Transfer Entropy for Feature Extraction in Physical Human-Robot Interaction: Detecting Perturbations from...

Conference Paper · October 2014

DOI: 10.1109/HUMANOIDS.2014.7041459

Erik Berger
Technische Universität Bergakademie Freiberg

David Vogt
Technische Universität Bergakademie Freiberg

Heni Ben Amor
Arizona State University

Transfer Entropy for Feature Extraction in Physical Human-Robot Interaction: Detecting Perturbations from Low-Cost Sensors

Erik Berger[1], David Müller[1], David Vogt[1], Bernhard Jung[1], Heni Ben Amor[2]

***Abstract*— In physical human-robot interaction, robot behavior must be made robust against forces applied by the human interaction partner. For measuring such forces, special-purpose sensors may be used, e.g. force-torque sensors, that are however often heavy, expensive and prone to noise. In contrast, we propose a machine learning approach for measuring external perturbations of robot behavior that uses commonly available, low-cost sensors only. During the training phase, behavior-specific statistical models of sensor measurements, so-called perturbation filters, are constructed using Principal Component Analysis, Transfer Entropy and Dynamic Mode Decomposition. During behavior execution, perturbation filters compare measured and predicted sensor values for estimating the amount and direction of forces applied by the human interaction partner. Such perturbation filters can therefore be regarded as virtual force sensors that produce continuous estimates of external forces.**

I. Introduction

Autonomous robots require accurate sensing capabilities in order to act in an intelligent and meaningful way within their environment. In particular human-robot interaction tasks require sensors for measuring physical contact with a human partner. Recorded measurements can be used by a robot to ensure safety during interactions and to react to physical perturbations. To this end, it is important that both the occurrence as well the magnitude of an external perturbation, e.g., a push, are reliably detected. Existing sensing technologies, such as force-torque sensors, are often heavy, expensive, and noise-prone. However, there are numerous affordable low-cost sensors available which, while not directly measuring perturbation forces, can be used to generate estimates of external perturbations.

In this paper, we present an approach for perturbation detection which is based on a combination of low-cost sensors and machine learning techniques. During a training phase, we extract a compact representation, called a *perturbation filter*, which specifies the evolution of sensor readings during regular execution of a motor skill. The extraction is guided by information-theoretic measures such as Transfer Entropy, that determine the relevance of a specific sensor w.r.t. the executed robot behavior. In contrast to our previous work [1], we will not use any higher level stability parameters, such as the center-of-mass, center-of-pressure, or zero-moment-point for learning. Instead, we will learn the perturbation filter from low-level sensor data, solely. As a result, no knowledge about the robot kinematics or dynamics is required.

[1]Institute of Computer Science, Technical University Bergakademie Freiberg, Bernhard-von-Cotta-Str. 2, 09599 Freiberg, Germany

[2]Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, 801 Atlantic Drive, Atlanta, GA 30332-0280, USA

Fig. 1: A NAO robot estimates the influence of external perturbations applied by a human interaction partner to its current behavior execution.

After a perturbation filter is learned, it is used to generate a continuous estimate of the amount of external human perturbations. During physical interaction between a robot and a human, the estimated perturbations can be used to compensate for the external forces or infer the intended guidance of a human interaction partner. The presented perturbation filter can be regarded as a virtual force sensor that produces a continuous estimate of external forces.

II. Related Work

In recent years, natural and intuitive approaches to HRI have gained popularity. Various researchers have proposed the so-called *soft robotics* paradigm: compliant robots that "can cooperate in a safe manner with humans" [2]. An important robot control method for realizing such a compliance is impedance control [3]. Impedance control can be used to allow for touch based interaction and human guidance. To this end, impedance controllers require accurate sensing capabilities, in the form of force-torque sensors. However, such sensors are typically heavy, expensive and suffer from significant noise. Other sensors, such as torque sensors are

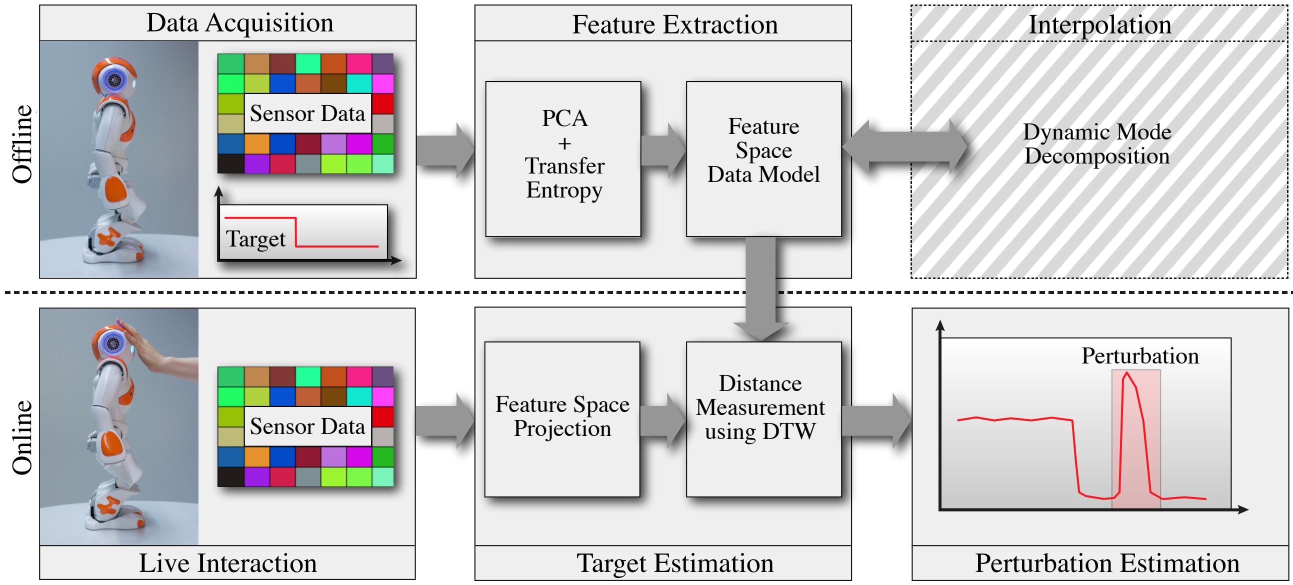

Fig. 2: An overview of the presented machine learning approach. Training data, together with a labeling target vector will be processed using *Principal Component Analysis*, *Transfer Entropy* and *Dynamic Mode Decomposition* algorithms, providing a training data model of vectors comprising the *Feature Space*. During live interaction, the recorded data is being projected into this space and mapped to the nearest data model vector and its target vector using *Dynamic Time Warping*.

even more prone to issues related to noise and drift.

Still, the ability to sense physical influences is at the core of recent advances made in the field of HRI. For example, Lee et al. [4] use impedance control and force-torque sensors in order to realize human-robot interaction during programming by demonstration tasks. Wang et al. [5] present a robot adapting its dancing steps based on the external forces exerted by a human dance partner. Ben Amor et al. [6] use touch information to teach new motor skills to a humanoid robot. Touch information is only used to collect data for subsequent learning of a robotic motor skill. Robot learning approaches based on such kinesthetic teach-in have gained considerable attention in the literature, with similar results reported in [7] and [8]. A different approach aiming at joint physical activities between humans and robots has been reported in [9]. Ikemoto et al. use Gaussian mixture models to adapt the timing of a humanoid robot to that of a human partner in close-contact interaction scenarios. This approach significantly improves physical interactions, but is limited to learning timing information.

Stückler et al. [10] present a cooperative transportation task where a robot follows the human guidance using arm compliance. In doing so, the robot recognizes the desired walking direction through visual observation of the object being transported. A similar setting has been investigated by Yokoyama et al. [11]. They use a HRP-2P humanoid robot with a biped locomotion controller and an aural human interface to carry a large panel together with a human. Forces measured with sensors on the wrists are utilized to derive the walking direction.

The main disadvantage of the above approaches is that they require special aural and visual input devices or force sensors, which are not present on many robot platforms. Additionally, none of the approaches using force-torque sensors addresses the problem of uncertainty in the measurements. As a result, all of these approaches assume high-quality sensing capabilities and low-speed execution of the joint motor task. We propose a new filtering algorithm that can *learn* the natural variation in sensor values as a motor skill is executed.

III. Approach

The objective of the presented method is to estimate the strength and direction of external perturbations caused by a human interaction partner. To infer these estimates from low-cost sensor readings, we condition behavior-specific perturbation filters. An overview of the approach can be seen in Figure 2. First, we record training data for a behavior with different parameter configurations, e.g., varying step lengths during walking. In this data acquisition phase, no external perturbations from humans are applied. Thereafter, the training data is used to create a *Feature Space data model* during feature extraction. Linear combinations of different sensors are weighted by their relevance to the observed parameter and projected into the low-dimensional Feature Space. In the following, the configuration parameter will be referred to as the *target vector*. The relevance of a specific sensor to the target vector is extracted using Transfer Entropy [12] (TE). In this context, TE is used as a measure of predictability and information flow between the target vector and the conduct of sensors. Sensors that have a high TE w.r.t. the robot's behavior are deemed more influential and relevant.

During behavior execution, an external perturbation is de-

tected by comparing the recorded training data to the current sensor data within the low-dimensional Feature Space. Dynamic Time Warping (DTW) [13] is used as a distance function in order to include the temporal pattern for the comparison. The estimation of a *perturbation value* is performed by comparing the current sensor readings to the sensor readings acquired during training. The perturbation value is then inferred from the difference between the currently configured behavior parameter, e.g., the currently employed step length, and the estimated behavior parameter which produced similar sensor readings during training.
In the following section, we will depict each step of our approach in more detail. We will describe how to perform feature space extraction and how to use the resulting embedding to estimate a continuous perturbation value.

A. Data Acquisition

The first step in our approach is to record training data that reflects the evolution of sensor values during the regular execution of a motor skill. To this end, we perform the investigated motor skill with varying parameter values, e.g. varying step lengths during walking. For generalization purposes, it is important to record the motor skill under large a set of possible target parameter configurations. However, since the parameter space may have a dynamic range, this can lead to a time-consuming recording phase, which in consequence leads to wear and tear of the robot hardware. To avoid a lengthy training session, Dynamic Mode Decomposition can be used to learn a model of the sensor data using few training samples. This process is not being detailed in this publication, the interested reader is referred to our previous work [14].
The training data is sampled equidistantly with $100Hz$. Please note, that we only record low-level sensor data. Preprocessed variables, such as center-of-mass or the zero moment point are not included in this process. In contrast to our previous work, we will automatically identify and combine relevant low-level sensor data.
To prevent a comparison between sensors of different units (i.e. comparing angles with pressure values), a sensor group is assigned to each sensor, enabling to deduce conclusions from their individual relations.

B. Feature Extraction

The next step in our approach is to extract relevant features from the stream of sensor data. While it is often possible to acquire a large number of different sensor values, we are typically faced with significant redundancy and noise. Additionally, it is often unclear which of these readings we should pay attention to. Feature extraction can help to single out important parts of a sensor stream.
For feature extraction, we first compute a low-dimensional embedding of the sensor data by performing a PCA-like procedure. We extract the principal components of the feature space using an eigenvector decomposition of the sensor data matrix. In traditional PCA the eigenvalues define how much information each eigenvector carries. The eigenvector with highest eigenvalue is the direction with highest variance. Hence, in traditional PCA the relevance of a feature is defined by the observed variance along that dimension.
Instead of using the variance to infer the relevance of a feature, we will focus on the relationship between the feature values and the future state of the robot. Features that have a strong statistical coherence with the future state of the robot are more likely to be relevant. In other words, a feature is deemed relevant, if its past activity is a good predictor of the robot's next state. From an information theoretic point of view, this type of relationship can be estimated using Transfer Entropy. [12]
We employ TE in order to measure the directed information transfer between the target and each PC vector separately. TE is a recently introduced information-theoretic measure, which has been used extensively in diverse fields of science [15][16][17].

$$TE_{J\rightarrow I} = \sum p(i_{t+1}, i_t, j_t) log_2 \frac{p(i_{t+1}|i_t, j_t)}{p(i_{t+1}|i_t)} \qquad (1)$$

TE quantifies the incorrectness of the assumption, that in the absence of information flow from system J to system I, the state of J has no influence on the transition probabilities on system I [12].
To compute the TE, the conditional probability as well as joint probability of co-occurrences in J and I is required. To estimate these probabilities without resorting to density estimation we quantize the sensor values and use a frequentist approach. The optimal parameters for the quantization are empirically determined. We quantize each PC vector into 200 value levels and use a quantile based transformation, which has the advantages of stability and independence of input value transformations.
Variations of the robot's target vector could possibly have a time-delayed impact on sensory data. However, the standard TE algorithm only allows to draw conclusions based on transitions of a one-step delay between samples of J and I. A more general approach can be implemented using *Delayed Transfer Entropy* [18] which introduces multiple possible time delays.

$$TE_{J\rightarrow I}(d) = \sum p(i_{t+1}, i_t, j_{t+1-d}) log_2 \frac{p(i_{t+1}|i_t, j_{t+1-d})}{p(i_{t+1}|i_t)} \qquad (2)$$

We calculate the Transfer Entropy peak values $TE^*_{T\rightarrow P_i} = \text{argmax}_{\forall j}\, TE_{T\rightarrow P}(d_j)$ between the target vector T and each principal component P_i over a number of time delays d_j within a preset time delay window D, $\cup_{\forall j} d_j = D$. See Figure 3 for further details. These TE^* peak values are then used to scale the previously acquired principal components, such that components of higher TE^* values are stretched and those of lower TE^* values are shrunk. The resulting scaled PCA space is the so-called *Feature Space*. As mentioned earlier, within the feature space, the relevance of variables depends on their influence on the target vector.
The feature space projection of the acquired training data set is now stored as the feature space data model.

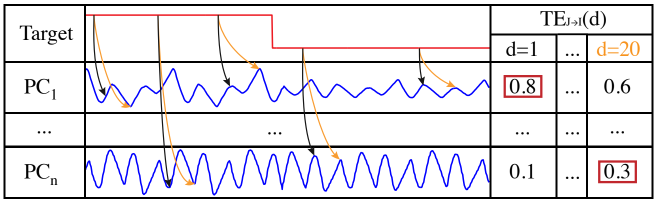

Fig. 3: The Transfer Entropy measures the target vector's (red graph) influence on each PC vector for each time delay (black arrow for d=1, orange arrow for d=20). The TE peak value of each PC component (marked by red squares) is the largest of each delayed TE value.

C. Target Estimation

The next step is to use this data model to continuously estimate current target behavior values and to detect and qualify possible perturbations upon live human-robot interaction. To do so, a time window of raw sensor data is projected into the previously specified low dimensional Feature Space. To identify the most similar segment of the projected training data, we use the *subsequence dynamic time warping* technique (SDTW) [19].

$$v = \operatorname{argmin} \Gamma(X, Y). \tag{3}$$

For this, the SDTW algorithm Γ is measuring the distance between two temporal sequences $X = (x_1, \ldots, x_N)$ and $Y = (y_1, \ldots, y_M)$ of length $N \in \mathbb{N}$ and $M \in \mathbb{N}$. In our specific case, the goal is to find the training data Y with minimal distance to the projection of the currently observed sequence X. As a result, the target behavior value v of the corresponding training data can be used as an estimation for the current one.

Since we captured the behavior for a discrete set of target behavior values we can only make estimations for these. An efficient way to expand our data model to cover continuous behavior parameters can be achieved using interpolation schemes. Therefore, we use a novel interpolation method from fluid dynamics, the so called *Dynamic Mode Decomposition* (DMD) [20][21]. A detailed explanation of DMD and how it can be used for the interpolation of robot sensor data can be found in our previous publication [14].

D. Reaction

Accordingly, we can generate an estimate for possible interfering external forces $\hat{E}$ by calculating the difference between the configured behavior parameter P used to control the robot and the estimated behavior parameter $\hat{P}$ identified by the learned model for each sensor group.

$$\hat{E} = \hat{P} - P \tag{4}$$

Thus, our approach can be used in scenarios where a robot has to detect and react to external perturbations in order to fulfill a specified task. Certain sensor groups will be suitable to qualify certain perturbations, allowing conclusions about perturbation characteristics. This and details about how the robot should react to a perturbation depends on the specific use-case and is left open for further research at this point.

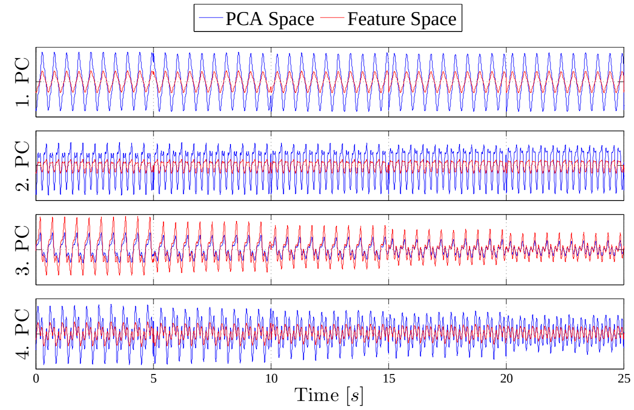

Fig. 4: The principle components (blue) and the principle components scaled with TE (red) of a walking gait's training data. The third principle component has the highest TE value and is stretched while the others are dampened. The scaled principle components comprise the low dimensional Feature Space.

IV. Experiments

In the following section, we evaluated our approach using a NAO robot from *Aldebaran Robotics*. To do so, we recorded a total of 52 seconds of training data from the robot's walking gait with step lengths between 3 cm and -3 cm. Each sample contains readings of the angle and pressure sensors. Next, we interpolated these samples with a resolution of 0.01 cm utilizing Dynamic Mode Decomposition. Retaining 95 % of information we applied Principal Component Analysis on the robots 24 angle sensors and its 8 foot pressure sensors separately resulting in a $4d$-angle space and a $6d$-pressure space. Finally, each principal component is scaled by its delayed Transfer Entropy whereas the target value equals the gaits step-length. The resulting dimensions of the low dimensional Feature Space are shown in Figure 4. While the first, second and fourth dimensions are dampened, the third dimension is stretched. This is due to the fact that PCA extracts the most characteristic properties of a behavior and not its dependency on the adjusted parameter.

Figure 5 shows the vector length of $PC \cdot TE^*$, the resulting *Feature Space vectors* compared to the simultaneous longitudinal center-of-mass which was used extensively in our previous research [1] [14]. Obviously, they are very similar even though our new approach has no knowledge about the kinematic chain or the mass distribution of the robot.

A. Estimation Quality

We utilize the learned low dimensional Feature Space data model during runtime while the robot frequently reduces its step-length. Figure 6 shows the resulting mean absolute error (MAE) for the angle and pressure sensor groups. The angles are especially influenced by spurious relationships, which are strongly dampened by PCA, even without TE.

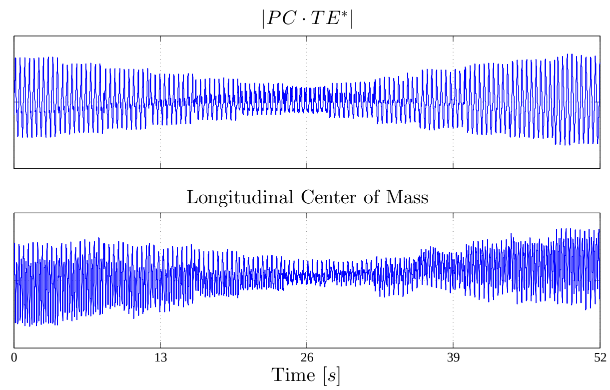

Fig. 5: The vector length of $PC \cdot TE^*$ compared to the simultaneous longitudinal center-of-mass. In contrast to the center-of-mass our approach does not need any knowledge about the robots kinematics or mass distribution to generate a comparable result.

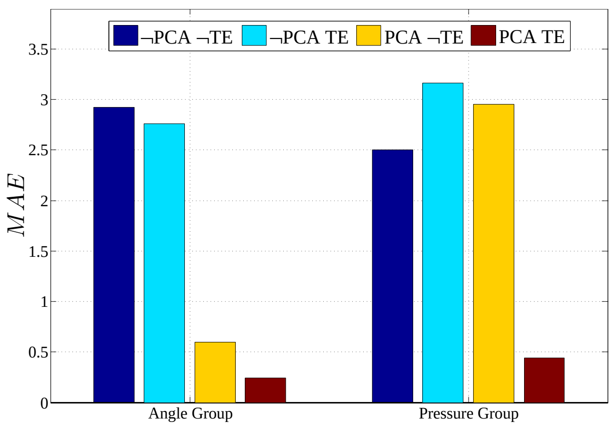

Fig. 6: The mean absolute error prediction results in cm. Left: The angle group error with all permutations of PCA and TE. Right: The pressure group error with all permutations of PCA and TE. Obviously, using a combination of PCA and TE increases the accuracy of the estimation.

Furthermore, as shown by the pressure group, PCA fails to identify the relations between the sensors and the behavior parameter. However, our Feature Space, combining PCA and TE, increases the accuracy of the estimation for both sensor groups.

B. Parameter Estimation

In this experiment, we measure the robot's hardware delay using the 52 seconds of angle training data without interpolation. For this, the step length is reduced from three to one centimeters over a period of 15 seconds with a sliding window of a 0.25 seconds. Figure 7 shows that the robot needs about one second to react to parameter reconfigurations. This indicates, that the robot has an average hardware delay of 0.75 seconds.

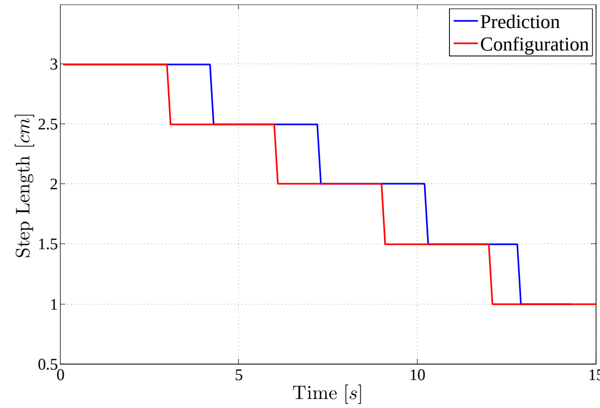

Fig. 7: The configured behavior parameter (red) decreases over time. The estimated behavior parameter (blue) recognized the robot's reactions with a time delay, since the robot needs to finish the current step before adapting to the new parameter.

C. Perturbation Estimation

In this experiment, the human perturbs the robot during execution of a walking gait. To verify the correctness and universality of our approach, the perturbations are applied to different parts of the robot. Figure 8 shows the resulting parameter estimations for the angle and pressure groups as well as the configured parameter value. Perturbation (a) is recognized by both sensor groups, because the angles as well as the force sensors are affected. If no external perturbation is recognized during (b), the parameter estimations of both sensor groups measure almost the configured parameter value. However, in (c) the perturbation does not affect angles and in consequence can't be measured by the angle but by the pressure group. Finally, perturbation (d) leads to a measurement of flat zeros for each pressure sensor, which is not part of the training data set and, consequently, can only be recognized by the angle group.
This confirms our assumption, that redundant sensor groups can help to recognize and qualify a variety of perturbations.

V. Conclusion

In this paper, we presented an approach for estimating external perturbations during physical human-robot interaction tasks. Instead of using expensive force-torque sensors, we leverage available information from low-cost sensors. To this end, we introduced a machine learning approach that can learn behavior-specific perturbation filters in software. In turn, these filters can be used to generate a continuous estimate of the inflicted external perturbations. An information-theoretic measure, in particular Transfer Entropy, is used to guide the feature extraction process. Given a set of low-level sensor data, our approach allows for the automatic identification of relevant sensor values by calculating the information flow from sensors to future robot states. We have shown, that this approach automatically leads to the emergence of features that are remarkably similar to the center-of-mass,

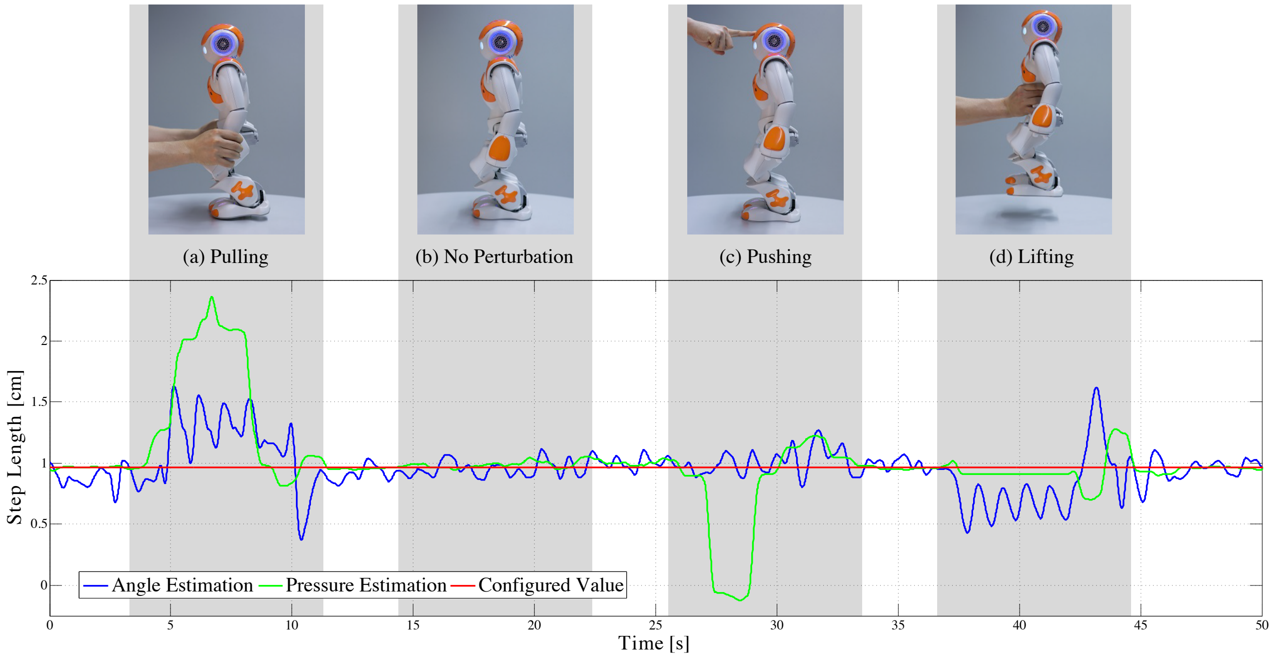

Fig. 8: The estimated perturbation value $\hat{E}$ for each of the external perturbations is the difference between the estimated parameter $\hat{P}$ and the configured parameter value P, as defined in Formula 4. (a) can be detected by both the angle and the pressure sensors while the perturbations in (c) and (d) can only be detected by one of the sensor groups.

without actually having to provide prior knowledge about the robot kinematics or mass distributions. The automatic determination of these features is important, since manufacturer-supplied mass distributions are effectively invalidated in tasks where the robot is carrying weights. Further characterization of causing effects and details about how to react to certain perturbations should be investigated in future work. Using spatial groupings of sensors on the robot could be used to localize the external influences.

References

[1] E. Berger, D. Vogt, N. Haji-Ghassemi, B. Jung, and H. Ben Amor, "Inferring guidance information in cooperative human-robot tasks," in *Humanoids'13*, 2013.

[2] A. Albu-Schäffer, O. Eiberger, M. Fuchs, M. Grebenstein, S. Haddadin, C. Ott, A. Stemmer, T. Wimböck, S. Wolf, C. Borst, and G. Hirzinger, "Anthropomorphic soft robotics from torque control to variable intrinsic compliance," in *Robotics Research*, ser. Springer Tracts in Advanced Robotics, C. Pradalier, R. Siegwart, and G. Hirzinger, Eds. Springer Berlin Heidelberg, 2011, vol. 70, pp. 185–207.

[3] S. Haddadin, *Towards Safe Robots - Approaching Asimov's 1st Law*. Springer, 2014.

[4] D. Lee and C. Ott, "Incremental kinesthetic teaching of motion primitives using the motion refinement tube," *Autonomous Robots*, vol. 31, no. 2-3, pp. 115–131, 2011.

[5] H. Wang and K. Kosuge, "Control of a robot dancer for enhancing haptic human-robot interaction in waltz," *IEEE Trans. Haptics*, vol. 5, no. 3, pp. 264–273, Jan. 2012.

[6] H. Ben Amor, E. Berger, D. Vogt, and B. Jung, "Kinesthetic bootstrapping: Teaching motor skills to humanoid robots through physical interaction," in *KI 2009: Advances in Artificial Intelligence*. Springer Berlin Heidelberg, 2009, pp. 492–499.

[7] S. Calinon, *Robot programming by demonstration: A probabilistic approach*. EPFL Press, 2009.

[8] J. Kober and J. Peters, "Policy search for motor primitives in robotics," *Machine Learning*, vol. 84, no. 1, pp. 171–203, 2011.

[9] S. Ikemoto, H. Ben Amor, T. Minato, B. Jung, and H. Ishiguro, "Physical human-robot interaction: Mutual learning and adaptation," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 24–35, 2012.

[10] J. Stückler and S. Behnke, "Following human guidance to cooperatively carry a large object," in *Humanoids'11*, 2011, pp. 218–223.

[11] K. Yokoyama, H. Handa, T. Isozumi, Y. Fukase, K. Kaneko, F. Kanehiro, Y. Kawai, F. Tomita, and H. Hirukawa, "Cooperative works by a human and a humanoid robot," in *Proceedings. ICRA '03. IEEE International Conference on Robotics and Automation 2003.*, vol. 3, 2003, pp. 2985–2991 vol.3.

[12] T. Schreiber, "Measuring information transfer," *Physical Review Letters*, vol. 85, no. 2, pp. 461–464, 2000.

[13] H. Sakoe, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, pp. 43–49, 1978.

[14] E. Berger, M. Sastuba, D. Vogt, B. Jung, and H. Ben Amor, "Dynamic mode decomposition for perturbation estimation in human robot interaction," in *RO-MAN, 2014 IEEE*, to appear 2014.

[15] S. Ito, M. E. Hansen, R. Heiland, A. Lumsdaine, A. M. Litke, and J. M. Beggs, "Extending transfer entropy improves identification of effective connectivity in a spiking cortical network model," *PloS one*, vol. 6, no. 11, p. e27431, 2011.

[16] M. P. Machado, C. W. Kulp, and D. Schlingman, "Composition and analysis of music using mathematica," *Mathematica in Education and Research*, p. 1, 2007.

[17] S. K. Baek, W.-S. Jung, O. Kwon, and H.-T. Moon, "Transfer entropy analysis of the stock market," *arXiv preprint physics/0509014*, 2005.

[18] S. Ito, M. E. Hansen, R. Heiland, A. Lumsdaine, A. M. Litke, and J. M. Beggs, "Extending Transfer Entropy Improves Identification of Effective Connectivity in a Spiking Cortical Network Model," *PLoS ONE*, vol. 6, no. 11, pp. e27 431+, Nov. 2011.

[19] M. Müller, *Information Retrieval for Music and Motion*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.

[20] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of Fluid Mechanics*, vol. 656, pp. 5–28, 8 2010.

[21] C. W. Rowley, I. Igor Mez, I. Shervin Bagher, R. Philipp Schlatte, and D. S. Henningson, "Spectral analysis of nonlinear flows," *Journal of Fluid Mechanics*, vol. 641, no. -1, pp. 115–127, 2009.