RECOVERING AND CLASSIFYING UPPER LIMB IMPAIRMENT TRAJECTORIES AFTER STROKE

Martín Méndez-López [1], *Alicia Fornés* [2] *and Antonio Agudo*[3]

[1]Universitat Pompeu Fabra, Spain
[2] Computer Vision Center, UAB, Spain
[3]Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Spain

ABSTRACT

Upper limb impairment is a loss of motor function after a stroke, leading to difficulties in performing daily tasks. With a low remission rate six months after stroke, monitoring during this critical period is essential. Telemonitoring with inertial sensors has become a common approach that requires the identification and recognition of specific movements. However, data quality in clinical databases is a challenge, making data augmentation necessary. In this work, we propose a novel method that can learn a trajectory subspace from partial 3D signals in an unsupervised manner. Our method is simple yet effective, producing novel human-feasible motions compatible with the estimated trajectory subspace. To this end, three approaches are introduced from global to local models that can capture a wide variety of human motions. Our method outperforms the results in the state of the art in both control and patient subjects.

***Index Terms*—** Trajectory subspace, Signal completion, Low-rank models, Upper limb impairment.

1. INTRODUCTION

Stroke is the second leading cause of death and the fourth leading cause of disability according to the World Health Organization, and its prevalence will increase with the aging population. Among the disabilities caused by stroke, the most common is upper limb impairment. This means that the upper limb experiences a loss of motor function and sensation. The patient unlearns the normal use of the limb and adopts new patterns such as compensations, making it difficult to perform daily tasks [1]. After a stroke, there is a critical six-week window for rehabilitation. During this time, neuronal plasticity supports movement recovery. However, if pain and muscle weakness lead to immobilization, patients may develop compensatory postures that are difficult to correct later [2]. The limited window for mobility improvement is further challenged by the need for patients to perform unsupervised home rehabilitation due to resource constraints. As a result, telemonitoring human motion (some visual instances of human motion are displayed in Fig. 1) has become a key factor for therapeutic protocols. To this end, some alternatives based on vision or inertial sensors [3], such as the Apple Watch, can be used to capture 3D human motion of the extremity. These tools track predefined target movements to monitor progress and adjust therapy and medication as needed [4, 5].

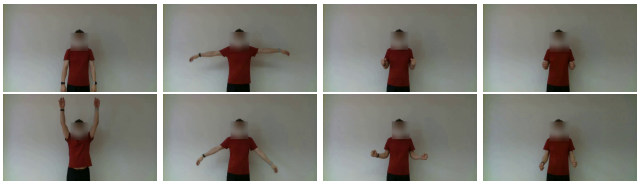

Fig. 1: **Human Activity Recognition**. Four types of human motion we use to analyze a patient evolution (two frame instances per column are displayed).

From a technical perspective, this is a Human Activity Recognition (HAR) problem, which requires signal preprocessing, segmentation, and feature extraction for classification [6]. Clinical HAR databases face challenges due to high data acquisition costs and interclass variability, complicating movement classification. A common strategy is data augmentation, though it is less studied for time series than for images. For inertial sensor signals, geometric transformations like Gaussian noise, resampling, scaling, and interpolation have shown an improvement in accuracy in standard datasets such as UCI-HAR and USC-HAR [7]. Another approach applies rotation to simulate different sensor positions [7]. In [8], Discrete Time Warping (DTW) is used for data generation, improving accuracy in some datasets. However, they use datasets from healthy subjects, simplifying classification, and do not ensure that the generated movements remain within humanly feasible ranges of trajectory, velocity, and acceleration, despite improving accuracy.

In recognition of motions in patients with upper limb impairment, no previous work has used data augmentation. Approaches such as Random Forest [9], DTW-based classification [5], and methods to distinguish goal-oriented

This work has been supported by the project GRAVATAR PID2023-151184OB-I00 funded by MCIU/AEI/10.13039/501100011033 and by ERDF, UE.

movements [10] have shown good results, with deep learning frameworks such as LSTM performing best [11]. However, these successes are due to the use of well-separated motions and patients at the same stage of impairment. In more realistic telemonitoring scenarios, the accuracy drops to 56% [12, 13, 14]. To address that, in this paper we propose a compact and interpretable formulation to learn a 3D trajectory subspace in patients with upper limb impairment from partial data and in an unsupervised manner. Thanks to that, we can synthetically generate more trajectories within the subspace at low cost, a key factor in this context as the amount of publicly available datasets is very limited and the real acquisition of new ones is expensive, when possible.

2. FITTING TRAJECTORIES AND GENERATION

Let $\bar{\mathbf{X}}$ be a possibly non-full $3M \times N$ matrix of 3D annotations where M and N are the number of trajectories and samples, respectively; and $\mathbf{O}$ the corresponding $M \times N$ observation matrix with $\{1, 0\}$ entries indicating whether a specific point is observed or not. Given both $\bar{\mathbf{X}}$ and $\mathbf{O}$, our goal is to learn a subspace to model the set of observed trajectories from partial data. To address this problem, in this paper, we propose optimizing a cost function that enforces the full trajectories $\mathbf{X}$ to lie in a linear subspace, i.e., that matrix is low rank. As rank minimization is a non-convex NP-hard problem [15], we employ a nuclear norm for a convex relaxation [16]. Taking that into consideration, we propose to handle the next optimization problem:

$$\begin{aligned} &\underset{\mathbf{X},\mathbf{C},\mathbf{B}}{\arg\min} \|(\mathbf{O} \otimes \mathbf{1}_3) \odot (\mathbf{X} - \bar{\mathbf{X}})\|_F^2 + \alpha \|\mathbf{X}\|_* \qquad (1) \\ &\text{subject to} \quad \mathbf{X} = \mathbf{CB} \end{aligned}$$

where $\otimes$ and $\odot$ indicate a Kronecker and Hadamard products, respectively, and $\mathbf{1}_3$ is a vector of ones. $\|\cdot\|_F$ and $\|\cdot\|_*$ denote a Frobenius and a nuclear norm, respectively. α is a weight coefficient. Our goal is to infer both $\mathbf{X}$ together with a subspace encoded by $\mathbf{C}$ and $\mathbf{B}$, which will be exploited to produce other trajectories later. In this paper, we propose an algorithm to minimize Eq. (1) by means of a two-stage approach in which: 1) complete missing entries $\mathbf{X}$; 2) retrieve a subspace to model the observed data.

The first stage can be sorted out by means of bilinear factorization [17, 18, 19], defining $\mathbf{X} = \mathbf{U}\mathbf{V}^\top$ where $\mathbf{U} \in \mathbb{R}^{3M \times K}$ and $\mathbf{V} \in \mathbb{R}^{N \times K}$ with $K \leq \min(3M, N)$ an upper bound on rank($\mathbf{X}$), and considering the next problem:

$$\begin{aligned} &\underset{\mathbf{X},\mathbf{U},\mathbf{V}}{\arg\min} \|(\mathbf{O} \otimes \mathbf{1}_3) \odot (\mathbf{X} - \bar{\mathbf{X}})\|_F^2 + \frac{\beta}{2}(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2) \\ &\text{subject to} \quad \mathbf{X} = \mathbf{U}\mathbf{V}^\top \qquad (2) \end{aligned}$$

with β a weight coefficient. That problem is efficiently solved via Augmented Lagrange Multipliers. Convergence is improved by initializing the entries of $\bar{\mathbf{X}}$ in every trajectory as the mean value of the observed data points.

Equipped with the full-trajectories matrix $\mathbf{X}$, we now have to learn a subspace to encode the set of observed data. In general, the problem to be tracked can be written as:

$$\underset{\mathbf{C},\mathbf{B}}{\arg\min} \|(\mathbf{X} - \mathbf{CB})\|_F^2, \qquad (3)$$

where $\mathbf{C}$ and $\mathbf{B}$ are $3M \times K$ and $K \times N$ matrices –unless otherwise stated– to encode the subspace of trajectories, with K its rank.

In this work, we propose the use of three formulations to constrain the subspace, where the dimensionality of both factors and their interpretability are different. Up to our knowledge, we are the first to use this type of representations in upper limb impairment modeling.

Principal Component Analysis (PCA). In the first case, data are projected onto a low-dimensional space of K principal components [20], which are orthogonal to each other and ordered by their eigenvalue. In particular, in this type of decomposition $\mathbf{C}$ represents a shape basis matrix since every vector includes information for all trajectories in a particular sample, with $\mathbf{B}$ the corresponding coefficients. Both factors can be obtained efficiently by PCA.

Discrete Cosine Transform (DCT). In the second case, we use a trajectory basis from signal theory. In particular, for $n = \{1, \ldots, N\}$ we define a K-dimensional vector $\mathbf{b}^n = [w_1^n, \ldots, w_K^n]^\top$ with:

$$w_k^n = \frac{\rho_k}{\sqrt{N}} \cos\left(\frac{\pi(2n-1)(k-1)}{2N}\right), \qquad (4)$$

where $\rho_k = 1$ for $k = 1$ and $\rho_k = \sqrt{2}$, otherwise. The collection of time-varying 3D trajectories for all N instances can then be written by a linear combination of trajectories as $\mathbf{X} = \mathbf{CB}$, where $\mathbf{C}$ is a matrix of unknown coefficient vectors, and $\mathbf{B} = [\mathbf{b}^1, \ldots, \mathbf{b}^N]$ is a known matrix with a predefined trajectory basis. This type of basis has shown its strength to capture human motion [21, 22, 23] from video, especially when the behavior of the signal can be adjusted globally through a small combination of vectors. However, we exploit a K $\times$ N trajectory basis instead of 3K $\times$ N, making our formulation more compact and robust when inferring the subspace coefficients (thanks to that, we can solve the problem by estimating a smaller number of coefficients without losing representation).

Piecewise trajectory basis. Finally, we propose the use of local curves to encode the full trajectories, building a global curve in pieces. To this end, following the work in [24, 25], we propose uniform B-spline basis functions where K control points are needed in order to define the number of pieces and, therefore, the final global curve. We consider N $3K$-dimensional $\mathbf{p}^n = [p_{x1}^n, \ldots, p_{xK}^n, p_{y1}^n, \ldots, p_{yK}^n, p_{z1}^n, \ldots, p_{zK}^n]^\top$ vectors to collect the K control point locations associated with the n-th sample point. The time-varying 3D trajectories for all N instances can be now written by a linear combination of trajectories as $\mathbf{X} = \mathbf{CB}$, where $\mathbf{C} \in \mathbb{R}^{3M \times 3K}$

is a matrix of known local B-spline functions [24] and $\mathbf{B} = [\mathbf{p}^1, \ldots, \mathbf{p}^N] \in \mathbb{R}^{3K \times N}$ is an unknown matrix with control point locations.

On balance, we propose three models to learn the trajectory subspace. Two of them are global (PCA and DCT), and the other is local. In PCA we need to infer both factors jointly, while in the others one factor is assumed as known in advance, making every model different with particular properties.

With these ingredients, we present three simple yet effective generators. The main idea is to include random Gaussian noise with zero mean and standard deviation σ to the *coefficients matrix* in every model. For PCA and B-Spline cases, new trajectories are generated by $\hat{\mathbf{X}} = \mathbf{C}(\mathbf{B}+\mathbf{N})$, where $\mathbf{N} = \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ is a noise matrix $K \times N$ or $3K \times N$ for PCA and B-Spline model, respectively. When the DCT model is considered, new trajectories are obtained as $\hat{\mathbf{X}} = (\mathbf{C} + \mathbf{N})\mathbf{B}$ where $\mathbf{N} = \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ is a $3M \times K$ noise matrix.

3. EXPERIMENTAL EVALUATION

We now present our experimental results on real data, providing both qualitative and quantitative evaluation and comparison with respect to competing techniques. For quantitative evaluation, we use recall, precision, and F1-score metrics.

Dataset. We use RPM3D [12], a dataset that contains the motion of 25 healthy (control) subjects and 4 upper-limb impairment patients. In total, there are four Fugl-Meyer motions (see one per column in Figs. 1- 2 for images and 3D trajectories, respectively) –an ictus clinical protocol– for classification: shoulder extension and flexion, shoulder abduction/adduction, internal/external shoulder rotation, and elbow flexion/extension [12, 13, 14]. The movements have some particularities. Movement #1 and #2 are simple and are up and down, while #3 and #4 are complex as they include the combination of several simple motions, implying multiple joints. It must be taken into account that control subjects will be able to perform those movements completely, while for patients, depending on their degree of impairment, they will perform a simplified version of the motion because they do not have a complete range of movement in the limb. However, impairment will include sudden increases in jerk in motion, over compensations and new learned motions (owing to the pain of performing the movement naturally) that will increase the variability between and within classes [5, 26]. All 3D trajectories were acquired by using an Apple Watch sensor placed on every individual's wrist over four sessions with the aim of monitoring the patient's progress. For every motion and session, the motion of both hands was acquired for each person, collecting some features such as acceleration, velocity, position, and gravity along the three axes, jerk, pitch, and roll, sampled at 100 Hz. Note that our method only needs the 3D trajectory of every individual's wrist.

Analysis. We first split control subjects from patients, creating for each group an input trajectory matrix $\bar{\mathbf{X}}$, that is, one for every distinct motion (see Table 1 to know the numbers of every case). We then solve Eq. (1) by considering the three models we present in section 2. The rank in every model is empirically selected, ensuring that the reconstructed matrix retains the 99.5% of the variance in the input matrix.

A qualitative evaluation of our reconstructed trajectories is shown in Fig. 3 where the best estimations are provided by the DCT and B-Spline models for control subjects. However, for patients, DCT do not reconstruct correctly as it has to deal with localities, which is not going to perform correctly, as the trajectory basis is global. It is worth noting that B-Spline still obtains a good reconstruction of the 3D signal.

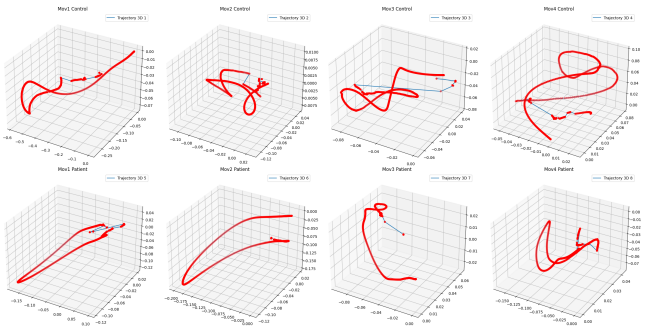

Fig. 2: **Some instances of 3D trajectories from upper-limb impairment patients.** Every column includes a type of Fugl-Meyer motion, displaying the control subjects and patients in the top and bottom parts, respectively. The red points and blue lines represent the continuous trajectory and steps between two red points, i.e., a non-soft trajectory can be captured.

	Control	Patient
Class 1	(987, 727)	(192, 597)
Class 2	(1050, 644)	(204, 573)
Class 3	(927, 710)	(192, 698)
Class 4	(1017, 576)	(192, 822)

Table 1: **Number of trajectories and samples for control and patient subjects.** The table reports the total number of trajectories M and the maximum number of samples N in $\bar{\mathbf{X}}$.

	Control	Patient
Class 1 - DCT	(21714, 727)	(7488, 597)
Class 2 - DCT	(13650, 644)	(7956, 573)
Class 3 - DCT	(8343, 710)	(7488, 698)
Class 4 - DCT	(18306, 576)	(7488, 822)
Class 1 - PCA	(38493, 727)	(7488, 597)
Class 2 - PCA	(40950, 644)	(3876, 573)
Class 3 - PCA	(36153, 710)	(1536, 698)
Class 4 - PCA	(37629, 576)	(2304, 822)
Class 1 - B-Spline	(38493, 727)	(7488, 597)
Class 2 - B-Spline	(40950, 644)	(7956, 573)
Class 3 - B-Spline	(36153, 710)	(7488, 698)
Class 4 - B-Spline	(39663, 576)	(7488, 822)

Table 2: **Synthetic trajectories and samples for control and patient subjects that have been generated.** The numbers for every generator (PCA, DCT, B-Spline) are reported.

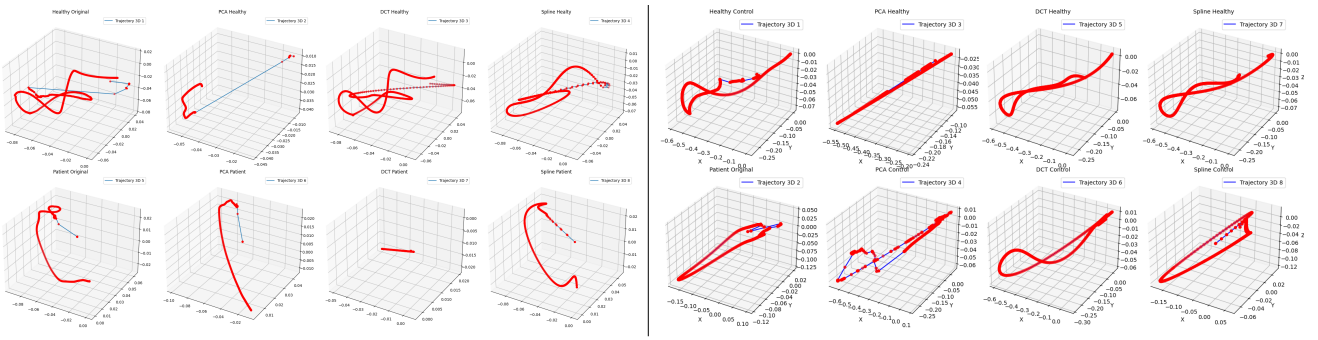

Fig. 3: **Motion types #3 and #1 reconstruction.** The same information is displayed in both sides. From left to right: the 3D trajectory captured by the sensor, and the reconstructed one by PCA, DCT and B-Spline models. Control subject and patients are considered in the top and bottom parts.

	Classifier	Recall				Precision				F1 Average
		Class 1	Class 2	Class 3	Class 4	Class 1	Class 2	Class 3	Class 4	
Reconstructed trajectories without noise										
Control - Baseline [12]	SVM	0.82	0.75	0.44	0.85	0.72	0.88	0.52	0.76	0.72
Control - DCT	SVM	0.77	**0.82**	**0.45**	0.83	**0.76**	0.79	**0.62**	0.69	0.71
Control - PCA	SVM	**0.89**	0.72	0.35	0.61	0.54	0.80	**0.65**	0.68	0.62
Control - B-Spline	SVM	0.78	**0.82**	0.42	0.85	**0.76**	0.79	**0.64**	0.68	0.70
Control - Baseline	MLP	0.89	0.85	0.62	0.85	0.80	0.89	0.69	0.83	0.82
Control - DCT	MLP	**0.90**	**0.93**	**0.79**	**0.87**	**0.89**	**0.901**	**0.82**	**0.88**	**0.85**
Control - PCA	MLP	**0.89**	0.84	0.53	0.34	0.58	0.74	0.55	**0.95**	0.63
Control - B-Spline	MLP	**0.93**	**0.88**	**0.76**	**0.87**	**0.85**	**0.93**	**0.82**	**0.86**	**0.85**
Patient - Baseline [12]	SVM	0.27	0.87	0.64	0.54	1.00	0.81	0.39	0.43	0.55
Patient - DCT	SVM	**0.58**	0.85	**0.73**	0.34	0.75	0.72	**0.53**	**0.52**	**0.62**
Patient - PCA	SVM	**0.33**	0.76	**0.76**	0.37	0.87	0.75	**0.46**	0.39	0.53
Patient - B-Spline	SVM	**0.53**	0.73	**0.76**	0.36	0.79	0.82	**0.49**	0.41	**0.61**
Patient - Baseline	MLP	0.53	1.00	0.64	0.64	1.00	0.83	0.50	0.58	0.70
Patient - DCT	MLP	**0.80**	0.93	0.25	0.23	0.47	0.57	**1.00**	**0.60**	0.50
Patient - PCA	MLP	**0.58**	0.95	0.62	0.59	0.72	0.73	**0.67**	**0.63**	0.66
Patient - B-Spline	MLP	**0.84**	0.90	**0.69**	**0.73**	0.78	**0.92**	**0.72**	**0.73**	**0.78**
Synthetic trajectories with noise										
Control - Baseline [12]	SVM	0.82	0.75	0.44	0.85	0.72	0.88	0.52	0.76	0.72
Control - DCT	SVM	**0.89**	**0.83**	0.25	**0.94**	0.73	0.84	**0.84**	0.65	**0.75**
Control - PCA	SVM	**0.85**	**0.84**	**0.48**	0.39	0.54	0.70	0.59	**0.89**	0.66
Control - B-Spline	SVM	0.82	**0.86**	**0.59**	0.85	**0.78**	0.79	**0.77**	0.80	**0.78**
Control - Baseline	MLP	0.89	0.85	0.62	0.85	0.80	0.89	0.69	0.83	0.82
Control - DCT	MLP	**0.96**	**0.96**	**0.76**	**0.99**	**0.90**	**0.96**	**0.98**	**0.87**	**0.92**
Control - PCA	MLP	0.73	0.79	**0.66**	0.34	0.63	0.70	0.48	**0.95**	0.62
Control - B-Spline	MLP	**0.98**	**0.91**	**0.95**	**0.94**	**0.90**	**0.97**	**0.92**	**0.99**	**0.94**
Patient - Baseline [12]	SVM	0.27	0.87	0.64	0.54	1.00	0.81	0.39	0.43	0.55
Patient - DCT	SVM	**0.56**	**0.88**	0.48	0.30	0.48	0.58	**0.69**	**0.54**	0.56
Patient - PCA	SVM	**0.87**	0.78	0.00	0.00	0.28	**0.85**	0.00	0.00	0.22
Patient - B-Spline	SVM	**0.78**	0.81	**0.70**	0.37	0.70	0.83	0.55	**0.58**	**0.67**
Patient - Baseline	MLP	0.53	1.00	0.64	0.64	1.00	0.83	0.50	0.58	0.70
Patient - DCT	MLP	**0.80**	0.90	0.66	0.72	0.64	0.75	**0.89**	**0.88**	**0.77**
Patient - PCA	MLP	**0.84**	**0.98**	**0.83**	**0.81**	0.72	0.80	**0.94**	**0.83**	**0.84**
Patient - B-Spline	MLP	**1.00**	**0.97**	**0.95**	**1.00**	**0.94**	**1.00**	**1.00**	**0.98**	**0.98**

Table 3: **Classification recall, precision and F1-score.** The table reports the numbers for the baseline [12] and for our three generators based on DCT, PCA and B-Spline. All cases are evaluated for control subjects and patients, and exploiting a classifier based on SVM or MLP. **Top:** Results for reconstructed trajectories by our methods. **Bottom:** Results after applying Gaussian noise to generate synthetic new trajectories.

As was said above, once we have applied our three models to learn a trajectory subspace, a Gaussian noise (σ=0.01) is applied to randomly produce more trajectories within the learned subspace, i.e., the matrix $\hat{\mathbf{X}}$. As our trajectories are represented in meters, this implies a deviation of 1 cm from the trajectory reconstructed without noise. A generated trajectory is defined as valid if its Mean Absolute Error (MAE) compared to the original trajectory is less than 1.5 times the MAE between the reconstructed trajectory without noise and the original one. After applying that, the set of synthetic trajectories that we obtain is reported in Table 2.

Without loss of generality, we propose to evaluate our trajectory generators on a classification task. To do so, we exploit two alternatives: an SVM-based approach as is standard in the literature [12] and an MLP-based approach with 100 layers, Adam solver with learning rate 0.001 and 1000 iterations. To this end, we propose a battery of experiments in which the three generators are considered in two situations: 1) the reconstructed trajectories are exploited for training and the original ones for testing, and 2) Gaussian noise is added to produce new trajectories that are exploited in training, again using the original ones for testing. That means training is always performed with synthetic trajectories and testing with real ones. To train and test both SVM and MLP classifiers, a 12-feature vector is obtained from each trajectory, corresponding to the maximum, minimum, mean and standard deviation of each x-, y-, z-channel of each trajectory. These features are evaluated based on the actual maximum number of frames for each trajectory, without considering the signal completion we propose in section 2. For comparison, we consider [12] with full trajectories, the same features for classification, and an 80-20 train-test split of the data to make a fair analysis.

First, we consider reconstructed trajectories without noise for training (see Table 3-top). As can be seen, for control subjects, the proposed approaches obtain a similar result with respect to the baseline without improving it when using an SVM classifier, but our DCT- and B-Spline-based solutions are better when an MLP classifier is considered. This effect is more visible for patients –a more challenging case–, where our B-Spline-based solution consistently outperforms the rest of the approaches for both classifiers.

Second, we report the same analysis but now including Gaussian noise to generate new trajectories for training (see Table 3-bottom). In this case, the most robust solution seems to be the one obtained with the B-Spline-based model, for control subjects and patients, and considering both classifiers, obtaining very precise classification results in some cases. For patients with SVM, some generators respond worst to noise, decreasing their performance compared to not adding noise, except for the B-Spline solution which obtains the major overpassing of the baseline. In contrast, when MLP is used, all three generators outperform the MLP baseline.

In general, for similar conditions, the MLP classifier provides better solutions than those reported by SVM. Additionally, our results show the effectiveness of our generators, especially that based on B-Spline, since piecewise modeling of trajectories gives greater versatility to the generator, producing more accurate adjustments and therefore generating new trajectories that are faithful to the real ones –which represents an effective encoding of the upper-limb impairment trajectory space in this context–. This shows that the generator is a feasible way to obtain new trajectories that even outperform the originals in classification. That is because our generator produces cleaner and smoother trajectories compared to the potentially noisy ones that are acquired with the sensor. The numbers of F1 score still low for SVM are supported by the fact that a single wrist sensor placed in patients obtains an accuracy around 40% which makes it necessary to place more sensors to improve the results, as well as by the type of classifier used [5]. For the case of MLP it obtains good metrics both for the control subjects and patients, and the generators improve the results better than in the case of SVM.

Then global methods such as DCT perform well in global trajectories with no localities as the ones in healthy motions, independently of whether it is a composed movement or not, while B-Spline works better in patients as it has a local basis. However, B-Spline also performs well in complete subjects. In addition, both the DCT and B-Spline methods respond well to the addition of Gaussian noise and improve the baseline. This is particularly notorious for B-Spline methods using MLP where it is near to a perfect classification.

Overall, what has been shown is that the trajectory generators that we propose are a framework that can augment upper-limb datasets for both patients and healthy subjects. We hope that our contribution will serve as inspiration for other works to increase the amount of data in a precise and interpretable manner, especially in those contexts where obtaining the data is complex and expensive.

4. CONCLUSION

We have provided a framework capable of generating trajectories from small- and high-variance trajectory datasets such as the one in RPM3D [12]. Our method just needs a set of incomplete 3D signals to learn a trajectory subspace with physical interpretation in an unsupervised fashion, which is then exploited to produce compatible trajectories. As has been shown, our method improves the baseline in this dataset for control and patient subjects, obtaining robust, consistent, and accurate trajectories that are properly classified by off-the-self classifiers. Further, this method overcomes geometric transformations in the sense that working with trajectories enables to create data that live in the subspace of feasible trajectories, not only for healthy subjects but for patient data. Future work should focus on incorporating additional constraints in our models to encode the features of motion in this context as well as the combination with other signals.

5. REFERENCES

[1] P. Raghavan, "Upper limb motor impairment after stroke," *Physical medicine and rehabilitation clinics of North America*, vol. 26, no. 4, pp. 599–610, 2015.

[2] A. Pollock, S.E. Farmer, M.C. Brady, P. Langhorne, G.E. Mead, J. Mehrholz, and F. van Wijck, "Interventions for improving upper limb function after stroke," *CDSR*, vol. 12, no. 11, 2014.

[3] C. Yang, A. Kerr, V. Stankovic, L. Stankovic, and P. Rowe, "Upper limb movement analysis via marker tracking with a single-camera system," in *ICIP*, 2014.

[4] D. J. Gladstone, C. J. Danells, and S. E. Black, "The fugl-meyer assessment of motor recovery after stroke: a critical review of its measurement properties," *NNR*, vol. 16, no. 3, pp. 232–240, 2002.

[5] N. J. Seo, K. Coupland, C. Finetto, and G. Scronce, "Wearable sensor to monitor quality of upper limb task practice for stroke survivors at home," *Sensors*, vol. 24, no. 2, pp. 554, 2024.

[6] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *CSUR*, vol. 46, no. 3, pp. 1–33, 2014.

[7] T. Um, F. MJ. Pfister, D. Pichler, S. Endo, M. Lang, S. Hirche, U. Fietzek, and D. Kulić, "Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks," in *ICMI*, 2017, pp. 216–220.

[8] G. Forestier, F. Petitjean, H. A. Dau, G. I. Webb, and E. Keogh, "Generating synthetic time series to augment sparse datasets," in *ICDM*, 2017, pp. 865–870.

[9] E. M. Bochniewicz, G. Emmer, A. McLeod, J. Barth, A. W. Dromerick, and P. Lum, "Measuring functional arm movement after stroke using a single wrist-worn sensor and machine learning," *JSCD*, vol. 26, no. 12, pp. 2880–2887, 2017.

[10] S. I. Lee, C. P. Adans-Dester, M. Grimaldi, A. V. Dowling, P. C. Horak, R. M. Black-Schaffer, P. Bonato, and J. T. Gwin, "Enabling stroke rehabilitation in home and community settings: A wearable sensor-based approach for upper-limb motor training," *JTEHM*, vol. 6, pp. 1–11, 2018.

[11] M. Li, G. Scronce, C. Finetto, K. Coupland, M. Zhong, M. E. Lambert, A. Baker, F. Luo, and N. J. Seo, "Application of deep learning algorithm to monitor upper extremity task practice," *Sensors*, vol. 23, no. 13, pp. 6110, 2023.

[12] A. Fornés et al., "The RPM3D project: 3D kinematics for remote patient monitoring," in *IGHM*, 2022.

[13] A. Bensalah et al., "Towards stroke patients' upper-limb automatic motor assessment using smartwatches," in *ICPRW*, 2021, pp. 476–489.

[14] A. Bensalah, A. Fornés, C. Carmona-Duarte, and J. Lladós, "Easing automatic neurorehabilitation via classification and smoothness analysis," in *IGHM*, 2022.

[15] B. Recht, M. Fazel, and P. A Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM review*, vol. 52, no. 3, pp. 471–501, 2010.

[16] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational Mathematics*, vol. 9, no. 6, pp. 717, 2008.

[17] F. Bach, J. Mairal, and J. Ponce, "Convex sparse matrix factorizations," *Technical report HAL-00345747*, 2008.

[18] R. Cabral, F. de la Torre, J. P. Costeira, and A. Bernardino, "Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition," in *ICCV*, 2013.

[19] A. Agudo, "Segmentation and 3D reconstruction of non-rigid shape from RGB video," in *ICIP*, 2020, pp. 2845–2849.

[20] Y. V. Lata, C. K. B. Tungathurthi, H. R. M. Rao, A. Govardhan, and L. P. Reddy, "Facial recognition using eigenfaces by pca," *IJRTE*, vol. 1, no. 1, pp. 587, 2009.

[21] A. Agudo and F. Moreno-Noguer, "A scalable, efficient, and accurate solution to non-rigid structure from motion," *CVIU*, vol. 167, no. 2, pp. 121–133, 2018.

[22] A. Zaheer, I. Akhter, M. H. Baig, S. Marzban, and S. Khan, "Multiview structure from motion in trajectory space," in *ICCV*, 2011.

[23] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade, "Non-rigid structure from motion in trajectory space," in *NIPS*, 2008.

[24] A. Agudo, "Spline human motion recovery," in *ICIP*, 2022, pp. 4138–4142.

[25] A. Agudo, "Piecewise bézier space: Recovering 3D dynamic motion from video," in *ICIP*, 2021, pp. 3268–3272.

[26] G. J. Kim, A. Parnandi, S. Eva, and H. Schambra, "The use of wearable sensors to assess and treat the upper extremity after stroke: a scoping review," *Disability and rehabilitation*, vol. 44, no. 20, pp. 6119–6138, 2022.