some (possibly infinitely many) user-defined functions of the states, so-called *observables*. An observable is a function $\psi : \mathcal{M} \rightarrow \mathbb{R}$ which belongs to typically an infinitely dimensional vector space \mathcal{F} . The Koopman operator $\mathcal{K} : \mathcal{F} \rightarrow \mathcal{F}$ is then defined as

$$(\mathcal{K}\psi)(x_k) = \psi(\mathcal{T}(x_k)) = \psi(x_{k+1}), \quad (4)$$

Thus, the operator \mathcal{K} advances every observable ψ from the time step k to $k+1$. One of the fundamental properties of \mathcal{K} is its linearity since for any two observables ψ^1 and ψ^2 and scalars α and β the following holds

$$\mathcal{K}(\alpha\psi^1 + \beta\psi^2) = \alpha\mathcal{K}(\psi^1) + \beta\mathcal{K}(\psi^2). \quad (5)$$

Therefore, using the Koopman operator, one can convert the analysis of finite-dimensional nonlinear system into the analysis of the infinite-dimensional linear operator. Additionally, the Koopman operator fully captures the behavior of the nonlinear system, provided the space of observables contains the coordinate identity mappings. Linear predictors in the form (2) can then be viewed as finite-dimensional approximation to the Koopman operator.

B. Linear Predictor Identification

We identify the linear predictor (2) of the system's dynamics (1) using the approach described in [21]. The approach is based on the Extended Dynamic Mode Decomposition (EDMD) algorithm for controlled systems. The EDMD was first introduced in [22], and then generalized to the control setting in [21]. Full explanation and further information regarding the algorithm can be found in [23] or [18].

The process starts with gathering p data collections, the measurements and inputs (y^i, y_+^i, u^i) , $i = 1, \dots, p$ from the nonlinear system satisfying

$$\begin{aligned} y^i &= h(x_k), \\ y_+^i &= h(f(x_k, u^i)), \end{aligned} \quad (6)$$

where h and f are from (1). Therefore, y^i and y_+^i are temporally ordered but y^i and y^{i+1} are necessarily not, so the measurements can be from different trajectories. Then, we select s lifting mappings ψ^j – the observables – and combine all observables into a single vector function

$$\Psi(y) = [\psi^1(y), \psi^2(y), \dots, \psi^s(y)]^\top. \quad (7)$$

We arrange the data collections in the following manner

$$Y = [y^1, y^2, \dots, y^p], \quad (8a)$$

$$Y_+ = [y_+^1, y_+^2, \dots, y_+^p], \quad (8b)$$

$$Y^{\text{lift}} = [\Psi(y^1), \Psi(y^2), \dots, \Psi(y^p)], \quad (8c)$$

$$Y_+^{\text{lift}} = [\Psi(y_+^1), \Psi(y_+^2), \dots, \Psi(y_+^p)], \quad (8d)$$

$$\Omega = [u^1, u^2, \dots, u^p]. \quad (8e)$$

The linear predictor (2) can be then obtained by solving

$$\min_{A, B} \|Y_+^{\text{lift}} - AY^{\text{lift}} - B\Omega\|_F, \quad (9a)$$

$$\min_C \|Y - CY^{\text{lift}}\|_F, \quad (9b)$$

where $\|\cdot\|_F$ is the Frobenius norm. The analytical solution to (9) is

$$\begin{bmatrix} A & B \\ C & 0 \end{bmatrix} = \begin{bmatrix} Y_+^{\text{lift}} \\ Y \end{bmatrix} \begin{bmatrix} Y^{\text{lift}} \\ \Omega \end{bmatrix}^\top \left(\begin{bmatrix} Y^{\text{lift}} \\ \Omega \end{bmatrix} \begin{bmatrix} Y^{\text{lift}} \\ \Omega \end{bmatrix}^\top \right)^\dagger, \quad (10)$$

where $(\cdot)^\dagger$ denotes the Moore–Penrose pseudoinverse [2].

IV. CONTROL METHODS

Having the linear predictor (2), we can design a feedback controller via classical linear control systems method. We use two control techniques, the Linear Quadratic Regulator (LQR), and the Model Predictive Control (MPC).

A. Linear Quadratic Regulator

The discrete-time Linear Quadratic Regulator (LQR) is a controller that minimizes the quadratic cost function

$$J(u, z) = \sum_{k=0}^{\infty} (z_k^\top Q z_k + u_k^\top R u_k), \quad (11)$$

subjected to the dynamics of the system (2). The matrices $Q \succeq 0$ and $R \succ 0$ represent the penalties on states and input, respectively. The control law minimizing (11) is

$$u_k = -K z_k = -K \Psi(y_k), \quad (12)$$

where K can be computed from the solution to the discrete-time Algebraic Riccati Equation.

B. Model Predictive Control

Use of linear prediction in conjunction with model predictive control was first introduced in [21], and subsequently coined as *Koopman* MPC (KMPC). KMPC for control of dynamical systems has been previously used, such as in [24], [25].

Model Predictive Control (MPC) is a discrete-time optimal control algorithm that uses optimization to find input sequences that minimize a given cost function $J(\cdot)$ over defined time frame – the *prediction horizon* N_p . Upon computing the optimal input sequence, the controller then applies only the first element from the sequence, and the process is repeated in the next time step. With linear predictor of the system's dynamics, the optimization problem is a convex quadratic program for which many high-performance solvers exist. We use the OSQP solver [26].

We formulate the MPC for tracking. Let e_k be the tracking error and r_k the reference. At every time step t , the MPC solves an optimization problem

$$\begin{aligned} \min_{u_k, z_k} \quad & J\left(\{z_k\}_{k=0}^{N_p}, \{u_k\}_{k=0}^{N_p}\right), \\ \text{subject to} \quad & z_{k+1} = A z_k + B u_k, \quad k = 0, 1, \dots, N_p - 1, \\ & e_k = r_k - C z_k, \\ & u_{\min} \leq u_k \leq u_{\max}, \\ \text{parameters} \quad & z_0 = \Psi(y_t), \\ & r_k = \text{given}, \quad k = 0, 1, \dots, N_p, \end{aligned} \quad (13)$$

where u_{\min}/u_{\max} are bounds on inputs. After the optimal sequence u_k is found, only the first element u_0 is used, new measurement is obtained, and the process repeats in next time step $t + 1$.

The loss function $J(\cdot)$ in (13) has a standard quadratic form for an MPC tracking problem

$$J(z_k, u_k) = \frac{1}{2} e_{N_p}^\top S e_{N_p} + \frac{1}{2} \sum_{k=0}^{N_p-1} (e_k^\top Q e_k + u_k^\top R u_k). \quad (14)$$

Matrices $Q \succeq 0$ and $S \succeq 0$ penalize the error inside and the end of the prediction horizon, respectively, while $R \succeq 0$ penalizes the input.

Furthermore, to reduce computational complexity, and due to computational considerations (see Remark 2), we rewrite the quadratic problem (13) into a *dense* and *delta input* formulation. For the sake of brevity, we omit the full derivation and present only the resulting formulas; see [27] or other classical textbooks on MPC for full explanation. Let

$$\begin{aligned}\Delta u_k &= u_k - u_{k-1}, \\ \Delta U &= [\Delta u_0^\top, \Delta u_1^\top, \dots, \Delta u_{N_p-1}^\top]^\top, \\ r &= [r_0^\top, r_1^\top, \dots, r_{N_p-1}^\top]^\top.\end{aligned}\quad (15)$$

The final quadratic problem solved by MPC at each time step t is

$$\begin{aligned}\min_{\Delta U} \quad & \frac{1}{2} \Delta U^\top H \Delta U + [z_0^\top r^\top] F^\top \Delta U, \\ \text{s.t.} \quad & b_{\min} \leq G \Delta U \leq b_{\max},\end{aligned}\quad (16)$$

where the matrices H , F , G , and vectors b_{\min} , b_{\max} can be derived from the state (A, B, C) and cost (Q, R) matrices.

Remark 2: We use the dense formulation so the dimensionality of the optimization problem (16) does not scale with the number of observables (the size of the lifted state z). The delta input formulation allows achieving offset-free control in case of external disturbances acting on the system.

V. EXPERIMENTS

In this section, we present experiments with a lattice structure built from the voxels, the *Voxel tower*. The Voxel Tower is chosen as the experimental structure as it is a basic geometry applicable to a variety of different application, such as soft-robotic arms, limbs, or smart architecture. The structure, together with the scheme of its architecture, is depicted in Fig. 6. The tower's motion is controlled by the actuated voxel at the bottom, and the movement is measured by sensing voxel mounted halfway up the tower.

The structure consist of eight vertically connected voxels of two types. The structure is primarily composed of semi-compliant voxels that enable tilting in one axis, while being stiff in the other. We additionally used one rigid voxel that serves as the housing for the sensor.

A. Control Synthesis

To identify the predictor (2), we gathered the inputs and outputs from the real system. In total, we acquired $p = 4 \times 10^4$ measurements points from six trajectories. Each trajectory consists of supplying a different open-loop input and measuring the tilt ϕ and tilt speed $\dot{\phi}$.

As observables, we used *delay-embeddings* (time-shifted copies) of the measured output. The use of delay embeddings has been documented and justified in [28]. For the stabilization task V-B we used $s = 2$ delay embeddings, while the tracking predictor yielded better results with $s = 3$. Upon introducing $\psi = [\phi, \dot{\phi}, D\psi]^\top$, where $D^n b_k = b_{k-n}$ is the

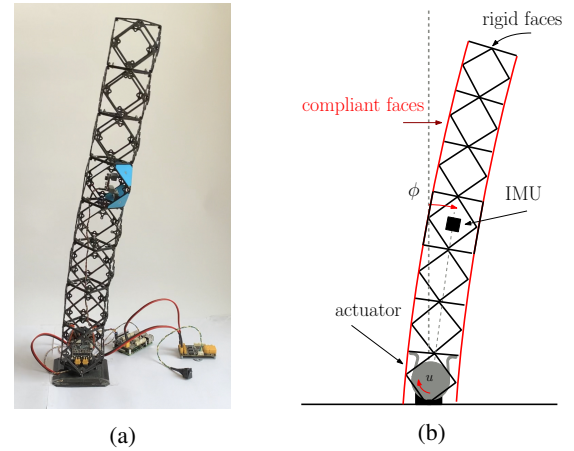


Fig. 6: The Voxel Tower: (a) a photo of the real prototype; (b) a graphic of the Voxel Tower system with its individual parts.

delay operator, we can write the particular lifting functions as

$$\Psi_{\text{stab.}}(\psi) = [D^2\psi, D\psi, \psi]^\top, \quad (17)$$

$$\Psi_{\text{trac.}}(\psi) = [D^3\psi, D^2\psi, D\psi, \psi]^\top. \quad (18)$$

The main tuning parameters for the design of the stabilizing LQR controller are the values of matrices Q and R in (11). Since penalizing delayed states or inputs is not meaningful, only the values of Q corresponding with the system's original states are non-zero and chosen to obtain the desired performance.

For the design of the KMPC controller, we still have the state and input penalty matrices Q and R at hand, but on top of that, there are two extra parameters, that we can use – the prediction horizon length N_p , and the final state penalty matrix S . We set the prediction horizon $N_p = 10$, and through a suitable choice of the matrix C in (13), we can again penalize only the original states ϕ and $\dot{\phi}$. The corresponding values in the Q and S matrices that result in the desired performance were chosen.

The KMPC gives us the option to bound the control action. We found the limit on the applied torque to be $\pm 0.5 \text{ N.m}$. This limit ensured the torque would not damage the actuated voxel, while still deforming the voxel enough to induce motion into the structure.

B. Stabilization

In the first testing scenario, presented in Fig. 7, we tilted the tower into an initial angle deviation: $\phi_0 = -20^\circ$ and compared the response in the controlled system to the uncontrolled system. The figure includes snapshots from the experiment. Even though the uncontrolled structure would also eventually stabilize due to damping, the controlled structure is able to stabilize much faster.

In the second scenario, presented in Fig. 8a, we tested the response in the case where neither of the state variables is zero. The figure compares the behavior of the controlled and

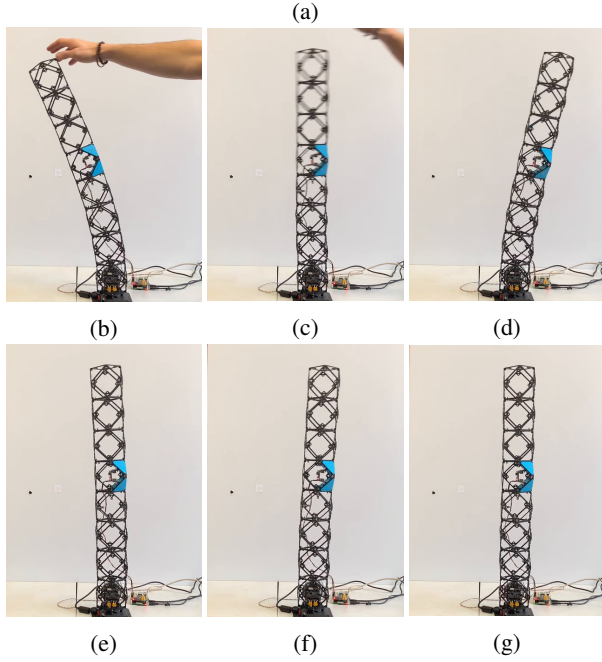
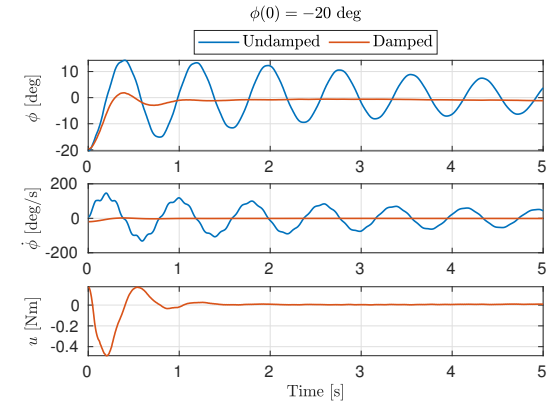


Fig. 7: (a) Response to the initial deviation $\phi_0 = -20^\circ$ from vertical position. Snapshots of the experiment with active damping: (b) $t = 0$ s; (c) $t = 0.2$ s; (d) $t = 0.5$ s; (e) $t = 0.8$ s; (f) $t = 1.0$ s; (g) $t = 1.4$ s.

uncontrolled systems. We first applied a precomputed sine wave torque excitation to the system (red area) to initiate movement. After 6 seconds, when the tower was already in motion, we switched to the damping control. Again, we can observe that the controlled system significantly outperforms the uncontrolled system in terms of settling time.

In the third scenario, presented in Fig. 8b, we applied two 0.5 s force pulses, simulating disturbances. The disturbance signal is shown in the bottom plot as a black dashed line. During the time of the disturbance (red areas), the motor acts as the disturbance. The figure also shows the response of the tower to the disturbance, comparing the behavior of the uncontrolled system with the controlled system. The measured results show that the designed controller is able to successfully dampen even higher-frequency oscillations caused by the abrupt disturbance.

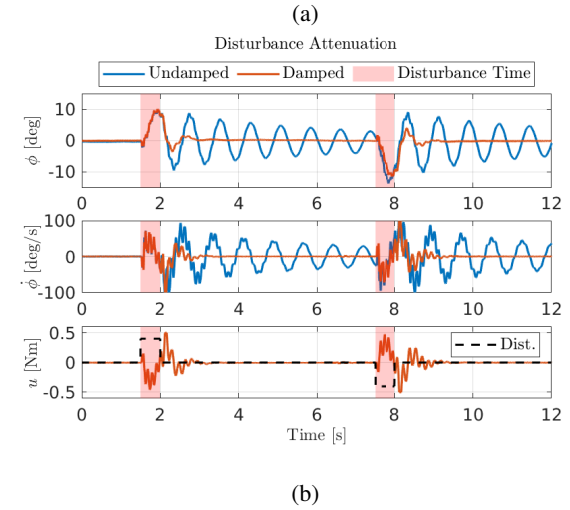
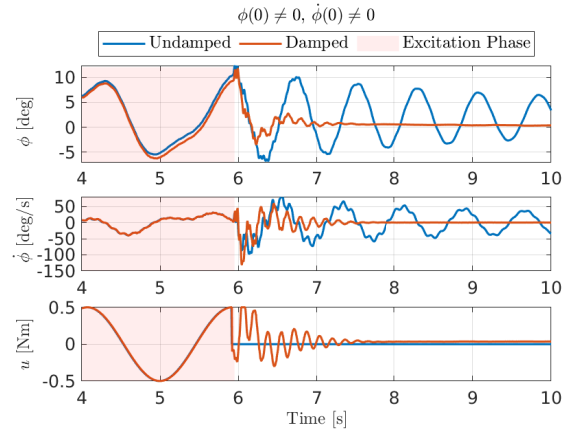


Fig. 8: Two experiments of stabilization

C. Tracking

To test the KMPC controller, we generated two reference trajectories. First, we required multiple consecutive step changes in the tilt angle. The second type of tested trajectories consist of gradual changes in the tilt angle. In both experiments, the reference velocity remained zero. This choice was made considering that the tower was intended to remain stationary or move slowly. We present the measured responses in Fig. 9. The outcomes of both experiment confirm that the KMPC is able to track the desired references and further confirms the efficiency of the developed method.

VI. CONCLUSION

This work introduced sensing and feedback control into lattice structures made from 3D printed cuboctahedron voxels. We presented a novel means for actuation, the *actuated* voxel, which uses local deformations to actuate a voxel structure. We build a structure from these digital materials—the Voxel Tower—and considered the problem of stabilization, disturbance attenuation, and reference tracking. To solve these tasks, we used the EDMD algorithm to obtain a linear predictor and implemented two regulators, the LQR and the Koopman MPC, and demonstrated their performance

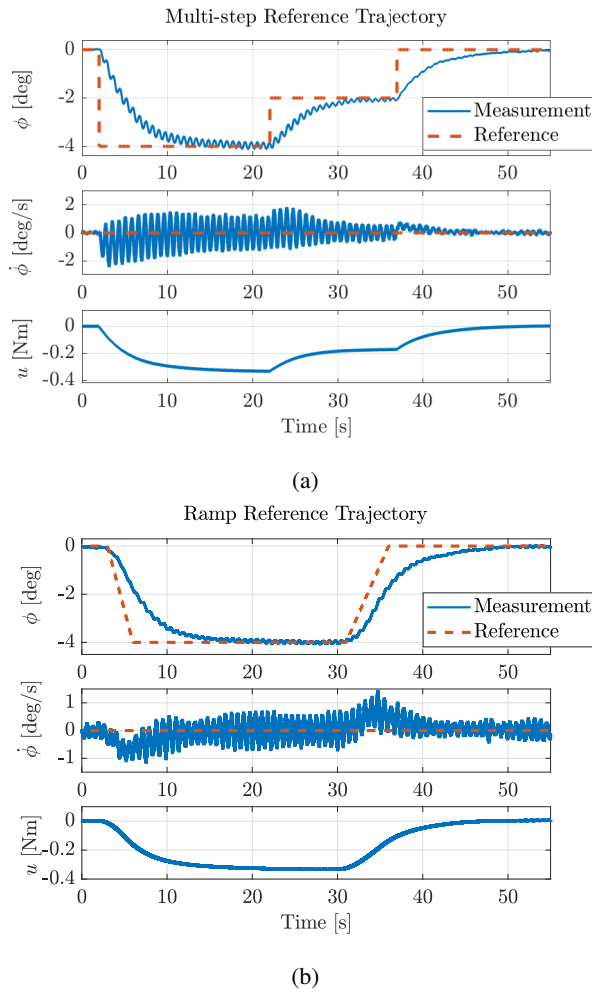


Fig. 9: Two experiments of reference tracking

on the physical system. The presented work expanded the capabilities of mechanical lattice structures beyond open-loop control and can serve as a stepping stone toward the utilization of digital structures in complex applications that demand more advanced control techniques. Future work will address controlling systems with more complex structure.

REFERENCES

- [1] Coulais C, Teomy E, de Reus K, Shokef Y, van Hecke M. Combinatorial Design of Textured Mechanical Metamaterials. *Nature*. 2016 Jul;535(7613):529–532.
- [2] Gregg CE, Kim JH, Cheung KC. Ultra-Light and Scalable Composite Lattice Materials. *Advanced Engineering Materials*. 2018;20(9):1800213.
- [3] Shaw LA, Sun F, Portela CM, Barranco RI, Greer JR, Hopkins JB. Computationally efficient design of directionally compliant metamaterials. *Nature Communications*. 2019 Jan;10(1):291.
- [4] Tyburec M, Doškář M, Zeman J, Kružík M. Modular-topology optimization of structures and mechanisms with free material design and clustering. *Computer Methods in Applied Mechanics and Engineering*. 2022 May;395:114977.
- [5] Jenett B, Cellucci D, Gregg C, Cheung K. Meso-Scale Digital Materials: Modular, Reconfigurable, Lattice-Based Structures. *American Society of Mechanical Engineers Digital Collection*; 2016. p. 1–11.
- [6] Preumont A, Dufour JP, Malekian C. Active damping by a local force feedback with piezoelectric actuators. *Journal of Guidance, Control, and Dynamics*. 1992;15(2):390–395.
- [7] Paupitz Goncalves PJ. Dynamic analysis and active control of lattice structures [Doctoral Thesis]. University of Southampton; 2007.
- [8] Parra Rubio A, Fan D, Jenett B, Del Águila Ferrandis J, Tourlomousis F, Abdel-Rahman A, et al. Modular Morphing Lattices for Large-Scale Underwater Continuum Robotic Structures. *Soft Robotics*. 2023 Aug;10(4):724–736.
- [9] Mirzaali MJ, Janbaz S, Strano M, Vergani L, Zadpoor A. Shape-matching soft mechanical metamaterials. *Scientific Reports*. 2018 Jan;8.
- [10] Langford W, Ghassaei A, Gershenfeld N. Automated Assembly of Electronic Digital Materials. In: *Volume 2: Materials; Biomanufacturing; Properties, Applications and Systems; Sustainable Manufacturing*. Blacksburg, Virginia, USA: American Society of Mechanical Engineers; 2016. p. V002T01A013.
- [11] Jenett B, Cheung K. BILL-E: Robotic Platform for Locomotion and Manipulation of Lightweight Space Structures. In: *25th AIAA/AHS Adaptive Structures Conference*. American Institute of Aeronautics and Astronautics; 2017. p. 1–12.
- [12] Gregg CE, Catanoso D, Formoso OIB, Kostitsyna I, Ochalek ME, Olatunde TJ, et al. Ultralight, strong, and self-reprogrammable mechanical metamaterials. *Science Robotics*. 2024;9(86):eadi2746.
- [13] Williams MO, Hemati MS, Dawson STM, Kevrekidis IG, Rowley CW. Extending Data-Driven Koopman Analysis to Actuated Systems. *IFAC-PapersOnLine*. 2016 Jan;49(18):704–709.
- [14] Jenett B, Cameron C, Tourlomousis F, Rubio AP, Ochalek M, Gershenfeld N. Discretely assembled mechanical metamaterials. *Science Advances*. 2020 Nov;6(47):eabc9943. Publisher: American Association for the Advancement of Science.
- [15] Smith MM. Recursive Robotic Assemblers. 2023 Jun.
- [16] Abdel-Rahman A, Cameron C, Jenett B, Smith M, Gershenfeld N. Self-replicating hierarchical modular robotic swarms. *Communications Engineering*. 2022 Nov;1(1):1–10.
- [17] Budišić M, Mohr RM, Mezić I. Applied Koopmanism. *Chaos: An Interdisciplinary Journal of Nonlinear Science*. 2012 Dec;22(4):047510.
- [18] Brunton SL, Budišić M, Kaiser E, Kutz JN. Modern Koopman Theory for Dynamical Systems. *SIAM Review*. 2022 May;64(2):229–340. Publisher: Society for Industrial and Applied Mathematics.
- [19] Koopman BO. Hamiltonian Systems and Transformation in Hilbert Space. *Proceedings of the National Academy of Sciences*. 1931 May;17(5):315–318.
- [20] Koopman BO, Neumann Jv. Dynamical Systems of Continuous Spectra. *Proceedings of the National Academy of Sciences*. 1932 Mar;18(3):255–263.
- [21] Korda M, Mezić I. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*. 2018 Jul;93:149–160.
- [22] Williams MO, Kevrekidis IG, Rowley CW. A Data-Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition. *Journal of Nonlinear Science*. 2015 Jun;25(6):1307–1346. Publisher: Springer Science and Business Media LLC.
- [23] Korda M, Mezić I. On Convergence of Extended Dynamic Mode Decomposition to the Koopman Operator. *Journal of Nonlinear Science*. 2018 Apr;28(2):687–710.
- [24] Do L, Korda M, Hurák Z. Controlled synchronization of coupled pendulums by Koopman Model Predictive Control. *Control Engineering Practice*. 2023 Oct;139:105629.
- [25] Arbabi H, Korda M, Mezić I. A Data-Driven Koopman Model Predictive Control Framework for Nonlinear Partial Differential Equations. In: *2018 IEEE Conference on Decision and Control (CDC)*; 2018. p. 6409–6414.
- [26] Stellato B, Banjac G, Goulart P, Bemporad A, Boyd S. OSQP: an operator splitting solver for quadratic programs. *Mathematical Programming Computation*. 2020 Dec;12(4):637–672.
- [27] Borrelli F, Bemporad A, Morari M. Predictive Control for Linear and Hybrid Systems. 1st ed. Cambridge New York Port Melbourne: Cambridge University Press; 2017.
- [28] Tu JH, Rowley CW, Luchtenburg DM, Brunton SL, Kutz JN. On dynamic mode decomposition: Theory and applications. *Journal of Computational Dynamics*. 2014 Nov;1(2):391–421.