Being safe around collaborative and versatile robots in shared spaces

Being safe around collaborative and versatile robots in shared spaces

Case story: Intendicate

Domain: Manufacturing, Logistics, Public, Consumer

Author: Paul Schweidler

Grant agreement no. 779966
Date 2019-02-27

1 Overview

In human-robot collaboration, safety comes in at least two variants. One of them is concerned with mitigating the consequences of a possible incident, it is called *hard safety*. The other one aims to prevent incidents from happening, which can be called *soft safety*. Soft safety of a system can be achieved or enhanced by proper training of the operator, behavioral guidelines and instructions, or a system-design that helps to prevent errors in the first place. Until today, the last part, i.e. designing robots (and cobots) in an intuitively error preventing manner, has gotten comparatively little attention. And it is easy to see why: the most common domain for cobot use is still the industrial shop floor, where guidelines, proper training, and rigid safety precautions are on the agenda. But to safely use cobots in a more casual and less constrained environment, a self explanatory and error preventing system design will be needed. Of course, the shop floor worker would also enjoy the benefits of such design.

My colleague Linda Onnasch, professor for Engineering Psychology at Humboldt University Berlin, and I, working at a private R&D company specializing in human-machine interaction, had been thinking and talking about this problem for some time. We were wondering: Why is it that robots are still inherently suspicious for dangerous, unforeseeable behavior? Why are there only few good design ideas aiming to reduce this scary imponderability? Wouldn't it be nice to have a predictable robot?

There was this Rethink Sawyer Robot used for research in Linda's Lab. When the robot was switched on, a small screen at head level would display a set of eyes. These eyes were somewhat concerning to us as engineering psychologists - on one hand, because the eyes had a cartoonish look to them, which we felt was humanizing the machine in an inadequate and uncanny way. On the other hand the eyes would blink and randomly look from left to right, but had no functional use whatsoever – they might have even been deceptive.

So we put two and two together and came up with the idea to give the robot a set of functional, non human-looking eyes, that would help to make the robot more predictable.

2 Challenges

A major concern for safety in human-robot interaction is the fact, that the human operator oftentimes does not know what the cobot is up to. The robotic arm may turn, extend or contract and unless the operator is perfectly in the loop, there is no way of knowing what comes next. So, generally our challenge was to enable the operator to intuitively read the robot's "intention". In order to get there, we first took a look at the mechanics of human-human interaction:

When humans interact with each other, they have to infer their counterpart's movement intention all the time to organize cooperative behavior. Next to body pose, reading each other's gaze is key to success. Humans read each other's eyes to gather information about what their attention is focused on and to make inferences about the next movement. Most of the time, we look to where we move. Thus, handovers between humans mostly come safely and without much effort. Following this, our more precise challenge reads as follows:

How can we adapt the natural safety feature of gaze-reading to human-robot cooperation in order to make the robot's movements more predictable for the human operator?

The challenge was further characterized by two problems. First, real eyes are 3D objects positioned in a 3D environment, whereas eyes on a screen are 2D objects positioned towards a 3D environment. Somehow, we had to make up for the loss of a dimension. Second, machines with eyes quickly become scary, eerie or at best funny. So we had to come up with a design, that would successfully circumnavigate the uncanny valley.

Another touchstone was feasibility. We aimed to make the design easily describable as functions and parameters, in order to reduce complexity and hardware requirements for the final solution.

3 Solution

The goal was, to display a dynamic animation on the robot's screen, from which humans could infer the intended direction of movement without much effort. To reach this goal, we formed a consortium consisting of three partners: one for project management and technical implementation, one for scientific evaluation, and one for creative production.

We then started with a creative breakout phase to produce a variety of abstracted eye designs (see fig. 1).

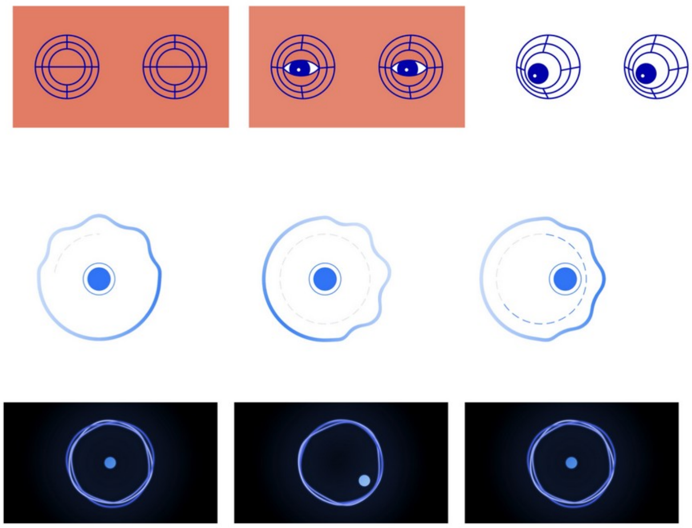

Figure 1: Examples of some initial desing ideas. The single eye designs (row 2 and 3) prove to be problematic in terms of spatial pointing.

After some iterations, two candidate designs were tested for predictability and uncanniness in a set of (hopefully) well-thought-out psychological laboratory experiments (two publications on the way). Both candidates performed sufficiently well, but one of them produced slightly better results. Luckily, that design also was presumably easier to implement – yet there were some caveats.

The heart of our solution is a C++/Qt application rendering the animation receiving three input values, namely the X, Y and Z positions of the gripper. Whereas X and Y could simply be projected onto the screen size to determine the direction of the gaze, handling the Z-value was less straight forward (due to the constraints introduced in section 2). To solve the problem, we converted the Z-Values to a range between 0 and 1, which then determined the degree of pupil displacement (see fig. 2 and 3).

It is noteworthy, that the application did not run on the robot itself, but on a notebook connected to the robot via Ethernet. This was, of course, only possible when the robot was in SDK mode. This again disabled all the convenient features for teaching and storing trajectories. So it took a lot of work that went into two further ROS / C++ applications to record and control the robot, until we were able to test our solution in a final experiment.

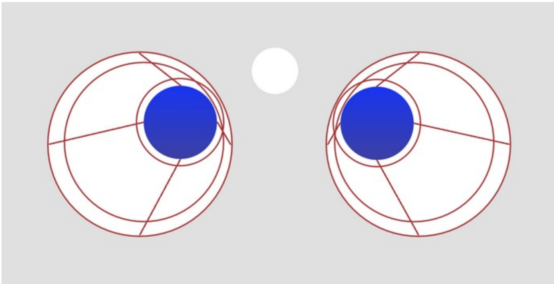

Figure 2: The grey circle marks target (normally not visible). Z-value = 0 leads to strong convergence.

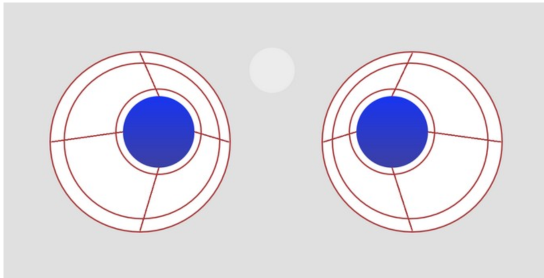

Figure 3: The same x and y coordinates as in figure 2, but with a higher z-value, to simulate a distant target

In this experiment, participants were sitting vis-à-vis to the robot where they had to solve time-critical memory-puzzle tasks on a tablet PC. In order to solve the tasks correctly, participants had to follow instructive movements of the robot. The time-criticality of tasks could be reduced when the participants were able to anticipate the instructive movement of the robot. The quicker the participant knew where the robot would move, the more time was left to solve the puzzle task. Participants were either instructed by the robot displaying our predictive eyes or by the robot displaying just a black screen.

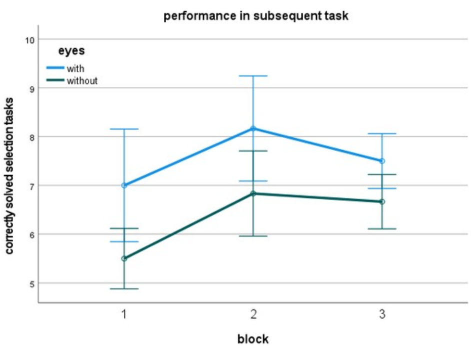

Figure 4: Task Performance on puzzle task. One Block equals 10 trials.

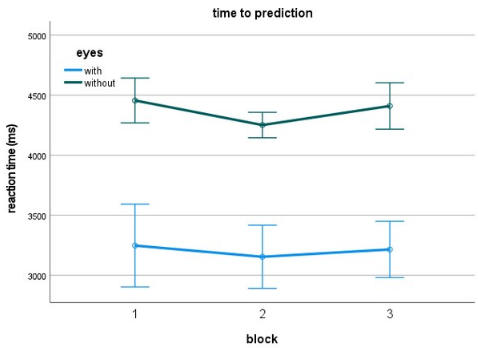

Figure 5: The time, until the participants knew, where the robot would move. Lower means better.

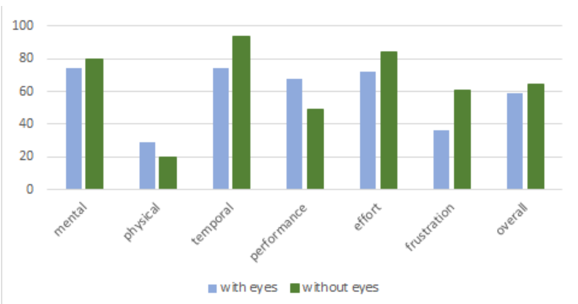

Figure 6: Task load (Hart & Staveland, 1988) experienced during puzzle task. Lower means better.

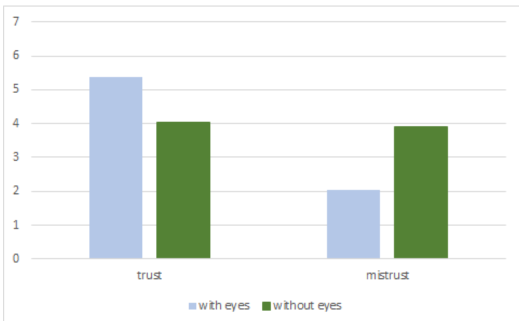

Figure 7: The robot with eyes was perceived more trustworthy (Pöhler, Heine & Deml, 2016)

The results of the study are striking: participants instructed by the predictive robot were able to correctly anticipate its movements, resulting in a lower error rate in the subsequent puzzle task (see fig. 4 and 5). Also, participants reported higher trust towards the predictive robot and felt less stressed and frustrated (see fig. 6 and 7).

Still, there are serious limitations: In our solution, we constrained the movement of the display itself (which normally has one degree of freedom) in order to reduce complexity. Also, there was no gripper action involved. Third, and most importantly, our software works as a standalone application which masks the rest of the software environment. Thus, normal usage of the robot is not quite possible.

4 Considerations

The solution we developed is an excellent proof of concept and a good demonstrator. We showed, that a simple natural interaction pattern can help to enhance the predictability of the robot leading to higher trust and less stress and errors. We implemented the interaction pattern in a way that it can generally be adapted by other cobot users.

Yet, it is far from being a productive solution. To get closer to real world application, we have to (and would love to) replicate the results in real industrial collaborative tasks. We are looking forward to extend the functionality of the software, so that more complex tasks (e.g. including handovers) can be performed.

The final step towards productive use would be to build a smoother system architecture that integrates better into the existing robot programming environment.

Figure 8: The Experiment: The robotic arm points at one of the squares, helping the participant to solve a puzzle game. The quicker the participant gets the color, the easier the puzzle gets. The robot's eyes indicate the trajectory's end point, easily enabling the participants to anticipate the motion.

References:

Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology* (Vol. 52, pp. 139-183). North-Holland.

Pöhler, G., Heine, T. & Deml, B. (2016). Itemanalyse und Faktorstruktur eines Fragebogens zur Messung von Vertrauen im Umgang mit automatischen Systemen. *Zeitschrift für Arbeitswissenschaft*, 70 (3), S. 151-160