



An Invariant Representation of Coupler Curves Using a Variational AutoEncoder: Application to Path Synthesis of Four-Bar Mechanisms

Anar Nurizada

Computer-Aided Design and Innovation Lab,
Department of Mechanical Engineering,
Stony Brook University,
Stony Brook, NY 11794-2300
e-mail: anar.nurizada@stonybrook.edu

Anurag Purwar¹

Computer-Aided Design and Innovation Lab,
Department of Mechanical Engineering,
Stony Brook University,
Stony Brook, NY 11794-2300
e-mail: anurag.purwar@stonybrook.edu

This paper focuses on the representation and synthesis of coupler curves of planar mechanisms using a deep neural network. While the path synthesis of planar mechanisms is not a new problem, the effective representation of coupler curves in the context of neural networks has not been fully explored. This study compares four commonly used features or representations of four-bar coupler curves: Fourier descriptors, wavelets, point coordinates, and images. The results demonstrate that these diverse representations can be unified using a generative AI framework called variational autoencoder (VAE). This study shows that a VAE can provide a standalone representation of a coupler curve, regardless of the input representation, and that the compact latent dimensions of the VAE can be used to describe coupler curves of four-bar linkages. Additionally, a new approach that utilizes a VAE in conjunction with a fully connected neural network to generate dimensional parameters of four-bar linkage mechanisms is proposed. This research presents a novel opportunity for the automated conceptual design of mechanisms for robots and machines.
[DOI: 10.1115/1.4063726]

Keywords: planar four-bar linkage, path synthesis, Fourier descriptors, wavelets, variational autoencoder, neural networks, machine learning, deep learning

1 Introduction

The challenge of path generation, which involves synthesis of linkage mechanisms, such as four-bar and higher-order ones, to follow a sequence of consecutive points (x_i, y_i) in R^2 , has been the subject of numerous studies, yielding a multitude of proposed solutions. Often these solutions hinge on optimization approaches. These methods represent the coupler point path by precision points and utilize an objective function to minimize the error between them. Despite their prevalent use, they suffer from significant drawbacks, including slow optimization, dependence on initial conditions, and lack of guaranteed results, often failing to capture the true shape of a given path [1].

The complexity of these mechanisms is further emphasized by the nonlinear relationships between input and dimensional parameters. Even minor changes to the input can lead to a significantly different mechanism. In the context of coupler-curve generation for a four-bar linkage synthesis, the problem appears to be defined but is often considered over-determined, leaving the synthesis problem without an analytical solution [2]. As the number of links in a mechanism grows beyond the simplicity of a four-bar linkage, it enables

the production of more complex motions, which makes the design of such linkages considerably more challenging compared to four-bar linkages [3].

Due to these reasons, there has been a growing interest in using neural networks (NNs) for mechanism synthesis. NNs are widely recognized for their ability to approximate a class of mappings defined in Euclidean space. This further translates to their application in learning a mapping from design specifications of mechanism design problems to the dimensional parameters of mechanisms. The use of NNs in mechanism synthesis is motivated by a desire to address fundamental issues that lack analytical or theoretical foundations, such as synthesizing circuit-, branch-, and order-defect-free mechanisms, translating user intent into well-defined problems, and satisfying additional kinematic and geometric constraints.

Variational autoencoders (VAEs), a class of generative deep neural network models, offer the potential to approximate these nonlinear relationships more effectively. Unlike conventional optimization methods, VAEs can provide a comprehensive representation of the entire path shape, offering numerous stable solutions. Once trained, VAEs eliminate the need for initial guesses, generate approximations from defect-free datasets, and provide a more robust response to minor input changes. Consequently, VAEs present a promising avenue for the synthesis of closed-loop linkage mechanisms, extending the potential for designing more complex systems with greater accuracy and efficiency.

¹Corresponding author.

Manuscript received February 11, 2023; final manuscript received October 6, 2023; published online October 27, 2023. Assoc. Editor: Stephen Canfield.

A key challenge in the use of NNs for mechanism synthesis is the representation of mechanisms and their properties, which form a non-Euclidean space. In the case of planar four-bar mechanisms, a crucial property is the coupler curve generated by a floating link. To date, most researchers using NNs have utilized a Euclidean embedding of the input path, leading to various representations of the input coupler curves, including point coordinates, Fourier descriptors, wavelets, and 2D pixel representations in an image. Of these, Fourier and wavelet representations are derived from the point coordinates only, and therefore, they can be considered features instead. However, for simplicity, we will forego this distinction and call them representations only for the purpose of this paper. Despite these representations having produced effective outcomes, there have been no studies investigating the relative merits of different representations and their impact on linkage mechanism synthesis. This lack of knowledge has impeded the field's progress in using modern machine learning algorithms for mechanism synthesis.

Apart from a comparative analysis of the representations, this paper also presents a machine learning approach for the generation of four-bar linkages that approximate a desired coupler curve. The approach uses a VAE and a fully connected neural network (FCNN) to generate a multitude of possible linkages. Four widely utilized representations, namely, Fourier descriptors, wavelets, point coordinates, and image-based representations, were investigated.

The input (desired) coupler curve is normalized to make it invariant with respect to translation, rotation, and scale. Each of the four representations is then fed to their respective pre-trained VAEs, which map the input curve to a latent space. The k -nearest neighbor (k -NN) [4] search in the latent space yields k similar coupler curves represented as latent points. These k latent representations are input to a fully connected neural network to generate k mechanisms. This is a major difference between the previous approaches which would only yield a single mechanism for a desired coupler curve. The use of the VAE allows us to cluster similar looking coupler curves together, thus, providing us with several mechanisms approximating the input curve. This process is illustrated in Fig. 1.

The generated coupler curves were compared to the desired coupler curves by computing the mean square error (MSE). The representation with the lowest MSE was determined to be the

most optimal representation. It is important to point out that the initial starting points of the mechanisms compared are the same, thus, there is a one-to-one mapping between the points of two curves compared. In case the initial starting points of the two curves would be different, the MSE for two exactly similar curves would be high. In those cases, it would be necessary to consider other methods of curve comparison, such as Frechet distance [5], dynamic time warping [6], etc.

The results demonstrate that all of the representations yield comparable outcomes, with the MSE values obtained from the wavelet and point coordinates-based approaches being the lowest and similar to each other. Although the Fourier- and image-based approaches yielded higher MSE values, the generated linkages still provided a reasonable approximation of the input coupler curves. The MSE served as a useful and meaningful metric because the compared curves were closed and subsequently sampled and parameterized identically.

The results also indicate that the VAE can serve as a standalone representation of a coupler curve and that a 5D latent space of the VAE is sufficient to describe a coupler curve of a four-bar linkage. It is crucial to emphasize that there is no inherent relationship between the latent space and the actual characteristics of the coupler curves. This gives rise to the possibility of using latent space as the invariant description of coupler curves, which normalizes several different representations used in the literature.

The results of this study suggest that all of the representations generate several mechanisms that approximate the input coupler curve well and produce comparable outcomes. The similarities among the mechanisms generated using different representations indicate that the latent space of a VAE can be used as an invariant representation of a coupler curve. The k NN search in the latent space, which led to similar curves and their mechanisms, provides the evidence that this mapping is locally Euclidean.

This paper also explores the effects of linear interpolation between two random curve latent representations. Previous research [7] has shown that reconstructing the interpolated latent representations directly from a VAE can result in unrealistic artifacts. In contrast, this study demonstrates that utilizing a FCNN in conjunction with a VAE can overcome this problem by ignoring the decoder and

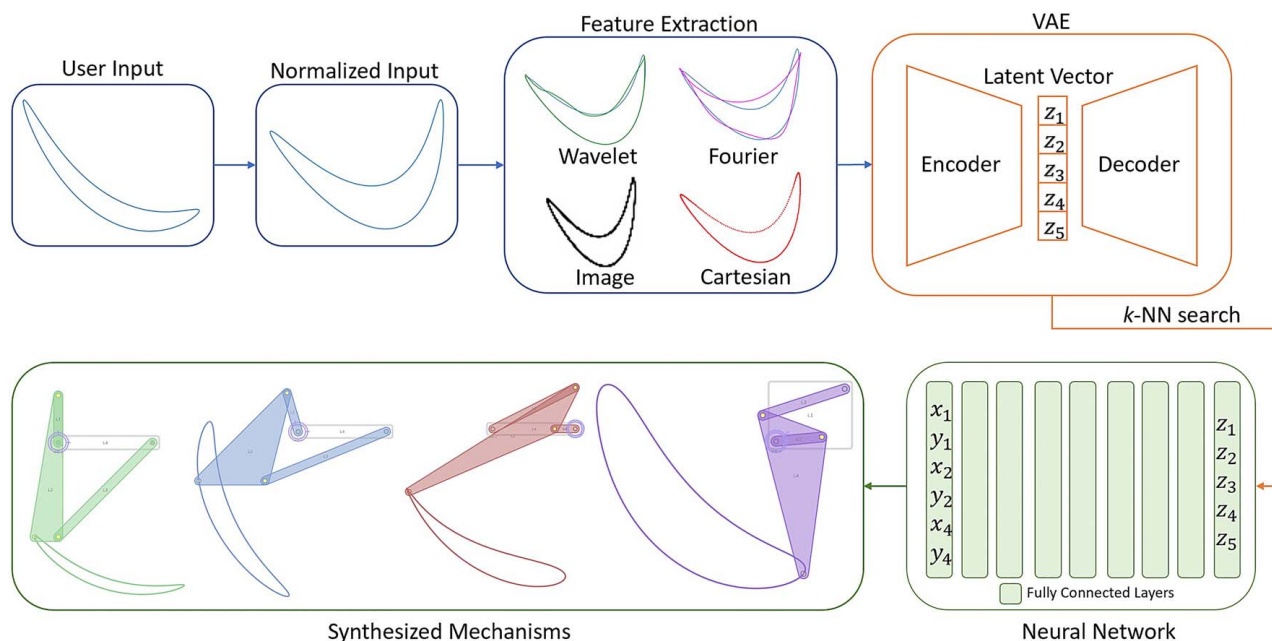


Fig. 1 An input curve consisting out of 360 (x, y) points is normalized, i.e., translated, scaled, and rotated. Once normalized, one of the representations, for example wavelet descriptors, is computed and fed into the trained VAE which maps it to its latent space. Performing k -NN search in the latent space yields k latent vectors of coupler curves similar to the desired coupler curve. Taking these latent representations through a fully connected neural network yields k mechanisms that approximate the desired input coupler curves. The output of the NN is a vector of unknown joint coordinates of mechanisms.

producing coupler curves that smoothly morph from one input coupler curve to another, resulting in mechanisms that transform from one to another without singularity.

This work makes two key contributions to the field. First, it investigates and compares effective representations of four-bar coupler curves and demonstrates that a normalized representation using a VAE can simplify storage and computation. Second, it proposes a combined VAE-FCNN architecture that learns an effective mapping from the task space (path) to the mechanism (dimensional parameters) and produces four-bar mechanisms by sampling and interpolating in the latent space of the VAE. Although this paper primarily focuses on the most widely used planar four-bar mechanisms, the approach presented can be extended to high-order single-degree-of-freedom mechanisms.

The remainder of the paper is organized as follows. First, the existing literature is reviewed in light of the goals of this study. The planar four-bar mechanism dataset and normalization techniques are presented in Sec. 3. Mathematical fundamentals of Fourier descriptors, wavelets, VAEs, and t-distributed stochastic neighbor embedding (t-SNE) are reviewed in Sec. 4, while the architecture of the neural networks used in this study is presented in Sec. 5. Finally, results of the study are presented via several examples in Sec. 6, followed by a conclusion section.

2 Neural Network Literature for Mechanism Synthesis

In the field of path synthesis and representation of coupler curves in four-bar mechanisms, several key works have been published. In this section, we present a review of these works, focusing on those relevant to the development of this paper.

Vermeer et al. [8] introduce a novel approach that combines reinforcement learning techniques with neural network-based policies and reward functions to achieve optimal link lengths for straight-line mechanism synthesis tasks. Fogelson et al. [9] graph convolution policy for high-order linkage graph optimization is an algorithm that utilizes machine learning techniques, such as hierarchical optimization and high-order linkage construction, to effectively generate feasible paths in complex environments.

Unruh and Krishnaswami developed a computer-aided design algorithm for synthesizing a four-bar linkage that best approximates a given closed trajectory with an infinite number of points [10]. The algorithm utilizes B-splines to store a large number of coupler curves in a database and an algorithm for fitting B-splines to closed curves. In contrast, Mcgarva and Mullineux used Fourier descriptors (FDs) to represent closed curves [11]. The authors normalized low-order coefficients to eliminate the difference between two curves that are either translated, rotated, or scaled versions of each other. Their results suggest that a curve can be represented by the fundamental and the first harmonic terms of FDs. Mcgarva later provided an algorithm for searching a catalog of coupler curves represented as FDs to find the best match to a desired input [12].

Ullah and Kota also used FDs for optimal synthesis of mechanisms for the path generation problem [13]. They introduced an objective function that finds the difference between two curves. Once the function is minimized, it provides an approximation to a candidate curve. However, FDs require the curve to be a closed one. Wu et al. devised a method for extending the FD approach to open curves by incorporating finite Fourier series in a curve-fitting scheme for approximation of periodic as well as non-periodic functions [14]. Li et al. extended the use of FDs for the motion generation problem, in which the coordinates (x_i , y_i) and orientation (θ_i) of the path points are given [15]. The authors show that the rotational component of a motion in combination with the translational part is enough to determine all of the necessary components of a four-bar linkage for approximate motion generation.

Sharma et al. exploited the relationship between the path and orientation of coupler of four-bar mechanisms to devise a solution to

the Alt-Burmester problem, which is essentially a mixed path points and position problem [16]. Leveraging this relationship, they translated this problem into a pure motion generation problem. Another drawback of using FD approach is its dependence on the time parametrization. This is usually overcome by considering the uniform parametrization of the data points, which in turn usually disregards the uniqueness of harmonic properties of a given coupler curve. Sharma and Purwar introduced a nonuniform parametrization that considers the harmonic properties of a coupler curve of a given four-bar linkage and allows imposing additional user-specified constraints [17].

Vasiliu and Yannou presented a method for synthesizing the parameters of four-bar linkages for path generation by combining Fourier descriptor feature extraction and machine learning [1]. This approach demonstrated promising results. The procedure involved first calculating Fourier descriptors from a desired closed coupler curve, which were then fed into a neural network that was trained to map the descriptors to bar linkage parameters such as the link coordinates in the initial position of the mechanism.

Similarly, Khan et al. proposed a method for determining the dimensions of a four-bar mechanism by analyzing the shape of its coupler curve [18]. The shape was represented using Fourier descriptors of cumulative angular deviation, which eliminated the need to consider the position or scale of the curve. An artificial neural network was trained to learn the relationship between these Fourier descriptors and the dimensions of the mechanism.

The use of Fourier descriptors as shape descriptors for representing curves is not the only approach available. Another family of shape descriptors, known as wavelet descriptors (WDs), has been employed in a variety of engineering disciplines. Chuang and Kuo were among the first researchers to explore the use of WDs, demonstrating the capability to decompose a closed planar curve into a hierarchy of scales that contain important information about the curve's features [19]. The study revealed that some of the descriptors capture the global features of the curve, while others focus on its more detailed aspects. Osowski and Nghia conducted a comparative study between Fourier descriptors and wavelet descriptors in the extraction of 2D pattern features, with their results suggesting that the wavelet approach outperforms the Fourier approach in situations where the input is noisy [20]. Nabout and Tibken applied WDs in object shape recognition, using puzzle-shaped animals as the input shapes [21–23]. The recognition process involved calculating the WDs for the input shapes and comparing them with the previously stored WDs of contour patterns.

Liu et al. presented a technique for the representation of open curves utilizing wavelets [24,25]. They proposed a path generator that considers the WDs extracted from the curve, rather than the curve itself, and outputs the actual parameters of a four-bar mechanism that approximates the input curve. Li et al. used wavelets and neural networks for flaw classification [26]. The 2D flaw shape was transformed into wavelet descriptors and utilized as the input to a neural network for classification. The study compared the performance of wavelet descriptors, Fourier descriptors, and principal component analysis [27] as feature extractors. The results indicated that the wavelet descriptors outperformed the other methods, particularly when the input was noisy, which was in accordance with the findings of Osowski and Nghia [20].

Galan-Marin et al. developed a pipeline for synthesizing the path of crank-rocker mechanisms through the use of wavelet-based neural networks [28]. Similar to the work done by Li et al. [26], the wavelet descriptors were extracted from a given coupler curve and utilized as feature vectors in the neural network to determine the parameters of the mechanism. A comparison was made between the use of Fourier descriptors and wavelet descriptors as feature extractors and the latter was found to produce superior results.

The previously discussed methods have demonstrated promising results; however, they are limited in their ability to provide design

solutions. Deshpande and Purwar addressed this issue by introducing an approach that leverages a pre-compiled database of four- and six-bar linkage parameters and a convolutional neural network-based deep generative machine learning model [29]. During the training process, the coupler-curve images of mechanisms stored in the database serve as inputs, and the NN learns the probabilistic distribution of the input data.

In another study, Deshpande and Purwar utilized the curvature integral as a signature for the prescribed path and motion [30]. These signatures were then compiled in a database. The k -NN algorithm was employed to find suitable mechanisms that would satisfy the requirements. The authors also made use of a conditional VAE that incorporates uncertain user input to provide the user with greater control over the design process, as described in Ref. [31]. Regenwetter et al. conducted a comprehensive review of the use of deep generative machine learning models in engineering design [32]. They examined several types of deep generative models that have been successful in engineering design applications, including structural optimization, materials design, and shape synthesis. More recently, in a review paper, Purwar and Chakraborty [33] highlighted the recent advances in deep neural networks for design of robot mechanisms and outlined future directions for research in this area.

3 Dataset Compilation and Normalization Procedure

One of the major challenges facing the engineering design community in the utilization of data-driven methods, specifically deep learning, is the scarcity of large, publicly accessible datasets. This presents a significant hindrance not only for kinematic design but also for other branches of engineering design. Nobari et al. [34] address the problem by introducing LINKS, a comprehensive dataset comprising a hundred million planar linkage mechanisms, created to support data-driven approaches in kinematic design. This dataset serves as a valuable resource enabling researchers to conduct extensive experimentation and analysis in the field of kinematic design.

Despite the comprehensiveness of the dataset, conducting numerous experiments utilizing it would necessitate significant time and computational resources. Therefore, in order to address this challenge, we have devised an algorithm for generating planar four-bar mechanisms, represented as a sequence of five points (joints). We imposed that the mechanisms in the dataset satisfy the Grashof condition [35], have reasonable link ratios, and produce a diverse set of coupler curves [36]. Computational experiments showed that a 9×9 grid with the joint locations at one of the grid points was optimal in providing a sufficiently large number of mechanisms while also ensuring a wide range of coupler curves.

A four-bar mechanism consists of four joints, two of which are fixed and two are moving joints, and a coupler point. These joint locations can be collected in a tensor as $\mathbf{c} = (j_0, j_1, j_2, j_3, j_4)$; $j_i = (x_i, y_i)$, where j_0 and j_3 are fixed joints' locations, j_1 and j_2 are free moving pivots connected to j_0 and j_3 , respectively, and, finally j_4 represents the coupler point. It was decided to keep the fixed link of unit length for all mechanisms by setting $j_0 = (0.0, 0.0)$ and $j_3 = (1.0, 0.0)$. Since the coupler curves' shape remains unchanged as long as the link ratios are the same, this choice did not exclude any mechanisms and kept the computation and storage more manageable. By varying the positions of the moving pivots and the coupler point on the grid, a large dataset of four-bar joint combinations was compiled satisfying the following conditions:

- (1) If two or more joints' locations were at the same grid spot, this mechanism would be discarded. For example, if locations of two moving pivots are the same, it would create a structure without relative movement of links.
- (2) To make sure that the dataset does not consist of mechanisms with the same link ratios, we introduced the following criteria. Denoting the ground link length as l_0 , l_1 as the input link length connecting j_0 and j_1 , l_2 as the coupler link length

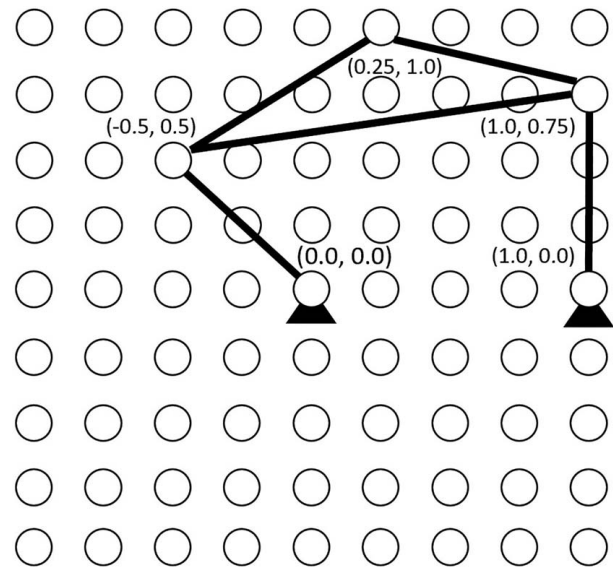


Fig. 2 An example combination of joints' locations. Each joint's location is shown in the brackets next to the joint. For each combination, moving pivots and coupler points can be at any point denoted with a circle, as long as the combination passes the filtering rules.

connecting j_1 and j_2 , and l_3 as the output link connecting j_2 and j_3 , we check that no two mechanisms have equal ratios ($l_1/l_0, l_2/l_0, l_3/l_0$).

- (3) Removing all nonviable and repetitive mechanisms, we perform the last filtering step removing all of the mechanisms not satisfying the Grashof condition.

Performing the aforementioned steps, we were able to compile 43,572 combinations of joint locations. Supplying each combination to our coupler-curve generator that takes in the information of the joints' locations, and outputs 360 (x_i, y_i) points on a coupler curve, we transformed these combinations into a dataset with information about these points and joints' locations. Checking the extreme coordinates corresponding to the normalized coupler curves from the database, we get that 99% of the generated coupler curves fit in a seven by seven unit square. Further examination of the dataset showed that the minimum link ratio of the bar mechanisms is equal to 1.1, whereas the maximum link ratio is 8.9, with most of the mechanisms' link ratios being between two and five. Figure 2 shows one example of the combinations. Fixed joints j_0 and j_3 are at locations $(0, 0)$ and $(1, 0)$, respectively. Moving pivots j_1 and j_2 are at locations $(-0.5, 0.5)$ and $(1, 0.75)$, whereas the coupler point is at location $(0.25, 1.0)$.

The compiled dataset was later partitioned into randomly selected training and testing sets, with proportions of 90% and 10%, respectively. Both sets exhibit similarity to one another since four-bar mechanisms do not generate significantly diverse coupler curves. Access to the dataset is available.² Upon closer examination of the dataset, it becomes evident that there exist curves that bear a striking resemblance to each other. Consequently, the training and testing datasets do not exhibit complete dissimilarity. The trained weights corresponding to each representation pipeline can be acquired upon request.

3.1 Normalization. In this work, each coupler curve is normalized to make sure it is invariant with respect to translation, scale, and rotation. The normalization procedure starts with ensuring that the curve is centered around $(0, 0)$ by subtracting the

²<https://www.kaggle.com/datasets/anamurizada/four-bar-coupler-curve-dataset>

mean $(\bar{x}, \bar{y})$ from each coupler point (x_i, y_i) . Scaling consists of dividing the coupler-curve points by the root mean squared variance in both X - and Y -directions. Normalization of the orientation requires finding the principal component axes. The principal component axes are eigenvectors of the covariance matrix C of a coupler path comprising of m two-dimensional points $\{x_j, y_j\}_{j=1}^m$, where C is given by

$$C = \begin{bmatrix} C_{xx} & C_{xy} \\ C_{yx} & C_{yy} \end{bmatrix} \quad \text{where} \quad (1)$$

$$C_{xx} = \frac{1}{m} \sum_{i=1}^m (x_i - \bar{x})(x_i - \bar{x}) \quad (2)$$

$$C_{yy} = \frac{1}{m} \sum_{i=1}^m (y_i - \bar{y})(y_i - \bar{y}) \quad (3)$$

$$C_{xy} = C_{yx} = \frac{1}{m} \sum_{i=1}^m (x_i - \bar{x})(y_i - \bar{y}) \quad (4)$$

To achieve full normalization of the coupler curve, the authors employed a rotation technique. This rotation involves aligning the principal component axes of the curve with the Cartesian coordinate system. By doing so, the curve is transformed to a standardized orientation. However, it is crucial to note that no normalization procedure was applied to address mirrored mechanisms and their associated coupler curves. In other words, mirrored mechanisms were treated as distinct entities, and their respective coupler curves were mapped to separate locations within the latent space.

4 Mathematical Fundamentals of Fourier Descriptor, Wavelet Descriptor, Variational Autoencoder, and t-Distributed Stochastic Neighbor Embedding

In this section, we review background material covering FDs, WDs, VAE, and t-SNE to the extent necessary for the development of this paper.

4.1 Fourier Descriptors. Fourier descriptors (FDs) are a common way of representing the shape of a closed curve at varying levels of detail. Regarding X - and Y -coordinates of a coupler curve as real and imaginary numbers, respectively, we can represent them as a complex function:

$$z(t) = x(t) + iy(t) \quad (5)$$

For a closed path curve, $z(t)$ is periodic and can be rewritten in a form of sum of series as follows:

$$z(t) = \sum_{m=-\infty}^{\infty} a_m e^{2\pi i m t} \quad (6)$$

where a_m is the Fourier descriptor and, in the case when a given curve is not a function but a sequence of points on the curve, can be calculated using the following approximation:

$$a_m = \frac{1}{N} \sum_{k=0}^{N-1} z_k e^{-2\pi i m (k/N)} \quad (7)$$

where z_k are the given points in their complex representation and N is the total number of points given. Descriptor a_0 is called the fundamental, whereas descriptors a_{-1} and a_1 are called first harmonic, etc. In other words, there can be an endless number of FDs, however, it has been shown that a task curve with $m=5$ captures the four-bar coupler path very accurately and is deemed sufficient for practical implementation [15]. In this work, we have also used a coupler-curve representation with $m=5$. In addition to that, two

more Fourier descriptor representations with $m=10$ and $m=20$ were also tested to see whether they yielded better results for mechanism generation. This choice was made to balance the input data load across different representations.

4.2 Wavelet Descriptors. Wavelets are mathematical functions that are used to analyze and represent data, particularly in the field of signal processing. They are based on the idea of analyzing a signal at different scales or resolutions, and are particularly useful for analyzing data with sudden changes or singularities. The most common type of wavelet is the Daubechies wavelet, named after Ingrid Daubechies [37], who first introduced them in 1988. The Daubechies wavelets are defined by

$$\psi_{a,b}(x) = \frac{1}{\sqrt{a}} \sum_{n=-\infty}^{\infty} h(n) e^{2\pi i ((nx-b)/a)} \quad (8)$$

where $h(n)$ is the wavelet coefficient, a is the scaling factor, and b is the translation factor. The Daubechies wavelets are characterized by the number of vanishing moments, which is the number of times the integral of the wavelet multiplied by a polynomial of a certain degree is zero. Daubechies wavelets are widely used in various fields such as image compression [38], audio processing [39], and medical imaging [40]. They are effective in capturing the important information in a signal while discarding the noise.

The formula for the wavelet coefficient in the Daubechies wavelet is given by the sequence $h(n)$, which is a parameter in the wavelet function definition. It can also be computed from the scaling function coefficients using the equation:

$$h(n) = (-1)^n \sum_{k=0}^n (-1)^k n k f(k) \quad (9)$$

where $f(k)$ is the sequence of scaling function coefficients, and n is the index of the wavelet coefficient. The computation of wavelet coefficient is typically done as part of the wavelet transform process, which decomposes a signal into its different frequency components using the wavelet function.

It is worth noting that the wavelet coefficient is different from the wavelet function, which is the function that represents the wavelet. While the wavelet function gives us a way to visualize the wavelet, the wavelet coefficients gives us a way to quantitatively represent the wavelet in the form of a sequence of numbers.

We treated x and y locations of the points on a coupler curve as two separate signals and performed five-level decomposition using Daubechies wavelets on them separately. Similar to the Fourier descriptor representation, we wanted to check how discarding coefficients affects the coupler-curve approximation; and thus, we chose three different representations with 38, 76, and 136 wavelet coefficients describing our initial input. We selected a representation of 38 coefficients as the initial approximation of the coupler curve, as it was determined to be the minimum number required to accurately capture the necessary details without significant loss of information.

4.3 Variational Autoencoder. VAE is a modified version of the traditional autoencoder [41] architecture, which comprises three essential components: an encoder, a latent space, and a decoder. The utilization of autoencoder architectures for dimensionality reduction has been well established in the literature [42]. The encoder component of VAE comprises a series of convolutional and/or fully connected layers, which are stacked on top of each other. The encoder retains only the key features of the input in the latent space and the resulting encoded vector comprises of the most salient information about the input. The decoder component of VAE comprises of transposed convolutional and/or fully connected layers, which take the latent space vector as input and transform it back to the original input. However, as it is impossible to retain all the information while keeping the dimensionality of the

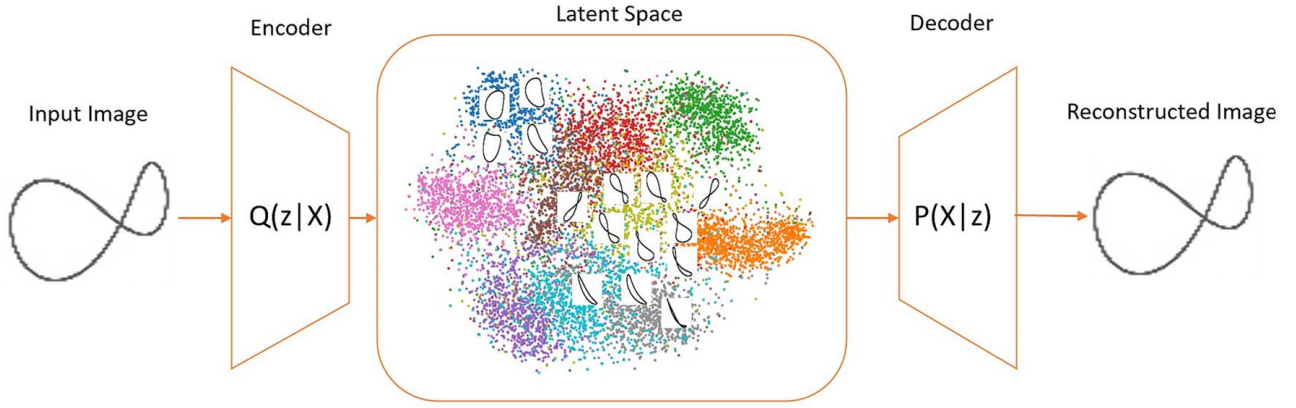


Fig. 3 VAE takes an input image of a coupler curve and maps it to a 2D latent space representation through the use of the encoder and t-SNE. This latent space shows that similar coupler curves are grouped together. By passing the latent representation of the input image through the decoder, the original input image can be reconstructed.

latent space much smaller than the input, the output tends to be lossy. It is worth noting that the latent space of an autoencoder is a single vector, in contrast to a VAE, thus, it is impossible to generate new data with it.

VAE differs from traditional autoencoder in that it approximates the probability distribution of the true distribution of the training data, rather than producing a single latent vector. VAE utilizes a mean and a standard deviation to generate multiple latent vectors by sampling from them. This allows for the generation of new data by interpolating in the latent space or by varying the mean and standard deviation values. This is a key feature of VAE, as it allows for the generation of diverse and realistic samples, which can be used for various applications such as image synthesis, image completion, and anomaly detection.

A VAE is trained by maximizing the expectation lower bound (ELBo) of the marginal likelihood, which is given by

$$\mathbb{E}_{\text{ELBo}} = \mathbb{E}_{Q(z|x^i; \theta_e)}(\log(p(x^i|z))) - D_{\text{KL}}(Q(z|x^i; \theta_e) \| p(z)) \quad (10)$$

where x^i represents a training image and z represents the vector sampled from the latent space. The first term on the right side of Eq. (10) represents reconstruction error and the second term is called the Kullback–Leibler divergence (KL divergence) [43] which ensures that our learned distribution $Q(z|x; \theta_e)$ is similar to the true prior distribution $p(z)$, which in our case was chosen to be a Gaussian distribution. Representing the reconstruction error as cross-entropy of original and reconstructed images, we can rewrite the first term of the right side of Eq. (10) as

$$\mathbb{E}_{Q(z|x^i; \theta_e)}(\log(p(x^i|z))) = x \log(\hat{x}) + (x - 1) \log(1 - \hat{x}) \quad (11)$$

where $\hat{x}$ is a reconstructed image vector. Since we assume that $p(z)$ is a Gaussian distribution, we can rewrite the second term of the right side of Eq. (10) as

$$D_{\text{KL}}(Q(z|x^i; \theta_e) \| p(z)) = \sum_i^k (\sigma_i^2 + \mu_i^2 - \log(\sigma_i) - 1) \quad (12)$$

where μ_i and σ_i are mean and standard deviation values from the latent space.

The latent space of a trained VAE clusters similar looking curves with each other. Figure 3 shows how analogous curves are mapped close to each other in the latent space. In other words, given a desired coupler curve, it is possible to map it to the latent space and take several neighboring latent representations to get several coupler curves that would approximate the desired coupler curve k -NN [4] algorithm was implemented to find k closest latent points given a latent representation of a coupler curve by calculating the Euclidean distance between the input coupler-curve latent

representation and all latent points available. It is important to mention that the results of the k -NN search are supplied to the second neural network in the pipeline and not the decoder since the decoder would reconstruct the input information rather than provide us with a mechanism capable of approximating the input curve.

4.4 t-Distributed Stochastic Neighbor Embedding. Visualizing high-dimensional data is important in order to understand how the VAE would cluster the given data in the latent space. The latent space dimensionality chosen for this work was equal to five; thus, it is mandatory to use a dimensionality reduction technique, such as t-SNE [44], to visualize the data in 2D or 3D maps.

The t-SNE technique consists of three major steps: (1) calculating a joint probability distribution that determines neighboring points, (2) calculating a joint probability distribution of a dataset of points created in the target dimension, and (3) using gradient descent to make these two joint probabilities as close as possible.

The conditional probability of a point x_i being a neighbor of x_j is given by

$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2 / 2\sigma_i^2)} \quad (13)$$

where σ_i is a standard deviation of a Gaussian distribution centered at x_i .

Gradient descent in t-SNE is performed by minimizing the KL divergence given by

$$\text{cost} = \text{KL}(P \| Q) = \sum_i \sum_j p_{i,j} \log \frac{p_{i,j}}{q_{i,j}} \quad (14)$$

where P is a joint probability distribution of the higher-dimensional space, whereas Q is a joint probability distribution of the lower-dimensional space that we would like to decrease our dimension to.

5 Neural Network Architectures of VAE-FCNN

In this section, we outline architectures for the VAE and the FCNN for each of the four representations. Due to the differences in the input vector and their cardinality, we trained a different VAE and FCNN for each representation. We tested and compared 13 different parameterizations of a coupler curve as follows: (1) Fourier descriptors with 5, 10, and 20 fundamentals, (2) wavelet descriptors with 38, 76, and 136 coefficients, (3) coordinates with 15, 23, 45, 90, 180, and 360 points on a curve, and (4) 64×64 image-based representation. The architecture of VAE for Fourier-, wavelet-, and point coordinates-based representations were the

Table 1 Fully connected neural network architecture

Layer (activation)	Input neuron #	Output neuron #
Dense (ReLU)	5	1280
Dense (ReLU)	1280	1280
Dense (ReLU)	1280	1280
Dense (ReLU)	1280	1280
Dense (ReLU)	1280	1280
Dense (ReLU)	1280	1280
Dense (ReLU)	1280	1280
Dense (ReLU)	1280	1280
Dense (ReLU)	1280	1280
Dense (Tanh)	1280	6

same as well except for the input and output layers since their dimension are a function of the chosen representation; for example, for a wavelet representation with 38 coefficients, the input and output layers have 38 neurons. The architecture of the FCNN was exactly the same for all of the representations since the input and output in all the cases were the same. Table 1 shows the architecture of the FCNN with rectified linear unit (ReLU) activation functions, where the input layer is 5D because the latent space of the VAE is 5D, while the output is a 6D layer, which represents the locations of three unknown joints of a mechanism.

In this study, the architecture for the image-based VAE differed from the other representations as it employed convolutional layers instead of fully connected layers. The architecture of the VAEs used for image and other representations are presented in Tables 2 and 3, respectively. In an effort to maintain comparability, the neural network architectures were kept as similar as possible. The latent dimensions for all VAEs were set to 5. All neural networks were trained for 200 epochs. The training loss for the image-based VAE composed of two components: reconstruction loss, which measures the difference between the reconstructed image and the original image, and the KL divergence loss, which compares the predicted distribution to a Gaussian distribution. The point coordinates-, Fourier-, and wavelet-based VAEs utilized flattened vectors for their input and output layers, with MSE as the reconstruction loss. Conversely, the image representation utilized binary cross-entropy as the loss function. The loss function for the fully connected neural network was set to MSE, which compares the predicted values to the actual values.

6 Results and Discussion

Figure 1 illustrates the overall approach, which we explain via a chosen representation, say FDs. A desired coupler curve consisting out of 360 (x, y) points is normalized following the steps described

Table 2 Image-based VAE architecture

Layer	Filter count	Filter size
Convolution 1	32	(4, 4)
Convolution 2	64	(4, 4)
Convolution 3	128	(4, 4)
Convolution 4	256	(4, 4)
Flatten 1	—	—
Mean—std. dev.	—	—
Sampling 1	—	—
Fully connected 1	—	—
Un-flatten 1	—	—
Transpose convolution 1	128	(5, 5)
Transpose convolution 2	64	(6, 6)
Transpose convolution 3	32	(6, 6)
Transpose convolution 4	1	(6, 6)

Table 3 Fourier-, wavelet-, and (x, y) point coordinates-based VAE architecture

Layer (activation)	Input neuron #	Output neuron #
<i>Encoder</i>		
Dense (ReLU)	^a	512
Dense (ReLU)	512	512
Dense (ReLU)	512	512
Dense (ReLU)	512	512
Dense (ReLU)	512	512
Dense (ReLU)	512	5
<i>Decoder</i>		
Dense (ReLU)	512	512
Dense (ReLU)	512	512
Dense (ReLU)	512	512
Dense (ReLU)	512	512
Dense (ReLU)	512	512
Dense (—)	512	^a

^aIndicates the variable size of the layer dependent on the representation.

in Sec. 3. Once normalized, Fourier descriptors are calculated and fed into the trained VAE which maps it to a latent representation. Performing k -NN search in the latent space yields k latent vectors of coupler curves similar to the input curve. Taking these latent representations through the fully connected neural network yields k mechanisms that approximate the desired input coupler curve. In case the wavelet representation is used instead of the Fourier descriptors, the only change would be in the input size to the VAE since there are a different number of wavelets coefficients being used than the FDs.

In order to compare the approaches using different representations, we measured the MSE between the input curve and the output curve obtained at the very end of our pipeline. Neural network outputs the locations of the moving joints, and knowing the locations of the fixed joints, since they were the same for all of the mechanisms, this is j_0 and j_3 , located at (0, 0) and (1, 0), it is possible to generate the final (output) coupler curve. The same generator was used to get the input and output curves; thus, there is a one-to-one mapping between the curves. Therefore, it is possible to use the MSE loss to compare the input with the output. A lower MSE implies better performance of a particular representation.

There were two kinds of error calculations performed. In the first calculation, for each input curve only one output curve was generated. In other words, k -NN search was not performed in this case. Five thousand random mechanisms from the dataset were chosen and run through each representation pipeline. The average MSE loss for each representation is shown in column “MSE loss” in Table 4. The second kind of error calculation involved the k -NN search, where for each input coupler curve, ten approximate output coupler curves were generated. Taking the average of these ten coupler curves, and averaging these losses for 5000 mechanisms, we get the results presented in column “ k -MSE loss” in Table 4. Table 4 also shows the number of parameters used in the whole pipeline. The higher the input number, the more parameters are needed since the input and output layers’ neurons’ number depends on the input number.

Average k -MSE results are higher than the respective MSE loss results as shown in Table 4, which is expected since the MSE loss is now averaged over a set of neighbors, which are at least some distance away from the input curve. If a larger number of neighbors are sampled, the MSE loss will be larger resulting in a worse approximation. A different number of Fourier descriptors as well as wavelet descriptors were chosen to see how they affect the output coupler curve approximation. The results suggest that increasing the number of the descriptors does not result in a better coupler curve generated since loss values fluctuate around the same number. Image-based approach showed the worst results

Table 4 Average MSE losses ($\pm 0.25\%$) for 5000 random mechanisms

Representation	Input #	Parameters	MSE	k -MSE	Noisy-MSE
Fourier descriptors	22	15,265,318	0.1175	0.4847	0.3285
	42	15,285,818	0.1337	0.4636	0.3196
	82	15,326,818	0.1481	0.4690	0.3397
Wavelet descriptors	38	15,281,718	0.1759	0.2384	0.2390
	76	15,320,668	0.1554	0.2333	0.2168
	136	15,382,168	0.1300	0.2092	0.2107
(x, y) points	720	15,980,768	0.0789	0.1892	0.1702
	360	15,611,768	0.1115	0.2191	0.1969
	180	15,427,268	0.0817	0.2004	0.2099
	90	15,335,018	0.0854	0.2322	0.1927
	46	15,289,918	0.0918	0.2062	0.2508
	30	15,273,518	0.0818	0.1741	0.2399
Image	4096	17,441,135	0.3850	1.1583	0.3760

having the highest MSE losses in both cases compared to the other three representations. This can also be attributed to the specific choice of the architecture rather than to the representation. Wavelet descriptors and point coordinates (x, y) approaches with different number of points on the coupler curve performed approximately the same when taking ten neighbors into account. Fourier descriptors representation had worse results when considering ten neighbors instead of one-to-one comparison. The results also suggest that not all 360 (x, y) points on the coupler curve are needed in order to get a good output since both MSE and k -MSE values fluctuated around 0.09 and 0.18, respectively. This suggest that one does not need too many points on the curve in order to get a good approximation.

Figure 4 shows four different input curves together with four output curves generated using a Fourier descriptor approach with five fundamentals, a wavelet approach with 38 descriptors, 360 (x, y) points approach, and 64×64 image-based approach. Also shown in the figure are the respective four-bar mechanisms that generate these coupler curves. The marker locations are related to the fixed joints which are the same for all of the mechanisms. These results suggest that our approach generates similar mechanisms regardless of the chosen representation.

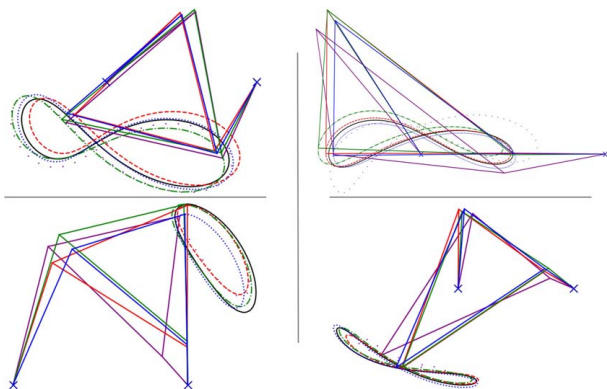


Fig. 4 Four different input coupler curves together with their corresponding curves generated using a Fourier descriptor approach with five fundamentals, a wavelet approach with 38 descriptors, 360 (x, y) points, and image-based approach. Input curve—solid curve; Fourier representation—dotted curve; wavelet representation—dashed-dotted curve; (x, y) representation—dashed curve; image representation—loosely dotted curve

Figures 5–8 show an input (black) curve and output curves generated through our pipeline using 360 (x, y) points-, wavelet-, Fourier-, and image-based approaches. While the k -MSE loss calculations used $k = 10$, we are showing only five ($k = 5$) nearest neighbor output curves as plotting all of them would have made the figure busy. The results shown are consistent with the numerical loss results presented; i.e., the 360 (x, y) points- and wavelet-based approaches perform equally well, whereas mechanisms obtained using the Fourier- and image-based representations quickly become worse. In Figs. 5 and 6, the input (black) curve can hardly be seen because approximations are so accurate that the input curve gets covered. It can also be seen that the higher the k -number, the further is the coupler curve from the best approximation. Choosing the representation with the lowest MSE values—point coordinates representation—we plot nine possible solutions to an input coupler curve using k -NN search together with the respective four-bar linkages in Fig. 9. It can be seen that the best result (top-left) gives the best approximation for the moving pivots' locations. The further the neighboring solution, the worse the approximation becomes.

Figures 10–13 show the results of a linear interpolation $\mathbf{L}(t) = (1 - t)\mathbf{L}_0 + t\mathbf{L}_1$, $t \in [0, 1]$, between the latent representations of

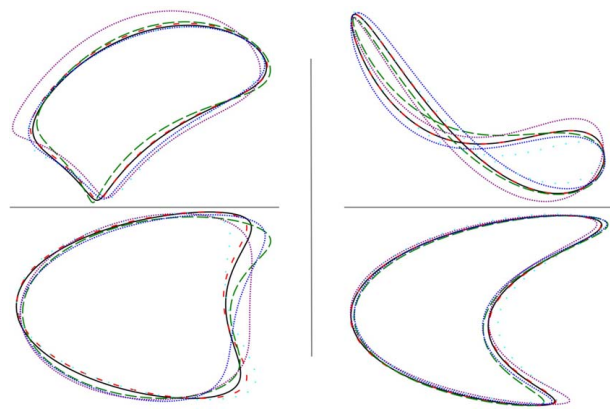


Fig. 5 Input (black) curve approximation obtained using the 360 (x, y) point-based approximation with $k = 5$. $k = 1$ —loosely dotted curve; $k = 2$ —dotted curve; $k = 3$ —densely dotted curve; $k = 4$ —long dashed curve; $k = 5$ —loosely dashed curve. Average MSE value: top-left—0.245, top right—0.152, bottom left—0.348, bottom-right—0.096.

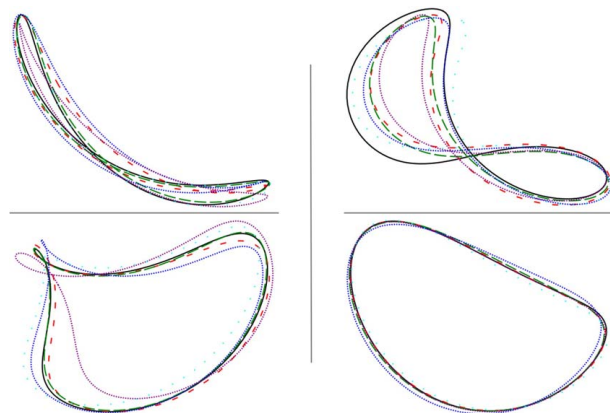


Fig. 6 Input (black) curve approximation obtained using the 38 wavelet coefficients-based approximation with $k = 5$. $k = 1$ —loosely dotted curve; $k = 2$ —dotted curve; $k = 3$ —densely dotted curve; $k = 4$ —long dashed curve; $k = 5$ —loosely dashed curve. Average MSE value: top-left—0.142, top right—0.169, bottom left—0.309, bottom-right—0.238.

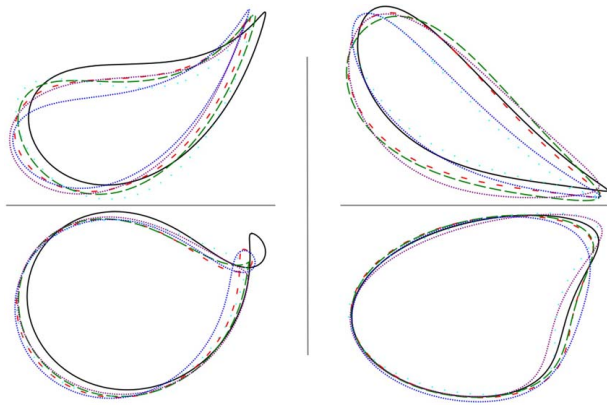


Fig. 7 Input (black) curve approximation obtained using the Fourier-based (five fundamentals) approximation with $k=5$. $k=1$ —loosely dotted curve; $k=2$ —dotted curve; $k=3$ —densely dotted curve; $k=4$ —long dashed curve; $k=5$ —loosely dashed curve. Average MSE value: top-left—0.209, top right—0.262, bottom left—0.207, bottom-right—0.157.

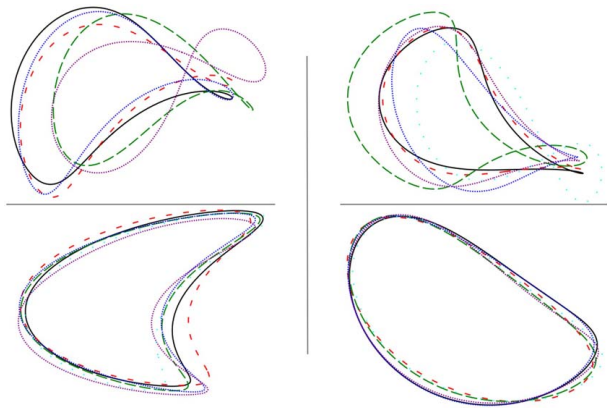


Fig. 8 Input (solid) curve approximation obtained using the image-based approximation with $k=5$. $k=1$ —loosely dotted curve; $k=2$ —dotted curve; $k=3$ —densely dotted curve; $k=4$ —long dashed curve; $k=5$ —loosely dashed curve. Average MSE value: top-left—0.510, top right—0.537, bottom left—0.429, bottom-right—0.347.

top-left and bottom-right coupler curves, given by latent vectors $\mathbf{L}_0$ and $\mathbf{L}_1$, respectively. It can be seen that the interpolation in the latent space and then mapping to the FCNN yield well-behaved transitions of mechanisms.

We also investigated the performance of our pipeline when presented with a noisy curve. To accomplish this, Gaussian noise with a mean of 0 and variance of 0.1 was introduced to the input and 5000 random mechanisms were subsequently processed through each representation pipeline. The results represented as the average MSE loss for each noisy representation are displayed in the column “noisy-MSE loss” in Table 4. Additionally, Fig. 14 illustrates the impact of noise on four input coupler curves, with comparison to coupler curves generated using a Fourier descriptor approach with five fundamentals, a wavelet approach with 38 descriptors, a 360 (x, y) points approach, and an image-based approach, along with the corresponding linkage mechanisms. It is worth noting that the noise introduced for the purpose of generating this figure has a variance of 0.01, rather than 0.1 used for loss calculation, as higher levels of variance resulted in a cluttered visual representation.

The findings indicate that the FD descriptor method exhibits inferior performance compared to the wavelet-based method when

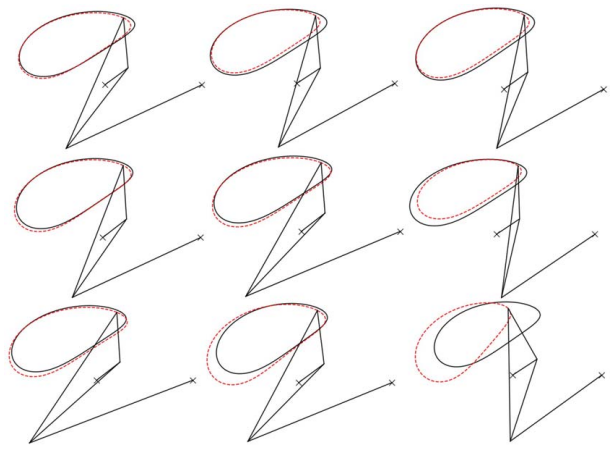


Fig. 9 Nine approximations (dashed line) of an input coupler curve (solid line) using a 360 (x, y) points-based approach obtained by sampling nine neighboring latent representations in the latent space. The best approximation is shown in the top left corner and the worst (farthest neighbor) approximation is shown in the bottom-right corner of the figure. The results show us that the approach provides several mechanisms that approximate the input well.

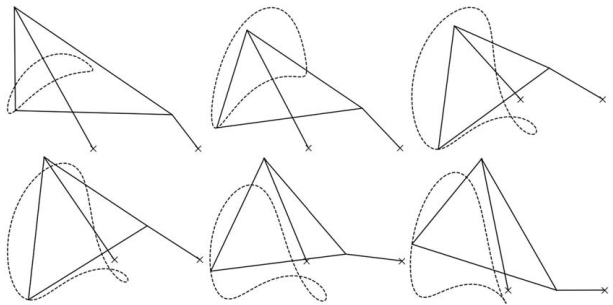


Fig. 10 Linear interpolation between the two coupler curves from top-left to bottom-right using 360 (x, y) points-based representation

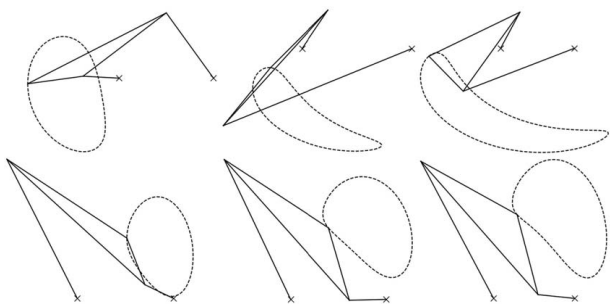


Fig. 11 Linear interpolation between the two coupler curves from top-left to bottom-right using 38 wavelet coefficient-based representation

exposed to a noisy input. This aligns with the outcomes reported in Ref. [20]. Conversely, both point-based and wavelet-based representations yield poorer results when presented with a non-smooth coupler curve, but they still demonstrate the most favorable outcomes. It is noteworthy that the image-based representation performs equally well in both scenarios: with a smooth curve and a non-smooth curve.

6.1 Falsification. The final stage of the study shows the results obtained by testing the pipeline on the coupler curves

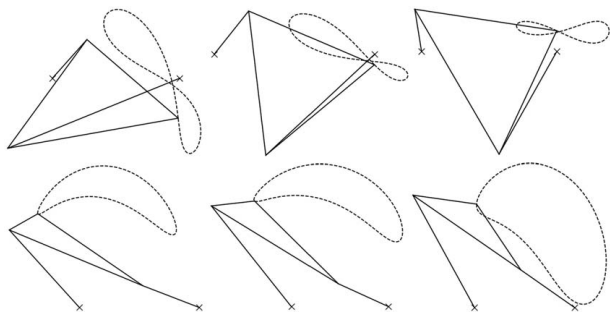


Fig. 12 Linear interpolation between the two coupler curves from top-left to bottom-right using Fourier descriptor-based representation with five fundamentals

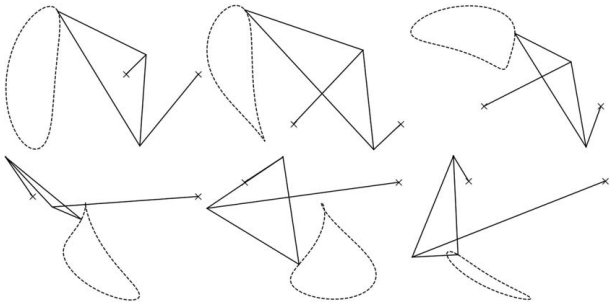


Fig. 13 Linear interpolation between the two coupler curves from top-left to bottom-right using image-based representation

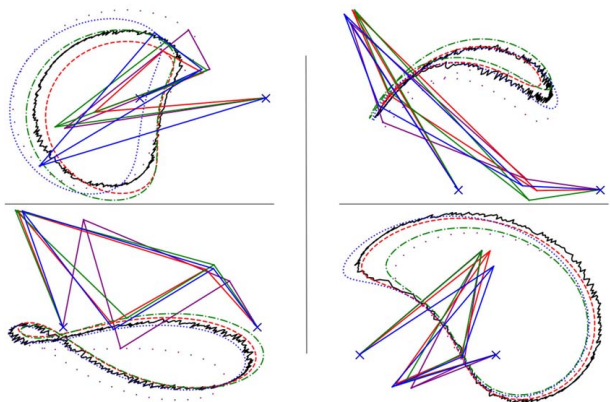


Fig. 14 Four different noisy input couple curves together with four coupler curves generated using a Fourier descriptor approach with five fundamentals, a wavelet approach with 38 descriptors, 360 (x, y) points approach, and image-based approach. Input curve—solid curve; Fourier representation—dotted curve; wavelet representation—dashed-dotted curve; (x, y) representation—dashed curve; image representation—loosely dotted curve.

that are substantially different from the ones the pipeline was trained on, i.e., the coupler curves that cannot be achieved using a four-bar mechanism. Figure 15 shows four different coupler curves in black color that are significantly different from the training and testing sets. The outcome shows us that the pipeline produces unsatisfactory results, which was anticipated, as the dataset did not consist of these types of curves during training. Some of the produced results are open curves although the training set consisted of Grashof mechanisms only. This happens because the predictions of the neural network might result in a joints' location of a non-Grashof mechanism.

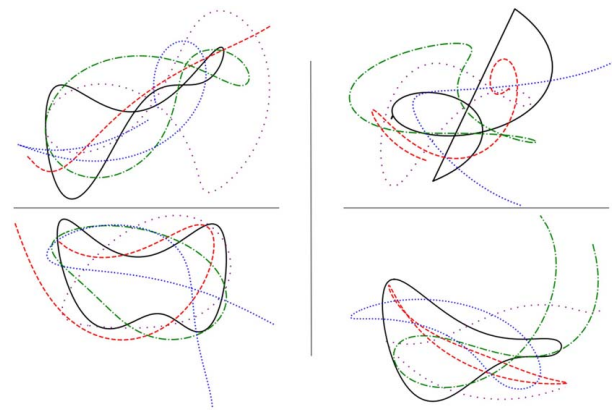


Fig. 15 Four different noisy input couple curves together with four coupler curves generated using a Fourier descriptor approach with five fundamentals, a wavelet approach with 38 descriptors, 360 (x, y) points approach, and image-based approach. Input curve—solid curve; Fourier representation—dotted curve; wavelet representation—dashed-dotted curve; (x, y) representation—dashed curve; image representation—loosely dotted curve.

7 Conclusions and Future Work

In this paper, we presented a novel methodology for generating planar four-bar mechanisms that approximate input couple curves. Four different representations of the coupler curves were analyzed, including Fourier, point coordinates, wavelets, and image-based. The findings of the study indicated that the wavelet and point coordinates representations produced the best approximations with only slight differences between the input and output curves. However, the Fourier and image-based representations resulted in higher errors. Nevertheless, all of the representations resulted in acceptable approximations, indicating that the latent space of the VAE could be used as an invariant representation of coupler curves.

Moreover, the study also tackled the issue of linear interpolation artifacts in the latent space by proposing a smooth interpolation solution. As a result, the study provides valuable insights for future research in this field. Possible avenues for future research may include exploring the use of the proposed pipeline for different types of linkage systems, such as six-bar, eight-bar, and spatial linkages.

Acknowledgment

This publication has research support from a National Science Foundation STTR phase II award (#2126882) to Co-PI Purwar who also holds stocks in Mechanismic Inc. The research findings included in this publication may or may not necessarily relate to the interests of Mechanismic Inc. The terms of this arrangement have been reviewed and approved by Stony Brook University in accordance with its policy on objectivity in research.

Data Availability Statement

The data and information that support the findings of this article are freely available.³

References

- [1] Vasiliu, A., and Yannou, B., 2001, "Dimensional Synthesis of Planar Mechanisms Using Neural Networks: Application to Path Generator Linkages," *Mech. Mach. Theory*, **36**(2), pp. 299–310.

³See Note 2.

- [2] Bai, S., and Angeles, J., 2015, "Coupler-Curve Synthesis of Four-Bar Linkages Via a Novel Formulation," *Mech. Mach. Theory*, **94**(3–4), pp. 177–187.
- [3] Plecnik, M. M., 2015, "The Kinematic Design of Six-Bar Linkages Using Polynomial Homotopy Continuation," Ph.D. thesis, University of California, Irvine, CA.
- [4] Taunk, K., De, S., Verma, S., and Swetapadma, A., 2019, "A Brief Review of Nearest Neighbor Algorithm for Learning and Classification," 2019 International Conference on Intelligent Computing and Control Systems (ICCS), Madurai, India, May 15–17, IEEE.
- [5] Efrat, G., Har-Peled, S., and Mitchell, M., 2002, "New Similarity Measures Between Polygons With Applications to Morphing and Polygon Sweeping," *Discrete Comput. Geom.*, **28**(4), pp. 535–569.
- [6] Efrat, A., Fan, Q. F., and Venkatasubramanian, S., 2007, "Curve Matching, Time Warping, and Light Fields: New Algorithms for Computing Similarity Between Curves," *J. Math. Imag. Vis.*, **27**(3), pp. 203–216.
- [7] Oring, A., Yakhini, Z., and Hel-Or, Y., 2020, "Faithful Autoencoder Interpolation by Shaping the Latent Space," *CoRR* abs/2008.01487.
- [8] Vermeer, K., Kuppens, R., and Herder, J., 2018, "Kinematic Synthesis Using Reinforcement Learning," Volume 2A: 44th Design Automation Conference, Quebec City, Canada, Aug. 26–29.
- [9] Fogelson, M. B., Tucker, C., and Cagan, J., 2023, "GCP-HOLO: Generating High-Order Linkage Graphs for Path Synthesis," *ASME J. Mech. Des.*, **145**(7), p. 073303.
- [10] Unruh, V., and Krishnaswami, P., 1995, "A Computer-Aided Design Technique for Semi-Automated Infinite Point Coupler Curve Synthesis of Four-Bar Linkages," *ASME J. Mech. Des.*, **117**(1), pp. 143–149.
- [11] Mcgarva, J., and Mullineux, G., 1993, "Harmonic Representation of Closed Curves," *Appl. Math. Model.*, **17**(4), pp. 213–218.
- [12] Mcgarva, J. R., 1994, "Rapid Search and Selection of Path Generating Mechanisms From a Library," *Mech. Mach. Theory*, **29**(2), pp. 223–235.
- [13] Ullah, I., and Kota, S., 1997, "Optimal Synthesis of Mechanisms for Path Generation Using Fourier Descriptors and Global Search Methods," *ASME J. Mech. Des.*, **119**(4), pp. 504–510.
- [14] Wu, J., Ge, Q. J., Gao, F., and Guo, W. Z., 2011, "On the Extension of a Fourier Descriptor Based Method for Planar Four-Bar Linkage Synthesis for Generation of Open and Closed Paths," *ASME J. Mech. Rob.*, **3**(3), p. 031002.
- [15] Li, X., Wu, J., and Ge, Q. J., 2016, "A Fourier Descriptor-Based Approach to Design Space Decomposition for Planar Motion Approximation," *ASME J. Mech. Rob.*, **8**(6), p. 064501.
- [16] Sharma, S., Purwar, A., and Ge, Q. J., 2019, "A Motion Synthesis Approach to Solving Alt-Burmester Problem by Exploiting Fourier Descriptor Relationship Between Path and Orientation Data," *ASME J. Mech. Rob.*, **11**(1), p. 011016.
- [17] Sharma, S., and Purwar, A., 2018, "Optimal Non-uniform Parameterization Scheme for Fourier Descriptor Based Path Synthesis of Four Bar Mechanisms," ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 2018, Quebec City, Canada, Aug. 26–29.
- [18] Khan, N., Ullah, I., and Al-Grafi, M., 2015, "Dimensional Synthesis of Mechanical Linkages Using Artificial Neural Networks and Fourier Descriptors," *Mech. Sci.*, **6**(1), pp. 29–34.
- [19] Chuang, G. C.-H., and Kuo, C.-C. J., 1996, "Wavelet Descriptor of Planar Curves: Theory and Applications," *IEEE Trans. Image Process.*, **5**(1), pp. 56–70.
- [20] Osowski, S., and Nghia, D. D., 2002, "Fourier and Wavelet Descriptors for Shape Recognition Using Neural Networks—A Comparative Study," *Pattern Recognit.*, **35**(9), pp. 1949–1957.
- [21] Nabout, A. A., and Tibken, B., 2005, "Wavelet Descriptors for Object Recognition Using Mexican Hat Function," *IFAC Proc. Volumes*, **38**(1), pp. 1107–1112.
- [22] Nabout, A. A., and Tibken, B., 2007, "Object Shape Recognition Using Mexican Hat Wavelet Descriptors," 2007 IEEE International Conference on Control and Automation, Hong Kong, China, Mar. 21–23.
- [23] Nabout, A. A., 2013, "Object Shape Recognition Using Wavelet Descriptors," *J. Eng.*, **2013**(3), pp. 1–15.
- [24] Sun, J., Liu, W., and Chu, J., 2015, "Dimensional Synthesis of Open Path Generator of Four-Bar Mechanisms Using the Haar Wavelet," *ASME J. Mech. Des.*, **137**(8), p. 082303.
- [25] Liu, W., Sun, J., Zhang, B., and Chu, J., 2018, "Wavelet Feature Parameters Representations of Open Planar Curves," *Appl. Math. Model.*, **57**(1), pp. 614–624.
- [26] Li, L., Cheng, W., Tsukada, K., and Hanasaki, K., 2004, "Flaw Classification by Using Artificial Neural Network and Wavelet," ASME Pressure Vessels and Piping Conference, San Diego, CA, July 25–29, pp. 59–65.
- [27] Mackiewicz, A., and Ratajczak, W., 1993, "Principal Components Analysis (PCA)," *Comput. Geosci.*, **19**(3), pp. 303–342.
- [28] Galan-Marín, G., Alonso, F. J., and Del Castillo, J. M., 2009, "Shape Optimization for Path Synthesis of Crank-Rocker Mechanisms Using a Wavelet-Based Neural Network," *Mech. Mach. Theory*, **44**(6), pp. 1132–1143.
- [29] Deshpande, S., and Purwar, A., 2020, "An Image-Based Approach to Variational Path Synthesis of Linkages," *ASME J. Comput. Inf. Sci. Eng.*, **21**(2), p. 021005.
- [30] Deshpande, S., and Purwar, A., 2019, "A Machine Learning Approach to Kinematic Synthesis of Defect-Free Planar Four-Bar Linkages," *ASME J. Comput. Inf. Sci. Eng.*, **19**(2), p. 021004.
- [31] Deshpande, S., and Purwar, A., 2019, "Computational Creativity Via Assisted Variational Synthesis of Mechanisms Using Deep Generative Models," *ASME J. Mech. Des.*, **141**(12), p. 121402.
- [32] Regenwetter, L., Nobari, A. H., and Ahmed, F., 2022, "Deep Generative Models in Engineering Design: A Review," *ASME J. Mech. Des.*, **144**(7), p. 071704.
- [33] Purwar, A., and Chakraborty, N., 2023, "Deep Learning-Driven Design of Robot Mechanisms," *ASME J. Comput. Inf. Sci. Eng.*, **23**(6), p. 060811.
- [34] Nobari, A. H., Srivastava, A., Gutfreund, D., and Ahmed, F., 2022, "LINKS: A Dataset of a Hundred Million Planar Linkage Mechanisms for Data-Driven Kinematic Design," Volume 3A: 48th Design Automation Conference (DAC), St. Louis, MO, Aug. 14–17.
- [35] Chang, W.-T., Lin, C.-C., and Wu, L.-L., 2005, "A Note on Grashof Theorem," *J. Marine Sci. Technol.*, **13**(4), pp. 239–248.
- [36] Kota, S., 1992, "Automatic Selection of Mechanism Designs From a Three-Dimensional Design Map," *J. Mech. Des.*, **114**(3), pp. 359–367.
- [37] Daubechies, I., 1988, "Orthonormal Bases of Compactly Supported Wavelets," *Commun. Pure Appl. Math.*, **41**(7), pp. 909–996.
- [38] Singh, P., Singh, P., and Sharma, R. K., 2011, "JPEG Image Compression Based on Biorthogonal, Coiflets and Daubechies Wavelet Families," *Int. J. Comput. Appl.*, **13**(1), pp. 1–7.
- [39] Popov, D., Gapochkin, A., and Nekrasov, A., 2018, "An Algorithm of Daubechies Wavelet Transform in the Final Field When Processing Speech Signals," *Electronics*, **7**(7), pp. 120–120.
- [40] Wahid, K., 2011, "Low Complexity Implementation of Daubechies Wavelets for Medical Imaging Applications," *Discrete Wavelet Transforms - Algorithms and Applications*, IntechOpen, London, UK.
- [41] Bank, D., Koenigstein, N., and Giryev, R., 2020, "Autoencoders," *CoRR*, abs/2003.05991.
- [42] Satapathy, S. C., Bhateja, V., Mohanty, J. R., and Udgata, S. K., 2020, "Smart Intelligent Computing and Applications," Proceedings of the Second International Conference on SCI 2018, Vol. 1, Bhubaneswar, Odisha, India, Dec. 21–22, Springer Singapore.
- [43] Kullback, S., and Leibler, R. A., 1951, "On Information and Sufficiency," *Ann. Math. Stat.*, **22**(1), pp. 79–86.
- [44] van der Maaten, L., and Hinton, G., 2008, "Visualizing Data Using t-SNE," *J. Mach. Learn. Res.*, **9**(86), pp. 2579–2605.