Exploiting Tactile Gestures for Intuitive Robot Programming and Control

Arne Muxfeldt, Daniel Kubus, Jan Haus, Konrad Kissener, and Jochen Steil

***Abstract*— Tactile surface sensors (TSSs) are often utilized for contact management in human robot collaboration scenarios but they may also be exploited as gesture input devices for online robot programming and control. To this end, we introduce a compact, invariant gesture representation as well as a two-stage gesture recognition and parameter extraction approach which is utilized to generate and parameterize robot motion and tool commands based on a set of 16 gestures. Our experimental results and the accompanying video show the viability of the proposed gesture-based online robot programming and control approach in an industry-related scenario.**

I. Introduction

In the robotics community, tactile surface sensors (TSS) attached to robot links have been used to facilitate contact detection and coarse tactile environment perception for decades. By increasing their spatial resolution, TSS can also be exploited as powerful gesture input devices for online robot control and thus accelerate the shift towards even more user-friendly robot programming paradigms. Over the last years, kinesthetic teaching (KT) has become an imperative when it comes to intuitive robot programming methods. In contrast to other approaches, KT does not require any intermediate input channels but the robot itself, which makes it a direct and natural way of interaction and teaching. While KT can easily be employed to teach poses and parameterize robot motion commands (e.g. [1], [2]), it is hardly suitable to generate and parameterize tool commands. Moreover, editing the program structure in KT scenarios requires additional input channels, e.g., manual control pendants (MCPs).

To bridge the gap between easy-to-use KT techniques and powerful textual (MCP) programming, we advocate gesture-based robot programming with gestures directly performed on TSSs attached to the links of a serial manipulator – as previously demonstrated in [3]. In contrast to torque sensors, which are usually employed in KT approaches, TSSs offer *spatially* resolved force measurements thus enabling much more sophisticated input options. Compared with most KT approaches, gesture-based programming permits users to program motion sequences involving *higher-level* motion commands *and* alter the program structure without resorting to the MCP or other textual programming interfaces.

Recently, (deep) learning approaches combined with classical image processing techniques have been proposed for gesture recognition in robot control applications (e.g. [4], [5]). These approaches typically require a large number of training samples to cover gesture executions differing w.r.t. location, orientation, size, and applied forces and yet do not achieve accuracies required for online robot programming and control. To overcome these drawbacks, we represent tactile gestures by space curves and exploit their invariance properties which facilitates both reliable gesture recognition and convenient parameter extraction for the parameterization of robot commands. Moreover, utilizing these invariance properties greatly reduces the number of required training samples and hence significantly decreases the cost of training data acquisition and processing.

In Sec. II, we describe our space curve based compact representation of gestures which is invariant w.r.t. translation, rotation, geometric, and time scaling as well as our two-stage gesture recognition approach. Sec. III addresses how this gesture recognition approach is integrated into a framework for online robot programming and control. Experimental results focusing on the gesture recognition performance are presented in Sec. IV while the online robot programming and control approach is illustrated in the accompanying video.

II. Gesture Representation and Recognition

A. Gesture Representation

Single- and multi-touch gestures can be described by space curves which in turn can be represented by a so-called $\phi\tau r$ representation, which is invariant w.r.t. translation, rotation, geometric and time scaling. The $\phi\tau r$-representation comprises the signed curvature, norm of the tangential vector and distance to the geometric median or the mean of the space curve.

The major steps involved in obtaining the $\phi\tau r$ representation from the TSS data are shown in Fig. 1. In Fig. 1 a) *accumulated* force data is shown for a double circle gesture (DCR, cf. Fig. 3) executed on a planar TSS. To increase resilience against disturbances and to facilitate stream-based processing of the tactile data, contact points in subsequent frames are represented by vertices in a sparse directed graph, which is depicted in Fig. 1 b). To connect corresponding contact points in subsequent frames, a shortest path search is performed.

Fitting a spline to the path results in the space curve depicted in Fig. 1 c). In contrast to Fig. 1 a), the shading in Fig. 1 c) shows that the forces applied in each time step are quite similar.

The invariant $\phi\tau r$ representation of Fig. 1 c) is depicted in Fig. 1 d). As the distance to the geometric mean in the xy-plane denoted by r_n remains approximately constant and the curvature measure ϕ_n is negative throughout the gesture,

The authors are with Technische Universität Braunschweig, Institut für Robotik und Prozessinformatik, 38106 Braunschweig, Germany {amu,dku,jha,kki,jsteil}@rob.cs.tu-bs.de

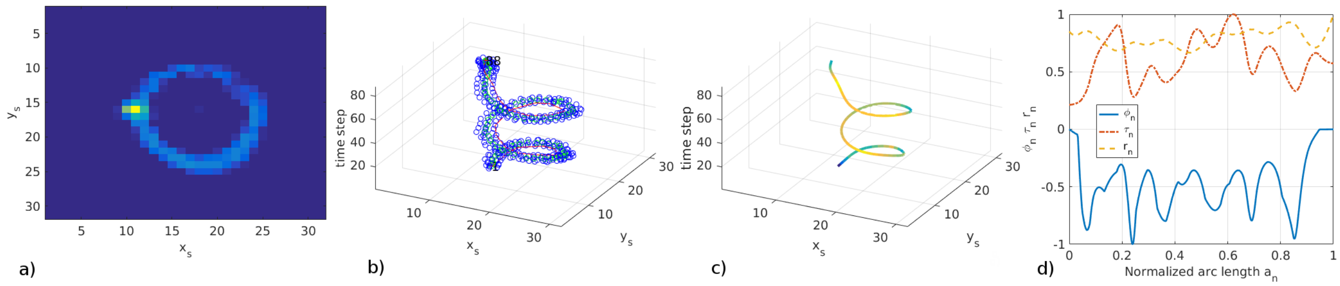

Fig. 1. Processing steps to transform TSS data to the $\phi\tau r$ representation illustrated for a DCR gesture (cf. Fig. 3). (a) Accumulated sensor forces projected onto the xy-plane. Shading indicates accumulated force. (b) Evolution of contact points over time as well as shortest path. (c) Space curve computed from shortest path in the graph. Shading indicates applied forces in contact points. (d) Invariant normalized $\phi\tau r$ representation of curve in (c).

the associated space curve tracks a clockwise circle in the xy-plane. Note that none of the curves will change shape if the gesture is scaled, rotated or translated. Exploiting these invariance properties enables robust gesture recognition based on a very small number of training samples!

B. Gesture Recognition

The gesture recognition approach depicted in Fig. 2 comprises a two-step classification strategy. First, individual curves are classified by the curve classifier (CC) which relies on the $\phi\tau r$ representation. As multi-touch gestures consist of several curves, a second classification (GC) is performed to classify these compound gestures. The GC uses weighted and aggregated curve score vectors $\boldsymbol{s}_c$ of all curves which occur within a specified time window as well as a few additional geometrical and statistical features $\boldsymbol{f}_{rc,n}$ and $\boldsymbol{m}_n$, primarily to distinguish *open* from *close* gestures.

During classification, gesture parameters, e.g. force profiles, are extracted. The label l_g and the parameters $\boldsymbol{p}_s$ of a gesture are forwarded to a command generator which compiles VAL 3 instructions. These instructions are finally sent to the VAL 3 interpreter of the robot controller.

III. Gesture-Based Robot Programming and Control

To demonstrate the potential of gesture-based robot programming and control in industrial scenarios, a *preliminary* gesture set as well as a gesture-based programming and control approach focusing on industry-related tasks have been devised.

A. Gesture Set

Fig. 3 shows the employed gesture set which consists of 16 programming gestures. The employed gestures can be distinguished according to the following characteristics: Gestures, which

- are performed with a single finger (s) or multiple fingers (m),
- are interpreted depending on their location (l), direction (d), sense of orientation (o), force profile (f), and/or velocity profile (v), and gestures whose semantics are independent (i) of these factors,
- trigger commands during execution (e) or after (a) execution.

As the semantics which can be assigned to such tactile gestures are much more fine-grained and richer than that of force-torque profiles interpreted in KT, more elaborate robot motion commands can be performed.

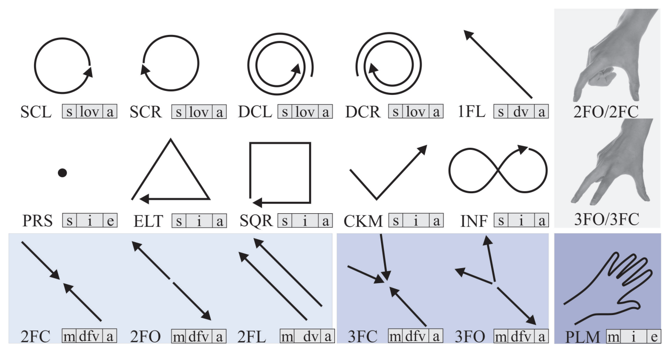

Fig. 3. Schematic representation of the used gesture alphabet. An arrow represents the tracked finger path. The gestures in the first two rows are single-touch gestures whereas the gestures in the third row involve multiple touch points as indicated by the number of arrows. The PLM gesture is performed by pressing the palm of the hand onto the TSS and therefore has no defined number of contact points. Note the symbols (s,m;l,d,o,f,v,i;e,a) assigned to each gesture.

B. Programming and Control Approach

The proposed preliminary gesture set can be used to specify low-level motion commands in both joint space and task space as well as more complex motion and tool commands involving various sets of parameters.

To exemplify the semantics currently assigned to the gestures, the 1FL, 2FL, 2FC and PRS gestures are described in more detail. 1FL is used to trigger task-space motion. The direction of motion is determined by the direction of execution of the gesture on the TSS. 2FL is used to trigger a plane alignment command. Again, its direction determines the plane the end-effector will align with. 2FC closes a gripper using the forces (and velocities) during gesture execution to parameterize the (velocity and) grasp force of the gripper. In contrast to 1FL, 2FL and 2FC, the PRS gesture triggers its associated command already during execution, viz. it stops the currently executed motion or tool command. While the latter gestures are used to specify robot and tool commands, ELT, SQR, CKM and PLM can be used

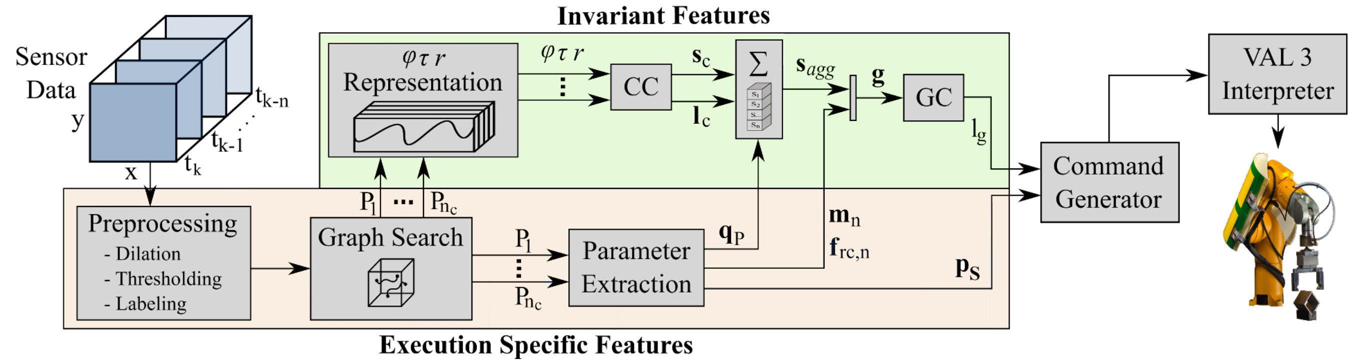

Fig. 2. Block diagram of the gesture recognition and parameter extraction approach. The approach is divided into a branch with invariant and one with execution-specific features. In the execution-specific branch, a quality measure q_P for each space curve , statistical $\boldsymbol{m}$ and geometric $\boldsymbol{f}_{n,rc}$ features as well as command parameters $\boldsymbol{p}_s$ are computed. The curve classifier (CC) receives the $\phi\tau r$ representation of the space curves and outputs a curve label vector $\boldsymbol{l}_c$ and score vectors $\boldsymbol{s}_c$ for each curve which are then aggregated to $\boldsymbol{s_{agg}}$, augmented by the features $\boldsymbol{m}$, $\boldsymbol{f}_{n,rc}$ and forwarded to the gesture classifier (GC). After combining the gesture label l_g with $\boldsymbol{p}_s$, a command is generated and forwarded to the VAL 3 interpreter of the robot controller.

to modify the structure of the generated robot program, for instance, by inserting or removing commands.

IV. Experimental Results

The used TSSs consist of a modified force sensing resistor array [6] of 32×32 taxels with a spatial resolution of $5mm$. The sampling rate is $42Hz$. These TSSs have been used for conducting a user study with 31 participants. In this study, 13304 valid gestures, which cover the whole gesture alphabet presented in Fig. 3, have been recorded for training of the CC and GC. The gestures have been performed on surfaces with different curvatures to incorporate possibly detrimental effects caused by curved robot links. Hence, only one classifier needs to be trained for gestures performed on robot links with different curvatures.

A cubic-SVM has been trained as CC classifier and an ANN with 80 neurons on its hidden layer as GC classifier. The overall accuracy of the classifiers is 98.94% for CC and 98.24% for GC using 5-fold cross validation. The whole gesture recognition process requires less than $40ms$ (*Intel Core i5-3570K*). The highest percentages of misclassifications occur for 3FO/3FC and 2FO/2FC gestures which are confused with each other in approx. 4% of the executions due to insufficient movements of the thumb.

To demonstrate the power of the space curve-based invariant $\phi\tau r$-representation and the low requirements w.r.t. the number of training samples, CC and CG have been trained using only the experimental data of 5 randomly selected subjects (2550 samples), which yielded an overall accuracy of 97.15% for the CC and 96.91% for the GC – a decrease by a mere $1.33pp$. If the number of training samples is reduced further, the advantages of the proposed approach become even more apparent. Even a linear SVM achieves an accuracy of 88.0% with a mere 10 samples per *single* finger gesture based on the $\phi\tau r$-representation whereas it reaches only an accuracy of 23.0% if classification is performed on the raw sensor data.

The accompanying video illustrates the gesture recognition approach and shows how a Stäubli TX40 industrial manipulator is reliably controlled using various gesture-based joint space and task space motion commands as well as tool commands.

V. Conclusion and Outlook

We proposed a compact and invariant representation for tactile gestures as well as a two-stage gesture recognition and parameter extraction approach, which first classifies space curves and extracts relevant parameters and subsequently aggregates the results to classify compound gestures. The gesture recognition performance has been evaluated experimentally and clearly surpasses that of previously proposed methods (e.g. cf. [3], [4]).

We integrated this approach into a control architecture for industrial manipulators to enable gesture-based online robot programming and control. Our current gesture-based robot programming and control framework is illustrated in the accompanying video. It is currently being evaluated and compared to MCP programming in a user study to examine its potential in industrial scenarios.

References

[1] S. Wrede, C. Emmerich, R. Grünberg, A. Nordmann, A. Swadzba, and J. J. Steil, “A User Study on Kinesthetic Teaching of Redundant Robots in Task and Configuration Space,” *Journal of Human-Robot Interaction*, vol. 2, no. 1, pp. 56–81, 2013.

[2] C. Schou, J. S. Damgaard, S. Bøgh, and O. Madsen, “Human-robot interface for instructing industrial tasks using kinesthetic teaching,” in *IEEE ISR 2013*.

[3] A. Muxfeldt, J. N. Haus, J. Cheng, and D. Kubus, “Exploring tactile surface sensors as a gesture input device for intuitive robot programming,” in *IEEE ETFA 2016*.

[4] D. Hughes, N. Farrow, H. Profita, and N. Correll, “Detecting and identifying tactile gestures using deep autoencoders, geometric moments and gesture level features,” in *2015 ACM on International Conference on Multimodal Interaction*, pp. 415–422.

[5] J. Cheng, M. Sundholm, M. Hirsch, B. Zhou, S. Palacio, and P. Lukowicz, *Application Exploring of Ubiquitous Pressure Sensitive Matrix as Input Resource for Home-Service Robots*. Springer International Publishing, 2015, pp. 359–371.

[6] J. N. Haus, A. Muxfeldt, and D. Kubus, “Material comparison and design of low cost modular tactile surface sensors for industrial manipulators,” in *IEEE ETFA 2016*.