

FULL PAPER

Estimation of perturbations in robotic behavior using dynamic mode decomposition

Erik Berger^{a*}, Mark Sastuba^b, David Vogt^a, Bernhard Jung^a and Heni Ben Amor^c

^aInstitute of Computer Science, Technical University Bergakademie Freiberg, Freiberg, Germany; ^bInstitute of Mechanics and Fluid Dynamics, Technical University Bergakademie Freiberg, Freiberg, Germany; ^cInstitute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA, USA

(Received 31 March 2014; revised 16 September 2014; accepted 14 October 2014)

Physical human–robot interaction tasks require robots that can detect and react to external perturbations caused by the human partner. In this contribution, we present a machine learning approach for detecting, estimating, and compensating for such external perturbations using only input from standard sensors. This machine learning approach makes use of *Dynamic Mode Decomposition* (DMD), a data processing technique developed in the field of fluid dynamics, which is applied to robotics for the first time. DMD is able to isolate the dynamics of a nonlinear system and is therefore well suited for separating noise from regular oscillations in sensor readings during cyclic robot movements. In a training phase, a DMD model for behavior-specific parameter configurations is learned. During task execution, the robot must estimate the external forces exerted by a human interaction partner. We compare the DMD-based approach to other interpolation schemes. A variant, sparsity promoting DMD, is particularly well suited for high-noise sensors. Results of a user study show that our DMD-based machine learning approach can be used to design physical human–robot interaction techniques that not only result in robust robot behavior but also enjoy a high usability.

Keywords: physical human–robot interaction; dynamic mode decomposition; model learning; external perturbation; usability in human–robot interaction

1. Introduction

The development of autonomous robots that adequately interact with their surroundings requires an accurate, reliable, and efficient sensing technology. Acquired sensory information needs to be included in the decision-making process in order to adapt to the current situation, or more generally, to influences from the environment. In particular, close contact physical interaction and cooperation between robots and humans require adequate sensory inputs. In such scenarios, forces and torques that are applied by the human partner can significantly perturb the execution of a robot's motor skills, resulting in a failure of the cooperative task. These external perturbations need to be estimated and addressed in the decision-making process to ensure a stable, safe, and successful execution of motor skills.

Hence, recent control approaches employ force sensing technology to realize compliant robot motions.[1,2] Impedance control in particular has been successfully applied to physical human–robot interaction. In the typical impedance control setup, a force–torque sensor mounted at the end effector is used to measure external perturbations and interactions with the environment. Mounting the sensing device in such a way restricts the compliant behavior to forces acting on the end effector. Various methods have

been proposed to extend the ‘sensitivity’ to the entire robot body.[3]

However, extending the sensitivity also makes it difficult to differentiate between natural variations in the sensor readings and external perturbations that need to be detected. In particular, during the execution of dynamic motions, e.g. walking or running, the sensors will have continuously varying readings that stem from the contact forces with the ground. In these situations, it can be challenging to discriminate between human inflicted perturbations, natural variation of the readings due to the execution of the behavior, and sensor noise. Also, in order to detect the degree by which an external perturbation occurred, we need to be able to calculate the difference between the expected state, referred to as the zero state, of the sensor readings and the current value.[4] As already pointed out, the zero state is constantly changing during the execution of dynamic, physical tasks. As a result, it becomes difficult to generate accurate estimates of the external perturbations acting on the robot.

In this paper, we present an alternative sensing approach that is based on machine learning. We are particularly interested in sensing human perturbation in dynamic tasks, in which a robot is physically interacting with the environment

*Corresponding author. Email: erik.berger@informatik.tu-freiberg.de

and a human partner at the same time. The proposed approach focuses on learning probabilistic, *behavior-specific* models of regular oscillations in sensor readings during motor skill execution. These models are used to (1) identify perturbations by detecting irregularities in sensor readings that cannot be explained by the inherent noise or the executed task and (2) to generate a continuous estimate of the amount of external perturbation. Due to the data-driven nature of the approach, no detection threshold needs to be provided by the user.

The presented *perturbation filters* can be regarded as virtual force sensors that produce a continuous estimate of external forces. In contrast to other approaches, perturbation filters can be used to extract accurate and reliable force estimates even from low-cost sensors. To this end, we use Sparsity-promoting Dynamic Mode Decomposition (SDMD) to learn a model of the system dynamics during the robot execution of a specific motor skill. During human–robot interaction, the model is then used to determine the existence and amount of irregularities in the sensor readings. By modeling the correlations as well as the time-dependent variation in the original sensor values, our filter can robustly deal with uncertainties in estimating the human physical influence on the robot. During task execution, the estimated perturbation value can be used to compensate for the external forces or infer the intended guidance of a human interaction partner. Experiments on a real robot show that learned models can be used to accurately determine even small disturbances.

2. Related work

In recent years, natural and intuitive approaches to human–robot interaction have gained popularity. Various researchers have proposed the so-called *soft robotics* paradigm: compliant robots that ‘can cooperate in a safe manner with humans.’[5] An important robot control method for realizing such a compliance is impedance control.[6] 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.

The ability to sense physical influences is at the core of recent advances made in the field of HRI.[7] For example, Lee et al. [8] use impedance control and force–torque sensors in order to realize human–robot interaction during programming by demonstration tasks. Wang et al. [9] present a robot adapting its dancing steps based on the external forces exerted by a human dance partner. Ben Amor et al. [10] 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 [11] and [12]. A different approach aiming at

joint physical activities between humans and robots has been reported in [13]. 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. The parameters of the interaction model are updated using binary evaluation information obtained from the human. This approach significantly improves physical interactions, but is limited to learning timing information.

Stückler et al. [14] 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. [15]. 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. Similarly, Bussy et al. [16] also use force–torque sensors on the wrists to adapt the robot behavior during object transportation tasks. Lawitzky et al. [17] also shows how load sharing and role allocation can be used to balance the contribution of each interaction partner depending on the current situation.

None of the approaches using force–torque sensors addresses the problem of uncertainty in the measurements, in particular, during the execution of dynamic tasks with many contacts. 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. Using predictive models learned by dynamic mode decomposition (DMD), the filtering algorithm also estimates the perturbation which best explains an observed set of new sensor values.

3. Approach

In our approach, the robot recognizes and automatically distinguishes between strength and direction of external perturbations which may be caused by a human interaction partner. An overview of the approach can be seen in Figure 1. First, we record training data for a behavior with different parameter configurations, e.g. walking with varying step lengths, in a controlled environment without external perturbations. The training data are used to learn a behavior-specific **perturbation filter**. Given the sensor values over time, the filter identifies the occurrence and degree of human perturbation using Gaussian process regression (GPR) and DMD.

During motor skill execution, external forces are detected by calculating the probability of the observed sensor values under a probabilistic model of the behavior. If the observed values cannot be explained by the natural variation in sensor readings, an event is triggered. Subsequently, an iterative optimization procedure is used to identify the behavior

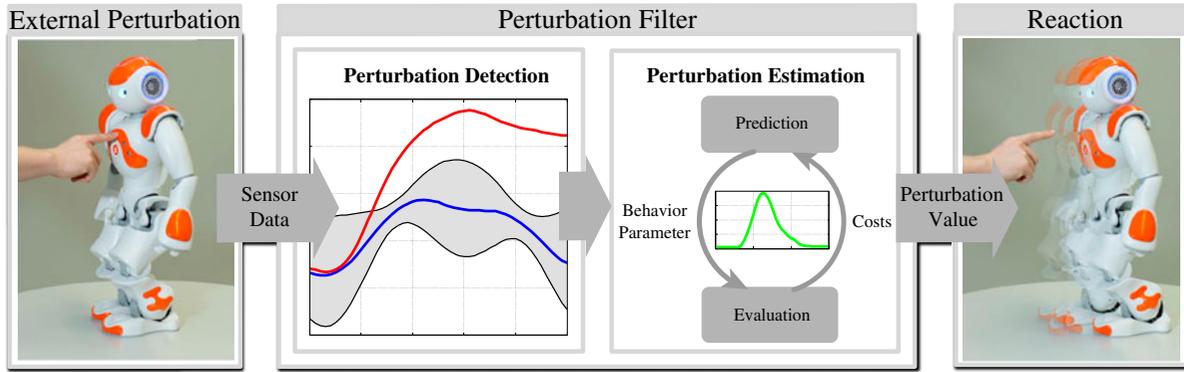


Figure 1. An overview of the presented machine learning approach. An external perturbation is filtered using a previously learned predictive model of behavior parameters. After detecting a perturbation, its strength and direction are estimated in the behavior parameter space. The resulting perturbation value can be used for generation of an adequate reaction.

parameters which best explain the observed values. Finally, the difference between the current behavior parameters and optimized behavior parameters is used to calculate a numerical **perturbation value**. The perturbation value is a continuous estimate of the external forces acting on the robot.

In the following, we will describe each step of our approach in more detail. Subsequently, we will discuss how perturbation detection, model learning, and perturbation estimation are computed in order to allow a whole variety of HRI scenarios.

3.1. Recording training data

The first step in our approach is to record the training data that reflect the evolution of sensor values during regular execution of a motor skill. It is important to record several executions of the behavior, since motor skills can often be executed with different parameters, e.g. varying the step lengths during walking. However, since we use machine learning methods, we will later see that the number of required training data can be limited to about five examples.

Each recorded example contains training data sampled with 100 Hz for one repetition of the modeled robot behavior. In our specific case of training a perturbation filter for walking, we record both the *center of mass* (CoM) and the proper acceleration of the robot for four seconds. Acquiring training data requires less than one minute in total.

3.2. Phase estimation

Since we are dealing with time-varying data, it is important to estimate the phase of the robot during the execution of a motor skill. Depending on the phase, e.g. the left leg is lifted and the variance in the sensor readings can change drastically. To determine the current phase, a time window of sensor values is captured and temporally aligned to the

training data. To this end, we use the *dynamic time warping* technique (DTW).[18] DTW is a time series alignment algorithm for measuring the similarity between two temporal sequences $\mathbf{X} = (x_1, \dots, x_N)$ and $\mathbf{Y} = (y_1, \dots, y_M)$ of length $N \in \mathbb{N}$ and $M \in \mathbb{N}$. In our specific case, the goal is to find the optimal correspondence between the sensor data \mathbf{Y} recorded during the training phase and the currently observed sequence \mathbf{X} , where M is much larger than N .

Due to this significant difference in length of \mathbf{X} and \mathbf{Y} , we formulate our task as finding a subsequence

$$\mathbf{Y}(a^* : b^*) = (y_{a^*}, y_{a^*+1}, \dots, y_{b^*}) \quad (1)$$

with $1 \leq a^* \leq b^* \leq M$, where a^* is the starting index and b^* is the end index that optimally fit to the corresponding subsequence \mathbf{X} . This technique is also known as *subsequence dynamic time warping* (SDTW).[19] To find the optimal subsequence, we first have to calculate the accumulated cost matrix D , which for SDTW is defined as

$$\begin{aligned} D(n, 1) &= \sum_{k=1}^n c(x_k, y_1), n \in [1 : N], \\ D(1, m) &= c(x_1, y_m), m \in [2 : M], \\ D(n, m) &= \min\{D(n-1, m-1), D(n-1, m), \\ &\quad D(n, m-1)\} + c(x_n, y_m) \end{aligned}$$

where c is a local distance measure, which in our case is defined as $c = |x - y|$. The goal of the SDTW algorithm is to determine the path with minimal overall costs C ending at (b^*, M) , where b^* is given by

$$b^* = \operatorname{argmin}_{b \in [1:M]} D(N, b). \quad (2)$$

To determine the warping path $p^* = (p_1, \dots, p_L)$ starting at $p_1 = (a^*, 1)$ and ending at $p_L = (b^*, M)$, a dynamic programming recursion is used. As illustrated in Figure 2, the resulting path p^* represents the optimal subsequence of \mathbf{X} in \mathbf{Y} . As a result, SDTW can be used to estimate the

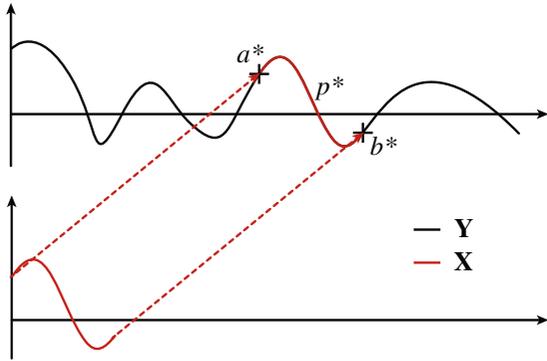


Figure 2. Given the recorded data (black) and the partial observation (red), we calculate the optimal warping path p^* between a^* and b^* .

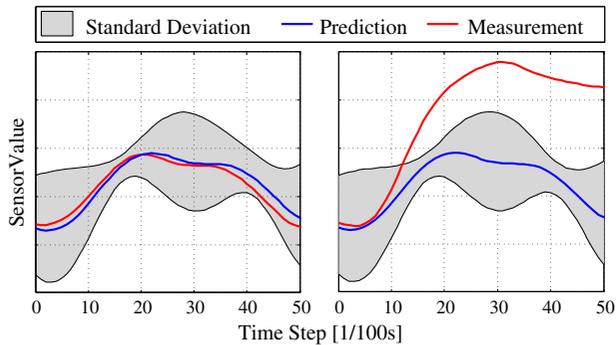


Figure 3. After estimating the current phase of the behavior, the deviation between the measured and predicted sensor values can be used to detect external influences. Left: There is no external perturbation. Right: An external perturbation is detected.

current state of a behavior using a subset of temporally measured sensor values which are mapped to the recorded data. In more detail, we use the subsequence p^* as prediction of sensor values at the current state.

3.3. Perturbation detection

Due to uncertainties in the real world, a motor skill is never executed twice in exactly the same way. To accommodate for natural noise in the behavior, we use learned, behavior-specific information about the temporal evolution of sensor variances.

Different approaches can be used to learn such a probabilistic model. One solution is to use GPR.[20] An important advantage of GPR is the ability to learn a probabilistic model from a small set of training data. When using large datasets, training a GP often becomes computationally demanding. Various solutions have been proposed in the literature to cope with these situations such as pseudo-inputs [21] and sample selection.[22] Details on how to train a GP for

perturbation detection, as well as a periodic kernel that is well suited for the tasks considered in this paper can be found in our earlier publication.[23]

In the remainder of this paper, we are going to employ a simpler solution which is computationally less expensive, can easily be implemented, and in practice produces similar results. More specifically, we are going to compute the standard deviation σ for each time step of the recorded data separately.

Given a probabilistic model as described above, we can detect a perturbation by calculating the likelihood of the current sensor readings. In our implementation, we trigger a detection when the sensor values are outside of the standard deviation σ . Figure 3 shows an example for a regular and a disturbed execution of a behavior.

3.4. Modeling robot dynamics using DMD

We are using DMD in order to teach a predictive model to describe the change in sensor values under different behavior parameters. DMD is a novel data processing technique from the field of fluid dynamics, which was introduced in [24] and [25]. This method presents a modal decomposition for nonlinear flows and features the extraction of coherent structures oscillating at a single frequency and growth/decay rate. The basic idea is that DMD computes a linear model which approximates the underlying nonlinear dynamics. Once DMD obtains the dynamics of an unknown system, they can be used to simulate sensor values under different parameter conditions.

An equidistant temporal snapshot sequence $N + 1$ of a vector of an observable (e.g. velocity field) $x = (u_1, \dots, u_M)^* \in \mathbb{C}^{M \times 1}$ which is stacked into two matrices $K_1 = [x_0 \dots, x_{N-1}] \in \mathbb{C}^{M \times N}$ and $K_2 = [x_1 \dots, x_N] \in \mathbb{C}^{M \times N}$, where $m \gg n$ is assumed. The matrices K_1 and K_2 are shifted by one time step Δt and can be linked via the mapping matrix (system matrix) $A \in \mathbb{C}^{M \times M}$ such that $K_2 = AK_1 = K_1S + re_N^T$ with the residual vector $r \in \mathbb{C}^{M \times 1}$ and the N th unit vector $e_N \in \mathbb{C}^{N \times 1}$.

Since the data are obtained from experiments, the system matrix A is unknown and for a very large system, it is computationally impossible to solve the eigenvalue problem directly as well as to fulfill the storage demand.[26] The idea is to solve an approximate eigenvalue problem by projecting A onto an N -dimensional Krylow subspace in order to compute the eigenvalues and eigenvectors of the resulting low-rank operator as described in [27]. One type of Krylow methods is the Arnoldi algorithm which needs no knowledge of A for the following variant: $x_N = a_0x_0 + a_1x_1 + \dots + a_{N-1}x_{N-1} + r$. The final snapshot x_N can be expressed as a linear combination of the previous ones $[x_0, \dots, x_{N-1}]$. The weighting factors $[a_0, \dots, a_{N-1}]$ are computed such that the residual r is minimized (least-squares problem) in order to form the companion matrix

$$S = \begin{pmatrix} 0 & & a_0 \\ 1 & 0 & a_1 \\ & \ddots & \vdots \\ & & 1 & 0 & a_{N-2} \\ & & & 1 & a_{N-1} \end{pmatrix} \in \mathbb{C}^{N \times N}. \quad (3)$$

In [24], the author describes a more robust calculation, also referred to as standard DMD, which is achieved by applying a *singular value decomposition* on K_1 such that $K_1 = U\Sigma W^*$. The *full-rank* matrix $\tilde{S} \in \mathbb{C}^{N \times N}$ is determined on the subspace spanned by the orthogonal basis vectors U of K_1 , described by $\tilde{S} = U^* K_2 W \Sigma^{-1}$. Solving the eigenvalue problem, $\tilde{S}\mu = \lambda\mu$ leads to a subset of complex eigenvectors μ . The DMD modes are defined by $\Phi = U\mu$, which implies a mapping of the eigenvectors from lower $\mu \in \mathbb{C}^{N \times N}$ to higher $\Phi \in \mathbb{C}^{M \times N}$ dimensional space. The complex eigenvalues λ contain growth/decay rates $\delta = \Re[\log(\lambda)]/\Delta t$ and frequencies $f = \Im[\log(\lambda)]/(2\pi \Delta t)$ of the corresponding DMD modes Φ . The discrete temporal evolution of the DMD modes is governed by the Vandermonde matrix

$$V_{and} = \begin{bmatrix} 1 & \lambda_1 & \cdots & \lambda_1^{N-1} \\ 1 & \lambda_2 & \cdots & \lambda_2^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_N & \cdots & \lambda_N^{N-1} \end{bmatrix}. \quad (4)$$

The DMD modes Φ must be scaled in order to perform a data recalculation of the first snapshot sequence $K_1 = \Phi D_\alpha V_{and}$. The analysis of V_{and} consequently shows that the first snapshot x_0 is independent of temporal evolution since $\lambda^0 = [\lambda_1^0, \dots, \lambda_N^0]^* = 1$. The scaling factors $\alpha = [\alpha_1 \dots \alpha_N]^*$ are calculated by solving $\Phi D_\alpha = x_0$, where $D_\alpha = \text{diag}\{a\}$.

The standard DMD is based on the linearization of the last snapshot, which is assumed to correspond to the underlying nonlinear dynamics. However, due to various reasons, the linearization fails with respect to a negligible small residual r in some cases. For example, huge data-sets may be problematic, especially when computational effort does not suffice. A memory-efficient variant that requires as few as two snapshots in memory at a given time was described in [28]. In [29], the authors describe different modifications of the standard DMD including a non-sequential time series approach, where the evolution operator is built up on the linearization of data pairs of consecutive snapshots.

Another variant uses the combination of *Proper Orthogonal Decomposition* (POD) and DMD as described in [30]. First, the POD extracts spatial coherent structures (topos), which are ranked by their associated fluctuating energy content. Second, the DMD is applied to the temporal weighting coefficients (chronos) of the POD. By changing the observable to the chronos, the linear operator no longer acts on the data space (e.g. velocity field). This reduces the dimension of the residual vector $r \in \mathbb{C}^{N+1 \times 1}$. Furthermore, the frequency spectra can be directly linked to the kinetic energy

spectra, which was only possible for perfectly permanent flow regimes in standard DMD.

The standard DMD algorithm provides a rank- N solution, which means that the approximate eigenvalues agree with the number of snapshots in K_1 . It turns out that by adding additional snapshots which do not increase the vector space, (they do not contribute any new information to the system), the number of approximate eigenvalues still increases. In [31], the authors developed a low-rank version of the standard DMD algorithm that provides a specified rank l solution with $l < N$. This means that when increasing the number of additional snapshots to a given data-set that do not expand the vector space, the number of approximate low-rank eigenvalues stays constant.

However, the low-rank DMD modes do not represent a subset of their standard DMD counterparts. In order to avoid this drawback, a new method was developed so as to find a subset of DMD modes that influences the quality approximation most.[31] Therefore, a new solution to find the scaling vector α was introduced.[32] Here, the scaling vector α is obtained by considering the temporal growth/decay rates of the DMD modes in order to approximate the entire data sequence K_1 optimally. Consequently, the problem can be brought into the following constellation

$$\min_{\alpha} J(\alpha) = |\Sigma W^* - \mu D_\alpha V_{and}|_F^2. \quad (5)$$

This expression is a convex optimization problem which can be transformed into

$$J(\alpha) = \alpha^* P \alpha - q^* \alpha - \alpha^* q + s \quad (6)$$

where $P := (\mu^* \mu) \circ \overline{(V_{and} V_{and}^*)}$, $q := \overline{\text{diag}(V_{and} W \Sigma^* \mu)}$ and $s := \text{trace}(\Sigma^* \Sigma)$. Its solution leads to the following equation:

$$\alpha = ((\mu^* \mu) \circ \overline{(V_{and} V_{and}^*)})^{-1} \overline{\text{diag}(V_{and} W \Sigma^* \mu)}. \quad (7)$$

For a detailed step-by-step description, the reader is referred to [32]. However, the key challenge is to identify a subset of DMD modes that captures the most important dynamic structures in order to achieve a good-quality approximation. SDMD [32] was developed in order to solve this problem. The sparsity structure of the scaling vector α is fixed in order to determine the optimal values of the non-zero amplitudes. Therefore, the objective function $J(\alpha)$ is extended with an additional term such that

$$\min_{\alpha} J(\alpha) + \gamma \sum_{i=1}^N |\alpha_i|, \quad (8)$$

where γ denotes a regularization parameter that indicates the focus on sparsity of α . As a result, instead of scaling the whole spectrum of available modes, the SDMD only concentrates on the most dominant modes for the entire series by setting the amplitudes of the negligible modes to zero. Larger values of γ increase the focus on sparsity (less extracted DMD modes) as illustrated in Figure 4 for an exemplary robot data-set with 51 samples. The higher

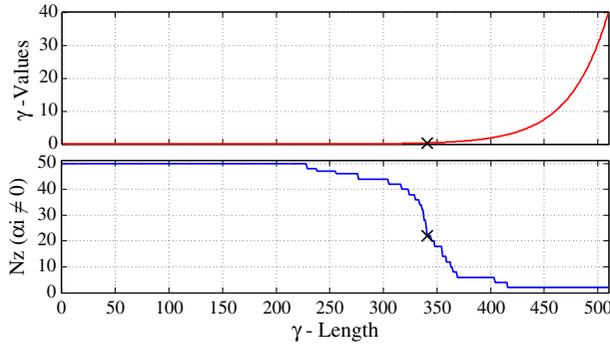


Figure 4. Regularization parameter and its influence on non-zero amplitudes. Top: Curve progression of the regularization parameter γ . Bottom: Number of non-zero amplitudes (Nz) as function of γ where the black cross marks $\gamma = 0.36$ resulting in 22 non-zero amplitudes.

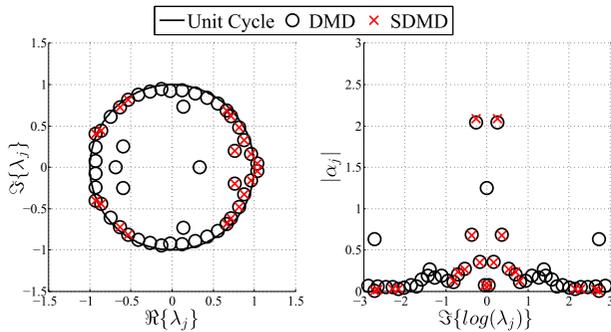


Figure 5. Left: Eigenvalues λ_j from standard DMD (black circles) and the subset of eigenvalues from sparsity-promoting DMD (red crosses). Right: Amplitudes α_j of standard DMD (black circles) and sparsity-promoting DMD (red crosses).

the value of γ (red curve), the lower the number of non-zero amplitudes Nz (blue curve) and the more the SDMD concentrates on the low-frequency modes. For the example presented here, $\gamma = 0.36$ which results in $Nz = 22$, compared to 50 standard DMD-modes. The results of the eigenvalue and amplitude distribution for both algorithms are displayed in Figure 5 left. Eigenvalues in the interior of the unit cycle are strongly damped and hence they influence early stages of the time evolution.

As already mentioned, the data presented here originate from low-cost sensors which may be affected by disturbance. Hence, forcing a low number of non-zero amplitudes in α can reduce the influence of noise in the approximation. As shown in Figure 5 right, the SDMD emphasizes 22 non-zero amplitudes. Furthermore, differences of the scaling vector α are noticeable for some modes due to different scaling approaches of DMD and SDMD.

For our implementation of DMD in a human-robot interaction scenario, the snapshot data $N + 1$ is represented by the sensor data recorded during training data acquisition.

Each column of the snapshot matrices, K_1 and K_2 , contains a fixed number of sensor values, i.e. the longitudinal CoM.

3.5. Calculating a continuous measure of perturbation

If the deviation between measured and predicted sensor values is larger than the allowed variance σ , we assume that an external perturbation is influencing the execution of the behavior. However, the question remains: how strong is the external perturbation?

To estimate the strength of the perturbation, we simulate different behavior parameters using the learned DMD model and select the one that produces sensor values similar to our current readings. For this task, we make use of the previously described SDTW method. As mentioned, the SDTW finds the optimal warping path p^* for a currently measured subsequence \mathbf{X} to a previous recorded data-set \mathbf{Y} . Whenever a perturbation is detected, we perform iterative optimization by generating predictions using a DMD model and calculating the warping costs using SDTW. The goal of this optimization process is to identify the behavior parameter that would best explain the currently observed sensor values. Optimization is performed using a stochastic optimization technique, i.e. *Covariance Matrix Adaptation Evolution Strategy* (CMA-ES). The warping costs C generated by SDTW are used as objective function. Figure 6 shows the warping costs C calculated during a walking task. The behavior parameter which produces least costs C is regarded as the true behavior parameter if human forces are taken into account. By calculating the difference between the behavior parameter used to control the robot and the behavior parameter identified by the learned model, we calculate a continuous *perturbation value*. The perturbation value is an estimate of the external (human) forces acting on the robot.

4. Experiments

In the following experiments, DMD, SDMD, and classical interpolation schemes were used to learn several distinct models of a robot's walking behavior. We evaluate and compare the quality of each of these models. The best model is then used to detect and estimate external perturbations during a human-robot interaction task.

4.1. Prediction quality

For the evaluation of DMD and SDMD, we make use of a walking data-set recorded on a Nao robot. The longitudinal CoM was recorded for a walking behavior with five different equidistant step lengths between $-cm$ and $+4cm$. The data are recorded with 100 Hz for four seconds. Both the DMD and SDMD algorithms were applied on this data-set, resulting in four DMD modes. Given the learned models,

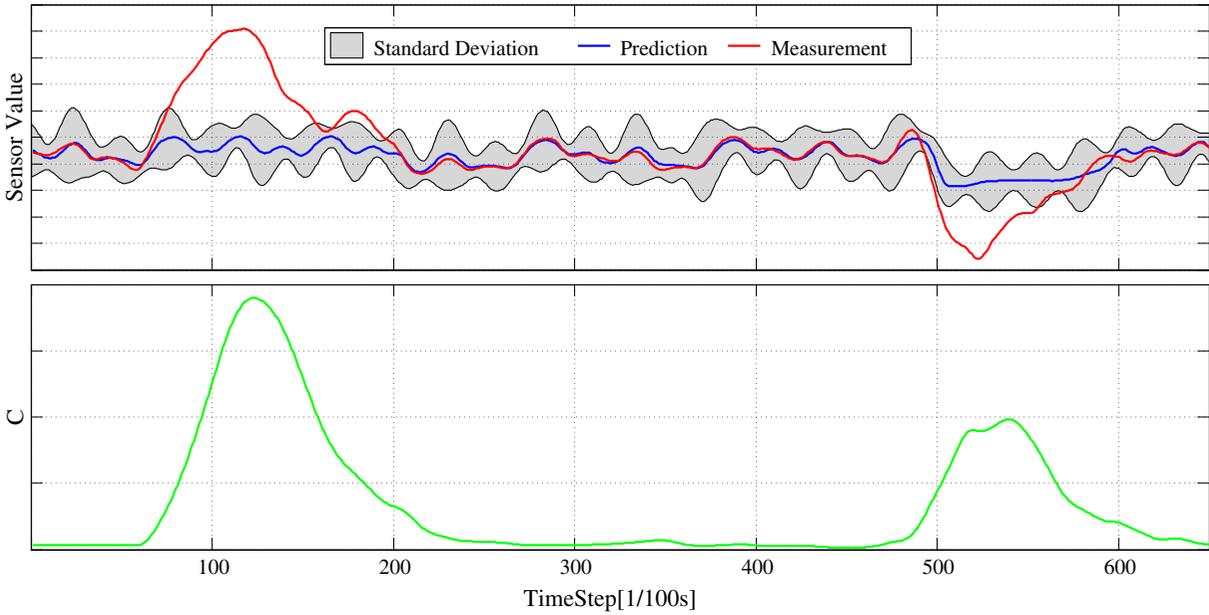


Figure 6. External perturbations which differ in strength and direction are increasing the overall warping costs C during behavior execution.

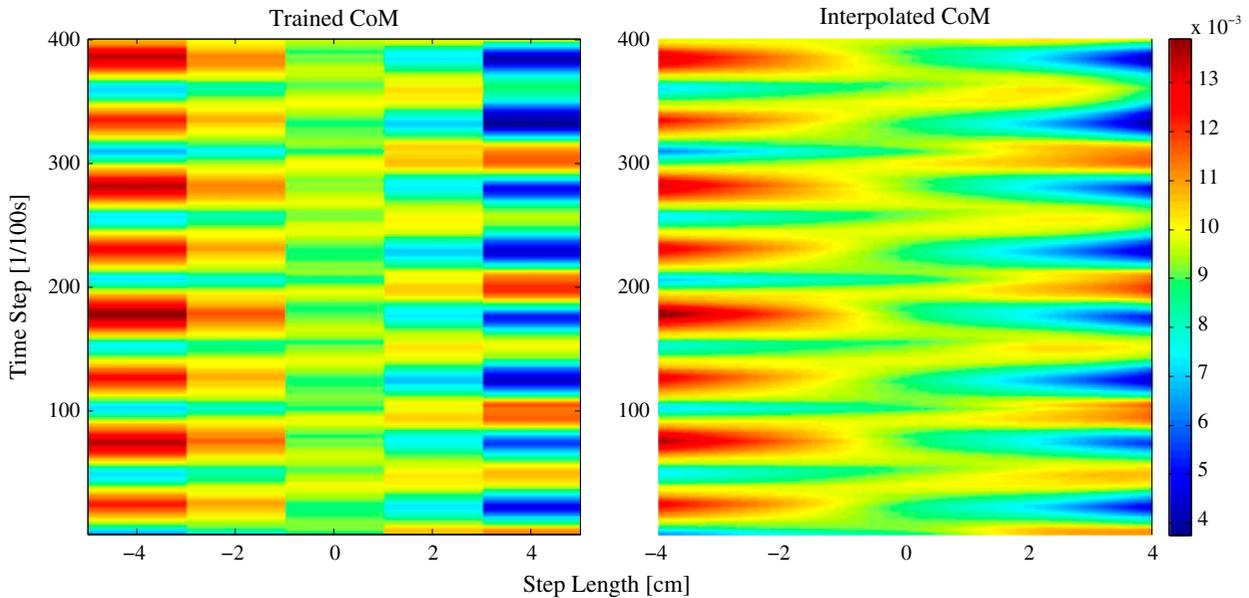


Figure 7. DMD is used to generate new sensor values for unknown parameter settings. Left: The training data consist of five equidistant samples of the longitudinal CoM during walking. Right: The longitudinal CoM is interpolated with an interval of 0.01cm resulting in predictions for 800 possible parameter configurations.

the goal is to generate new sensor values for step lengths that were not recorded during training.

Figure 7 shows the five training samples of the longitudinal CoM and the generated model which was interpolated with an interval of 0.01cm . To evaluate the precision of the generated data, we additionally recorded test samples with step lengths in an interval of 1cm and measured their mean relative error (MRE) w.r.t. the corresponding

generated data. We also compared the results with a set of classical interpolation schemes. For the CoM, Figure 8 shows that DMD scores the highest accuracy among all methods. SDMD reduces the number of used modes to three and results in a slightly less accurate model.

As an alternative to using the CoM for perturbation estimation, and as an example of a noisy low-cost sensor, we also recorded the robot’s longitudinal acceleration and

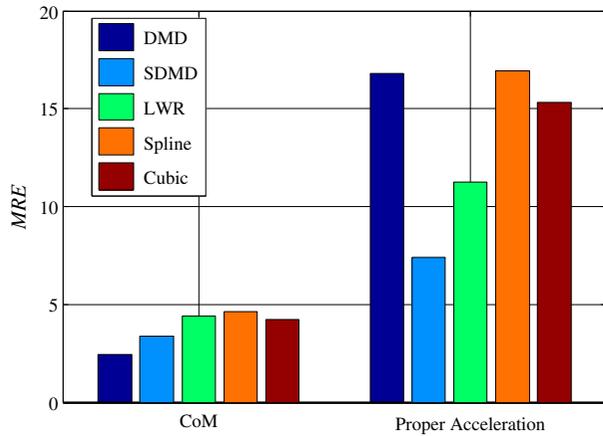


Figure 8. The DMD techniques are compared with a set of classical interpolation schemes. Left: The DMD shows the highest accuracy for the CoM. Right: In the presence of high noise, which is the case for proper acceleration estimates, SDMD produces higher accuracy than DMD or classical interpolation schemes.

applied the same data generation techniques as above. Again, the original DMD uses all extracted modes to predict new sensor values. However, these predictions are corrupted by the fact that some of these extracted modes mainly contain noise. As a result, the prediction performance of DMD deteriorates to about the same level as classical interpolation schemes. In contrast, SDMD concentrates on the three DMD modes that best approximate the sensor data. In this case, one mode was set to zero which apparently contained strong noise. From this, we can conclude that for rather noisy sensors, SDMD-based models will exhibit a better prediction quality than DMD-based models. For the following experiments, we use the DMD model in conjunction with the CoM because of its minimal MRE across all conditions.

4.2. Perturbation detection

In the following experiments, we detect external perturbations while the robot performs a walking behavior with a step length of 0.5cm for 35 seconds. While the robot walks, the human perturbs the robot by touching and pushing it as shown in Figure 9. Figure 9(a) and (c) shows slight pushes, which just marginally disturb the walking behavior. We also applied strong pushes as shown in Figure 9(b) and (d). Especially, the strong push from the back shown in Figure 9(d) significantly affected the robot's stability during walking.

We use the DMD model to generate the predicted sensor values for the current step length. During behavior execution, the longitudinal CoM is measured with 100Hz and saved in a sliding window with 10 measurements. To estimate the current walking phase, we calculate the optimal warping path from this subsequence in the predicted data using SDTW. The resulting path is used as time-dependent prediction of the longitudinal CoM for the currently measured values. Figure 11 shows the measured and predicted longitudinal CoM for the external perturbations a–d as shown in Figure 9. A perturbation is detected when the measured longitudinal CoM is outside the variance of the predicted one.

4.3. Perturbation estimation

If a perturbation is detected, we have to find another behavior parameter and its corresponding sensor evolution, which has minimal mapping costs C for the SDTW. Figure 12 shows the overall costs C for all possible step lengths of our DMD model during the peaks of the external perturbations as shown in Figure 9(a)–(d). As can be seen, backward pushes (Figure 9 top row) result in minimal mapping costs for negative step lengths, whereas forward pushes (Figure 9 bottom row) lead to positive step lengths. As a result, the

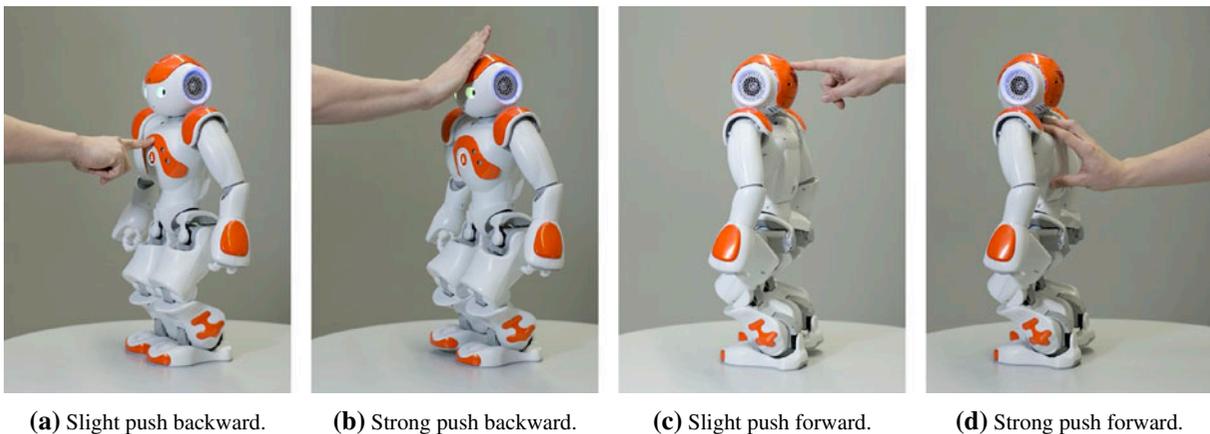


Figure 9. The human touches and pushes the robot during the execution of a walking behavior. The estimated perturbation values differ in strength and direction and reflect the amount of force applied on the robot.

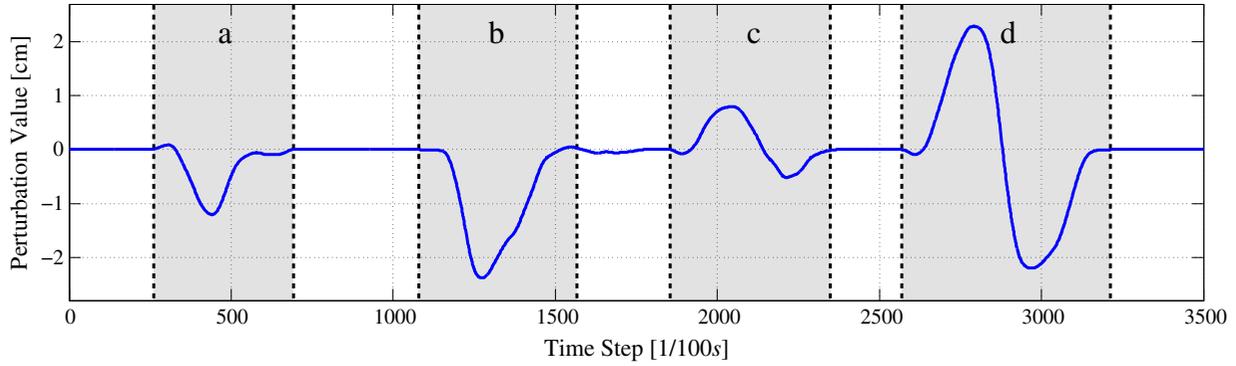


Figure 10. The perturbation value for the external perturbation a-d is the difference between the predicted parameter with minimal costs and the current behavior parameter. Perturbation d produces a large oscillation which is damped over time.

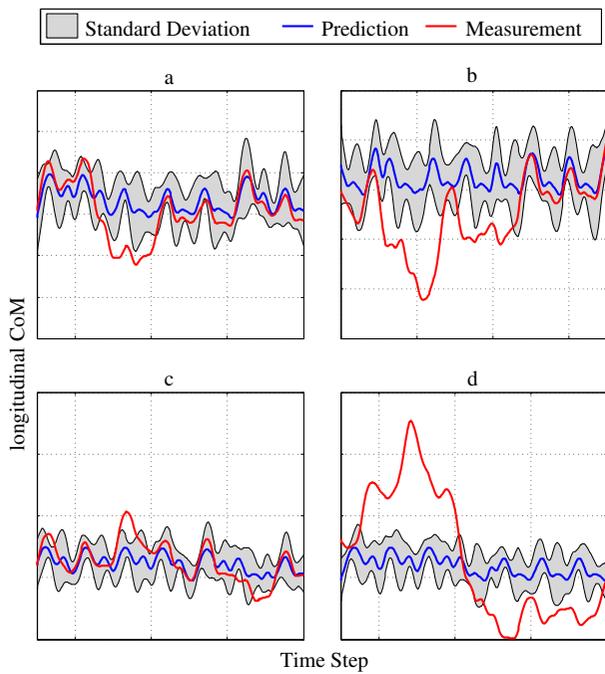


Figure 11. Perturbation detection during walking using the robot’s longitudinal CoM. Top Left: Slight push backward. Top Right: Strong push backward. Bottom Left: Slight push forward. Bottom Right: Strong push forward.

parameters with minimal costs can be seen as behavior parameters which counteract the external perturbation. Finally, the perturbation value is calculated from the difference of the current step length of 0.5 cm and the predicted step length. Since the behavior parameter is specified in cm, the measuring unit for the perturbation value is also in cm. The perturbation value for the complete behavior execution is shown in Figure 10.

4.4. Other scenarios

Generally, our approach can be used in scenarios where a robot has to detect and react to external perturbations. As

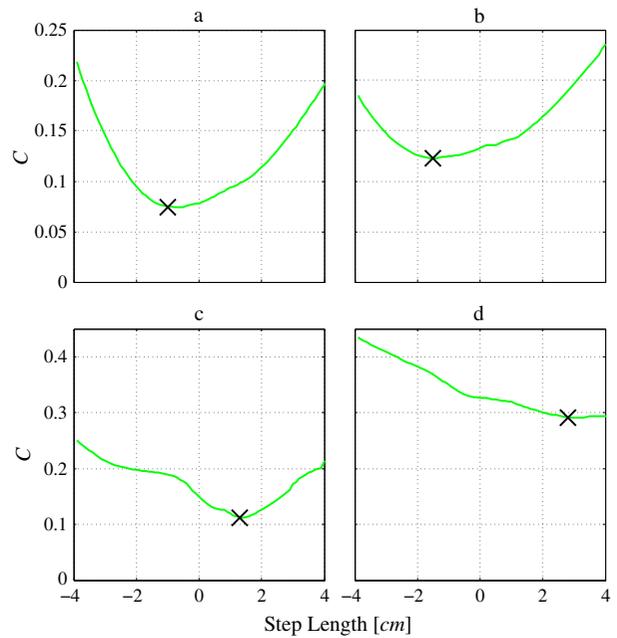


Figure 12. The overall costs C for all possible parameters during the peaks of the external perturbations a-d. The step length which produces the minimal costs (black crosses) is the predicted step length which is used to calculate the perturbation value.

investigated in a previous publication [23], it can be used to follow the human guidance in a cooperative transportation task as shown in Figure 13. A video can be found on Youtube¹. Furthermore, our approach can be used to implement collision detection and safety constraints. In addition, the method can also be used to measure the weight of a carried object during a manipulation task. In general, behavior-specific filtering allows for a variety of close-contact interactions with the environment.

4.5. Sensitivity analysis

In order to evaluate the sensitivity of the perturbation filter and the quality of the produced perturbation values, we

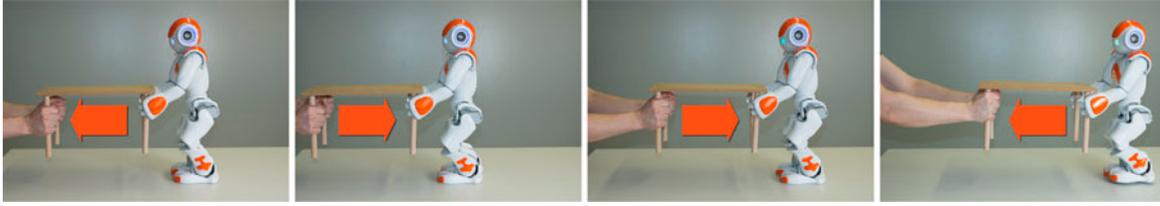


Figure 13. During a cooperative transportation task, a humanoid robot continuously estimates the amount and direction of external perturbations in order to follow the human guidance.

Table 1. Comparison of perturbation values when applying different forces to the robot during walking in place.

Applied force [N]	Perturbation value [cm]
1.8	0.9579
1.5	0.5507
1	0.1707
-1	-0.2443
-2	-0.4507
-3	-0.5579
-4	-0.8279
-5	-2.1586
-6	-2.2021

conducted an experiment in which we applied different forces on the robot. In the experimental setup, a string is used to attach a weight to the robot thereby applying a specific force. The robot then walks in place and the perturbation value is measured using the introduced perturbation filter approach. Table 1 shows the variation of the perturbation value for different applied forces (in Newton). Positive values for the applied force correspond to forward pushes, while negative values correspond to pulling the robot from the back. Forces beyond the $[1.8\text{ N}, -6\text{ N}]$ range caused the robot to fall over.

Until -4 N , the perturbation values decrease in an approximately linear fashion. At around -5 N , the perturbation values jump to about -2.2 cm . According to our observations, forces beyond -4 N introduce significant shake into walking gait of the robot rendering the behavior unstable. As can be seen from the table, the calculated perturbation values provide a reasonable estimate of the amount of external perturbation acting on the robot. Also, the perturbation values are measured in the parameter space of the behavior and can therefore readily be used to counteract the measured external influences.

5. User study

In order to investigate the usability of our approach, we used the well-known *System Usability Scale* (SUS) created by John Brooke [33]. SUS is a 10-item questionnaire for quickly measuring a system’s usability with a score ranging from 0 to 100. The SUS questionnaire is a method to

effectively determine a system’s usability even with a smaller number of test subjects.[34]

20 male and 9 female subjects (ages 13 – 18, mean = 15.28) participated in this user study. All participants were high-school students and were divided into two groups. The first group consisted of 21 students who had never before interacted with the NAO robot. This group will be referred to as *beginners*. The second group containing 8 students had participated on a one day NAO workshop before and, thus, will be referred to as *experts*. The total time per participant including instructions, experiment, and questionnaire was about 10 minutes.

5.1. Experimental conditions

In the experiment, the walking model described above was evaluated. Subjects had to steer the robot by physically touching and guiding it to several target positions. As a starting condition, the robot walks in place with a step length of zero centimeters. Next, the subjects were asked to steer the robot along its longitudinal axis to three specified points on the floor by physically pushing the robot forwards and backwards, as illustrated in Figure 14. In order to accomplish this task, a forward-to-backward and a backward-to-forward transition had to be completed. While walking toward the target points, the robot could be accelerated and decelerated.

While walking backward, the CoM is strongly moved toward the front of the robot. Moreover, this effect is further increased by the human guidance. In consequence, a backward-to-forward transition strongly affects the robot’s stability and is more challenging than a forward-to-backward transition. Considering this, the first waypoint was defined before the robot, the second behind it, and the third was the robot’s starting point. This design results in an increasing interaction difficulty during the experiment.

5.2. Evaluation

As suspected, most of the participants perceived the backward-to-forward transition as more challenging what often led to falling over of the robot. In this case, the subject was allowed to try the complete experiment once again. During the second execution, nearly all subjects were able to

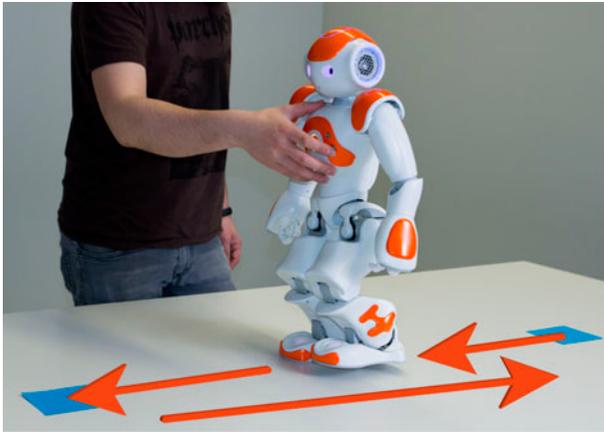


Figure 14. The experimental conditions. First, the subjects have to lead the robot forward to the blue marker and fulfill a forward-to-backward transition. Next, they have to lead the robot backward to another blue marker and fulfill a backward-to-forward transition. Finally, they have to lead the robot back to its initial position.

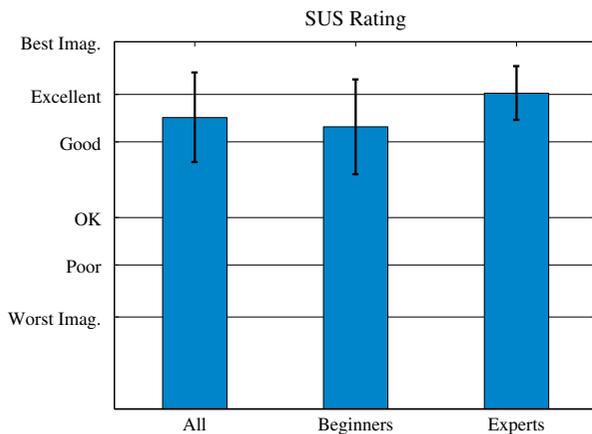


Figure 15. The results of the SUS questionnaire. Obviously, the usability depends to the users' knowledge of the robot platform. The overall rating is between *good* and *excellent* and confirms the usability of the proposed approach.

successfully execute both walking transitions. We attribute this fast-learning curve to a high usability of our approach.

Statistical analysis of the SUS questionnaire further underlines the usability of our approach. For this, we computed the mean and standard deviation of the SUS score for the beginners, the experts, and all participants and compared them in respect to the rating proposed by Bangor et al. [34]. Hence, the SUS score of 0 to 100 is assigned to values between *worst* and *best imaginable*. As shown in Figure 15, the experts assign an *excellent* mean score, while the beginners evaluate our approach with a *good* mean score. The better rating by the experts can be explained by their familiarity with the robot hardware and that they have less fear of contact than the beginners. This observation is confirmed by the measured standard deviation which, for the experts is smaller than for the beginners. In general, the user study

confirms a *good* to *excellent* usability and a steep learning curve of the proposed approach.

6. Conclusion

In this paper, we presented a new approach for learning behavior-specific filters that can be used to accurately identify human physical influences on a robot. The approach uses DTW and DMD/SDMD in order to (1) detect an external perturbation and (2) to quantify the amount of external perturbations. The estimated *perturbation value* can then be used by a robot to adapt its movements to the applied forces or interpret a human command such as 'walk backwards.'

In our experiments, we showed that the learned perturbation filter can be used to accurately estimate touch information from noisy, low-cost sensors. Our approach produces a continuous perturbation value that can be used to detect even subtle physical interactions with a human partner. Since we are using a data-driven approach, no thresholds need to be defined by the user. At the core of our approach lies DMD which, so far, has mostly been used in other fields of science, particularly fluid mechanics. We conclude that DMD is a highly promising method for robotics.

A drawback of the proposed approach is that no guarantees regarding the robustness and accuracy of the perturbation detection can be provided. In particular, in situations when safety is critical, it is important to use additional measures to ensure safety. However, even in these situations, our method can be used to generate a redundant estimate of the external perturbation. Conflicts between the different estimates can be used to halt execution of a behavior or trigger an emergency stop.

In our future research, we hope to hierarchically combine several filters in a mixture-of-experts approach, to generalize perturbation estimation to new, unseen behaviors. We are currently also investigating the application of this approach to industry-grade robots and collaborative assembly tasks.

Supplemental data

Supplemental data for this article can be accessed at <http://dx.doi.org/10.1080/01691864.2014.981292>.

Note

1. <http://youtu.be/wHZYx6Dzswk>.

Notes on contributors



Erik Berger is a research scientist and PhD student at the Institute of Computer Science at the Technical University Bergakademie Freiberg, Germany. He received his Master degree in computer science in 2011 and is currently working at the Virtual Reality and Multimedia Group Freiberg. His research was nominated for the Kazuo Tanie Award IEEE RO-MAN 2014 and won the BuddyPaddy

competition at the Symposium on Emergent Trends in Artificial Intelligence & Robotics 2013. His research interests include robotics, human-robot interaction, machine learning, robot assistance and virtual reality. In particular he is currently focusing on force recognition in physical human-robot interactions.



dynamical data.

Mark Sastuba is a research scientist at the Technical University Bergakademie Freiberg, Germany. He received his Diploma degree in Mechanical Engineering in 2011 and has been working on his PhD in Fluid Mechanics and Computer Science since 2012. His research focuses on 3D tomographic reconstruction methods, particle image velocimetry, as well as modern analyzing techniques for complex



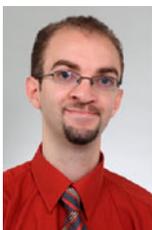
focusing on character animation in direct human-agent interactions.

David Vogt is a research scientist and PhD student at the Institute of Computer Science at the Technical University Bergakademie Freiberg, Germany. He received his Master degree in computer science in 2011 and is currently working at the Virtual Reality and Multimedia Group Freiberg. His research interests are around human-robot and human-agent interaction. In particular he is currently



agents and robots as well as a Habilitation degree for a thesis on intelligent virtual environments. From 2003 to 2005 he was professor for Media Informatics at University of Lübeck's International School of New Media. Since 2005, Bernhard Jung chairs the Virtual Reality and Multimedia group of the TU Bergakademie Freiberg. Prof Jung's research interests are in the fields of Virtual Reality, Large Data Visualization, Human-Computer Interaction, and Advanced Robotics.

Bernhard Jung is a professor at the Institute of Computer Science at the Technical University Bergakademie Freiberg, Germany. He studied computer science at the University of Stuttgart, Germany, and the University of Missouri, Saint Louis. He received his doctorate from the University of Bielefeld with a thesis on dynamic knowledge representations for situated artificial communicators such as virtual



University Darmstadt, Germany. His research has won several awards, such as the Bernhard-v.-Cotta Award 2011, and the CoTeSys Best Paper Award at IEEE RO-MAN 2009. Since 2011, he is also a Daimler-Benz fellow. His research interests include robotics, virtual reality, machine learning, motor skill acquisition and human-robot interaction.

Heni Ben Amor is a Research Scientist at the Institute for Robotics and Intelligent Machines at GeorgiaTech in Atlanta. He received his MSc from the University of Koblenz-Landau, and his PhD from the Technische Universitaet Bergakademie Freiberg. Heni has been a visiting researcher with the Intelligent Robotics Group of the University of Osaka, Japan and a postdoctoral scholar with the Technical

References

- [1] Rozo L, Calinon S, Caldwell DG, Jimenez P, Torras C. Learning collaborative impedance-based robot behaviors. In: AAAI Conference on Artificial Intelligence; 2013; Bellevue, Washington, USA.
- [2] Fuchs S, Haddadin S, Keller M, Parusel S, Kolb A, Suppa M. Cooperative bin-picking with time-of-flight camera and impedance controlled DLR lightweight robot iii. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2010 Oct; Taipei, Taiwan. p. 4862–4867.
- [3] Ott C, Nakamura Y. Base force, torque sensing for position based cartesian impedance control. In: IEEE/RSJ international conference on intelligent robots and systems; Vol. 2009; 2009 Oct 11–15; St. Louis, MO, USA; 2009:3244–3250.
- [4] Nof SY. Handbook of industrial robotics. 2nd ed. New York (NY): Wiley; 1999.
- [5] Albu-Schäffer A, Eiberger O, Fuchs M, Grebenstein M, Haddadin S, Ott C, Stemmer A, Wimböck T, Wolf S, Borst C, Hirzinger G. Anthropomorphic soft robotics – from torque control to variable intrinsic compliance. In: Pradalier C, Siegwart R, Hirzinger G, editors. Robotics research. Vol. 70, Springer Tracts in Advanced Robotics. Berlin: Springer; 2011. p. 185–207.
- [6] Haddadin S. Towards safe robots - approaching Asimov's 1st law. Berlin: Springer; 2014.
- [7] Special issue: Physical human-robot interaction through force interface. Adv. Robot. 2011;25:511–673.
- [8] Lee D, Ott C. Incremental kinesthetic teaching of motion primitives using the motion refinement tube. Autonom. Robots. 2011;31:115–131.
- [9] Wang H, Kosuge K. Control of a robot dancer for enhancing haptic human-robot interaction in waltz. IEEE Trans. Haptics. 2012;5:264–273.
- [10] Amor HB, Berger E, Vogt D, Jung B. Kinesthetic bootstrapping: teaching motor skills to humanoid robots through physical interaction. In: Proceedings of the 32nd annual german conference on advances in artificial intelligence. KI'09. Paderborn, Germany; Berlin, Heidelberg: Springer-Verlag; 2009. p. 492–499.
- [11] Calinon S. Robot programming by demonstration: a probabilistic approach. EPFL/CRC Press; 2009. EPFL Press ISBN 978-2-940222-31-5, CRC Press ISBN 978-1-4398-0867-2.
- [12] Kober J, Peters J. Policy search for motor primitives in robotics. Machine Learn. 2011;84:171–203.
- [13] Ikemoto S, Ben Amor H, Minato TJung B, Ishiguro H. Physical human-robot interaction: mutual learning and adaptation. IEEE Robot. Autom. Mag. 2012;19:24–35.
- [14] Stückler J, Behnke S. Following human guidance to cooperatively carry a large object. In: Humanoids; 2011; Bled, Slovenia. p. 218–223.
- [15] Yokoyama K, Handa H, Isozumi T, Fukase Y, Kaneko K, Kanehiro F, Kawai Y, Tomita F, Hirukawa H. 2003. Cooperative works by a human and a humanoid robot. In: IEEE International Conference on Robotics and Automation, 2003. Proceedings. ICRA '03; Sept; Taipei, Taiwan. Vol. 3. p. 2985–2991.
- [16] Bussy A, Gergondet P, Kheddar A, Keith F, Crosnier A. Proactive behavior of a humanoid robot in a haptic transportation task with a human partner. In: 2012 IEEE RO-MAN; 2012; Paris, France. p. 962–967.

- [17] Lawitzky M, Mortl A, Hirche S. Load sharing in human-robot cooperative manipulation. In: 2010 IEEE RO-MAN; 2010; Viareggio, Italy. p. 185–191.
- [18] Sakoe H. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoust. Speech Signal Process.* 1978;26:43–49.
- [19] Müller M. Information retrieval for music and motion. Secaucus (NJ): Springer-Verlag; 2007.
- [20] Rasmussen CE, Williams CKI. Gaussian processes for machine learning (adaptive computation and machine learning). Cambridge, Massachusetts: The MIT Press; 2005.
- [21] Snelson E, Ghahramani Z. Sparse gaussian processes using pseudo-inputs. In: Scholkopf B, Platt JC, Hoffman T, editors. *Advances in neural information processing systems*. Cambridge, MA: MIT Press; 2006. p. 1257–1264.
- [22] Lawrence ND, Platt JC, Jordan MI. 2005. Extensions of the informative vector machine. In: *Proceedings of the first international conference on deterministic and statistical methods in machine learning*, Sheffield, UK; Berlin, Heidelberg: Springer-Verlag. p. 56–87.
- [23] Berger E, Vogt D, Haji-Ghassemi N, Jung B, Ben Amor H. Inferring guidance information in cooperative human-robot tasks. In: *Proceedings of the international conference on humanoid robots (humanoids)*; 2013; Atlanta, USA.
- [24] Schmid PJ. Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.* 2010;8:5–28.
- [25] Rowley CW, Igor Mez I, Shervin Bagher I, Philipp Schlatter R, Henningson DS. Spectral analysis of nonlinear flows. *J. Fluid Mech.* 2009;641:115–127.
- [26] Bagheri S. Analysis and control of transitional shear flows using global modes [PhD thesis]. Stockholm, Sweden: KTH, Mechanics; 2010.
- [27] Chen K, Tu J, Rowley C. Variants of dynamic mode decomposition: boundary condition, koopman, and fourier analyses. *J. Nonlinear Sci.* 2012;22:887–915.
- [28] Tu JH, Rowley CW. An improved algorithm for balanced POD through an analytic treatment of impulse response tails. *J. Comput. Phys.* 2012;231:5317–5333.
- [29] Tu JH, Rowley CW, Luchtenburg DM, Brunton SL, Kutz JN. On dynamic mode decomposition: theory and applications; *J. Comput. Dyn.* 2014;1:391–421.
- [30] Cammilleri A, Gueniat F, Carlier J, Pastur L, Memin E, Lusseyran F, Artana G. Pod-spectral decomposition for fluid flow analysis and model reduction. *Theor. Comput. Fluid Dyn.* 2013;27:787–815.
- [31] Jovanović MR, Schmid PJ, Nichols JW. 2012. Low-rank and sparse dynamic mode decomposition. In: *Center for turbulence research annual research briefs*. Stanford, USA. p. 139–152.
- [32] Jovanović MR, Schmid PJ, Nichols JW. Sparsity-promoting dynamic mode decomposition. *Phys. Fluids (1994-present)*. 2014;26:024103.
- [33] Brooke J. SUS: A quick and dirty usability scale. *Usability evaluation in industry*. London: Taylor and Francis; 1996.
- [34] Bangor A, Kortum PT, Miller JT. An empirical evaluation of the system usability scale. *Int. J. Human-Comput. Interact.* 2008;24:574–594.

Copyright of Advanced Robotics is the property of Taylor & Francis Ltd and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.