COVR Award Agreement: AA9342566381

Collaborative Robots' Perceived Safety
CROPS

Deliverable 2.5: Perceived Safety Toolkit

Date: 30. 6. 2021

Authors: *Kristina Nikolovska, Luka Komidar, Anja Podlesek, Matjaž Mihelj, Sebastjan Šlajpah*

University *of Ljubljana*
Faculty *of Electrical Engineering*

Univerza *v Ljubljani*
Filozofska fakulteta
Oddelek za psihologijo

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 779966.

Contents

1 Introduction

In the future, our daily lives will be more and more affected by the robots' presence. Interaction between humans and robots (human-robot interaction – HRI) is a growing field that has already been introduced to the industry but can also be found outside the factories. It represents a broad interdisciplinary research field that tries to understand, design, and evaluate robot systems used by or with humans.

For a successful human-robot interaction, two distinct types of safety need to be considered: physical safety and perceived safety. When it comes to physical safety, many technologies have been and are being developed (safe mechanical design, safe planning and control, wide range of sensors to ensure safety). But on the other hand, research covering perceived safety is scarce. Perceived safety describes the user's perception of the level of danger induced by the robotic system and the user's level of comfort during the interaction. If we want to increase robots' acceptance and thus provide a safe and productive working environment, it is necessary that people feel safe around robots.

The aim of the project was to research an area of perceived safety, gather users' responses under different controlled conditions, and develop a toolkit for assessing perceived safety based on processed data. The Perceived Safety Toolkit (PST) aims to help robots' developers and integrators create applications that are perceived as safe and readily accepted by the users. Such an approach is a step towards a successful human-robot interaction.

In this deliverable, developed Perceived Safety Toolkit is presented.

2 Perceived Safety Toolkit

The purpose of the designed PST is to help robot integrators understand how designed applications will affect the perceived safety of the user and how successful human-robot cooperation will be. In case when the toolkit predicts very low perceived safety, suggestions, and hints on how to make the applications' perceived safety higher are given. The toolkit takes five input parameters (velocity of the robot, movement type, tool type, distance from the robot, type of human-robot interaction). It predicts four output parameters of the perceived safety of the robot (pleasure, arousal, perceived safety and willingness to cooperate with the robot). In this chapter toolkit's backend and frontend designs are presented.

2.1 Design of the Perceived Safety Toolkit

For the prediction model of the PST we used a design based on neural networks. Neural networks learn (or are trained) by processing examples, each of which contains a known "input" (or combination of "inputs") and a "result", forming probability-weighted associations between the two, which are stored within the data structure of the net itself. The neural network is constructed from three types of layers: input layer, hidden layers, and output layer. As we want to design our prediction model based on the datasets, we have gathered during both experiments presented in Deliverable 1.4 and Deliverable 2.4, the machine learning based on neural networks presents a suitable solution. From the data from both experiments, we have created a training data set. In the dataset, every input has five parameters. In total, we had 84 different sets of inputs, and for each input, we had 30 sets of outputs. Altogether we used 2520 different sets to train our model.

Every output had four different parameters. Part of the data set is shown in Table 1. For each parameter in the dataset, we have defined a numerical value used in the dataset. In Table 2, the defined numerical values for movement types are presented. Numerical values for tool types (safe tool, dangerous tool, combination of both) are stated in Table 3. Distance between the user and the robot was categorised into four categories; numerical values are presented in Table 4. Human-robot interaction was categorised as coexistence or collaboration. In Table 5 numerical values for the type of HRI are shown.

Table 1: Dataset example used for teaching the model (P – pleasure, A – arousal, S – perceived safety, C – willingness to cooperate; data were obtained during Experiment 1 and Experiment 2 with a 9-point scale).

#C	Velocity [m/s]	Movement type	Tool	Distance [m]	HRI	Participant 1					Participant 30			
						P	A	S	C		P	A	S	C
0	0.3	1	1	3	1	9	1	9	9		9	2	9	9
1	0.3	1	2	3	1	8	3	8	9		7	4	6	8
2	1.0	1	1	3	1	6	2	8	9	…	8	3	6	8
3	1.0	1	2	3	1	6	2	9	7		7	5	5	6
		….					…					…		
81	0.75	7	1	0	2	7	2	7	7		8	2	9	9
82	0.75	7	2	0	2	5	2	4	4		9	2	9	9
83	0.75	7	3	0	2	7	2	7	8		9	1	9	9

Table 2: Numerical value of the movement type.

Movement type	Numerical value
Linear forward/backward	1
Linear left/right	2
Linear up/down	3
Circular forward/backward	4
Circular left/right	5
Random	6
Random linear	7

Table 3: Numerical value of the tool type.

Tool	Numerical value
Safe tool	1
Dangerous tool	2
Safe + dangerous tool	3

Table 4: Numerical value of distance from the robot.

Distance [m]	Numerical value
0 - 0.5	0
0.5 - 2	1
2 - 4	2
> 4	3

Table 5: Numerical value of collaboration.

HRI type	Numerical value
Coexistence	1
Collaboration	2

For building the neural network model, we used the python's deep learning library Keras. The model is made of a dense layer, activation function, and optimizer. Activation function is used to determine the output of a neural network. It defines how the weighted sum of the input is transformed into an output from a node or nodes in a layer of the network. We have used a ReLu activation function. The ReLu is half rectified (from bottom). The function output is zero when the output is less than zero and is equal to the output number when the output number is above or equal to zero. Because we do not have a negative number, this activation function fits our model. As an optimizer we have used Adam, an optimization algorithm that can be used instead of the classical stochastic gradient descent procedure to update network weights iteratively based on training data.

2.2 User interface

To make the toolkit user friendly, an application with GUI was created using the python tkinter library. The GUI is presented in Figure 1. The application includes drop-down menus with which the user can describe his/her robotic application and thus define the inputs for the prediction model. Users need to define the velocity of the robot, movement type, tool type, distance from the robot, and type of human-robot interaction. These parameters are further described in chapter 4.1 Input parameters.

GUI also included two buttons. Button `Help` opens a new window with a detailed description of the parameters, as presented in Figure 2. The primary purpose of this window is to provide ad-hoc information to the user by further elaborating the meaning of individual parameters. Button `Calculate safety metrics` opens a new window that shows the results of the perceived safety assessment. Detailed description of the predicted parameters is presented in chapter 4.2 Output parameters.

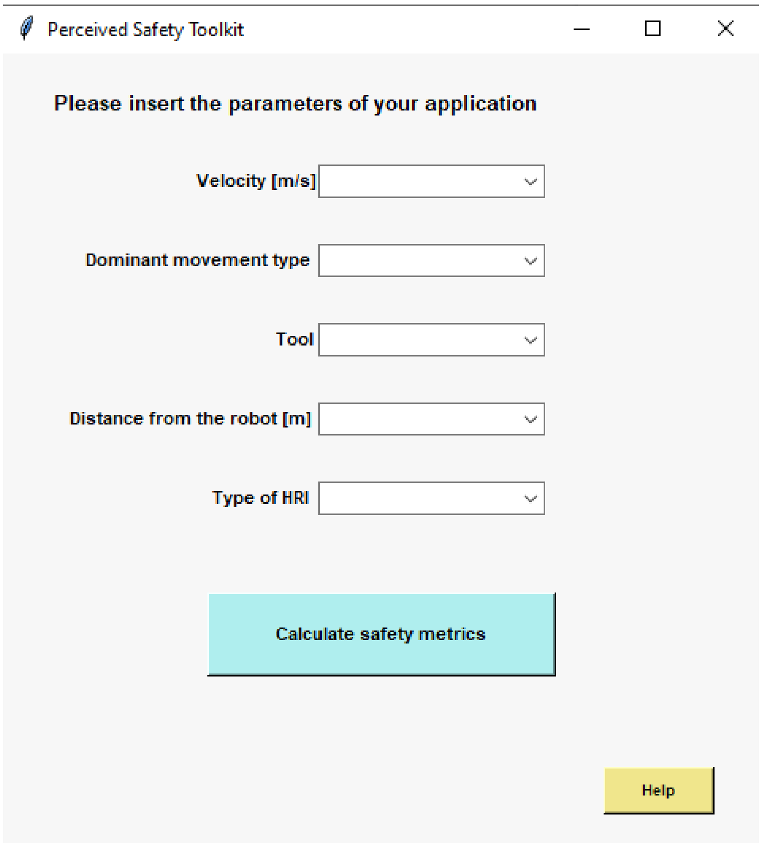

Figure 1: Graphical user interface where user can select parameters describing the robotic application.

Help

Instructions for filling in the parameters

Velocity [m/s]

The velocity that is most commonly used in your application. For selecting the velocity not stated in options check the examples below:

Example 1: For the robot's TCP velocity 0.22 m/s select 0.2 m/s.

Example 2: For the robot's TCP velocity 0.25 m/s select 0.3 m/s.

Dominant movement type

Insert the movement type that is most commonly used in your application. Select the movement type regarding the position of the user.

Linear forward/backward: robot is moving towards/away from the user.

Linear left/right: the robot is moving left-right from the user's perspective.

Linear up/down: the robot is moving up-down from the user's perspective.

Circular forward/backward: the robot is moving in forward-backward circular movement from the user's perspective.

Circular left/right: the robot is moving in left-right circular movement from the user's perspective.

Random: the robot is moving in all direction with linear and circular movements

Random linear: the robot is moving in all direction with linear movements

Tool

Safe tool: tool that has soft, flexible, and rounded construction (e.g., sponge).

Dangerous tool: tool that has sharp edges, rough structure (e.g., kitchen knife).

Safe + Dangerous tool: when you use both types of tools (e.g., robot uses tool changer to change tools during application).

Distance from the robot [m]

The distance between the human and the robot that is most commonly used.

If the user position is borderline, select category with shorter distances (e.g., for distance 0.5 m select category 0 m – 0.5 m).

Type of HRI

Coexistence: the robot and the user work alternately in the same workspace.

Collaboration: the robot and the user work on the same task at the same time in the same workspace.

Figure 2: Help window with additional explanation of the parameters.

3 Validation of the prediction model

The developed machine learning algorithm was tested with proper validation data. In this chapter the test sample and validation of the algorithm are presented.

3.1 Validation sample

The validation sample was based on assessed data that was not used for training the algorithm. We have collected this data during the measurements and were excluded from the training dataset. Again, we had 84 inputs, and for each input we had five outputs. Altogether we had 420 different sets to validate the developed model. The validation dataset was created in the same way as the training dataset. Part of the dataset is presented in Table 6.

Table 6: Dataset example used for testing the model (P - pleasure, A - arousal, S - perceived safety, C - willingness to cooperate).

#C	Velocity [m/s]	Movement type	Tool	Distance [m]	HRI	Participant 1					Participant 5			
						P	A	S	C		P	A	S	C
0	0.3	1	1	3	1	9	1	9	9		9	2	9	9
1	0.3	1	2	3	1	8	2	8	8		7	2	6	9
2	1.0	1	1	3	1	8	1	8	8	…	8	2	7	9
3	1.0	1	2	3	1	7	1	8	8		7	3	5	7
		….					…					…		
81	0.75	7	1	0	2	7	2	9	8		8	2	9	8
82	0.75	7	2	0	2	5	1	8	8		9	2	9	9
83	0.75	7	3	0	2	7	1	7	8		9	1	9	8

3.2 Results

To test the algorithms' accuracy, a validation method was used during training. Pre-build validation function for the models returns the loss and mean squared error. These parameters need to be low which means that results are good. Mean squared error is the average squared difference between the estimated values and the actual value. From the validation process we got the following results:

- prediction of P (pleasure): loss: 0.071, mean squared error: 0.071;
- prediction of A (arousal): loss: 0.099, mean squared error: 0.099;
- prediction of S (perceived safety): loss: 0.178, mean squared error: 0.178;
- prediction of C (collaboration): loss: 0.155, mean squared error: 0.155.

In Figure 3 mean squared errors obtained during training are presented for all four assessed parameters P, A, S, and C. It can be seen that all errors are converging with small possibility for over-fitting. From the results of the validation we can conclude that our model is sufficiently trained and performs good predictions.

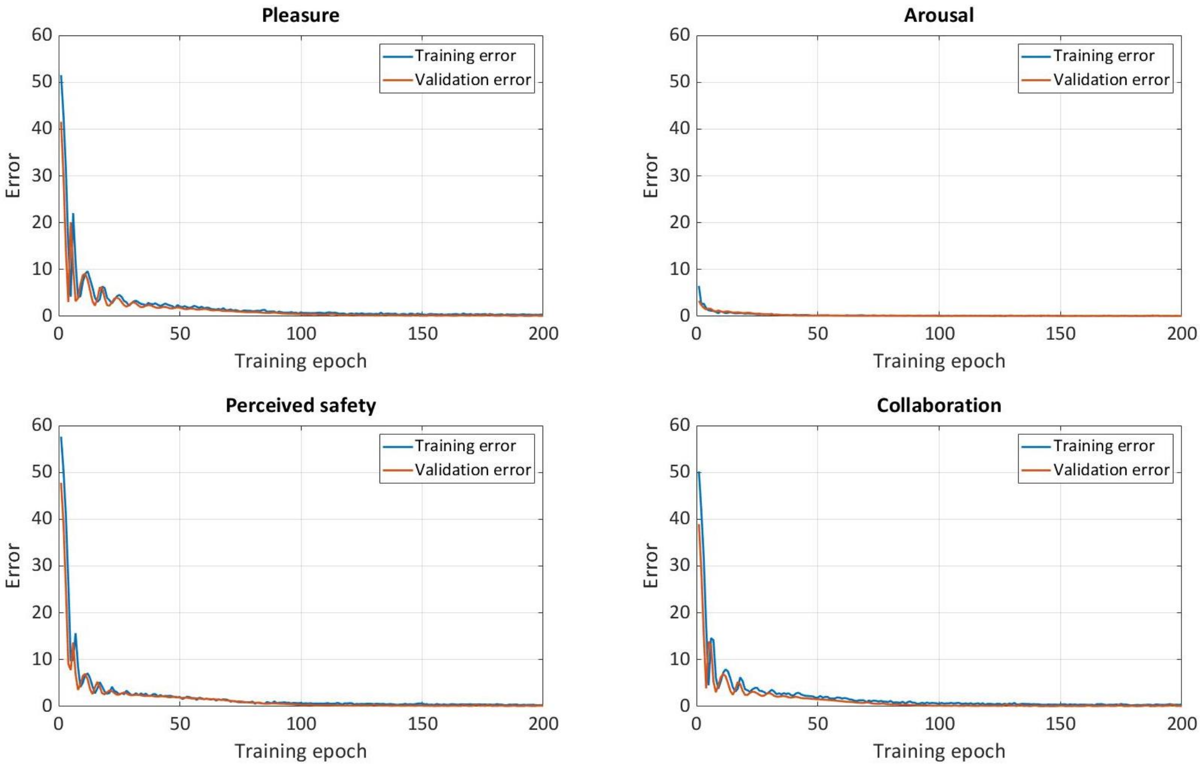

Figure 3: Mean squared error for all four predicted parameters regarding the number of training epochs.

Additionally, we have tested the model by using the test data presented in the validation sample. We have picked random combinations and checked matching of the predictions with the actual data. The result showed matching of 67 % (SD of error 0.71) for pleasure, 69 % (SD of error 0.55) for arousal, 71 % (SD of error 0.76) for perceived safety, and 64 % (SD of error 0.76) for collaboration. Few examples of the used validation data with predicted data are shown in Table 7.

Table 7: Presenting the predicted values and the actual value for randomly chosen combinations.

#C	Predicted value				Actual value			
	P	A	S	C	P	A	S	C
2	7	3	7	6	7	3	7	7
8	7	2	8	8	7	2	7	7
16	8	2	8	8	7	2	8	7
42	7	2	8	8	7	2	7	8
62	7	2	7	7	7	2	7	7
84	7	2	8	8	8	2	8	8

In Figure 4 predictions of perceived safety and arousal regarding the velocity and selected tool for linear front/back movement type during coexistence at a distance 0.5 m – 2 m is presented. The blue solid line presents perceived safety when we have a safe tool (on the graph marked as S-ST) and the blue dotted line presents perceived safety when we have a dangerous tool (on the graph marked as S-DT). The solid red line presents arousal with a safe tool (on the graph marked as A-ST) and the red dotted line presents arousal with a dangerous tool (on the graph marked as A-DT). From the graph we can see that the predicted parameters of perceived safety are lower when we have higher velocity and arousal is higher with higher velocity. Also we can see the impact of the tool on the result. We can see that the predicted perceived safety has lower values with dangerous tool (dotted line) than with safe tool (solid line). Given results indicated that prediction of the parameters is in line with the expected results.

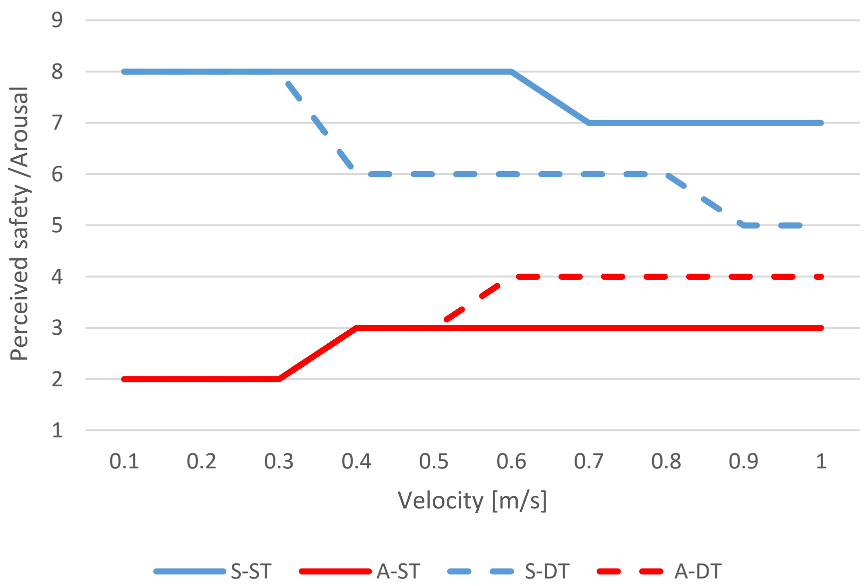

Figure 4: Perceived safety (blue lines, S) and arousal (red lines, A) regarding velocity. The solid lines present the arousal and perceived safety with selected safe too (ST)l. The dotted lines present the arousal and perceived safety with selected dangerous tool (DT). The presented parameters are for linear front-back movement, coexistence, and distance between 0.5 m and 2 m.

Figure 5 represents predictions of perceived safety and arousal regarding the velocity and selected tool for random movement type during coexistence at a distance 0.5 m – 2 m. The blue solid line presents perceived safety when we have a safe tool and the blue dotted line presents perceived safety when we have a dangerous tool. The solid red line presents arousal with a safe tool and the red dotted line presents arousal with a dangerous tool. Again, we can see the similar effect of the velocity on a perceived safety and arousal as in Figure 4.

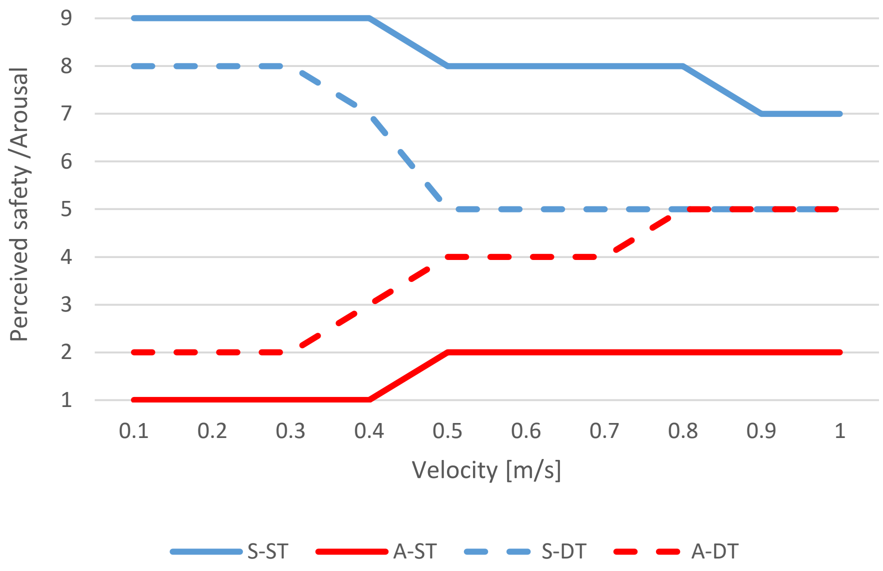

Figure 5: Perceived safety (blue lines, S) and arousal (red lines, A) regarding velocity. The solid lines present the arousal and perceived safety with selected safe tool (ST). The dotted lines present the arousal and perceived safety with selected dangerous tool (DT). The presented parameters are for random movement, coexistence, and distance between 0.5 m and 2 m.

4 How to use the Perceived Safety Toolkit

In this chapter an instruction manual for Perceived Safety Toolkit is presented. It includes a detailed description of input parameters and tips on how to choose them properly. Also, the interpretation of the output parameters is presented with included hints to improve the perceived safety of the robotics systems.

4.1 Input parameters

There are five input parameters describing the robotic application: velocity, predominant movement type, tool, distance from the robot, and type of HRI. The graphical design of the toolkit is presented on Figure X.

4.1.1 Velocity

Parameter Velocity describes the TCP velocity that is most used in your application in m/s. In the drop-down menu there are velocities ranged from 0.1 m/s to 1 m/s, increasing with step of 0.1 m/s. For selecting the velocity that is not in the options, select closest available as presented in the following examples:

- Example 1: For the robot's TCP velocity 0.22 m/s select 0.2 m/s.
- Example 2: For the robot's TCP velocity 0.25 m/s select 0.3 m/s.

4.1.2 Dominant movement type

Parameter Dominant movement type describes the most used movement type in the considered application. Select the movement type regarding the pose of the user.

- Linear forward/backward: robot is moving towards/away from the user.
- Linear left/right: the robot is moving left-right from the user's perspective.
- Linear up/down: the robot is moving up-down from the user's perspective.
- Circular forward/backward: the robot is moving in forward-backward circular movement from the user's perspective;
- Circular left/right: the robot is moving in left-right circular movement from the user's perspective.
- Random: the robot is moving in all direction with linear and circular movements;
- Random linear: the robot is moving in all directions with linear movements.

4.1.3 Tool

Parameter Tool describes the tool that is mounted on the robot and is used in the application. The user can pick from three options that are presented below:

- Safe tool: tool that has soft, flexible, and rounded construction (e.g., sponge)
- Dangerous tool: tool that has sharp edges, rough structure (e.g., kitchen knife)

- Safe + Dangerous tool: when you use both types of tools (e.g., robot uses tool changer to change tools during application).

Decision in which safety category the tools comply with is made solemnly on personal preferences and experience. Use of common sense is advised.

4.1.4 Distance from the robot

Parameter Distance from the robot describes the closest distance between the user and the TCP of the robot/tool mounted on the robot that is most used in m. If the user is moving, pick the closest distance between the user and the robot. If the user is standing in front of the robot, select the standing position as reference position. Four categories are available:

- 0 m – 0.5 m
- 0.5 m – 2 m
- 2 m – 4 m
- > 4 m

If the user position is borderline, select category with shorter distances (e.g., for distance 0.5 m select category 0 m – 0.5 m).

4.1.5 Type of HRI

Parameter Type of HRI describes the type of human-robot interaction used in the application. Two categories are available:

- Coexistence: the robot and the user work alternately in the same workspace.
- Collaboration: the robot and the user work on the same task at the same time in the same workspace.

4.2 Output parameters

Perceived Safety Toolkit predicts four different parameters describing the perceived safety. Values for all parameters are in range from 1 to 9:

- Pleasure: 1 – unpleasant, 9 – very pleasant
- Arousal: 1 – calm, 9 – nervous
- Perceived safety: 1 – very unsafe, 9 – very safe
- Willingness to cooperate: 1 – not at all, 9 – most certainly

In addition to the four parameters, PST also includes interpretation of those parameters from the perceived safety point of view. Three different cases are defined and presented with colour code.

In the first case (green) the application would be perceived as safe as presented in Figure 6. Here the pleasure, perceived safety, and willingness to cooperate are high (numbers close to 9), while arousal is low (value close to 1).

The second case (yellow) covers the applications that are on the verge of perceived safety as presented in Figure 7. The predicted values of pleasure, perceived safety, and willingness to cooperate parameters are close to middle of the scale (e.g., 5), and the value for the arousal is between 3 to 5.

The third case (red) includes the applications that are perceived as dangerous as presented in Figure 8. In this case the values of pleasure, perceived safety, and willingness to cooperate are low while the arousal is high.

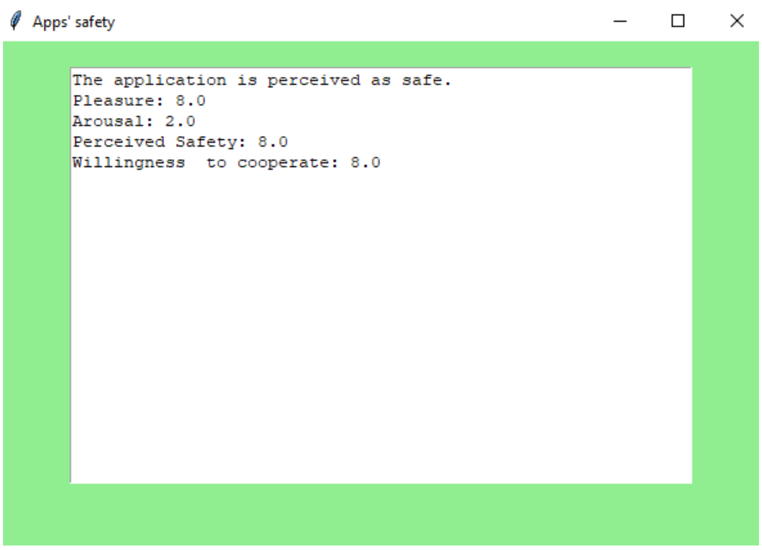

Figure 6: Output example for safe application.

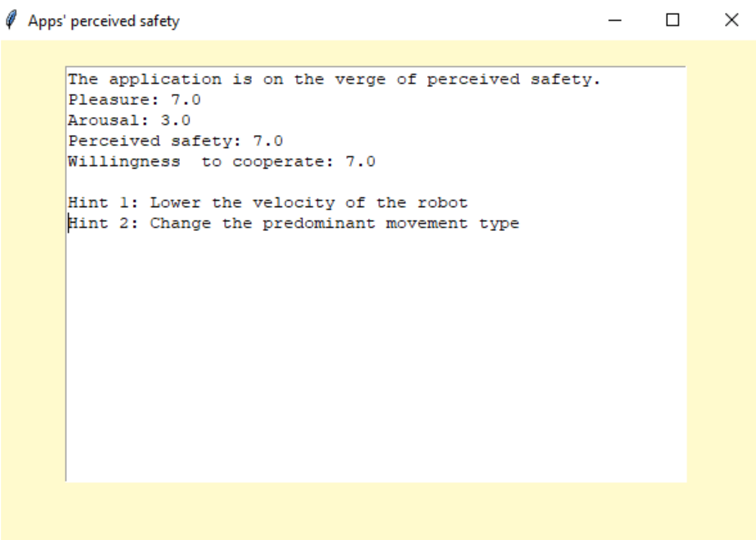

Figure 7: Output example for application that is on the verge of being perceived safe.

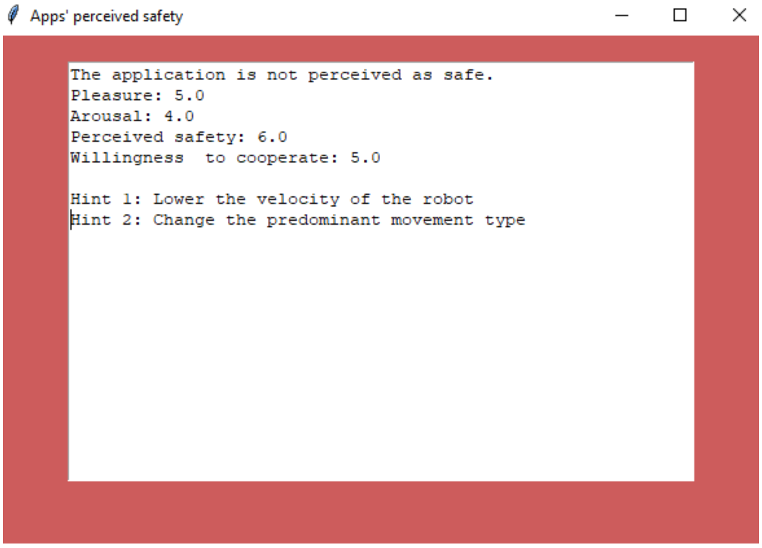

Figure 8: Output example for application that is not perceived as safe.

4.3 Hints

If the robotic application, described with the given parameters, is predicted not to be perceived as safe, additional hints are displayed. Hints can be used as a help for the integrators to identify which parameter should be changed for the application to be perceived as safe.

For each robotic application that is not perceived safe, background prediction of perceived safety with parameters shuffle is performed. Based on the results, proper hints are displayed. Some examples of the hints are:

- Hint 1: Lower the velocity of the robot.
- Hint 2: Change the predominant movement type.
- Hint 3: Change type of the HRI from Collaboration to Coexistence.

The hints should be used only as guidelines. Every user/integrator should consider what is possible to be changed in the current application.

5 Conclusion

Perceived safety has a major role when trying to design a robotic cell that will provide stress-free and comfortable interaction for the human. Presented Perceived Safety Toolkit is a tool that can be used in early stages of the development of robotics systems to help to guide towards user-friendly design. It can also be used during integration phase or even when the robotic cell is already operational. Here small changes in the robotic program (e. g., smaller velocities or a little bit different approach for picking up parts) can lead to better experience for the operator of the robotic cell.

Perceived Safety Toolkit was designed with a user-friendly interface. Users can describe the robotic application with only five parameters for which no special knowledge is needed. The predicted parameters from the PST are translated to simple information that can be easily interpreted by the user. In addition, the results are also colour coded for better understanding. If the addressed application is not assessed as being perceived as safe, additional hints should help the user to understand what can be improved.

Perceived Safety Toolkit presents a valuable tool for improving work environments that include collaborative robots and helps with easier integration of collaborative technologies to every-day use.

References

J. Brownlee, 'How to Choose an Activation Function for Deep Learning', Machine Learning Mastery, Jan. 17, 2021. https://machinelearningmastery.com/choose-an-activation-function-for-deep-learning/ (accessed Jun. 21, 2021).

'Activation Functions in Neural Networks | by SAGAR SHARMA | Towards Data Science'. https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6?gi=bc06f00c1a13 (accessed Jun. 21, 2021).

'Adam — latest trends in deep learning optimization. | by Vitaly Bushaev | Towards Data Science'. https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c (accessed Jun. 21, 2021).

J. Brownlee, 'Gentle Introduction to the Adam Optimization Algorithm for Deep Learning', Machine Learning Mastery, Jul. 02, 2017. https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/ (accessed Jun. 21, 2021).

'Core Layers - Keras 2.1.5 Documentation'. https://faroit.com/keras-docs/2.1.5/layers/core/ (accessed Jun. 21, 2021).