Animating the Adelino Robot with ERIK

The Expressive Robotics Inverse Kinematics

Tiago Ribeiro
INESC-ID, Instituto Superior Técnico,
Universidade de Lisboa
Lisbon, Portugal
tiago.ribeiro@gaips.inesc-id.pt

Ana Paiva
INESC-ID, Instituto Superior Técnico,
Universidade de Lisboa
Lisbon, Portugal

ABSTRACT

This paper presents ERIK, an inverse kinematics technique for robot animation that is able to control a real expressive manipulator-like robot in real-time, by simultaneously solving for both the robot's full body expressive posture, and orientation of the end-effector (head). Our solution, meant for generic autonomous social robots, was designed to work out of the box on any kinematic chain, and achieved by blending forward kinematics (for posture control) and inverse kinematics (for the gaze-tracking/orientation constraint). Results from a user study show that the resulting expressive motion is still able to convey an expressive intention to the users.

CCS CONCEPTS

• **Computing methodologies** → **Animation**; *Robotic planning*;
• **Human-centered computing** → *Interaction techniques*;

KEYWORDS

Robot Animation, Autonomous Social Robots, Human-Robot Interaction, Inverse Kinematics

ACM Reference Format:

Tiago Ribeiro and Ana Paiva. 2017. Animating the Adelino Robot with ERIK: The Expressive Robotics Inverse Kinematics. In *Proceedings of 19th ACM International Conference on Multimodal Interaction (ICMI'17).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3136755.3136791

1 INTRODUCTION

When we think of animated characters, what immediately comes to our mind are the characters seen on TV and in movies. These characters were artistically crafted either by hand or using computer graphics (CGI) and design techniques, in order to convey the illusion that they are alive (e.g. [10, 17]). Currently however, robots are becoming a new form of animated characters in order to be used in social applications backed up by technology and artificial intelligence (AI), in fields such as education, entertainment or assisted living.

These *social robots* are a class of robots to which "people apply a social model to, in order to interact with and to understand" [7]. In a more technical interpretation, social robots can be seen as a new form of human-computer interface, that provides the computer part with a physically expressive and perceptive embodiment, through which a sociable artificial intelligence agent engages in an interactive application with the human user. The ultimate goal of our work is to understand how these social robots, through their physically expressive embodiment, and considering their autonomous capabilities, may be able to convey the illusion of life just as movie characters do, while interacting with humans. The key to this goal is in establishing a new form of animation, called *robot animation*.

In the context of social robotics, our understanding is that *robot animation* is more than just making a robot move. It is about turning the robot into an animated character, and making it *seem* alive while interacting with humans in particular tasks or applications. Van Breemen had initially defined animation of robots as "the process of computing how the robot should act such that it is believable and interactive" [8]. We complement his definition by stating that *robot animation consists of all the processes that give a robot the ability of expressing identity, emotion and intention during autonomous interaction with human users.* The two keywords, in this definition, that guide our stance, are *expressing* and *autonomous*, i.e. *robot animation* is closely related to autonomous expression. The idea behind expressing *intention* is that an animated robot should be able to portray its motivation (i.e. *story*, purpose of existence), throughout its actions, in a way that the human interactors are able to understand it, and therefore to interpret the robot's motivation during their interaction.

1.1 Expressive Kinematics

Animating a real manipulator-like robot in a way that it is able to simultaneously be expressive while tracking an orientation constraint (e.g. gaze target) is not a trivial problem. The animation algorithm would be solving for two constraints which in most cases, are not simultaneously satisfiable: the expressive posture of the robot, i.e. the configuration of angles for each degree-of-freedom (DoF) that results in a given posture; and the global orientation of the endpoint node, i.e. the configuration of angles for each DoF such that the endpoint node faces towards a given orientation (in world coordinates). Moreover, such algorithm must be fast in order to provide a responsive interaction with humans, the resulting motion must seem smooth and continuous in order to exhibit naturalness, and we also want it to be extensible and adaptable to other embodiments.

Our belief is that a solution to this problem will allow to create social robots that are more capable of conveying their social intentions and overall motivation to human interactors, while performing other tasks such as gazing or pointing.

ICMI'17, November 13–17, 2017, Glasgow, UK

ACM ISBN 978-1-4503-5543-8/17/11…$15.00
https://doi.org/10.1145/3136755.3136791

In this paper, we present ERIK - Expressive Robotics Inverse Kinematics as a solution to this problem. The algorithm mixes techniques used both in CGI and in robotics in order to cope with the various needs that we are addressing. The next section presents relevant work related to inverse kinematics on which we have grounded our algorithm. The third section presents the ERIK algorithm, followed by a section on the Adelino robot. The fifth section reports a user study performed to verify if using ERIK, Adelino was able to expressively convey an intention to users, while still performing gaze-tracking, and exhibiting the illusion of life.

2 RELATED WORK

In general, the computation for the animation of a hierarchical, articulated structure (kinematic chain) is done through Forward Kinematics (FK) and Inverse Kinematics (IK). This section briefly introduces some fundamental concepts and techniques regarding these processes.

We start by introducing the lexicon and fundamental concepts used in this paper, regarding both FK and IK. Figure 1 provides visual guidance on each of the elements that compose a kinematic chain. The chain is a sequence of N segments connected through joints, starting at a segment S_1, which is connected to the Origin (O) of the world-frame through joint J_1 (or J_{Root}). The tip of the last segment is called $EndEffector$. Each joint J_i is located at world-frame coordinates P_i, and allows for a rotation α about an arbitrary axis $R_i! = \vec{0}$ with angular limits such that $min\alpha_i \leq \alpha \leq max\alpha_i$. A *Kinematic Solution (KS)* is a configuration of angles $\alpha_1, ..., \alpha_N$ applied to each joint $L_1, ..., L_N$. A *Posture* is a a given set of world-space positions $P_1, ..., P_N$ for each joint $L_1, ..., L_N$. Forward Kinematics allows to compute the final *Posture* achieved from a given *Kinematic Solution*, while Inverse Kinematics allows to compute the *Kinematic Solution* that allows to achieve a given *Posture*. In reality, IK is generally used to compute the KS that allows solely the end-effector S to achieve a given target T . The transform of an end-effector $S = S_{pos}S_{ori}$ that moves in 3D space may contain up to six DoFs: three for a position in world-space, and three for an orientation in world-space. Therefore most IK techniques created to date allow to calculate the KS that allows the chain's end-effector to achieve either a given position S_{pos}, or a given orientation S_{ori}, or both.

A comprehensive summary of the most popular IK techniques has already been gathered by Aristidou and Lasenby [2]. Given our goal, we are especially interested in understanding if and how currently existing techniques can be used in real-time, with any robotic embodiment, with joint limits and collision avoidance, without off-line training, and allowing to control the full-body expressive posture along with the end-effector orientation.

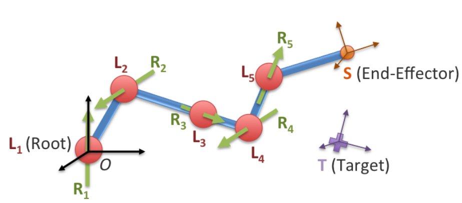


Figure 1: An articulated structure (kinematic chain) as used in both Forward Kinematics (FK) and Inverse Kinematics (IK). Also shown is a given target T that is to be reached by the end-effector S.

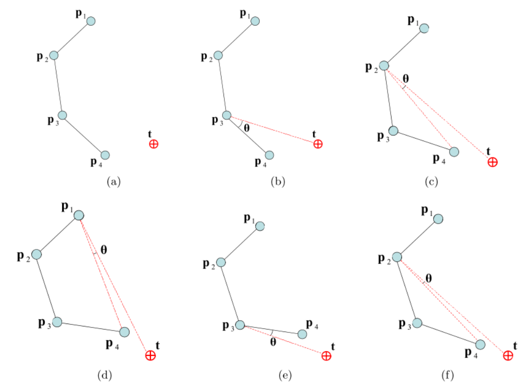


Figure 2: An example of visual solution of the IK problem using the CCD algorithm. (a) The initial position of the manipulator and the target, (b) find the angle Îÿ between the end effector, joint p3 and the target and rotate the joint p4 by this angle, (c) find the angle Îÿ between the end effector, joint p2 and the target and rotate joints p4 and p3 by this angle, (d), (e) and (f) repeat the whole process for as many iterations as needed. Stop when the end effector reaches the target or gets sufficiently close. Description and image cited verbatim from [2].

The classic Jacobian methods provide linear approximations to IK. They are based on the computation and inversion of the Jacobian matrix which contains the partial derivatives of the entire chain system, relative to the end-effectors. Due to space restrictions we will not describe the methods here, as an extensive explanation is already provided by Buss [9]. Conclusions drawn from the comparison of several Jacobian techniques (e.g., Jacobian Transpose, Damped Least Squares (DLS), Selectively Damped Least Squares (SDLS)), both by Buss [9] and by Aristidou [2] are that the Jacobian methods are mostly appropriate for single end-effector situations, not always suitable for time-critical situations (e.g. real-time computation) and the incorporation of constraints using this family of methods is neither straightforward nor controllable towards an optimal solution.

Cyclic Coordinate Descent (CCD) is a popular IK technique, both in computer graphics animation, robotics, and even in protein science [2, 18]. Some of its main advantages are that it is very easy to implement, fast to compute, and has linear-time complexity regarding the number of DoFs. In each iteration it starts from the end-effector, and moves inwards towards the base, adjusting each joint angle at a time, in order to minimize the distance between the end-effector and the target position. This procedure is repeated until either the error is considered to be minimal, or until a maximum number of iterations has been ran. Despite its simplicity and efficiency, the enforcing of constraints remains as a difficult problem. Constraints are applied locally, and it does not provide an intuitive way to enforce them globally. Figure 2 illustrates the execution of the algorithm.

FABRIK is an iterative method that takes on a geometric approach to the IK problem [1]. It borrows the idea of iterating through each joint individually as in CCD, but instead works in the joint-position space (instead of angles), and each iteration includes a forward step (traversing from the end-effector to the base) followed by a backward step (that traverses from the base back to the end-effector). Figure 3 illustrates the execution of the algorithm. Following the

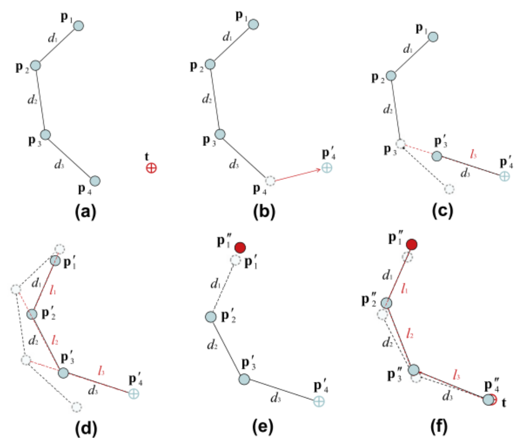


Figure 3: An example of a full iteration of FABRIK for the case of a single target and 4 manipulator joints. (a) The initial position of the manipulator and the target, (b) move the end effector p4 to the target, (c) find the joint p03 which lies on the line l3 that passes through the points p04 and p3, and has distance d3 from the joint p04, (d) continue the algorithm for the rest of the joints, (e) the second stage of the algorithm: move the root joint p01 to its initial position, (f) repeat the same procedure but this time start from the base and move outwards to the end effector. The algorithm is repeated until the position of the end effector reaches the target or gets sufficiently close. Description and image cited verbatim from [3].

notation of Figure 1, and adding that $d_i = |P_{i+1} - P_i|$, for $i = 1, ..., N$ is the length of each segment i, it starts by moving the end-effector S to the target position T which, for algorithmic purposes, will be referred to as P_{N+1}. This is an operation that can only be performed in virtual space, as it intentionally breaks the kinematic configuration of the system by stretching the last segment. However, after this initial move, each successive link L_i is moved to a new position, towards P_{i+1}. After the forward phase, the Root joint will most likely end up in a position that is not the Origin of the space as it was initially.

In order to bring the kinematic chain back to the Origin, the backward phase starts by moving the Root L_1 so that $P_1 = O$. Just as in the first step of the forward phase, this operation also stretches (or shrinks) the first link to an invalid length. So again, but now in inverse order, each joint is traversed and moved to reset the segments to their initial length, while keeping the Root centred at the Origin, and having successfully pulled the end-point closer to the target position. By working directly in the joint-position space, FABRIK avoids calculation of angles, which is one of its main advantages, making it even faster to compute than CCD. Other of its main features are that it does not suffer from singularity problems, produces naturally smooth and continuous motion, and emphasises movement in the joints closer to the base, supports multiple end-effectors, and joint limits can be enforced following the method described in [5].

3 ERIK - EXPRESSIVE ROBOTICS INVERSE KINEMATICS

ERIK is a multi-pass algorithm for inverse kinematics (IK) that was created especially to solve for arbitrary articulated structures of 1-DoF joints, such as the ones used by real robots. It provides, however, a novel joint model, that allows it to use techniques that had been initially developed for computer animation (CGI) and not for robotics. In particular, because it is based on quaternion and vector calculus, it not only minimizes the use of calculation of matrices or trigonometric methods, but can also be used with techniques that solve for cartesion (position-based) solutions and only angle-based solutions.

It is important to note that CGI techniques can facilitate on some of the calculation, as vertexes and joints can move around freely in virtual 3D space. Robots however rely in angular solutions, as there is no way to instruct a segment to translate to a position in real 3D space. Instead, all the instructions must be broken down into individual motor angles. This has allowed animated characters and intelligent virtual agents (IVAs) to acquire complex and even realistic non-verbal behaviours, while robots have been constrained to their mechanical form and motion.

One of the major peculiarities and contributions of ERIK is that it can provide solution for problems that require an arbitrary kinematic chain to orient its endpoint towards a target, while also providing some level of expressive control over the posture of the overall chain. Using it allows complex characters, both virtual or robotic, to perform multi-modal non-verbal behaviour, that would require interactive or procedural control of their expressive posture, along with inverse kinematics to adapt to the user and the task (e.g. gaze at the user or locations).

An animation engine can thus provide an underlying expressive posture for the character to exhibit, while also specifying an orientation for it to face (e.g. gaze if it's a head, point if it's an arm). The technique has been implemented within Nutty Tracks [16], meant to run on each frame of a typical CGI update-draw loop, to render at a rate of about 30 to 50 Hz, for smooth, real-time interactions. The same loop is used by Nutty Tracks for both CGI characters and for robots, following an open-loop control system [12].

3.1 ERIK Components and Overview

The three major components of ERIK, as seen in Figure 4, are the Solver, the Joint Model, and a Motion Filter.

The Solver consists of three passes: one pre-processing pass and two solving passes. We first use a modified version of FABRIK to solve for the target expressive posture. The output of FABRIK is then used as an initial solution for the CCD algorithm which is ran in order to orient the end-effector towards the target direction.

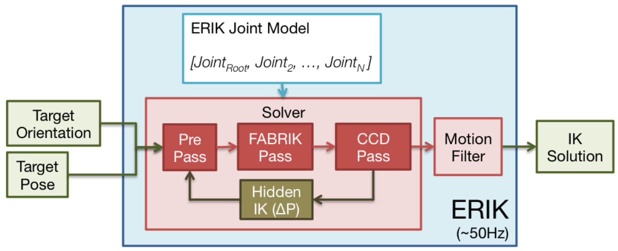


Figure 4: An overview of the ERIK technique and its major components.

This FABRIK algorithm was initially created for virtual characters and as such, runs in cartesian space. However, Nutty Tracks runs in angular space (in order to allow the use of real robots). Therefore one of the roles of the *Pre-Pass* is to convert the *Target Pose* (given in angles) to a cartesian representation. While this first conversion is trivial, the final solution would also be in cartesian space, which can not be directly applied to a robot.

The ERIK Joint Model (EJM) was created to provide a conversion mechanism from cartesian solutions to angular solutions, with consideration for each joint's rotation axis and angular limits. In particular, the EJM was created to cope with series of 1-DoF joints, which can introduce not only singularities (which can be mitigated through the use of quaternions), but also indeterminations.

Our initial attempts to calculate joint angles from cartesian positions using existing methods (e.g. Swing-Twist decomposition, [5]) lead to partial loss of a rotor's information in various situations, when attempting to transform an arbitrary world-space quaternion into an angle-axis pair in which the axis matched the specified joint's rotation axis, and the angle was within the joints' limits. The mathematical details involved in EJM are out of the scope of this paper and will be presented in a separate publication.

4 ERIK AND ADELINO

In order to fully challenge and test the capabilities of ERIK, we designed and built a custom robot. The robot was build with several goals in mind:

- Demonstrate ERIK in a custom built robot;
- Demonstrate ERIK with an autonomous craft robot;
- Promote the design and creation of craft robots for Do-It-Yourself (DIY) audiences;
- Provide animation software for Craft/DIY robots;
- To understand how to design and build robots that balance expressivity and affordability, in order to promote them to animation artists in the future;

We started by designing the concept of the robot using 3d animation software, as was previously done by Hoffman & Ju [15]. This can bee seen in Figure 5. The robot was designed as a line shape, as it is common for traditional and 3D animators to start their learning process by animating *lines of action*. These lines of action are "the first line indicated in a pose, that shows the basic overall posture, prior to adding the rest of the details" [11].

The actual robot is pictured in Figure 6. Its structure was hand crafted using balsa and pine stripwood, screws, washers, nails, some

Figure 5: The concept design of the Adelino robot. It was initially modelled and animated using 3D animation software, to explore the size and placement of each segment and articulation, in order to maximize its expressive capabilities. Note that in this image, at some points, there are articulations that can rotate more than 90 degrees to each side. This feature was not maintained in the actual robot due to typical servos' limitation.

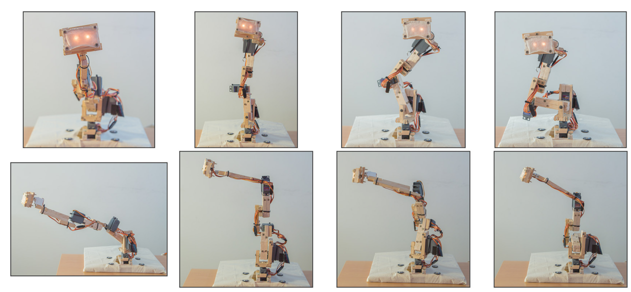


Figure 6: The Adelino robot, in four different expressive postures. The top row shows front views of each posture. Under each is the corresponding side view.

aluminium wire, one bearing, a hammer, a saw, and a drill/screwdriver. It is controlled using five hobby-grade servos connected to an Arduino[1], and an extra 5V, 2.0A power supply to feed the servos. The servos were chosen in order to be the cheapest ones that could handle the expected load. The commonly available and low-end motors are, however, restricted to 180 degrees of motion, thus allowing each joint to rotate only 90 degrees in each direction, given a rest pose that shapes the robot to the form of a vertical line. It also contains two small LEDs on the tip, allowing it to act as a face, so that it can portray the impression of gazing towards a given direction by pointing the tip towards that direction.

Our motivation on building such a robot was not to make it a prototype, nor solely to lower the cost of its production. On one hand, we wanted to build a robot that could appeal to non-technical audiences, such as animation artists, and to a construction process that did not require complex machinery or 3D printing. At the same time, because our goal is to challenge and test our algorithm, we wanted the final evaluation to heavily rely on our software, and

[1] https://www.arduino.cc

not on the use of expensive motors or precision machinery. As such, one of the major motivation for our craft approach was to demonstrate ERIK using the lowest-quality robot we *could* build. That way, we argue that there is only space for improvement on the resulting animation quality, given that the motors, structure, and build process can all be upgraded.

Using Adelino to illustrate the capabilities of ERIK, and taking as example Figure 7, given an expressive posture, the character is able to gaze towards different directions while attempting to maintain the given expressive posture. This figure demonstrates it by showing the virtual view of the robot's skeleton, holding an expressive posture while shifting the orientation target, as if the robot was gaze-tracking the user with an expressive stance. In another demonstration, Figure 8 shows the robot holding an orientation while shifting between postures, as if the robot was gazing towards a stationary user or location, and solely shifting its expressive stance. Note that during an interaction, the transition between postures may even give an impression that it was pre-designed. However, with pre-designed animations it would not be possible to maintain gaze-contact with the user.

4.1 The ERIK Solver Execution

The modification introduced into FABRIK was the use of the EJM to generate solutions in the Cartesian space that do consider the kinematic requirements of each joint, without stalling on singularities or indeterminations. It therefore also allows FABRIK to compute a solution that is simultaneously in angles and in positions. For the following paragraphs, please recall Figure 4.

On each execution's first pass, it runs the modified version of FABRIK, with joint limits, for multiple end-effectors, to find a joint configuration (in angles) for a full-body target posture (in Cartesian joint positions). The use of the EJM allows it to solve for a Cartesian solution, while intrinsically take joint limits into account during the solving process, and not as a post-step. The multiple end-effector version of FABRIK is used as an expressive full-body solver, by treating each joint as an end-effector, and to solve so that each of them attempts to satisfy its own Cartesian position constraint, which is given by the position of the joint in the target posture.

We refer to the result of the FABRIK pass as the *Expressive Solution*, which brings the embodiment to a posture that is as close as possible to the given cartesian configuration (resulting from the Pre-Pass), without violating the kinematic constraints.

The *Expressive Solution* is then used as input to a similarly modified version of CCD, which orients the chain's endpoint towards the given *Target Orientation*, resulting in the *Oriented Solution*.

At this point, a *Posture Gradient* (ΔP) is calculated and stored as the *Hidden IK* component. The ΔP is a set of global quaternions, one for each joint, that represent each local difference between the *Expressive Solution* and the *Oriented Solution*, i.e, the effective result of the CCD pass.

Finally, the *Oriented Solution* is output to the *Motion Filter*, which works as a signal processing algorithm and primarily guarantees continuity and smoothness between consequent solutions.

On the following execution (frame), the previously computed ΔP is applied to the *Target Pose* in the *Pre-Pass*, to warp the given pose towards a cartesian solution that is expected to be already close to the final one.

Given that the ΔP contains the amount of motion that CCD computed for each joint, in the previous frame, to orient the previous *Target Pose* to the previous *Target Orientation*, and that in most cases (e.g. face-tracking), the *Target Orientation* changes incrementally with small steps, we start by providing FABRIK with a hypothetically nearly-correct cartesian pose in order for it to solve towards its angular solution, considering the kinematics constraints of the articulated structure.

This feature was added because FABRIK results in more consistent, natural and continuous poses than CCD. Therefore we use CCD to calculate (in most cases) smaller increments in orientation changes, and then hand that information as a "hint" to FABRIK, in order for this algorithm to produce better overall results throughout the execution. Thus the unnatural effects of CCD are smaller, while its advantages for the orientation constraint are kept. By considering this pass, it is important to note that the input to FABRIK ends up not to be the given *Target Pose* (which would already be its own solution), but instead, the *Target Pose* warped by (ΔP), which we can call the *Pre-Oriented Target Pose*.

5 AHOY - THE PANTOMIMIC EXPRESSIVE MANIPULATOR

Upon developing ERIK and performing initial tests in the robot, we considered that further evaluation with users should be taken to assess the expressivity of system. Because in most cases there exists no solution that simultaneously satisfies the target posture and target endpoint orientation, the computed solution will tend to satisfy the orientation constraint, while slightly allowing to alienate the intended expressive posture. Therefore the main question we tried to address was

> Can an expressive posture be encoded into a face-tracking manipulator-like robot, using ERIK, in order to convey a purposeful intention to the user?

In order to answer this question, we developed an interaction scenario where a human plays a pantomime game with a robot. On each round, the robot performs an animation that represents a word that must be guessed by the user within a time limit. The robot used was Adelino, presented in the previous section. Gaze-tracking was performed in real-time using a Microsoft Kinect and the ERIK algorithm with Nutty Tracks.

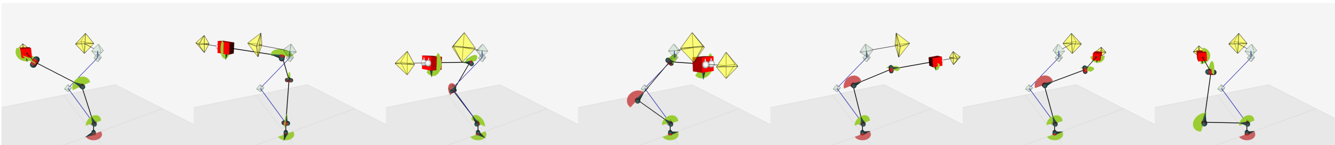
A Wizard (Controller) was given the role of selecting the expressive posture, which the IK algorithm tried to maintain while tracking the user, during the guessing phases. A study was performed with two conditions:

C1 - Intention-expressive : During the guessing of the word by the user, the robot changes its posture depending on a hot-cold measure, in order to provide hints to the player about their guess;

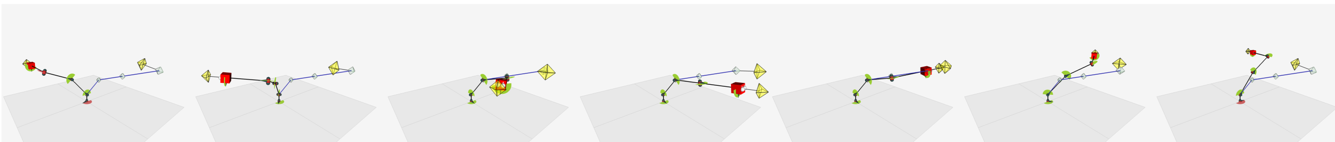
C2 - Non-intention-expressive : The robot maintains the same posture ('neutral') throughout all the guesses.

No other verbal or nonverbal communication was used for the game. Upon this, we have formulated the following hypotheses:

H1 The robot is perceived to play the game similarly well, in both conditions. *Measure: Performance Mean (1.*)*

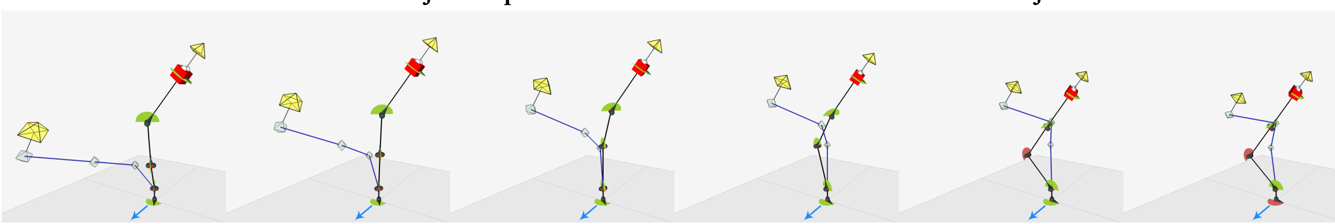


(a) An *unengaged* stance from -180°to 180°in 60°steps.



(b) An *engaged* stance from -180°to 180°in 60°steps.

Figure 7: Demonstration of the ERIK algorithm. Two different expressive postures are presented. For each posture the ERIK orientational target was swept 360°. The images were captured directly from ERIK's visualizer in Nutty Tracks. The blue lines represent the target expressive posture which remains fixed throughout each frame. The yellow arrows represent the target orientation. The circular sector at each joint represents its rotational limits. Red means that the joint is at its rotational limit.



(a) A shift from an *engaged* posture to an *unengaged* posture in an *extreme* angle, upwards and towards the side of the character.

Figure 8: Demonstration of orientation hold during posture shifts. Two shifts of posture are shown, one for a common *easy* orientation, and another for a more complicated, *extreme* orientation. The blue arrow represents the character's frontal *direction*. The remaining elements are described in the caption of Figure 7.

H2 The robot's animation conveys the illusion of life, in both conditions. *Measure: Animation Mean (2.*)*

H3 In the Intention-expressive condition (C1), players perceive the robot's posture changes as its intention of providing hints to the player. *Measure: Intention Recognition (3.1.a)*.

H4 In the Intention-expressive condition (C1), players correlate the intended hint of the robot with their performance in the game. *Measure: Intention Legibility Mean (3.2.*))*.

5.1 Sample

For this study we had a total of 42 university students (22 males and 20 females) with ages ranging from 18 to 34 ($M = 22.71$; $SD = 2.95$). 31% of the participants had interacted with a robot before and 25% frequently played the pantomime game. Half of the participants were randomly assigned C1, and the other half to C2. One participant from each condition were later excluded due to not complying with the questionnaire instructions.

5.2 Procedure

Upon arrival to the experimental setting participants were given an explanation about the study and as they enter a room where the robot was, the robot would see him/her and start interacting. A word category would be projected on a wall, and the robot would pantomime it through an animation. As in a regular pantomime game, the participant should verbally keep on guessing the correct word until either a correct answer was performed, or until time was out (40 seconds per word). The robot would then let them know if they either got it right, or if they were unable to answer within the time limit, and would then move on to the next word, until it finished. In total there were seven rounds. They were told they would understand when the game was finished, and could then leave the room, as the experimenter would be waiting right outside. Upon the interaction, the participant was led to a private room to answer the questionnaires. At the end of the study, a lottery was ran to draw thank you gifts between the participants (12 movie tickets were drawn within the 42 participants). In C1, upon each guess, the robot changes its posture depending on a hot-cold measure, in order to hint the player about their guess; C2, in which the robot maintains the same posture ('neutral') throughout all the guesses.

In a physically separate room, two wizards (W1 and W2) teamed to replace an artificial intelligence capable of quickly assessing the hot-cold quality of the participants' guesses. This design was chosen due to the an open-ended game vocabulary, and because participants were not equipped with a wearable microphone. W1 listened to the player's guesses through a hidden microphone and based on a predefined list of words that are semantically similar to the correct guess, W1 would perform a measure of hot, warm or cold. The list for each answer contained approximately 30 words, ordered

Table 1: The questions used in each specific measure on the Ahoy study.

Performance	The robot was good at pantomiming the words.
	I was able to think of words that were perhaps being pantomimed by the robot.
Animation	
Quality	During the pantomime, the motion of the robot seemed natural.
	The robot's movement while I was attempting to guess was smooth and natural.
Lifelikeness	The robot seemed to be alive.
	The robot reminded me of the characters I know from movies.
Staging	The robot performed the pantomimes in a way that it was easy for me to see what he was doing.
	The robot was moving in tune with me while I was trying to guess the correct answer.
Thought	The robot seemed to understand the concept of the words it was pantomiming.
	The robot thought of every word before it performed the pantomime.
Motivation	The robot was enthusiastic with my attempts to get the right answer.
	The robot wanted me to get the right answer.
Intention	
Recognition	The robot gave me tips while I was trying to guess the right answer.
Legibility	I was able to understand, through the robot's tips, if I was far or close to the right answer.
	The robot's tips helped me to get the right answer.
	The robot's tips seemed coherent with my guesses.

alphabetically, organized into hot, and warm words and formatted in order to provide a fast visual search by the wizard. They were gathered beforehand, both based on an online resource[2], and on the experimenters' intuition. Any word not on the list was considered to be cold. Upon assessment, W1 would verbally notify W2 of the result (hot, warm or cold), and W2 would use a simple WoZ interface to quickly trigger a new posture for the robot. Both wizards were previously trained by the experimenters, both informally, and in an initial pilot version of the study. Stable communication between the wizard-room and experimentation-room was guaranteed by an ethernet cable and audio extension cord (about 30 meters each).

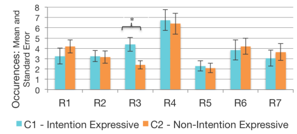
5.3 Measures

We considered questions concerning three types of specific subjective measures: ***Performance***, ***Animation*** and ***Intention***. ***Performance*** was included to assess how well the robot was perceived to play the game. ***Animation*** was measured looking at the following aspects: *Quality*, *Lifelikeness*, *Staging*, *Thought* and *Motivation*. ***Intention*** was measured using the perception of *Recognition* and *Legibility* of the motion. Table 1 presents the questions used in each specific measure. The subjective measures taken from literature were ***Perceived Message Understanding*** and ***Co-Presence***, from the *Networked Minds* questionnaire [13], the ***Inclusion of Other in Self*** (IOS) measure [4], ***Perceived Adaptability*** from the *Almere model* [14], and finally, the dimensions of ***Perceived Intelligence***, ***Animacy*** and ***Likeability*** from the *Godspeed* questionnaire [6]. All the questionnaire scales except IOS were answered in a 6-point Likert scale and when necessary, items were shuffled to mask for their dimensions. The IOS measure was answered in a 7-point scale.

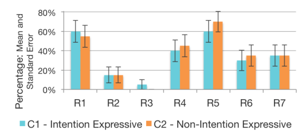
5.4 Results

To understand if our algorithm allowed Adelino to convey an intention while gaze-tracking users, statistical analysis was performed on the subjective data collected through the questionnaires, and the objective data collected during the interactions. The Shapiro-Wilk test was used to test of distributions are normal or non-normal. Where normal distribution existed, we performed a t-student for independent samples. When the normality assumption was not met we used a Mann Whitney U test.

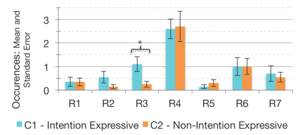
[2] http://swoogle.umbc.edu/SimService/top_similarity.html



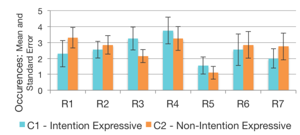
(a) Average number of *Guesses*, per round.



(b) Percentage of participants able to guess the *Correct* word, per round.



(c) Average number of *Hot* guesses, per round.



(d) Average number of *Cold* guesses, per round.

Figure 9: Objective data from the Ahoy study: analysed per round.

Regarding the specific subjective measures, we expected to find significant differences in the Intention measure. However, we also expected that all the other measures would not be affected by our algorithm, supporting solely that the participants of C1 should perceive the robot to have the intention of providing hints, while those of C2 had not. We immediately start by verifying that the *Intention Mean* measure (mean of both *Recognition* and *Legibility*), revealed a significant difference between the two conditions, showing that in C1 the hint-providing intention of the robot was perceived to be higher than in C2. However, given that the mean value for C1 was not very high we also analysed the *Intention Recognition* and *Intention Legibility* measures in separate. Both presented significant differences between conditions, with *Recognition* providing a much more accentuated result.

Neither *Performance Mean* nor *Animation Mean* presented significant results between conditions. That means that the use of the expressive postures in this scenario did not make the robot seem either better at playing the game, nor more animate, and both of these measures had positive results. When we broke down the *Animation* sub-measures, we did however find a significant difference between conditions for the *Staging* measure.

Regarding the non-specific subjective measures taken from literature, only the *Animacy* dimension of the *Godspeed* questionnaire reported significant differences between the two conditions. As to the objective measures, differences between conditions were reported only for the **Round**-measures *Duration*, *Number of Guesses* and *Hot Guesses* in Round 3. Figure 10 summarizes the objective data collected, concerning ammount and quality of guesses. These graphs concern the average of all the complete sessions.

5.5 Discussion

The data collected through the Ahoy study provides evidence to supported all of our hypothesis. **H1** is supported given that there were no significant differences between conditions for the *Performance* measure, and **H2** for the *Animation* measure. In particular, related with H1, although the mean scores were not very high, we

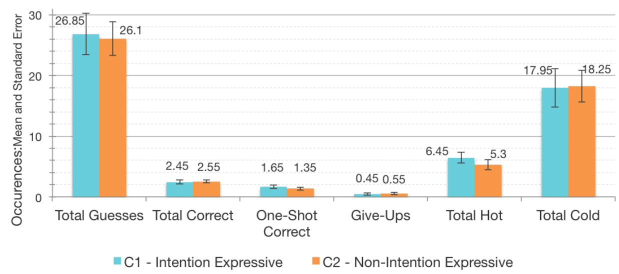
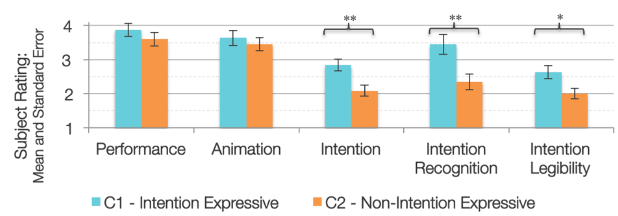
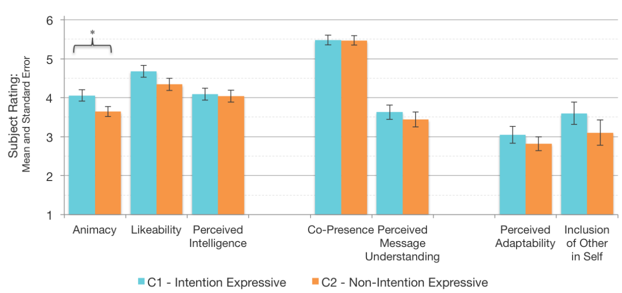


Figure 10: Objective data from the Ahoy study: Mean value of each objective measure (except *Total Duration*), per condition.



(a)



(b)

Figure 11: The results collected from the specific measures (subfigure a) and the measures taken from literature (subfigure b). An asterisk (*) reports a $p - value < 0.05$. A double-asterisk (**) reports a $p - value < 0.01$.

still found that the use of our algorithm did not bias the perception how well the robot played the game. H3 and H4 are supported as both the Intention-Recognition and Intention-Legibility measures rendered significant differences.

5.5.1 Specific Subjective Measures. The results collected for the subjective specific measures are presented in Figure 11a. Individually, we feel that most of these measures and sub-measures did not report strong values, i.e., the mean values are not very far from the scale's median value, and on some cases are even below.

The results collected for the subjective measures taken from literature are presented in Figure 11b. Within the non-specific measures, some had been considered in order to bring additional support to the specific ***Intention*** measure, while other were considered in order to support the specific ***Animation*** measure.

5.5.2 Objective Measures. The objective measures collected presented (Figure 9)no relevant differences among conditions. Although statistically significant differences were reported on some measures in one of the round, we considered it to be an isolated occurrence most likely due to a small sample size. Moreover, Round 3 was noted to be the most complicated round because most people were unaware of what the pantomimed word was at all (it was a Metronome, a device used during musical training). Not only did we see only one participant to be able to guess the word, but during informal post-experimental conversation with some participants, most of them asked us what it was. Therefore we should not draw conclusions regarding any factor reported solely on Round 3. The fact that there were no other differences reported means that while the participants in C1 did perceive that the robot was trying to help them, in the end that help did not translate into a better outcome for the participants' performance in the game.

6 CONCLUSIONS AND FUTURE WORK

In this paper we have presented the use of ERIK, an innovative inverse kinematics algorithm that allows an articulated robotic embodiment to be expressive while orienting in a given direction. In order to demonstrate the algorithm, we crafted the Adelino robot, which has a 5-dof articulated embodiment with a simple face, in order to be used as an expressive interactive character. The Ahoy pantomime game was developed, featuring the Adelino robot. This game was used as an interactive scenario in a user study aimed at discovering if the ERIK algorithm was capable of producing motion that provides the robot with the illusion of life, while still exhibiting useful and meaningful expressive behaviour that players were able to decode, understand and use throughout the task.

This experiment has provided us with initial evidence that our design methodology and animation techniques allows us to build robots that are able to interact autonomously with users, while *knowing* how to animate themselves correctly in order to convey meaningful expression, even with complex articulated embodiments, and while solving for various expressive goals (e.g. posture and gaze orientation).

The Ahoy scenario presented, was, however, not fully autonomous, and despite the initial positive evidence, we believe further developments will allow us to achieve stronger results. As such, our next step will be to create a new interactive scenario, featuring a fully autonomous Adelino-like robot, and tailored based on the lessons learned from this experience. Further objective tests will also be conducted with the ERIK algorithm, in order to allow us to both properly evaluate and refine the quality of the resulting motion.

ACKNOWLEDGMENTS

This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with references UID/CEC/50021/2013 and SFRH/BD/97150/2013.

REFERENCES

[1] Andreas Aristidou, Yiorgos Chrysanthou, and Joan Lasenby. 2016. Extending FABRIK with model constraints. *Computer Animation and Virtual Worlds* 27, 1

(2016), 35–57. https://doi.org/10.1002/cav.1630

[2] Andreas Aristidou and Joan Lasenby. 2009. Inverse Kinematics: a review of existing techniques and introduction of a new fast iterative solver. *University of Cambridge* (2009).

[3] Andreas Aristidou and Joan Lasenby. 2011. FABRIK: A fast, iterative solver for the Inverse Kinematics problem. *Graphical Models* 73, 5 (2011), 243–260. https://doi.org/10.1016/j.gmod.2011.05.003

[4] Arthur Aron, Elaine N. Aron, and Danny Smollan. 1992. Inclusion of other in the self scale and the structure of interpersonal closeness. *Journal of Personality and Social Psychology* 63, 4 (1992), 596–612.

[5] Paolo Baerlocher and Ronan Boulic. 2001. Parametrization and range of motion of the ball-and-socket joint. *IFIP Advances in Information and Communication Technology* 68 (2001), 180–190. https://doi.org/10.1007/978-0-306-47002-8

[6] Christoph Bartneck, Dana Kulić, Elizabeth Croft, and Susana Zoghbi. 2009. Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots. *International Journal of Social Robotics* 1, 1 (2009), 71–81. https://doi.org/10.1007/s12369-008-0001-3

[7] Cynthia Breazeal. 2008. Towards Sociable Robots. *Robotics and Autonomous Systems* 42, 3-4 (2008), 167–175.

[8] Albert Van Breemen. 2004. Animation engine for believable interactive user-interface robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems - IROS '04*, Vol. 3. 2873–2878. https://doi.org/10.1109/IROS.2004.1389845

[9] Samuel R. Buss. 2009. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. *University of California, San Diego, Technical Reports.* (2009). https://doi.org/10.1016/j.neuroscience.2005.01.020 arXiv:NIHMS150003

[10] John Canemaker. 1996. *Tex Avery: The MGM years, 1942-1955.* Turner Publishing.

[11] E. Goldberg. 2008. *Character Animation Crash Course!* Silman-James Press. https://books.google.pt/books?id=dwWePAAACAAJ

[12] Jesse Gray, Guy Hoffman, Sigurdur Orn Adalgeirsson, Matt Berlin, and Cynthia Breazeal. 2010. Expressive, interactive robots: Tools, techniques, and insights based on collaborations. In *ACM/IEEE International Conference on Human-Robot Interaction - HRI '10 - Workshop on What do Collaborations with the Arts Have to Say About Human-Robot Interaction.*

[13] Chad Harms and Frank Biocca. 2004. Internal Consistency and Reliability of the Networked Minds Measure of Social Presence. *Seventh Annual International Workshop: Presence 2004* (2004), 246–251. http://cogprints.org/7026/

[14] Marcel Heerink, Ben Kröse, Vanessa Evers, and Bob Wielinga. 2010. Assessing acceptance of assistive social agent technology by older adults: The almere model. *International Journal of Social Robotics* 2, 4 (2010), 361–375. https://doi.org/10.1007/s12369-010-0068-5

[15] Guy Hoffman and Wendy Ju. 2014. Designing Robots With Movement in Mind. *Journal of Human-Robot Interaction* 3, 1 (2014), 89. https://doi.org/10.5898/JHRI.3.1.Hoffman

[16] Tiago Ribeiro, Doug Dooley, and Ana Paiva. 2013. Nutty Tracks: Symbolic Animation Pipeline for Expressive Robotics. *ACM International Conference on Computer Graphics and Interactive Techniques Posters - SIGGRAPH '13* (2013), 4503.

[17] Frank Thomas and Ollie Johnston. 1995. *The Illusion of Life: Disney Animation.* Hyperion. 576 pages.

[18] L.-C.T. Wang and C.C. Chen. 1991. A combined optimization method for solving the inverse kinematics problems of mechanical manipulators. *IEEE Transactions on Robotics and Automation* 7, 4 (1991), 489–499. https://doi.org/10.1109/70.86079