# TITLE

Supplementary Material of “Real-Time Proxy-Control of Re-Parameterized Peripheral Signals Using a Close-Loop Interface”

# AUTHORS AND AFFILIATIONS

Vilelmini Kalampratsidou1, 2,, Steven Kemper3

1Center for Cognitive Science, Rutgers University, New Brunswick, NJ, USA

2Department of Computer Science, Rutgers University, New Brunswick, NJ, USA

3Music Department, Mason Gross School of the Arts, Rutgers University, New Brunswick, NJ, USA

Email Addresses of Authors:

Vilelmini Kalampratsidou, [vilelmini.kalabratsidou@gmail.com](mailto:vilelmini.kalabratsidou@gmail.com)

Steven Kemper, [skemper@mgsa.rutgers.edu](mailto:skemper@mgsa.rutgers.edu)

# VARIATIONS OF THE PRESENTED CLOSE-LOOP INTERFACE

The design of the generic interface presented in the main text is based on the equipment provided. However, several instruments can be used to register the biophysical signals and replace the sample equipment described above. We acquire brain (Central Nervous System, CNS) signals, bodily kinematics (Peripheral Nervous System, PNS) signals and heart (Autonomic Nervous System, ANS) signals.

Furthermore, LSL (LabStreamingLayer, <https://github.com/sccn/labstreaminglayer>) is an open access platform that allows the synchronized collection of data that is streaming from various equipment in near real-time. Although in our design the use of LSL is considered vital for the synchronous recording of data, other designs may satisfy the same principle. Instances of such designs could be the direct streaming of the data to a python interface, whereby the synchronization could take place hand in hand with the data analysis (in such case, we would have steps 1, 3, and 4 of **Figure 3** of the main text**,** step 2 would not be needed).

Other possible variations are that someone could create different experimental concepts and procedures, applied to different populations, or use different devices and methods to create sensory augmentation/substitution.

## Sensory Feedback Augmentation

Numerous variations could be applied on the way that sensory-feedback augmentation is created. In the interface example 1 (see section “EXAMPLE 1, Supplementary File), we take the inter-peak-interval (IPI) times of the heartbeats and use their length to define the speed of the played song. This process is happening in real-time and the updates of the song are taking place continuously during the experiment. As a result, the song is changing speed constantly. Alternatively, we could extract audio features from the MMS trains or the original signal, using the MATLAB MIR toolbox or other methods, and use them to alter different features of the sound.

Likewise, with the visual feedback (*e.g.* provided by the avatar, see section “Example 2”, Supplementary File) we can alter the real-time motions that we endow the avatar with, using noise portrays extracted from databanks of motion data collected from the person. We can also endow the avatar with the veridical motions from the user, while delaying them to test the person’s limits in gaining awareness of the delays. We can manipulate other properties such as color, brightness, and contrast of the 3D rendering, and as in this example, add audio to enhance the visuo-motor experience.

Other sensory modalities could be used as a means of feedback in similar interfaces. Modern technology allows us to generate sensations such as haptic, olfactory, to name a few, which could be incorporated in a co-adaptive interface

# EXAMPLE 1: AUDIO CLOSE-LOOP INTERFACE OF A REAL DYADIC INTERACTION

Two salsa dancers performed a well-rehearsed routine staging a choreography and a spontaneously improvised dance. The dancers had to perform the original version of the song and a version blended with the real-time speed of the heartbeat stream.

The proxy control interface used to blend the speed of the heartbeat with the speed of the song is illustrated in **Figure 4** of the main text. We used the female dancer’s heart signals to extract the heartbeat times and alter the music. In real time, we performed signal processing of the ECG signal to extract the times of the R peaks and estimate their IBI timings. Then, we streamed this information to MAX where it defined the speed of the played song. This way, the played song was altered by the biophysical signals. This process led to further alterations of the motions and heartbeat signals, which we were continuously re-parameterizing.

## Participants

Two experienced salsa dancers (a male and a female, 30 years old) participated in the experiment which was approved by the Rutgers University Institutional Review Board in accordance with the Helsinki Act.

## Set-up of the Audio Close-loop Interface

Set-up kinematic and EEG equipment -ANS, PNS, and CNS

A 32-channel EEG system was used to record brain activity as well as the HR of the female dancer. Twelve inertial measurement units (IMU) were also used to record bodily activity. The latter signals are used offline and currently are not part of our online streaming and analyses described in our interface. Their analysis is presented in 1.

To prepare the EEG system, follow the steps presented in sections 2.2 and 2.3 of “Set-up of the Close-Loop Interface” of the main text.

Preparation and set-up of LSL for synchronized data recording and streaming of data

For the synchronous recording of the data in LSL, we followed the step presented in sections 2.5 and 2.6 of “Set-up of the Close-Loop Interface” of the main text.

The real-time analyses of data and monitoring of the human system

A Python interface is designed to continuously receive chunks of data streamed by the LSL at a frequency of 500Hz. Once the interface receives the first chunk, it sends a signal to start the Max interface. At the beginning of the recording, the interface buffers 2000-frame-long data (equal to 4sec, enough to detect R-peaks of the ECG signal), which is utilized as an assistive vector. Once it collects those frames, it starts processing the data to detect peaks and stream those to Max.

Specifically, on each frame, the system adds the new value to the end of the buffer and dismisses the oldest value.  The updated buffer with the ECG data is filtered using the Butterworth IIR band pass filter for 5-30Hz at 2nd order. The range of the band pass filter is selected based on the finding that a QRS complex is present in the frequency range of 5-30Hz. Then, we use the *peakutils.indexes Python* function (threshold is set 0.7 and minimum distance is equal to 110) to detect the R peaks’ amplitude values and their times of occurrence. Each time the interface detects a new peak in the buffer, it sends 1 to Max, otherwise it sends 0.

To achieve the closest to real-time streaming, we designed our algorithm to stream 1 when detecting a peak at the most recently added frame, the last one, while accounting for the time it takes to filter and process these values. Such time periods are variable because they require forecasting the future trend of the signal. As a result, we had to detect peaks at a consistently earlier frame and estimate forward. To that end, we empirically examined different values of n number of earlier frames and investigated the limiting cases, so that n had the smallest possible value (be as recent as possible) and the corresponding ECG value had enough neighbour values to be filtered and be processed properly for reliable forward estimation. Initial stages to build the interface took extensive experimentation, upon which we settled on n = 10. Clearly, this choice will depend on sampling resolution and needs to be individualized. In the present examples, this value provides good buffering while delaying 1/50 sec. Future work will involve researching other methods of time series forecasting and quick detection methods, to optimize the buffering-window size. The interface can also save the raw data collected during a session for later analysis. These data were used to aid calibrate the buffering time in a personalized manner.

Finally, once the recording is complete the interface sends a signal to stop the Max interface.

The code that performs this analysis can be found at

<https://github.com/VilelminiKala/CloseLoopInterfaceJOVE>.

Generation of the sensory feedback

The peaks detected by the Python interface are sent to the playback interface. Specifically, we sent 1 each frame a heartbeat is detected, and we sent 0 otherwise. This is done with software designed using the musical programming language(cycling74.com). Filtered heart R peaks are transmitted in real time to Max from the Python script via Open Sound Control (OSC) (<http://opensoundcontrol.org/>). Additionally, signals to start and stop playback, as well as an index value of peak numbers used for testing the system are also transmitted to Max via OSC. Within the playback interface, the inter-peak intervals are measured and converted to beats per minute. This data is scaled to create a playback speed scaling factor between 0 (slowest possible playback) and maximum playback speed (fastest possible playback). On this continuum, 1 equals "normal" playback speed, 0.5 equals half speed, and 2 equals double playback speed. There are two modes for data scaling, a positive correlation linear scale, where an increase in the dancers’ heart rate will produce faster playback, and a negative correlation linear scale, where an increase in the dancers’ heart rate will produce slower playback. The playback speed scaling factor is continuously adjusted in real time based on the incoming heart rate data. **Supplementary** **Table 1** shows how the duration of the songs varies from repetition to repetition. Using Max’s sfplay~ object, any audio file may be loaded into the software. Playback is controlled in one of two modes: playback speed, where slower playback speed produces a lower pitch, as in the playback of analog recordings, or time-stretch mode, where playback speed and pitch are decoupled. The latter is what was used in the current experimental procedure.

**[Place supplementary Table 1 here]**

## Experimental Procedure

The task consisted of two parts. Part 1 was to perform a well-rehearsed routine staging a choreography. Part 2 was to spontaneously improvise. The dancers had to perform each song at each original version (baseline) and then do so two more consecutive times, while the song was blended with the real-time speed of the heartbeat stream.

The design of this experiment was extended to works 1-5

# EXAMPLE 2: AUDIO-VISUAL CLOSE-LOOP INTERFACE OF AN ARTIFICIAL DYADIC INTERACTION

The participant faces a large screen monitor and sees the 3D rendered image of the moving avatar (as in **Figure 2B** of the main text) endowed with the real-time movements’ output registered by the motion-capture system using active LEDS. As the person moves mirroring the avatar, we manipulate the environment by embedding position-dependent sounds within the region surrounding the person. The person is not aware of these sound-triggering positions, denoted regions of interest, RoI. We define such RoIs across the space where the person moves. As some body part (e.g. the hip) passes through a RoI, the computer program toggles music play. If the person sustains the pose with the body part in that RoI, the computer program continuous playing the music, but if the person moves the body part away from the RoI, then the music stops playing. This can turn the search for RoIs denoting musical spots into a game across space. The idea is to raise awareness about several aspects of actions and environmental cues that we automatically process beneath awareness.

To further play with the person’s ability to build maps between body poses and musical bits, we also may anchor the RoI to a specific body part and play the music faster or louder, etc. as a function of the person’s movement. For example, we can play the music faster or slower, depending on how close or far the hand is by the center of the RoI. Once the person finds the RoI, we can shift its location in space and define a new RoI to gamify the session and evoke curiosity. Importantly the person is naïve to these experimental goals and has to figure them out by self-discovering what the goals are (as autistic children did in 6).

## Participants

Six young adults participated in this experiment. They signed the consent form approved by the IRB of Rutgers University, in compliance with the Helsinki act.

## Set-up Audio-Visual Close-Loop Interface

Set-up kinematic and EEG equipment -ANS, PNS, and CNS

A motion-capture system was used to record bodily positions using an LED based motion-tracking system. The EEG was used to record the brain activity and heart signal. To prepare the equipment, we followed all the steps presented in sections 2.1, 2.2, and 2.3 of “Set-up of the Close-loop Interface” of the main text.

Preparation and set-up of LSL for synchronized data recording and streaming of data

For the synchronous recording of the data in LSL, we followed the steps presented in section 2.4, 2.5, and 2.6 of “Set-up of the Close-Loop Interface” of the main text.

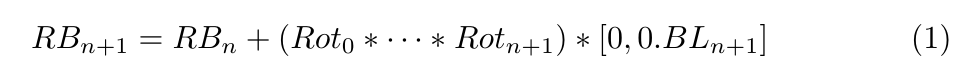
The real-time analyses of data and monitoring of the human system

We developed our methods to parameterize the fluctuations in the efferent motor output

Visual 3D representation of the human body

To represent the human body in 3D, we developed an avatar using MATLAB graphics. To that end, we obtained the positions of 23-body-part skeleton streamed from motion-capture system (**Figure 8** step 2 of the main text) to LSL and then from LSL to MATLAB (**Figure 8** step 3 of the main text). The position data of the skeleton coming in chunks were used to estimate the rotations between the body-parts and build a forward kinematic map. The full-body forward kinematic map was developed based on the arm model described in 7. Using the hip as the origin of the reference frame to build the skeleton (bone 0, **Figure 8** step 3 of the main text), we developed 5 kinematic chains spine-head, right arm, left arm, right leg, left leg. The left and right arms were initiated by the bone 2 (**Figure 8** step 3 and 4 of the main text), whereas the rest of the chains were initiated by the hips (bone 0, **Figure 8** step 3 of the main text).

To estimate the positions of the rigid bodies in MATLAB, we set the position of the origin rigid body equal to the position of bone 0, at the hips. Thus, *RBo=B0,* where *RBn* is the position of rigid body part *n* and *Bn* is the position of bone *n* as it is originally streamed from LSL. To estimate the positions of the rest of the rigid bodies we used Equation (2).



(2)

where *BLn* in the 3x1 vector is the bone length of bone *n* and the 3x3 matrix *Rotn* is the rotation matrix of bone *n.* Equation (2) is the generalized model of the forward-kinematic map presented in 7 which was designed to model human arm motions only. We here extend the previous work to a full-body model. In the arm model the kinematic chain starts from shoulder, whereas our full body version has 5 kinematic chains, three of which start from the hips, bone 0, and 2 start from the bone 2. Shown in **Figure 8** step 4 (of the main text) after the kinematic chain of spine-head is estimated.

We developed the forward kinematic map of the Avatar in MATLAB (version 2016b, The MathWorks, Inc., Natick MA, USA.). The code can be found here:

<https://github.com/VilelminiKala/CloseLoopInterfaceJOVE>

Audio parameterization

The parameterization of the audio feedback was different across conditions. These are listed in **Supplementary Figure 1**.

**[Place supplementary Figure 1 here]**

In the 1st condition, the sound was activated when the participant walked through a specific area defined by the proxy controller. The 3D center position of this area was entered by the researcher by clicking the proper button on the proxy controller (user interface) during the recording setting. The instant position of the hips of the moving participant was then set as the center of the song-activation body part. The system could also use other body parts as the anchor to manipulate the music. The researchers could also manually set the radius of that RoI before starting the recording to define a volume between the desired position and the position of the body part. In our example, the radius was set to 50cm. As the participant found and maintained their position within the RoI, the song would play for as long as the participant stayed within that volume.

In the 2nd condition, the hips once again controlled the activation of the song, but the sampling rate of the song’s playback was defined by the distance between the center of the activation area and the hand. Specifically, we created a grid of zones using distances at every 10cm. In the case of radius = 50cm, there were 5 zones of distance. The min sampling rate was set to a minimum of 95 frames per second and the maximum sampling rate was set to the original sampling rate. The interface could create a grid of n zones around the hip, 5 in our example, based on hand-to-activation-point distance of each person, and accordingly, modulate the sampling rate of the music playback.

In the 3rd condition, the speed of the participant’s hips, was used to control the sampling rate of the song. In this condition, the interface created 5 zones of sampling rate ranging from a minimum of 1000 frames per second to the maximum set as the original sampling rate. The sampling rate was set to minimum when speed was smaller than Vmin value and to maximum when speed was greater than Vmax. To define these two parameters, the speed ranges of the hip of each participant were previously recorded and extracted from their walking pattern. Thus, Vmin and Vmax was chosen (by the researchers) to belong in the individuals speed range but also based on how sensitive or stable the system should be, depending on its use. The completion of each task was defined by the participant’s self-discovery of the interaction rules, while maintaining their speed steady, to be able to listen to the song. We underscore that the participants were naïve to what they needed to self-discover.

Generation of the Sensory Feedback

Visual Feedback

The proxy controller generates a real-time 3D representation of the participant, an avatar, which is used as a mirror, with the purpose of enhancing the person’s kinesthetic feedback and self-awareness. **Supplementary Figure 1** shows the setup of the visually driven proxy control.

The MATLAB code (version 2016b, The MathWorks, Inc., Natick MA, USA.) that uses the presented forwards kinematic map and generates the avatar presented in **Supplementary Figure 1** and **Figure 8** of the main text can be located in <https://github.com/VilelminiKala/CloseLoopInterfaceJOVE>

Audio Feedback

The music was played back through the Matlab interface (Matlab version 2016b, The MathWorks, Inc., Natick MA, USA.). The MATLAB function used for playing the song was: *audioplayer(Y, Fs)*, where Y is the time series song signal and Fs is the sampling frequency.

## Experimental Procedure

The participants were naïve as to the purpose of the study. They had to walk around the room and figure out how to control the sound that would surprisingly emerge as they passed by a RoI that the proxy controller defined.

In condition 1, the position of the hips controls the activation of the music. As the participant’s hips entered a RoI that the proxy controller defined, the music played. The music playing occurred as a direct consequence of the person’s action (i.e. moving into the RoI). Since this action’s consequence surprised the participant, she initiated the search for the “magic” position that played music (i.e. the antecedent to the consequence.) During this exploratory motion, the activity corresponding to the search in the space outside the RoI was processed and compared to the activity harnessed while being inside the RoI. The exploratory *vs.* target activity bear different contextual signatures of motor-based feedback. As such, it was important to characterize them for each person.

In condition 2, the position of the hips controls the activation of the music, as in condition 1, but now the distance of the right hand from the center of the hip-anchored RoI served to set the control of the sampling rate to play the song back faster or slower.

In condition 3, the walking speed at the hips was used to control the sampling rate of the music playback: faster speed led to faster song playback.

The design of the interface was developed as part of the first author’s PhD Thesis 2, and it is also presented also in 3,8. It stems from US patented technology found at

<https://patents.google.com/patent/US10176299B2/en?inventor=Elizabeth+B.+TORRES>, which was initially applied to new sensory-motor based interventions for autism 6.

# DATA TYPES AND ANALYSES

In our representative examples, the signal analyses (the results of which are presented in section “Representative Results” of the main text) consist of two main steps: first we create a data type called the micro-movement spikes (MMS trains) and second, we use the spikes as input to a Gamma process. We use this process to update the stochastic signatures denoting the evolution of the noise-to-signal ratio and track the changes in the shape and the scale of the empirically defined family of continuous probability distribution functions (PDFs).

## Micro-movement Spikes

The MMS are derived from the fluctuations in the amplitude of the peaks of the time series signal. They can also be derived from the fluctuations in the inter-peak interval times. In this work, we use the MMS trains derived from the fluctuations in both amplitude and timing related signals. For example, the raw peak amplitude is normalized using equation (1) 9:

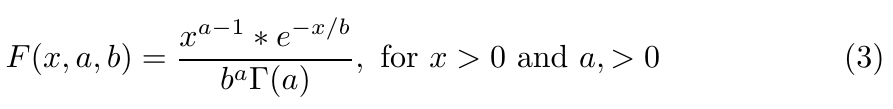
(1)

Here the Peak refers to the amplitude of the local peak and min to min refers to the two neighboring minima surrounding the local peak. All points sampled between these local minima (including the local peak) are averaged and added to the local peak, to build the denominator term. Importantly, the fluctuations are computed relative to the empirically derived overall mean of each window of data (under independent identically distributed iid assumption.)

The MMS, which are real numbers ranging in the [0,1] real interval maintain the information of the peak timing and/or amplitude. At the same time, they enable us to treat the time series as a random process. We use a Gamma process under the general rubric of Poisson random processes. Poisson-based analyses are commonly used in the field of computational neuroscience aiming to analyze binary spikes (*e.g.* from cortical signals.) We have adapted this approach to study continuous MMS trains derived from peripheral signals 10-14

## ****Gamma Distribution****

A random variable *X* that is Gamma distributed with shape *a* and scale *b* is denoted by *X~Γ(a,b) = Gamma(a,b)* with probability density function:



Here x is the random variable of interest and Г is the Gamma function. The Gamma mean *μ = a\*b* and the variance with the noise to signal ratio given by equation (4)

*NSR = σ/μ = b* (4)

The NSR is thus the scale parameter in the continuous Gamma family. In this work, we use the

continuous data stream of normalized peak values derived from the fluctuations in the heart

signal amplitude and the music to estimate the Gamma mean for each block of data buffered (as explained above) and to obtain the Gamma moments.

We track the shape and scale parameters empirically estimated each time step, using MLE with 95% confidence intervals.

Maximum Likelihood Estimation for each condition shows best fitting for the continuous Gamma

distribution with maximum value at 0 gradient convergence, see **Supplementary Figure 2**. Frequency histograms of the MMS are shown as insets with fitted distributions using the MATLAB distribution fitting app.

**[Place supplementary Figure 2 here]**

## Stochastic Analyses

These methods have been explained in previous work4,8,10,11,13-18 We define the Gamma parameter plane with axes described by the shape and scale parameters of the Gamma probability density functions (PDFs) that we empirically estimate using the MMS (see **Figure 9**  of the main text). We also plot the corresponding Gamma moments because in our experience, they offer good visualization and automatic clustering of various populations and/or contexts, parameters, etc. Specifically, on a four-dimensional graph, we plot the mean, variance and skewness along the X, Y, Z dimensions respectively (see **Figure 6C** of the main text). We represent the point by a marker and scale the size of the marker proportional to the kurtosis, as the fourth dimension. We may also color the marker’s face and edge using raw data ranges and derive other indexes to represent the phenomena in more compact formats using other parameter spaces.

For each case, the MMS trains are gathered in a frequency histogram. Then empirical distribution fitting is performed using maximum likelihood estimation (MLE) with 95% confidence intervals. We estimate the best continuous family of PDFs for the data at hand. In MMS derived from human biorhythms, the continuous Gamma family of PDFs has been adequate to fit these frequency histograms, see **Supplementary Figure 2 and 3**. Then we represent the Gamma parameters in a parameter plane: the shape denoting the distribution’s shape, and the scale denoting the dispersion (noise to signal ratio), see **Supplementary Figure 3**.

This empirical estimation captures the signatures of biorhythmic fluctuations of the person’s nervous systems for each of the signals that each of the instruments registers. This approach contrasts with traditional methods assuming *a priori* some theoretical distribution. In human data, it has been our experience that a log-log transform of the data aligns the points along a line of unity across multiple decades of the shape parameter values (a power law) with tight fit. In such cases, we reduce the parameters of interest to one (the noise to signal ratio, or Gamma scale) since, knowing the noise, we can safely infer the shape of the distribution 19,20. These estimations are performed for each context and condition and tracked in real time across the experimental session. **Supplementary Figure 3** shows the pipeline of these analyses using traces from the hip’s motions as an example. We have characterized across disorders of the nervous systems the healthy regimes of noise and identified phase transitions denoting learning and adaptation regimes differentiating novice- from expert-like signatures (see **Supplementary Figure 3F** and 19).

**[Place supplementary Figure 3 here]**

# FIGURES AND TABLES

**Supplementary Table 1: The duration of the original and the altered songs.** The speed of the altered songs is continuously updated during the experiment based on the ongoing speed of the heartbeat. Therefore, none of the altered recording matches neither with the original song nor with other altered versions of the same song

**Supplementary Figure 1:** **Conditions of the audio-visual interface**. In the first condition, the position of hips activates the music when they are in a spherical RoI (radius 50cm around the hip which is set as the origin). In the second condition, the position of hips activates the music as in condition 1 and the distance of the right hand from the center of the RoI defines the sampling rate that plays the song. In the third condition, the speed of the hips walking around the room sets the sampling rate of the song playback.

**Supplementary Figure 3: Standardized MMS data type and analytical pipeline.** (A) Sample 3D trajectory from a light emitting diode (LED) located on one of the hips, harnessing motion as the person dances to music. Red dots indicate the peaks of the linear speed. (B) Linear speed profiles with peaks signaling maxima, change in slope (positive to negative) as the body moves (accelerates and decelerates) for 166.66 seconds (2.77 minutes, sampling at 480Hz with the motion-capture system active LED-camera system). (C) Micro-movement spikes (normalized peaks extracted from the linear speed, marked in red in (B) using equation (1) and obtaining the absolute deviations from the mean amplitude. Full MMS trains including all peaks and pauses of the 4.5x104 frames. (D) Frequency histograms of the raw (cm/frame) and normalized (unitless) MMS. The bin size of the histograms is defined by the Scott’s rule. (E) Best fitting Gamma PDFs using MLE and 95% confidence intervals to estimate the shape and scale parameters and the corresponding histograms of two extreme cases. The bin size of the graphs has been estimated by Scott’s rule (F) Gamma parameter plane showing the three corresponding (shape, scale) points and their empirical determined interpretation along the color scale.

# REFERENCES

1 Kalampratsidou, V. & Torres, E. B. Sonification of heart rate variability can entrain bodies in motion. *Proceedings of the 7th International Conference on Movement and Computing.* 10.1145/3401956.3404186 Article 2, (2020).

2 Kalampratsidou, V. *Co-adaptive multimodal interface guided by real-time multisensory stochastic feedback*, Rutgers University-School of Graduate Studies, (2018).

3 Kalampratsidou, V., Kemper, S. & Torres, B., Elizabeth. Real time streaming and closed loop co-adaptive interface to steer multi-layered nervous systems performance. *48th Annual Meeting of Society for Neuroscience.* (2018).

4 Kalampratsidou, V. & Torres, E. B. Bodily Signals Entrainment in the Presence of Music. *Proceedings of the 6th International Conference on Movement and Computing.* 3, (2019).

5 Kalampratsidou, V., Zavorskas, M., Albano, J., Kemper, S. & Torres, E. B. Dance from the heart: A dance performance of sounds led by the dancer's heart. *Sixth International Symposium on Movement and Computing.* (2019).

6 Torres, E. B., Yanovich, P. & Metaxas, D. N. Give spontaneity and self-discovery a chance in ASD: spontaneous peripheral limb variability as a proxy to evoke centrally driven intentional acts. *Frontiers in integrative neuroscience.* **7** 46, (2013).

7 Torres, E. B. *Theoretical Framework for the Study of Sensori-motor Integration.*, University of California, San Diego, (2001).

8 Kalampratsidou, V. & Torres, E. B. Body-brain-avatar interface: a tool to study sensory-motor integration and neuroplasticity. *Fourth International Symposium on Movement and Computing, MOCO.* **17**, (2017).

9 Lleonart, J., Salat, J. & Torres, G. J. Removing allometric effects of body size in morphological analysis. *J Theor Biol.* **205** (1), 85-93, (2000).

10 Torres, E. B., Vero, J. & Rai, R. Statistical Platform for Individualized Behavioral Analyses Using Biophysical Micro-Movement Spikes. *Sensors (Basel).* **18** (4), (2018).

11 Torres, E. B. & Denisova, K. Motor noise is rich signal in autism research and pharmacological treatments. *Scientific Reports.* **6** 37422, (2016).

12 Torres, E. B. Two classes of movements in motor control. *Experimental Brain Research.* **215** (3-4), 269-283, (2011).

13 Torres, E. B. *et al.* Characterization of the statistical signatures of micro-movements underlying natural gait patterns in children with Phelan McDermid syndrome: towards precision-phenotyping of behavior in ASD. *Frontiers in integrative neuroscience.* **10** 22, (2016).

14 Wu, D. *et al.* in *APS March Meeting Abstracts.*

15 Torres, E. B. *Objective Biometric Methods for the Diagnosis and Treatment of Nervous System Disorders*. (Academic Press, Elsevier, 2018).

16 Torres, E. B. *et al.* Toward Precision Psychiatry: Statistical Platform for the Personalized Characterization of Natural Behaviors. *Frontiers in neurology.* **7** 8, (2016).

17 Kalampratsidou, V. & Torres, E. B. Outcome measures of deliberate and spontaneous motions. *Proceedings of the 3rd International Symposium on Movement and Computing.* 9, (2016).

18 Kalabratsidou, V. & Torres, E. B. Invariant and variable relations emerge with degrees of difficulty within habitual and surprise touch-pointing motions. *Journal of Vision.* **14** (10), 418-418, (2014).

19 Torres, E. B. Signatures of movement variability anticipate hand speed according to levels of intent. *Behavioral and Brain Functions.* **9** 10, (2013).

20 Torres, E. B. *et al.* Autism: the micro-movement perspective. *Frontiers in integrative neuroscience.* **7** 32, (2013).