Integrating Animation Artists into the Animation Design of Social Robots

An Open-Source Robot Animation Software

Author version

Etienne Balit, Dominique Vaufreydaz, Patrick Reignier
Inria - LIG
Université Grenoble-Alpes, France
{firstname}.{lastname}@inria.fr

***Abstract*— Movements are an important part of robot design and we need dedicated tools to design them. As previous research has shown, 3D animation techniques are of great use to animate a robot. However, most robots don't benefit from an animation tool and therefore from animation artists knowledge. We present an open-source robot animation software to address this issue and describe some principles for designing a robot animation software derived from the specificity of robot as a medium.**

Keywords—human-robot interaction; animation; robot animation; expressive movement; expressive gestures

I. Introduction

Movements and appearance are an important part of a robot interface. They are what users see and have a great influence on its familiarity and credibility, two major design goals of a social robot. Moreover, these two components are intertwined, which makes them difficult to design separately.

Robots appearance has made great progress over the years, but movements are often an afterthought in the design process. This often leads to social robots that move mechanically and fail to achieve the Illusion of Life. This expression comes from the title of a book, written by two animation masters from Disney, which is a reference in character animation [1]. Animators are trained to breathe life into characters. Robotics would greatly benefit from their skills and knowledge by integrating them in the design process of social robots.

A first strategy is developed in [2]. The authors propose a design methodology starting with the creation of sketch on paper followed by an animated 3D sketch of the prospective robot. A physical prototype is then built, and software is written to generate the movements displayed in those 3D animated sketches. An issue with this approach is that robots present additional challenges compared to 3D animation. Robots are physical objects subject to the laws of physics. They can fall and wobble, their motors make noise and are limited in speed and acceleration. All these constraints specific to robotics can modify the expressiveness of a movement if they are not accounted for. Moreover, many robots that are not designed with movements in mind could still benefit from animators knowledge.

Another strategy is to create tools to animate robots using the paradigm that animators already use for 3D animation for movies or video games. How to design robot animation software has been explored in [3, 4, 5]. But these softwares are designed for a specific robot and often proprietary (ex: Choregraphe, from Aldebaran). This leads to a situation where every research team designing a new robot would need to implement an animation tool to benefit from the animators knowledge and art. Because of the complexity of the task, animation tools for a given robot are often limited or inexistent.

We try to address this issue by developing an open source software that could become a common software for animating robots and a starting point for the community to iterate upon and investigate the optimal user interface for robot animation [Fig. 1]. We also derive some principles for the user interface of robot animation software from the specific constraints and possibilities of a robot.

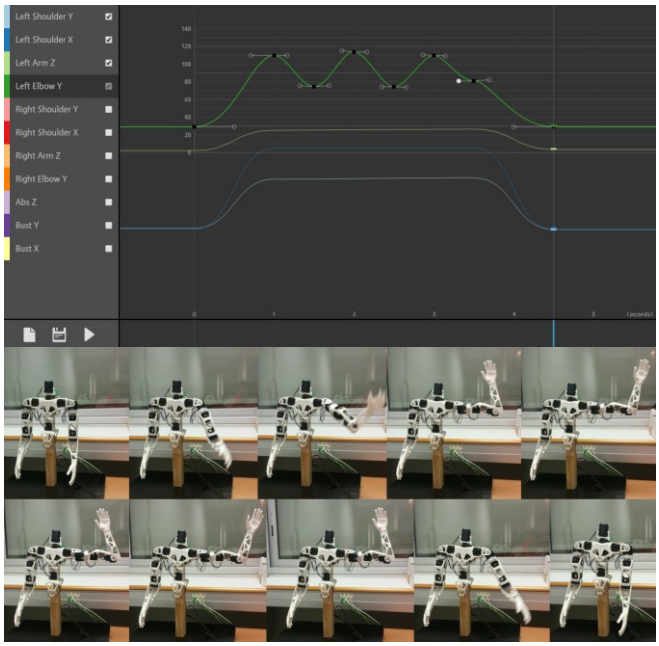


Fig. 1. Top: Preview of the interface of the animation software showing the degrees-of-freedom list on the left, the spline editor on the right and the timeline on the bottom. Bottom: Example of live preview of a movement.

II. An Open Source Robot Animation Software

A. Principles

Keyframes with Bezier interpolation: Most computer graphics (CG) animation software use keyframing with spline interpolation to describe the variation in time of a parameter. We use the same paradigm, as it has proven to be effective and animators master it.

Direct Manipulation and Live Preview: There has been an attempt at creating tangible input device for 3D character articulation [6], as it can be difficult to define the pose of a 3D character with many degrees-of-freedom (DOF) through a 2D interface. Using motors that can act as rotation

This work was partly funded by the Agence Nationale de la Recherche (ANR) under the project Amiqual4Home ANR-11-EQPX-0002.

sensors, we can use the robot as a tangible interface to define its key poses. Our system emulates gravity compensation by deactivating the torque if it reaches a given limit. The user can thus move the robot parts to bring the robot into the desired pose [Fig. 2] and then create a keyframe with it. Our system also uses the robot to preview the movement and interpolated poses between keyframes. It enables the user to preview the result, to edit the pose in place and create a new keyframe to precise the motion if necessary.

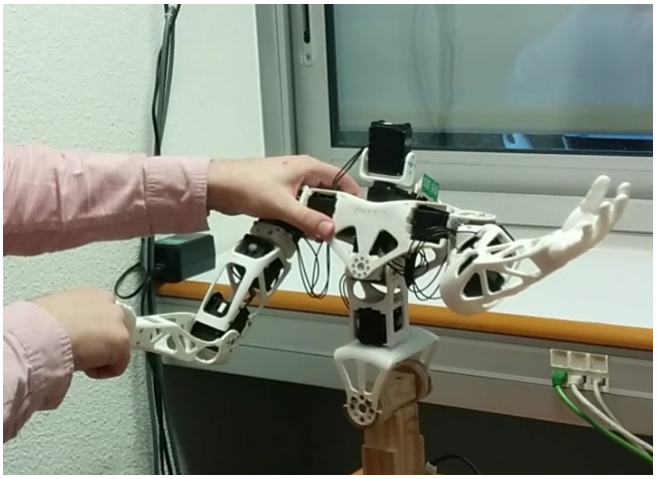


Fig. 2. The robot, here a Poppy Torso, can be manipulated to define new poses or edit a keyframe.

Constraints violations communication: As we noted in the introduction, robots have mechanical constraints. We propose that the user interface should make those constraints visible to the animator. This can take the form of a permanent visual aid or warnings when there is a violation, as it exists in programming Integrated Development Environment (IDE) for example. We implemented two types of constraints handling. Angular position limits are made visible by displaying the grid only on the valid range. Angular speed and acceleration limits violation are communicated through warnings [Fig. 3].

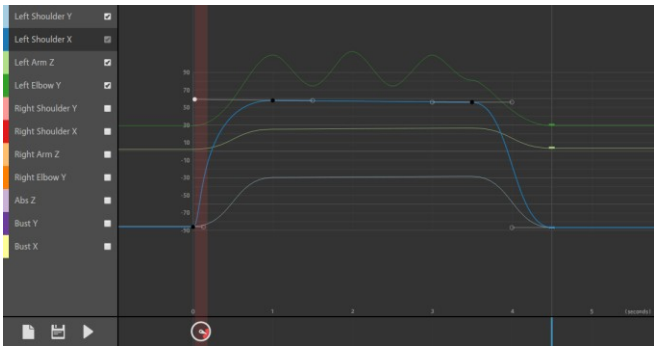


Fig. 3. Example of constraint handling. The grid makes visible the valid range of the angular position and the area in red indicate a violation of angular speed constraints.

B. Implementation

Our system is implemented with adaptability in mind as we aim at creating a software generic enough that the community can adapt it to diverse use cases and build upon it. It also means that it should be simple and easy to modify.

The user interface of our system is implemented using the Web Platforms standards - more specifically JavaScript, HTML and SVG – because these are multiplatform standards. Another reason is that it allows the software to be easily adapted as a desktop application or as a web application served directly from the robot's computer.

The backend of the system is developed in Python and uses the pypot library from the Poppy Project [7] to control the motors. This library is primarily targeted at controlling Robotis Dynamixel motors but is extendable to other actuators. We choose not to base our system on a middleware like ROS to keep the architecture simple but the backend could also be substituted with a ROS node if necessary.

III. Future Work

The system described could be improved in multiple ways. A first way would be to make other parameters of the motors available to animation. Animating the compliance of motors would bring new possibilities for designing movement with physical interaction with an object or with a person like a handshake. A second way to improve upon this system would be to include a method to compute the position of degree-of-freedom (DOF) from another, as it can be useful to simplify the animation of many coupled DOF like a bird's wings. The last way that we envision to improve upon this software is to create a parametric animation software for robots to enable animators to create movements and gestures adaptable to the context.

IV. Conclusion

We think this open robot animation software can enable the community to integrate animators and benefit from their knowledge.

The system that we presented draws inspiration from CG animation software for its user interface and introduces new principles derived from the specificity of robot as a medium. While the ideas drawn from CG animation are well tested in CG and in robotics, we need to evaluate the effectiveness of the new principles described in this paper and their compatibility with those inspired by CG.

This evaluation is ongoing, working with animation artists to animate the Poppy Robot [7]. But we also need feedback from the community to reach the goal of this project to propose a common software for animating robots.

REFERENCES

[1] F. Thomas and O. Johnston, The Illusion of Life: Disney Animation. New York, NY: Hyperion, 1995.

[2] G. Hoffman and W. Ju, "Designing robots with movement in mind" in Journal of Human-Robot Interaction, vol. 3, no 1, pp. 89-122, March 2014.

[3] A. J. van Breemen and Y. Xue, "Advanced animation engine for user-interface robots" IROS 2006.

[4] E. Pot, J. Monceaux, R. Gelin and B. Maisonnier, "Choregraphe: a graphical tool for humanoid robot programming" RO-MAN 2009

[5] J. Saldien, B. Vanderborght, K. Goris, M. Van Damme and D. Lefeber, "A motion System for social and animated robots" in International Journal of Advanced Robotic Systems, vol. 11, 2014

[6] A. Jacobson, D. Panozzo, O. Glauser, C. Pradalier, O. Hilliges and O. Sorkine-Hornung, "Tangible and modular input device for character articulation" UIST 2014.

[7] M. Lapeyre, P. Rouanet, J. Grizou, S. Nguyen, F. Depraetre, A. Le Falher and P. Y. Oudeyer, "Poppy project : open-source fabrication of 3D printed humanoid robot for science, education and art" in Digital Intelligence 2014, p. 6, 2014