An Intelligent Design Interface for Dancers to Teach Robots

Heather Knight, Reid Simmons[1]

***Abstract*— Dancers are human *Expressive Motion* experts and could theoretically help robots communicate their state to people, e.g., rushed, confused, curious. The problem is twofold: first, dancers are trained in human-motion whereas many robots are non-anthropomorphic, and second, most dancers are not programmers. This is where the present interface is useful: the robot demos a batch of motions, in person, and the dancer, who knows expressive motion when she sees it, rates each path's success at communicating a particular state. Using an evolutionary algorithm, the interface – where feedback is recorded on the robot's screen and motion is demonstrated via the robot – calculates a new batch of motions that explore variations of the top-rated paths from the previous generation. This approach addresses the challenges of visualizing the expressive potential of non-anthropomorphic robots, while also ensuring path characteristics are reproducible via the robot's motion controller. The purpose of the interface is to help a non-expert negotiate a high-dimensional space of robot motion expression. Thus, it also has interactive functionality enabling users to freeze a feature value they like, or reset all features to begin again. To illustrate the system, this paper includes the results of two dancers designing motions for an omni-directional mobile robot, showing convergence with every generation. In reality, motion designers may have many authoring styles – exploring multiple solutions before honing in, or being satisfied easily versus getting each detail exactly right. By combining human-in-the-loop machine learning with direct authoring, we create a kinetic conversation between the robot and the dancer, and gain the ability to model knowledge from complementary fields.**

I. Introduction

Machine learning approaches are great for finding patterns in data, but struggle to model complex concepts like social behaviors. People, on the other hand, are highly adept at performing and recognizing nuanced and contextualized social behaviors. This paper combines machine learning and human authoring to enable non-programmers to design social behaviors for robots. It presents a hybrid robot-motion design interface in which a robot demonstrates iterative batches of possible motions to the human trainer, collects their ratings for each trajectory, and recombines the attributes of the top-rated paths. The desired robot state communication (e.g., curious) is decided before the program begins, and the trainer can choose to save a feature value that they like, or re-initialize the features to random values if the system gets stuck in a local minima, or they want to try something different.

*This work was supported by (while the second author was working at) the National Science Foundation

[1]The authors completed this work at the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, United States. knighth@oregonstate.edu, reids@cs.cmu.edu

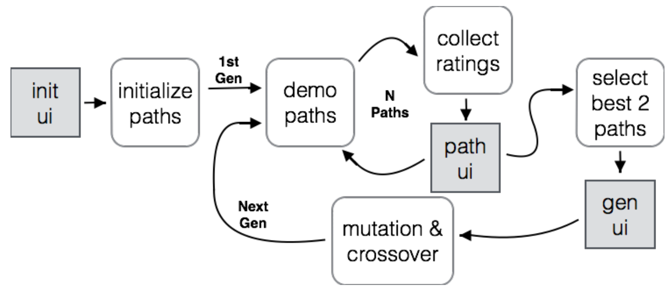

Fig. 1. Interactive Training Mode: the dancer can define subsets of the feature space to explore, save feature values they like, or reinitialize features individually

The reason dancers were deemed valuable contributors to robot Expressive Motion design is because their expertise is entirely focused on using motion, and also because the authors were adapting a system from dance and theater training to simple robots – the Laban Effort System. As there exist Certified Motion Analysts specializing in this method, the authors sought to incorporate their knowledge into their robots' motion design. The creation of this interface was motivated from the conversations with these experts which identified difficulty to visualize robot motions, and translation of qualitative expertise to the quantitative information as key objectives.

Many machine learning techniques are designed for expensive off-line training stages and fast runtime phases. Although solutions have been proposed in the context of online learning [1] or interactive image processing [2], these methods usually rely on a strong human supervision or expensive training procedures. We instead propose a novel intelligent design interface by combining humans-in-the-loop with an evolutionary algorithm. The evolutionary algorithm helps to interpret in real-time the high-dimensional space of the robot motion expression: it is intuitive to understand, and allows for human feedback between each generation.

A pragmatic advantage of this approach over direct expert motion demonstrations (e.g., [3]) is that the robot's motion will always be constrained to motions that a robot can execute. The *generations* also help the dancers intuitively understand the robot's feature space and motion capabilities (Fig. 2). Whatever algorithmic approach future variants of this interface use, they need to be capable of batching demonstrations, and receiving human-in-the-loop feedback.

While human motion experts may not have initially been able to imagine what motions were possible for our non-anthropomorphic mobile robot, during training, one dancer

Fig. 2. CoBot robot following oscillating pathway down a hallway with in a shared environment with people.

said, “I Know It When I See it.” This inspired the title of the system, **KIWIS: a Know-It-When-I-See-it interface**.

The paper begins with related work (Section 2), next presenting KIWIS (Section 3), the motion-design interface. KIWIS provides two modes for the dancer to explore the robot motions: basic and interactive. The former is purely algorithmic, while the second allows the dancer to hand set feature parameters, reducing the state space for exploration, helping speed convergence or nudging the system out of local minima. The robot test platform and available motion features (Section 4) leverage the authors’ past work applying the Laban Effort System to mobile robots [4] [3]. To demonstrate the system (Section 5), the paper includes four sample sequences of dancers designing motions for an omni-directional mobile robot. It finishes with system reflections (Section 6), and conclusions about the promise of combining human and machine expertise (Section 7).

II. Background

A. Why Expressive Motion Matters

In this work, *Expressive Motion* refers to parameterizing of a robot’s motion features in order to communicate a particular robot state. Like previous work in animation [5], Expressive Motion seeks provides “adverbs” to the robots “verbs.” This concept is further defined in [6].

People treat robots similarly to a human co-worker or service agent, judging them by human social rules [7]. Thus, the successful integration of robots in human contexts will depend, in part, on how well they can interact with people. As described by Don Norman [8], author of The Design of Everyday Things, “Cognition is about understanding the world, emotion is about interpreting it, saying good/bad, safe/dangerous, getting us ready to act... That’s why we can tell the emotions of somebody else, because their muscles are acting subconsciously.” Robots will be more effective if they can establish rapport with people, and almost certainly rejected if they regularly irritate or offend us [9] [10].

B. Related Work in Robot Motion

While much of the early work in expressive robotics emphasized robots with complex morphologies [11] [12] our work focuses on the simple robots that are currently entering human environments, like vacuum cleaners [13], service robots [9], or information kiosks [14]. It turns out that people automatically anthropomorphize the motion of even very simple shapes (see Fig. 2), applying storytelling to interpret their social significance [15]. In fact, studies show that people create stories about the attitudes and social behaviors of computer graphics [15] [16], point-light displays [17], the motions of a single-axis door [18], ottomans [19], flying robots [20], rolling balls [21], and even a robotic stick [22]. Thus, even low degree-of-freedom robots can use motion to communicate aspects of their internal states, from boredom to confidence, urgency to curiosity [3].

Inspirations for this project include interactive motion training for robot arms with fuzzy logic [23], and screen-based motion simulators with feature sliders [24] [5]. Previous researchers have collected human motion demonstrations to aid robot motion design. They used a variety of interfaces: sympathetic, remote control, direct joint movement [22], and motion tracking [3]. This work takes a different approach, in that the robot is demonstrating motions to people. This approach also unique in that it learns expression parameters via robot demonstration and interactive algorithms.

III. The Know-It-When-You-See-It Interface

The goal is to enable ”motion conversations” between motion trainers and robots, which originated as a desire to adapt the insight of dancers into the motion demonstrations of a mobile robot. For example, most people would agree that a robot that is rushed would move at higher velocity than one that is not. But what seems like a socially appropriate transit velocity down a hallway might seem like a threatening choice if directed at a person. These examples highlight the importance of contextualized motion, at which dancers are adept.

In this interface, the robot demonstrates batches of possible motions and asks the dancer for their feedback, integrating that data into future demonstrations. It uses an evolutionary algorithm, which is well adapted to human-in-the-loop labels. At a high level, evolutionary algorithms create batches of N variants at each generation, selects two samples which best exemplify the category (decided here by highest ratings), and the sub-features of these samples are randomly combined into the next generation of N samples. The trainer can also provide input about both the important features to explore, reducing or expanding the feature space before each generation; and what their values should be, saving a feature to a particular path value or re-initializing its value).

This method requires no knowledge of programming or robotics, ideally leveraging the expertise of the dancers directly into the kinetics of the moving machine. This section explains how the evolutionary algorithm chooses motion feature parameters in conjunction with a human

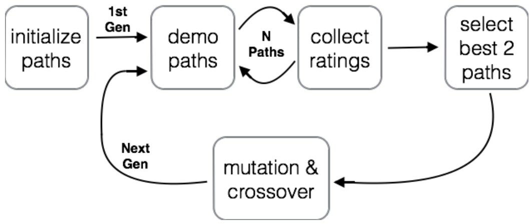

Fig. 3. Basic Training Mode: While evolutionary algorithms are not particularly efficient, they are easy to understand, so the dancer can rapidly grasp how the system works and what motions are possible.

motion coach. It incorporates the dancer's knowledge of 'what looks right' into the basic training mode (optimization only) and an interactive training mode (optimization + dancer input).

Basic training mode

In the basic training mode (Fig. 3), the robot demonstrates a set of motion examples (a generation) by sampling its feature space and asking the dancer to rate each instance on a 1-5 scale. The robot performs $\mathbf{N}$ motions in each generation. The motivation behind using a training algorithm with batches is that the dancer gets a sense of the feature space (and how it ideally reduces over time), and sees the influence of their input. After each generation, $\mathbf{G}$, the algorithm combines the features of the dancers top-rated paths to create the next set of examples, using evolutionary computation [25].

At startup, the dancer is asked to identify the state they are trying to train (similar to Fig. 7). The initial paths randomly sample each robot motion feature to create $\mathbf{N}$ sets of feature parameters. During each generation, the robot enacts the paths corresponding to each feature set one at a time, asking the dancer to rate each path after each demonstration (see Fig. 11 for a sample training sequence). Ratings can be between 1 and 5, where 1 is not at all that state, and 5 is very much that state. For instance, for curious,

1-Not Curious
2-Probably not Curious
3-Neither Curious nor not Curious
4-Somewhat Curious
5-Curious

At the end of the $\mathbf{N}$ demonstrations and ratings, the evolutionary algorithm combines the features of the two top rated paths to create the next set of examples (Fig. 4), the metaphor being of two parents. This continues until the dancer is satisfied (as indicated by their ratings), gives up, or the program reaches the maximum number of generations.

Evolutionary computation is easy to explain to the dancer because of the parallels to human genetics, in which children recombine the chromosomes of their parents (Fig. 5) following:

$$f\langle i\rangle_{child} = \text{random}\langle f\langle i\rangle_{p1}, f\langle i\rangle_{p2}\rangle \tag{1}$$

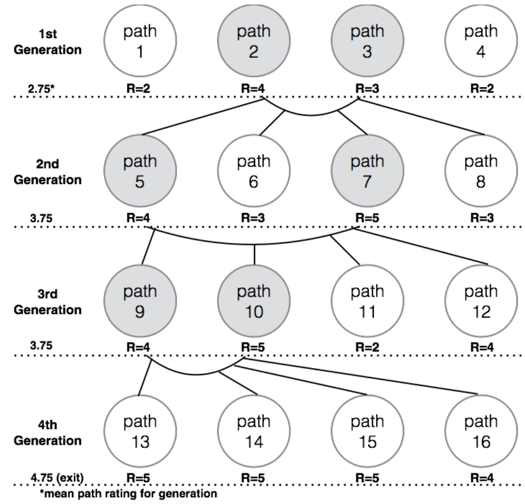

Fig. 4. Illustration of Best Path Selection from Ratings

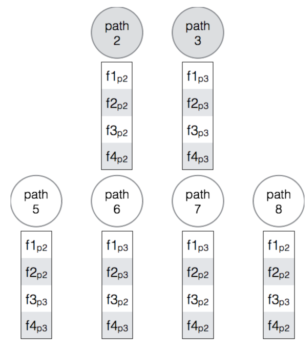

Fig. 5. Example of Crossover (no mutation).

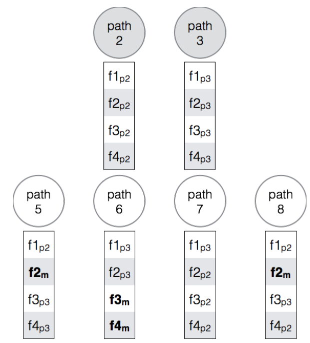

Fig. 6. Crossover with Mutation (rate=0.25).

To avoid local minima and aid in exploration, genetic algorithms also allow for the possibility of mutation (Fig. 6), in which we replace the previous equation with the following:

$$f\langle i\rangle_{\text{child}} = \begin{cases} \text{random}\langle f\langle i\rangle_{\text{p1}}, f\langle i\rangle_{\text{p2}}\rangle & \text{if } 1 - p_{mutate} \\ \text{resample}(f\langle i\rangle) & \text{if } p_{mutate} \end{cases} \tag{2}$$

In other words, the feature value will reinitialize from its full range of possible parameterizations with the mutation probability.

Interactive training mode

In the interactive training mode, the dancer can define subsets of the feature space to explore, save feature values they like, or reinitialize features individually (Fig. 1). This mode uses human expertise to decrease the feature space for exploration, ensuring more rapid convergence, and helping nudge the system out of any local minima it might get stuck in.

Dancer interaction occurs at three points. Upon startup, the Initialization UI requests dancer id and training state (Fig. 7). After each path, the Path UI requests dancer ratings, and provides the possibility of saving or reinitializing one or more features (Fig. 8). To review, reinitialize means

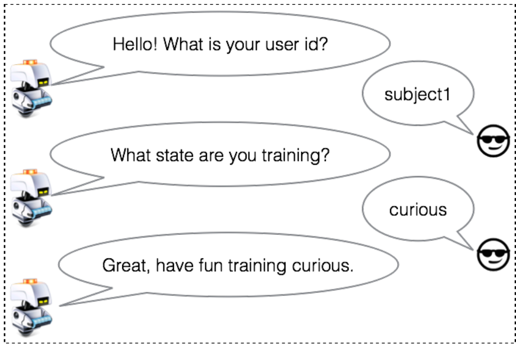

Fig. 7. Initialization UI

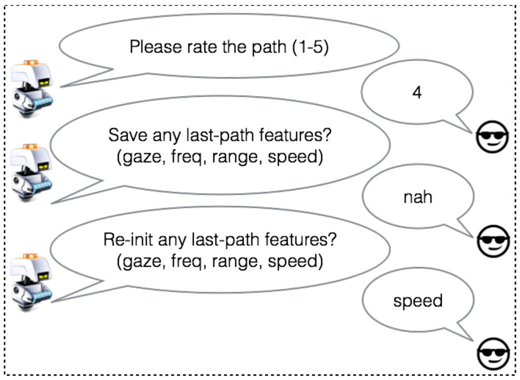

Fig. 8. Path UI

randomly sampling the feature from the future range of allowed feature values. And before each generation, the Generation UI asks the dancer if they would like to limit the feature space and/or have any comments about their design experience and targets (Fig. 9). These comments were not used for training, but rather as qualitative results that help us interpret the trainer strategies and their reactions to the interface.

IV. Test Platform and Motion Features

A. The CoBot

The mobile robot testbed is the CoBot [26]. The Expressive Motion implementations are built on top of its

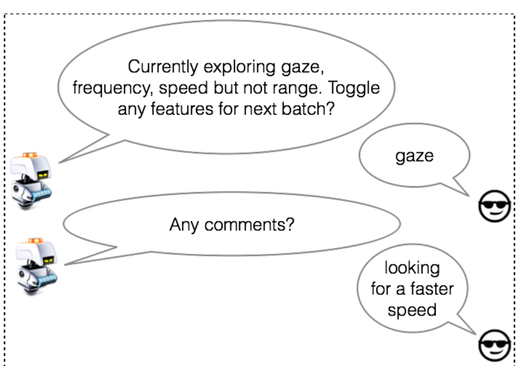

Fig. 9. Generation UI

pre-existing software, sensing and autonomous navigation behaviors. The CoBot robots (Fig. 2) have omni-directional bases that control orientation independently of x, y position [26]. During typical hallway motions, CoBot operates within a corridor of safe travel that is 0.5 meters wide, has a minimum velocity of 0.2 m/s, and a maximum of 1.0 m/s.

Hallway motion is important to this platform. Anyone with a computer science account can reserve tasks on CoBot: delivering a message; picking up and transporting an object [27]; meeting a visitor at an elevator and escorting them to a destination ; and semi-autonomous telepresence, in which dancers are remotely present but use autonomous navigation and obstacle avoidance [28]. Moreover, in earlier work [29], researchers found that the changes in robot velocity during hallway transit on Halloween interrupted the likelihood of people to take candy from the costumed robot.

B. Computational Laban Effort Features

The evaluated motion features come from ongoing work operationalizing a system from dance and theater training to robot Expressive Motion. For the purposes of this paper, the underlying motion features used from the above system were exposed, including velocity, oscillation and orientation features inspired by the Laban Time, Space, Flow Efforts (Fig. 10).

The Space Effort feature set includes linear versus oscillating **path shape**, and **orientation along the path**, (as in section 6.2), explicitly ascribing the oscillation amplitude to the Laban Flow Effort feature set. For example, a shyer robot could express indirect space via muted oscillations, versus one that is more outgoing. Finally, the Laban Time Effort explores the robot's attitude toward time, sudden to sustained, for which we again include a speed feature, but also add oscillation frequency.

The robot's net motion is calculated in a robot-centric coordinate system, in which the robot's travels along this $\mathbf{X}-\mathbf{axis}$, with side-to-side oscillation along the $\mathbf{Y}-\mathbf{axis}$. We calculate **velocity** along the $\mathbf{X}-\mathbf{axis}$ because that is the value that represents the robot's speed down the hallway. If the robot has side-to-side motion, the net-velocity may exceed the $\mathbf{x}-\mathbf{velocity}$.

The oscillation amplitude and **frequency** settings represent sinusoidal variation along the $\mathbf{Y}-\mathbf{axis}$. The **amplitude** feature corresponds to the height of the side-to-side oscillation. Sinusoidal frequencies are constrained to start and end at the center-line of the motion to enable constant start and end positions. To do this, they are limited to discrete values for **frequency** corresponding to whole and half sine waves.

The robot samples its waypoint-based motion commands for **pathshape**, **amplitude** and **orientation** following the equations in section 6.2. As the robot motions occur in the coordinate system of the building map, it also transforms these calculations back into the global coordinate system so the robot can execute them correctly.

V. System Demonstration

We outline the procedure and results from two dancers, who use the interactive training mode to train two defined

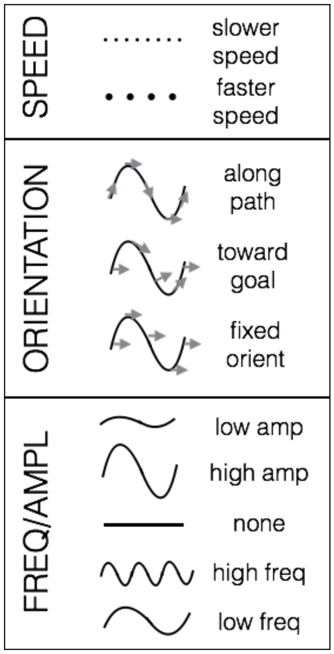

Fig. 10. Illustration of the Feature Space included in this iteration of our Expressive Motion design program.

states: **curious** and **rushed**. We also briefly discuss dancer comments about the training system and its usability.

Training Procedure

When the dancer arrived, they were introduced to the robot and design procedure, and given a handout of Fig. 10 to keep track of the feature space available to them for design. Motion features included speed, orientation condition (covered at length in section 6.2), and sinusoidal oscillation features producing the robot's path-shape, namely, frequency and amplitude. Trainers were invited to design two motions: **curious** and **rushed**. As this was an illustration of the system rather than a formal human-robot interaction study, we did not randomize the motion design ordering.

The first generation acts as a 'practice round,' in which the trainer can experience the design options and ask questions. After the robot demonstrates each path, the guide records the trainer ratings, reviewing any questions they have about the experiment along the way, and recording any comments they have about their goals or experiences at the end of each generation. The training session ends when the trainer ratings are sufficiently high or when the trainer wants to stop. Afterwards, the trainers can share any additional feedback they have about the experience. For this system demonstration we used one layperson (Trainer 1) and one Laban-trained motion expert (Trainer 2).

Training Results

The sequence of demonstrated robot motions in the four training sessions (2 trainers, 2 states) are shown in Figures 11- 14. Blue (solid) lines depict the robot path along the hallway, the spacing between the red horizontal dots represent speed, and the text identifies the orientation setting. The speed data has 0.2 seconds passing between each point, thus, closely spaced dots represent slower speeds, and widely spaced dots are faster speeds. The axis units are in meters.

Overall, trainers spent 10-20 minutes training the motion for each state projection. Trainer 1 continued the program until the final generation paths were almost identical, while Trainer 2's final path retains feature variability outside what may have been the dominant features (i.e. high velocity for rushed and wide range and path-orientation for curious).

Curious: The algorithms allow dancers to choose their own strategies for achieving the kind of robot motion they find to best express a particular state. Trainer 1 parametrized curious features one at a time, first setting oscillation features, then orientation setting, then speed. In contrast, Trainer 2 settled quickly on a wide-range of oscillation and path settings, but said "any speed was good as long as it wasn't too slow" and rated a variety of frequencies highly.

Trainers used the comments to describe their current goal ("Looking for slower speed") or make suggestions about how the algorithm could work ("Last one from this generation was the best representation and the first one from the first generation. Don't know if those 2 can give birth?"), both of which could help explain the choices they are trying to make.

Rushed: Both trainers included low-range but visible oscillation in their final rushed paths. While Trainer 1 did not have a purely linear path to choose from, Trainer 2 chose a variety of oscillation types, but also preferred low-range visible oscillation (as expressed via comments).

The randomness of the algorithm means that desired features do not always appear, and in other cases they appear rapidly. For example, after the 2nd generation of rushed, Trainer 1 says, "didn't see low range," and after the 4th, "finally got good range, was starting to give up." On the other hand, Trainer 2 said, "I had an idea of what I wanted, but then something else came out. I enjoyed finding out the kind of rushed I liked most."

VI. Possible Extensions

The diversity of solutions within our pilot dancers raises the concept that there may be several possible training designs that could work well to express particular states. Trainers also suggested features to add, ("I want the robot to look around more") or labels for paths ("that one seemed cautious") that could potentially suggest future extensions of our feature space or UI, e.g., allowing dancers to add labels to particular paths irrespective of the current training category. Another desired addition was the ability to hand-set particular feature values, rather than just reinitialize them.

Future versions of this system could also leverage direct manipulations of the Laban Setting, perhaps in continuous space. The outputs of the current system could be applied to calibrating Laban motion features, or establishing a corpus of expert-designed robot motions. Beyond training specific state communications, one could also use this system for two other practical purposes:

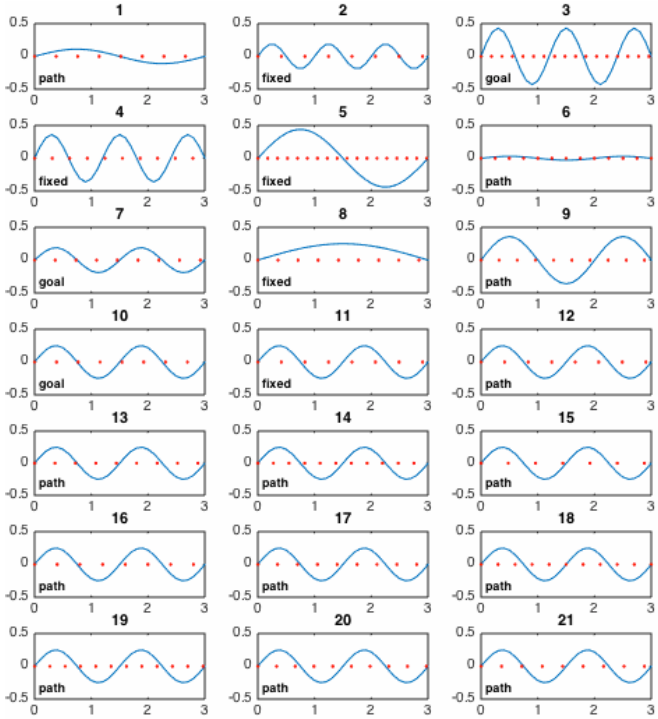

Fig. 11. Curious Training Session, Trainer 1 (interactive). Each row corresponds to a generation, G, of motion traversal demonstrations. In this example, G_1 consists of paths 1, 2, and 3, and G_7 includes paths 19, 20, and 21. Trainer provides a rating after each path and has the option to make comments at the end of each generation.

1) Calibration - tuning the feature levels such that it clearly communicates the desired Effort value (relevant to both binary and continuous Effort Setting representations)
2) Degree - feature levels may vary non-linearly between one Effort pole and another, thus exploring the space between the extremes may benefit from a direct characterization of these non-linear variations.

VII. Conclusions

This paper outlines an intelligent robot motion design interface, using people's ability to read motion at a glance. The robot demonstrates batches of possible motions and asks the motion trainers for their feedback, modeling their preferences to inspire more targeted demonstrations in the next batch. It also provides direct authoring capabilities, in which people can save a feature value they like, reinitialize a feature, or toggle how many features the system explores at a time.

KIWIS has a format that was inspired to enable human creatives to participate: it is both kinetic and computational. Actors and dancers are highly competent at crafting expressive motions that communicate inner state, and dancer comments during the system demonstrations illustrate the ability of dancers and machine learning algorithm to work together, allowing the human to take the creative lead.

Such techniques are highly relevant at a time when innovation in robots for human environments is exploding.

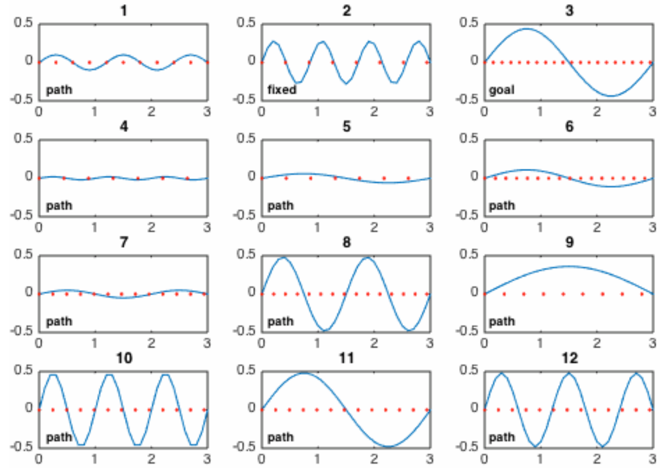

Fig. 12. Curious Training Session, Trainer 2 (interactive).

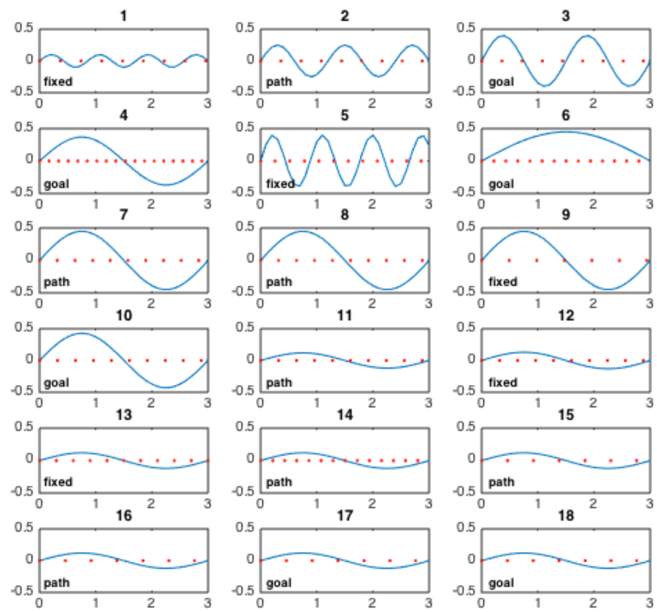

Fig. 13. Rushed Training Session, Trainer 1 (interactive)

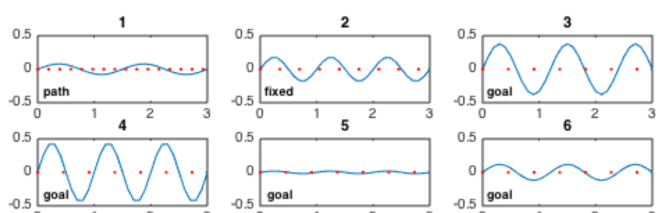

Fig. 14. Rushed Training Session, Trainer 2 (interactive)

The ability of robots to smoothly incorporate expressions into their task motions could make them seem more socially intelligent and aid in their functionality. Think of a security guard robot that is generally authoritative but acknowledges human coworkers in a friendly manner, or a hospital robot that uses urgency features to help make an important delivery under a deadline.

There are many more problems out there that could benefit from combining human and machine insights. One day, a robot's user could KIWIS or something similar to "program" their own robot. One can also imagine giving motion feature feedback to an autonomous car, e.g., if you are uncomfortable with how it seems to take turns at the last possible second, or if you find it agonizing that it stops for three full seconds at every stop sign. Robots should adapt their behavior in different social or relational contexts, and perhaps they could customize to particular users.

It is already a lot of work to get robots to do simple things in human environments. Roboticists should take all the help they can get, and dancers know a lot about motion.

ACKNOWLEDGMENT

This research was funded by a grant from Hasso-Plattner Design Thinking Research Program.

REFERENCES

[1] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[2] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM transactions on graphics (TOG)*, vol. 23, no. 3. ACM, 2004, pp. 309–314.

[3] H. Knight and R. Simmons, "Expressive motion with x, y and theta: Laban effort features for mobile robots," in *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, 2014, pp. 267–273.

[4] H. Knight, R. Thielstrom, and R. Simmons, "Expressive path shape: simple motion features that illustrate a robots attitude toward its goal," in *2016 IEEE International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016.

[5] C. Rose, M. F. Cohen, and B. Bodenheimer, "Verbs and adverbs: Multidimensional motion interpolation," *IEEE Computer Graphics and Applications*, vol. 18, no. 5, pp. 32–40, 1998.

[6] H. Knight, "Expressive motion for low degree-of-freedom robots," 8 2016.

[7] A. Sauppé and B. Mutlu, "The social impact of a robot co-worker in industrial settings," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2015, pp. 3613–3622.

[8] D. A. Norman, *The Design of Everyday Things*. New York, NY, USA: Basic Books, Inc., 2002.

[9] B. Mutlu and J. Forlizzi, "Robots in organizations: the role of workflow, social, and environmental factors in human-robot interaction," in *Human-Robot Interaction (HRI), 2008 3rd ACM/IEEE International Conference on*. IEEE, 2008, pp. 287–294.

[10] R. Mead and M. J. Matarić, "Robots have needs too: How and why people adapt their proxemic behavior to improve robot social signal understanding," *International Journal of Social Robotics*, 2016.

[11] C. Breazeal, "Emotion and sociable humanoid robots," *International Journal of Human-Computer Studies*, vol. 59, no. 1, pp. 119–155, 2003.

[12] C. Breazeal, C. D. Kidd, A. L. Thomaz, G. Hoffman, and M. Berlin, "Effects of nonverbal communication on efficiency and robustness in human-robot teamwork," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 708–713.

[13] J. Forlizzi and C. DiSalvo, "Service robots in the domestic environment: a study of the roomba vacuum in the home," in *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*. ACM, 2006, pp. 258–265.

[14] R. Gockley, A. Bruce, J. Forlizzi, M. Michalowski, A. Mundell, S. Rosenthal, B. Sellner, R. Simmons, K. Snipes, A. C. Schultz *et al.*, "Designing robots for long-term social interaction," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 1338–1343.

[15] F. Abell, F. Happe, and U. Frith, "Do triangles play tricks? attribution of mental states to animated shapes in normal and abnormal development," *Cognitive Development*, vol. 15, no. 1, pp. 1–16, 2000.

[16] F. Heider, "Social perception and phenomenal causality." *Psychological review*, vol. 51, no. 6, p. 358, 1944.

[17] A. P. Atkinson, W. H. Dittrich, A. J. Gemmell, and A. W. Young, "Emotion perception from dynamic and static body expressions in point-light and full-light displays," *Perception*, vol. 33, no. 6, pp. 717–746, 2004.

[18] W. Ju and L. Takayama, "Approachability: How people interpret automatic door movement as gesture," *International Journal of Design*, vol. 3, no. 2, 2009.

[19] D. Sirkin, B. Mok, S. Yang, and W. Ju, "Mechanical ottoman: how robotic furniture offers and withdraws support," in *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 2015, pp. 11–18.

[20] J. R. Cauchard, K. Y. Zhai, M. Spadafora, and J. A. Landay, "Emotion encoding in human-drone interaction," in *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2016, pp. 263–270.

[21] L. Boccanfuso, E. Barney, C. Foster, Y. A. Ahn, K. Chawarska, B. Scassellati, and F. Shic, "Emotional robot to examine different play patterns and affective responses of children with and without asd," in *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2016, pp. 19–26.

[22] J. Harris and E. Sharlin, "Exploring the affect of abstract motion in social human-robot interaction," in *2011 Ro-Man*. IEEE, 2011, pp. 441–448.

[23] A.-A. Samadani, S. Burton, R. Gorbet, and D. Kulic, "Laban effort and shape analysis of affective hand and arm movements," in *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on*. IEEE, 2013, pp. 343–348.

[24] K. Perlin, "Real time responsive animation with personality," *IEEE transactions on visualization and Computer Graphics*, vol. 1, no. 1, pp. 5–15, 1995.

[25] T. Bäck, D. Fogel, and Z. Michalewicz, "Handbook of evolutionary computation," *Release*, vol. 97, no. 1, p. B1, 1997.

[26] J. Biswas and M. Veloso, "The 1,000-km challenge: Insights and quantitative and qualitative results," *IEEE Intelligent Systems*, vol. 31, no. 3, pp. 86–96, 2016.

[27] B. Coltin, "Multi-agent pickup and delivery planning with transfers," Ph.D. dissertation, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, May 2014, available as technical report CMU-RI-14-05.

[28] M. Veloso, J. Biswas, B. Coltin, S. Rosenthal, T. Kollar, C. Mericli, M. Samadi, S. Brandao, and R. Ventura, "Cobots: Collaborative robots servicing multi-floor buildings," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5446–5447.

[29] H. Knight, M. Veloso, and R. Simmons, "Taking candy from a robot: Speed features and candy accessibility predict human response," in *Robot and Human Interactive Communication (RO-MAN), 2015 24th IEEE International Symposium on*. IEEE, 2015, pp. 355–362.