Visualization of Forces and Torques for Robot-Programming of In-Contact Tasks

Johannes Hartwig, Sabine Fischer, and Dominik Henrich

Lehrstuhl für Angewandte Informatik III (Robotik und Eingebettete Systeme), Universität Bayreuth, Universitätsstraße 30, D-95440 Bayreuth, Germany
johannes.hartwig@uni-bayreuth.de

Abstract. A variety of automation tasks require an accurate specification of forces and torques over time and space. Traditionally, only experts can provide such specifications, contrasting the need for intuitive robot programming in small and medium enterprises. Simulations and visualizations are typical approaches to increase the usability of robot programming frameworks. Thus, we extend our robot programming approach with force and torque visualizations, enabling non-experts to verify and adjust programs for in-contact tasks. In this paper, we contribute a comprehensive comparison of different force/torque visualization techniques and discuss their applications in robot programming systems. Furthermore, we evaluate visualizations of forces and torques generally as well as specifically adapted to our programming concept utilizing two user studies. Vector arrows for forces and curved arrows for torques showed promising results.

Keywords: robotics, non-expert robot programming, programming system, graphical user interface, simulation

1 Introduction

In industrial environments, experts usually program robot manipulators via a textual programming language. This leads to a time- and cost-consuming programming process, where users must be highly experienced in programming and executing the task. These issues limit the acceptance of robotic systems in small and medium enterprises (SMEs) [1]. Therefore, a fast and easily re-programmable robot system is needed to be profitable for small batch sizes and flexible production. We can solve this by using the programming by demonstration paradigm with kinesthetic teaching because the user only needs to guide the robot, which can generally be considered intuitive [2]. With this approach, it is traditionally only possible to enter trajectories, but it is also necessary to allow program structures and adaptions to the robot motions. For this purpose, extended playback programming can be used, which enhances the concept of playback programming by a graphical programming interface allowing these operations [3]. After the programming process, feedback for the user is also helpful. Here, the graphical user interface (GUI) with a robot simulation can help non-experts verify whether the robot program will solve a task satisfyingly.

Some tasks require an accurate specification of forces and torques over time and space (in-contact tasks). Such tasks are, for example, planing wood or tightening screws. To date, many GUIs simulate only position-related robot motions. For easy use, the visualization of in-contact robot tasks should extend the GUI to allow for verification before they are executed on the real robot. The simulation is extended by visualization of forces and torques if a user programs such motions. While some approaches exist to visualize forces and torques (see Section 2), their usability and intuitiveness still need to be evaluated. Moreover, it needs to be evaluated how and which visualization methods can be combined best to create an intuitive representation of in-contact motions. In addition to visualizations, forces and torques can be transmitted to the user by haptic signals or audio signals [4, 5]. Nevertheless, this work is limited to visualizations since they can be directly integrated into the GUI.

In this paper, we evaluate supportive visualizations of forces and torques exerted by the robot in our programming system through two user studies. The goal is to provide the highest usability and intuitiveness for non-experts. Section 2 gives an overview of the different visualization methods of forces and torques. We then describe the foundations and assumptions for our approach and embed the possible visualization methods within them in Section 3. Thereupon, Section 4 discusses the user studies conducted and their results. Section 5 summarizes the paper and discusses future work.

2 State of the Art

Their Magnitude, direction, and application (or reference) point describe forces and torques fully. Nevertheless, there exist various visualization methods in education and robotics. The suitability of these representations for forces and torques of an in-contact motion depends, among other things, on which of the above properties they represent.

Forces are visualized in teaching various natural and engineering sciences in both school and university. Different effects of visualization types were investigated in direct comparison for educational applications. One study found that to illustrate Newton's law, displaying an animated object is better for force visualization than displaying the same positions as text [6]. However, the transferability of this finding to robot simulations is limited due to the different application areas and by the nature of the textual description, which does not occur in any other application from education or robotics. Also, it has been shown that arrows animated and displayed at runtime are better suited as visual force feedback when training surgeons than a time-force graph displayed after completing the task [7]. However, due to the different times of display and application domain, it cannot be concluded that arrows are also more suitable for force visualization in robot simulations. Unlike for forces, there are no studies evaluating different visualizations for torques. In terms of usability, both force visualizations and torque visualizations have yet to be quantitatively evaluated.

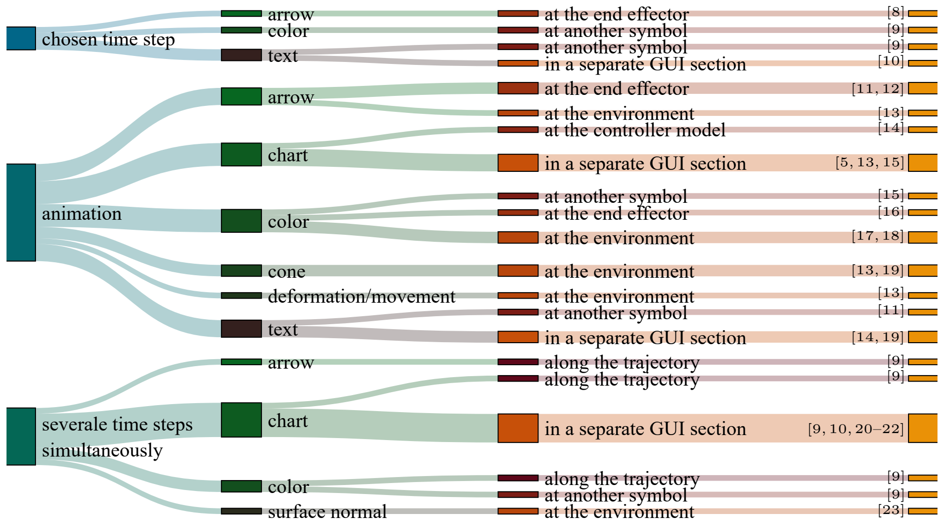


Fig. 1: Classification of reviewed visualizations of forces in robot simulations. The representations are split up by the categories (left to right): time courses, the symbols used and their placement.

In robotic-related applications, forces and torques are nearly always considered over time. The visualizations are used here as a tool during the programming process and are rarely reviewed or evaluated. There, we find different programming systems with a variety of visualizations. They can be categorized by the way they represent time courses (single chosen time step, progressive animation, several time steps simultaneously), the symbols used (e.g., arrow, text, diagram), and the placement (e.g., on an object, in a separate GUI section). Figure 1 depicts such an overview for the reviewed force visualizations represented as a tree [5,8–23]. The torque visualizations [10,11,13,14,20,21,24–26] are illustrated in the same classification in Figure 2. In Section 3, the identified visualizations are reviewed and applied to our general programming approach. Subsequently, these are evaluated for their suitability for an intuitive robot programming system in Section 4, as there have been no findings in this area.

3 Visualization Methods

In order to determine which type of visualization is suitable for our approach, we first outline the requirements of the robot programming concept, which is based on playback programming [3]. This concept extends playback programming by enabling users to edit the trajectory, simulate the robot's motion, and add program structure. It shall be expanded to support in-contact tasks. This concept is intended to be usable in a general, task-agnostic manner. It does not

PREPRINT

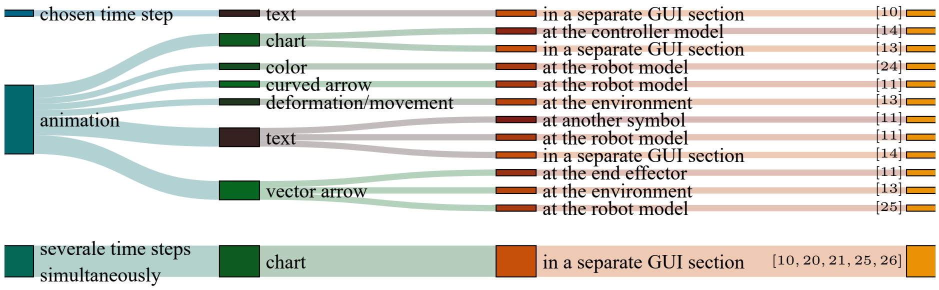


Fig. 2: Classification of reviewed visualizations of torques in robot simulations. The representations are split up by the categories (left to right): time courses, the symbols used and their placement.

use environment modeling, as it requires either a high level of online perception or previous offline modeling by an expert. Therefore, the included robot simulation models only the robot, not the environment. This circumstance eliminates all visualizations that require an environment representation.

Additionally, all parameters of the in-contact motion are known as a user directly demonstrates it and modifies it using the GUI. Therefore, we can formulate these demonstrated in-contact motions as hybrid force-/motion-controlled motions defined by [27]. For simplification, our formalization of in-contact motions omits the control part since we only want to visualize them. Using the 6D-vector representation of forces and torques, we get the wrench $\mathcal{F}(t)$ at a given time step t of our desired in-contact motion in the robots task frame C:

$$\mathcal{F}(t) = S(t)\mathcal{F}_{\text{position}}(t) + (\mathbb{I} - S(t))\mathcal{F}_{\text{force}}(t) \tag{1}$$

The wrench $\mathcal{F}_{\text{position}} \in \mathbb{R}^6$ generates the movement of the dimensions, which have no natural constraints to generate a force or torque (so they are not in-contact). Wrench $\mathcal{F}_{\text{force}} = (f_x, f_y, f_z, t_x, t_y, t_z)$ represents the desired forces and torques. Matrix $S(t)$ is the compliance selection matrix describing whether a dimension of the wrenches is position- or force-controlled. For the forces and torques, only $(\mathbb{I} - S(t))\mathcal{F}_{\text{force}}$ needs to be visualized. Since we have no modeled environment to use, e.g., for a physics simulation, we simulate all degrees of freedom for $\mathcal{F}_{\text{position}}$ of the motion in every time step t.

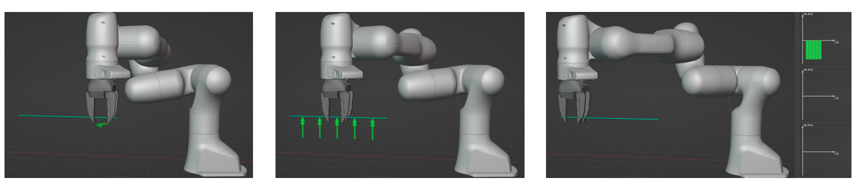
Combining the reviewed visualizations (see Section 2) and our requirements, possible visualization methods are: an arrow at the end effector, arrows along the trajectory of the end effector, a bar chart along the trajectory of the end effector, or three bar charts aligned in their GUI section, which plotted the forces along the world coordinate axes over time. Both curved and vector arrows are applicable for torques, while forces were only represented via vector arrows. The magnitude of the forces and torques is scaled using the maximum magnitude. The direction depends on the direction of the sum over all dimensions for the arrows, and for the curved arrows, the rotation axis of the sum of all torques (using the right-hand rule). For the diagrams, we need no direction as the dimensions are represented separately. The application (or pivot) point is the origin of the task frame. We choose either a single time step or sampled points on the robot's trajectory for the temporal courses.

4 Evaluation

4.1 Pre-study

We used an online questionnaire based on the System Usability Scale (SUS) [28] to gauge the usability of different visualizations for forces and torques. To allow for an assessment of the chosen visualizations (see Section 3) without fully incorporating them within the GUI of a robot simulation, each of them was represented by two picture series. Thus, the questionnaire showed snapshots of a linear movement with a constant force or torque in one direction (see Figure 3) and what attaching the second of four screws along a circular trajectory would look like.

Since we are specifically interested in the visualizations, we modified the statements of the SUS to refer to the visualization instead of the system throughout the questionnaire. Furthermore, "I found the various functions in this system were well integrated." was omitted. Finally, the questionnaire was translated into German. The overall SUS score is robust against these kinds of modifications [29].



(a) curved arrow at the end effector (b) vector arrows along the trajectory (c) bar charts over time in a separate GUI section

Fig. 3: Visualization snapshots for a linear movement with constant torque.

In the online questionnaire, 32 participants rated the visualizations, out of whom 21 identified as male and the rest as female. Their age ranged between 19 and 79 years. Most participants were non-experts. About 40.5% of the participants reported having no experience with robots. Further 43.8% reported having used robots at least once but not frequently.

Figure 4 shows the resulting SUS scores of force and torque visualizations. Similar visualizations are ranked similarly based on their median scores for both quantities. Notably, the visualizations that reach the highest median scores use an arrow at the end effector. Based on a one-way repeated measures ANOVA

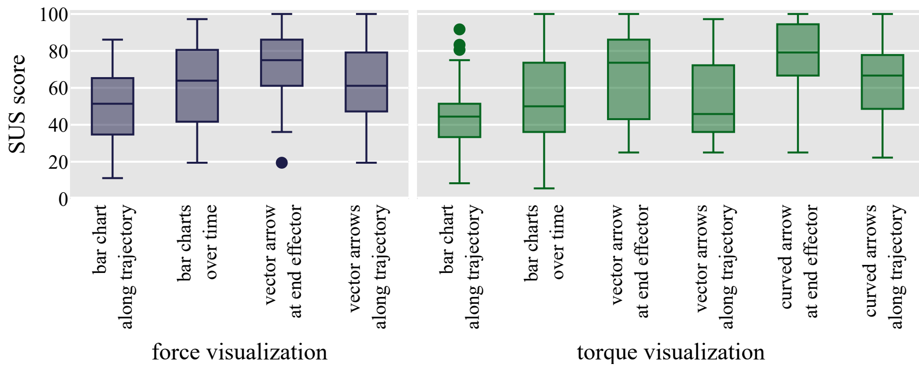


Fig. 4: Box-plot of SUS scores of the force/torque visualizations in the pre-study.

the SUS scores of the force visualizations differ significantly ($p \approx 0.0001$). A Tukey HSD test shows that the SUS score of bar charts along the trajectory differ significantly($p_{adjusted} \approx 0.0002$) from the SUS score of vector arrows at the end effector. There is also weak evidence that the SUS score charts along the trajectory differ from those of charts over time ($p_{adjusted} \approx 0.07$) and from those of vector arrows along the trajectory ($p_{adjusted} \approx 0.08$). Similarly, the SUS scores of the torque visualizations differ significantly based on a one-way repeated measures ANOVA ($p \approx 0.0000$). A Tukey HSD test shows that the SUS scores of curved arrows at the end effector differ significantly from the SUS scores of bar charts along the trajectory ($p_{adjusted} = 0.0$), vector arrows along the trajectory ($p_{adjusted} \approx 0.0001$) and bar charts over time ($p_{adjusted} \approx 0.0002$). Furthermore, there are statistically significant differences between the SUS scores of bar charts along the trajectory and of vector arrows at the end effector ($p_{adjusted} \approx 0.002$) as well as between the SUS scores of bar charts along the trajectory and curved arrows along the trajectory ($p_{adjusted} \approx 0.002$).

Other than the described significant results, only tendencies can be derived based on the surveys for the individual aspects, the representation of the temporal courses, the symbols used, and their placement. This is also due to the sample size (N=32). Overall, the survey showed a clear tendency to prefer arrows over diagrams as symbols for both forces and torques. Concerning the symbol used, the visualization selected for forces via vector arrows at the end effector is consistent. This observation also fits with the results of the study from Section 2 on visualization of forces in surgical procedures [7]. For the curved arrows, which should be used to represent torques based on this survey, only one proof of feasibility has been provided so far, in which these arrows were localized differently [11]. The two visualizations rated highest by participants on the SUS, vector arrows at the end effector for forces and curved arrows at the end effector for torques, were based on progressing animations. They received significantly higher scores than visualizations that also use arrows but visualize the complete time course at a glance. These results suggest that animation is best suited for visualizing temporal sequences for robot programming systems.

4.2 User-study

We conducted a second user study on a real robot system to further investigate the usability of force and torque visualizations to represent in-contact motions. To this end, we integrated arrows at the end effector into the GUI of our robot simulation (see Figure 5). We used vector arrows for forces and curved arrows for torques as our findings from Section 4.1. As these represent only a single time step in the animation, we added continuous line charts over time for both forces and torques in a different section of the GUI to evaluate if a simultaneous overview of the whole movement helps the users. Thus, we evaluated three visualization combinations using only arrows in animation, line charts, or both. All methods display all forces and torques exerted by the end effector simultaneously.

The user study consisted of three parts, one per combined force/torque visualization. First, participants were asked to recreate two in-contact motions displayed in the GUI by kinetically guiding a Franka Emika Panda robot. Together these two motions consisted of four parts with different force/torque components, namely no forces and torques, a force along the x-axis, a force with components along both the x- and the z-axis, and a force along the z-axis in combination with a torque around the z-axis, all with respect to the world coordinates. The sequence of these parts and the sense of direction of the forces/torques varied between sections. For each replication attempt, we recorded the poses of the robot and the forces and torques measured between the flange and end effector using an additional force/torque sensor. At the end of each section, participants answered the SUS questionnaire [28] translated to German but otherwise unmodified.

Half of the 12 participants identified as female, and the other half as male. Their age ranged from 20 to 28 years. Most participants were non-experts. Seven participants reported having no experience with robots. Another three participants reported having used robots at least once but not frequently.

The recorded forces and torques revealed that participants only applied forces and torques correctly in 60.6% of the in-contact sections where forces/torques were displayed. At about 45.7%, the correct reproductions rate is also relatively low for the torques around the z-axis. In some cases, participants applied forces and torques along the correct axis but in the wrong direction. As shown in Figure 7 this happened for all combinations of forces and visualizations but was also more prevalent for torques than for forces. The rate of correctly recreated forces and torques differed between visualization methods. Only displaying arrows at the end effector produced the highest rate overall and in terms of the single forces/torques tested. This rate is also higher for tasks displaying charts and arrows than for only line charts. This observation indicates that arrows are better suited to visualize the forces and torques of in-contact motion than charts.

Considering the absolute SUS score (see Figure 6), it is high for arrows and can be rated as acceptable [30]. Thus, the perceived usability is to be rated high. On the other hand, the connection between actual usability and intuitiveness

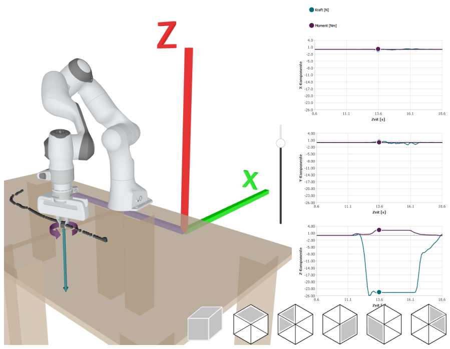


Fig. 5: Force/torque visualization in our robot simulation with arrows and line charts.

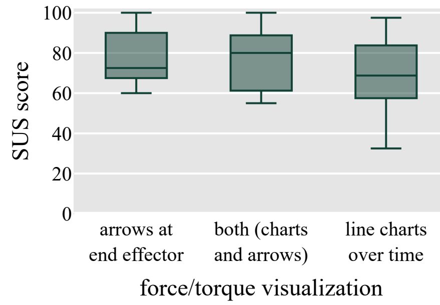


Fig. 6: Box-plot if SUS scores of force/torque visualizations. Arrows consist of vector arrows for forces and curved arrows for torques.

needs to be further investigated because of the comparatively low correct execution rate, especially for torques (see Figure 7). Here, it should be noted that the participants had no feedback on task completion. In addition, the experiment showed that the execution of the task is also motorically challenging.

An analysis of the SUS scores of the visualization methods confirmed this trend. A one-way repeated ANOVA test showed significant differences between the SUS scores of the three force/torque visualizations ($p \approx 0.036$). However, a pairwise comparison of the SUS scores of these visualizations using a Tukey HSD test did not reveal statistically significant differences. Nonetheless, as shown in Figure 6 there is a clear trend that visualizations including arrows perform better than the visualization only using charts. Interestingly, less information (arrow only, single time step only) yields better results in the trend. This result leads us to the design decision for our programming concept: For the visualization, we use arrows in the simulation and add an expert mode with line graphs.

5 Conclusion

Through two user studies, we evaluated supportive visualizations of forces and torques for robot programming systems. As a result, we provide a visual method with the highest usability and intuitiveness for non-experts shown by our studies. Users perceive the visualization as highly usable, but the task performance shows that intuitiveness can still be improved. Future work may include integrating these visualizations in a skill-based visual programming framework (e.g. [31]) to validate our conclusions. Furthermore, we could investigate the gap between perceived usability and task completion if we repeat the experiment and provide the participants feedback on whether the task was successful. Finally, utilizing a less mathematical representation of forces and torques, e.g., using intuitive words (up-down, left-right) instead of axis labels, the intuitiveness of our approach could be increased even further.

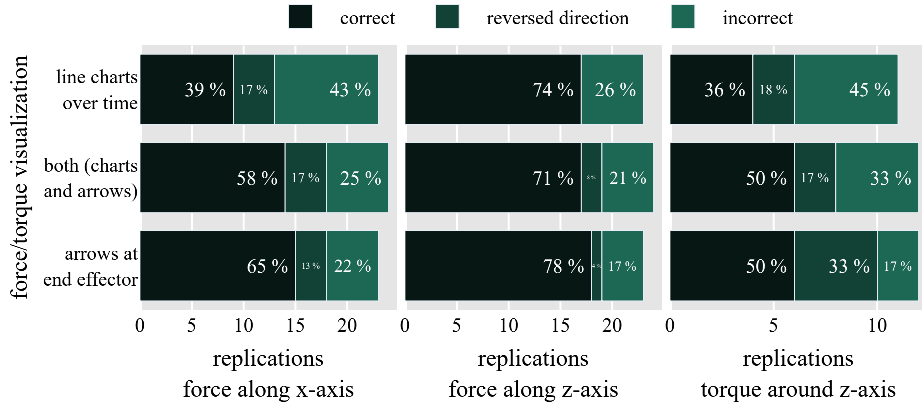


Fig. 7: Correctness of the forces and torques applied by hand guiding the robot based on the visualized in-contact motions excluding replications where the transmission of measured forces/torques was interrupted.

References

1. Kildal J, et al. (2018) Potential users' key concerns and expectations for the adoption of cobots. Procedia CIRP 72.
2. Villani V, et al. (2018) Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. Mechatronics 55.
3. Riedl M, Henrich D (2019) A Fast Robot Playback Programming System Using Video Editing Concepts. In: Tagungsband des 4. Kongresses Montage Handhabung Industrieroboter. Springer, Berlin, Heidelberg.
4. Ebrahimi A, et al. (2018) Real-Time Sclera Force Feedback for Enabling Safe Robot-Assisted Vitreoretinal Surgery. In: 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC).
5. Yusof AA, Kawamura T, Yamada H (2012) Evaluation of Construction Robot Telegrasping Force Perception Using Visual, Auditory and Force Feedback Integration. J. Robotics Mechatronics 24.6.
6. Rieber LP, Tzeng S-C, Tribble K (2004) Discovery learning, representation, and explanation within a computer-based simulation: finding the right mix. Learning and Instruction 14.
7. Rodrigues SP, et al. (2014) Influence of visual force feedback on tissue handling in minimally invasive surgery. British Journal of Surgery 101.
8. Liu H, et al. (2018) Interactive Robot Knowledge Patching Using Augmented Reality. In: 2018 IEEE International Conference on Robotics and Automation (ICRA).
9. Leutert F, Schilling K (2015) Augmented Reality for Telemaintenance and -inspection in Force-Sensitive Industrial Robot Applications. IFAC-PapersOnLine 48.
10. Toz M, Kucuk S (2010) Dynamics simulation toolbox for industrial robot manipulators. Computer Applications in Engineering Education 18.
11. Hulin T, Hertkorn K, Preusche C (2012) Interactive Features for Robot Viewers. In: Intelligent Robotics and Applications. Springer, Berlin, Heidelberg

12. Ortmaier T, et al. (2007) Robot Assisted Force Feedback Surgery. In: Advances in Telerobotics. Springer, Berlin, Heidelberg
13. Miller AT, Allen PK (2004) Graspit! A versatile simulator for robotic grasping. IEEE Robotics Automation Magazine 11.
14. Williams LEP, et al. (2002) Kinesthetic and visual force display for telerobotics. In: Proceedings 2002 IEEE International Conference on Robotics and Automation.
15. Talasaz A, Trejos AL, Patel RV (2017) The Role of Direct and Visual Force Feedback in Suturing Using a 7-DOF Dual-Arm Teleoperated System. IEEE Transactions on Haptics 10.
16. Akinbiyi T, et al. (2006) Dynamic Augmented Reality for Sensory Substitution in Robot-Assisted Surgical Systems. In: 2006 International Conference of the IEEE Engineering in Medicine and Biology Society.
17. Erickson Z, et al. (2017) What does the person feel? Learning to infer applied forces during robot-assisted dressing. In: 2017 IEEE International Conference on Robotics and Automation (ICRA).
18. Haouchine N, et al. (2018) Vision-Based Force Feedback Estimation for Robot-Assisted Surgery Using Instrument-Constrained Biomechanical Three-Dimensional Maps. IEEE Robotics and Automation Letters 3.
19. Vogl W, Ma BKL, Sitti M (2006) Augmented reality user interface for an atomic force microscope-based nanorobotic system. IEEE Transactions on Nanotechnology 5.4.
20. Fonseca Ferreira NM, Tenreiro Machado JA (2000) ROBLIB: An Educational Program for Robotics. IFAC Proceedings Volumes 33.
21. Gil A, et al. (2015) Development and deployment of a new robotics toolbox for education. Computer Applications in Engineering Education 23.
22. Tenreiro Machado JA, Galhano AMSF (1994) A program for teaching the fundamentals of robot modelling and control. In: IFAC Proceedings Volumes. Elsevier, Capri, Italy
23. Quintero CP, et al. (2018) Robot Programming Through Augmented Trajectories in Augmented Reality. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
24. Manring L, et al. (2020) Augmented Reality for Interactive Robot Control. In: Dervilis N (ed) Special Topics in Structural Dynamics & Experimental Techniques, Volume 5. Springer, Cham
25. Rossmann J, Jung TJ, Rast M (2010) Developing virtual testbeds for mobile robotic applications in the woods and on the moon. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems.
26. Gupta V, Chittawadigi RG, Saha SK (2017) RoboAnalyzer: Robot Visualization Software for Robot Technicians. In: Proceedings of the Advances in Robotics. Association for Computing Machinery, New York, NY, USA.
27. Raibert MH, Craig JJ (1981) Hybrid Position/Force Control of Manipulators. Journal of Dynamic Systems, Measurement, and Control 103.
28. Brooke J (1996) SUS-A quick and dirty usability scale. Usability evaluation in industry 189.194.
29. Lewis JR (2018) The System Usability Scale: Past, Present, and Future. International Journal of Human–Computer Interaction 34.
30. Bangor A, Kortum PT, Miller JT (2008) An Empirical Evaluation of the System Usability Scale. International Journal of Human-Computer Interaction 24.
31. Riedelbauch D, Sucker S (2022) Visual Programming of Robot Tasks with Product and Process Variety. Annals of Scientific Society for Assembly, Handling and Industrial Robotics 2022 (to appear).