In: 2006 IEEE International Conference on Robotics and Automation, May 15-19, 2006, Orlando, Florida USA

Modeling Intuitive Behavior for Safe Human/Robot Coexistence and Cooperation

Dominik Henrich and Stefan Kuhn

Lehrstuhl für Angewandte Informatik III (Robotik und Eingebettete Systeme)
Universität Bayreuth, D-95445 Bayreuth, Germany
E-Mail: {dominik.henrich|stefan.kuhn}@uni-bayreuth.de
http: //ai3.inf.uni-bayreuth.de/

Abstract **– An intuitive behavior model for safe human/robot coexistence and cooperation is presented. For this purpose, we classify the robot behavior into four cooperation states, which are characterized by two criteria: motion context and interaction context. The motion context can be gross or fine and the interaction context can be free or guided. We describe basic principles relevant to modeling safe and intuitive behavior for robots. The desired robot behavior is detailed for each of the four states along with the necessary transitions between those states.**

Index Terms – industrial robot, safe human/robot cooperation, guided motion, distance controlled velocity

I. Introduction

Humans are skilled at complex tasks and are able to react very flexibly to unknown situations. Industrial robots are very strong, fast, persevering and accurate. The skills of both can be combined in close collaboration.

Today's safety standards prohibit direct collaboration between human and robot because of the unsafe technology involved and the many accidents with robots that have occurred in the past [6]. In industrial settings robots must remain strictly separated from humans.

In order to bring human and robot together and thus combine the skills of both, new methods and safety technology for direct collaboration without separation have to be investigated. There is great demand for safe and ergonomic human/robot coexistence and cooperation, such as flexible handling of heavy workpieces, pick-and-place tasks and programming of multi robot systems.

On the one hand, *coexistence* means that the human and the robot share the same workspace at the same time while working on different tasks. On the other hand, *cooperation* implies coexistence while working on the same task. Both coexistence and cooperation must be safe for the human worker. Cooperation furthermore requires a kind of intuitive behavior on the part of the robot so the human can comprehend the robot motions.

Safety technology avoids unintentional contacts – called collisions. A classification of different types of contacts can be found in [3]. There, contacts are classified into the categories task, control and collision contacts.

Many collision avoidance approaches have been explored, with most of them using sensors to provide local information. For example, in [10] and [7] capacitance sensors are used as an artificial skin. In [9], algorithms for whole-arm collision avoidance for robots with artificial skin are presented. In [12], a wrist-mounted laser scanner is used. In [5] an overview is presented classifying those approaches according to the strategy of collision detection and reaction. Using such an approach permits humans and robots to work in coexistence; they can share one workspace at the same time without being separated. In [4] an implementation of such a system is presented that combines whole-arm collision detection with a global collision avoidance strategy. This approach is based on multiple cameras and a difference image method. A similar system was presented by [1] and [2] but without evasive movements.

Safety technology is insufficient for substantial human/robot cooperation. At least a simple intuitive behavior on the part of the robot is indispensable. Humans must be able to directly understand the behavior of the robot. Our goal is not to complicate the task with complex dialogs but rather to keep it basic. The robot must perform tasks without colliding with the environment and without disturbing the human operator. If necessary, the human operator should be able to guide the robot, for example by directly contacting and leading its Tool Center Point (TCP) via force-following [11].

The presented approach aims to deal with manifold tasks like transporting objects and processing workpieces controlled by the robot program as well as by the human operator. On closer examination, it appears that those tasks have different and even conflicting demands on the behavior of the robot. For example, when transporting an object controlled by the robot program, contact with the environment (apart from the transported object) must be strictly avoided. However, contact is needed when the object ist placed or assemblied. Even worse, permanent contact is needed when transporting an object controlled by the human operator via force-following. For this reason, overall behavior is divided

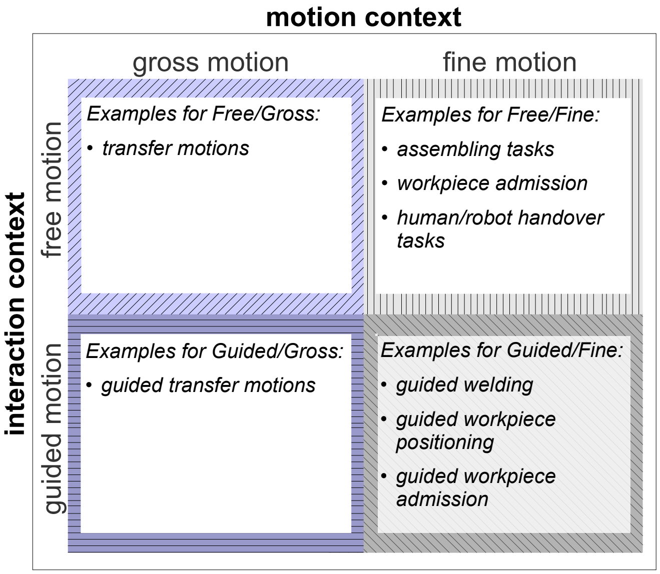

Fig. 1: The combination of the different contexts results in the four major states of cooperation. Additionally, some application examples are mentioned for each state.

into parts with no conflicting demands. It is necessary to explain the behavior within the individual parts, the so-called *cooperation states*, and how to switch from one to another part.

We describe our concept for human/robot coexistence and cooperation using a state transition diagram. The major cooperation states are identified and explained in Section II. The concrete behavior of the robot within each of the derived states is described in Section III (intrastate behavior). Two basic principles that are relevant to modeling safe and intuitive behavior are formulated. In Section IV, the necessary transitions between the states are pointed out (interstate behavior). Finally, in Section V we summarize and draw some conclusions.

II. States of cooperation

As mentioned, we present our concept using a state transition diagram. This section introduces its major states.

In order to solve the problem of conflicting demands on the robot's behavior we divide the overall behavior into parts containing no conflicting demands. Therefore, we define two criteria, motion context and interaction context. The *motion context* can be either fine or gross robot motion; the *interaction context* can be either free or guided robot motion. In the following the different contexts are explained.

During *free robot motions* the robot can move autonomously within all of its available degrees of freedom as long as the movement does not obstruct the achievement of objectives. For example, this can be a movement initiating a workpiece admission, during which contacts with the human operator must be avoided in any case.

With a *guided robot motion*, the temporal or spatial freedom of robot movement is determined by the human operator. This can be useful for example with guided welding, whereby the human operator defines the welding velocity and the robot controls the welding angles.

During *gross robot motions*, the distances to obstacles are larger than the sum of all system tolerances (robot positioning errors, system measurement errors, etc.). This includes for example transporting objects over large distances.

During *fine robot motions*, the distances to obstacles are similar to the tolerances of the system. Therefore, the tolerances must be considered explicitly so that a desired contact between workpiece and environment can be established. Examples of fine robot motion are assembly tasks, workpiece admission and the handing over of objects between humans and robot.

The combination of the different contexts results in four major states of cooperation in our state transition diagram. By introducing these major states, the conflicting demands mentioned are dissolved. In Fig. 1, some application examples are listed.

III. Behavior within the states

In the previous section, we identified the four major cooperation states. We will now describe two principles enabling us to formulate the concrete robot behavior within each state.

The most important precondition for human/robot coexistence and cooperation is safety of humans. In order to guarantee the safety, robot motions have to be supervised.

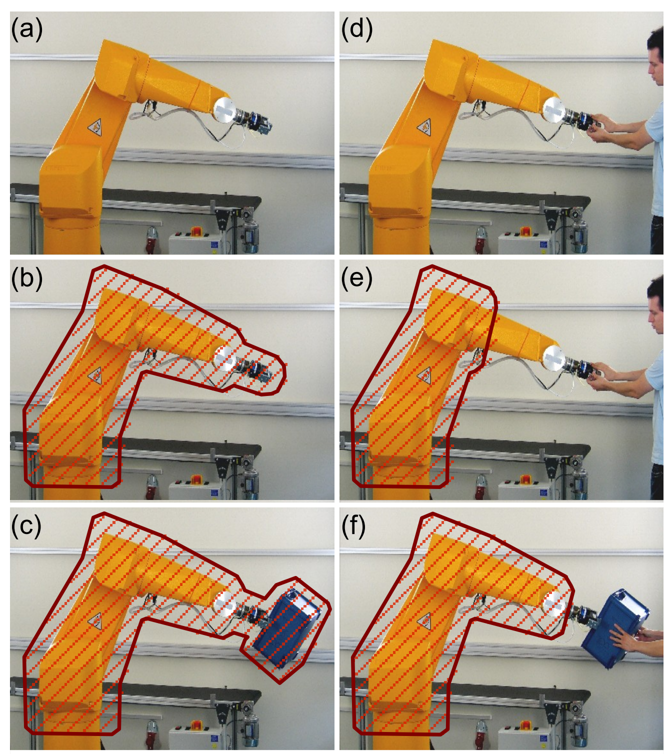

Fig. 2: Six photos illustrating the adaptation of surveillance area: The left column represents free motions and the right one illustrates guided motions. (a) and (d) show the robot with no surveilled area; (b) the surveilled area encompasses the entire robot; (c) the entire robot and the object being moved are enclosed by the surveillance area; (e) and (f) only the rear part of the robot is under surveillance, the human has to take care of the rest by himself.

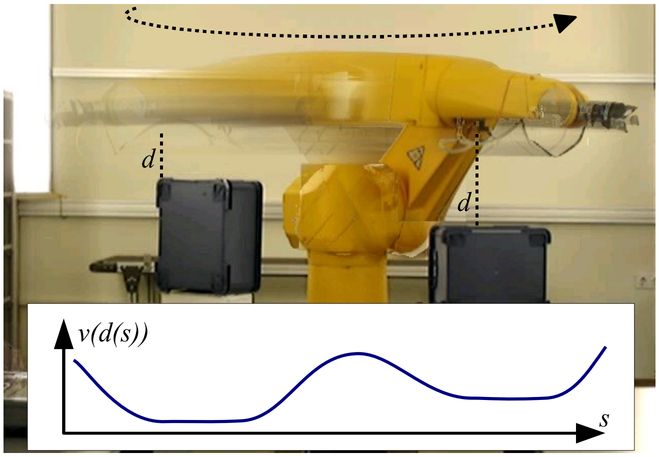

Fig. 3: Photo showing a robot movement and its associated velocity given by the principle 'distance-controlled velocity': The diagram shows the velocity *v(d(s))* for the covered distance *s* dependent on the distance *d(s)* between the surveilled area and the nearest obstacle. The first obstacle reduces velocity more than the second one.

The robot must not unintentionally get in contact with the environment, in particular with humans. This type of contact is called a *collision*, whose avoidance is of utmost priority. However, there are situations in which a contact with the environment or with the human operator is desired (e.g., if the human operator wants to control the robot via force following). Even in this situation, unintentional contacts between the occluded part of the robot, which is moved indirectly, and the environment must be avoided.

For this purpose, the principle of *adapted area of surveillance* is used, which adapts the supervised area depending upon the current requirements. Figure 2 illustrates this principle. The two pictures in the first row show the robot with no surveillance area, thus all safety is neglected with regard to collisions during robot motions. Images (b) and (c) show the robot along with all moved objects enclosed by the surveillance area. Such settings are useful for free gross robot motions. Pictures (e) and (f) show a trimmed area of surveillance – adjusted to the needs of guided robot motions via direct contact between human and robot. The forearm of the robot and the moved object are not supervised. The human operator has to take care of those parts.

As mentioned in Section I, safety is necessary but alone it is insufficient for substantial human/robot cooperation, as mutual understanding of the counterpart's behavior is a prerequisite for cooperation. Thus, we require intuitive behavior on the part of the robot to enable cooperation. A very simple but effective principle for intuitive behavior is *distance-controlled velocity*. Comparable to human behavior, it is desirable that the robot reduces its velocity when the surveilled area approaches an object or obstacle. The disruption of the human operator can thereby be reduced. Figure 3 illustrates this principle using a sample robot movement traveling from left to right in the picture.

The combination of both principles provides the basic requirements for safe cooperation between humans and robot.

Now, we can proceed to complete our state transition diagram (Fig. 4). The behavior of the robot based on the two principles is represented by the *distance/velocity function* in each state. The x-axis is labeled with the distance d and the y-axis with the maximum permissible velocity v_{max}. The velocity is controlled by the robot program in the case of a free robot motion while for guided robot motion, velocity is controlled by the human operator. In both cases the actual velocity is limited by the *distance/velocity function* $v_{max}(d)$.

During guided motions, additionally, the direction velocity of the robot movement is specified by the human operator. In the following, we think of this operator input as of an abstract guiding force. For example, the deflection of a joystick can provide the amount of guiding force and give its direction. More relevant for practice is the use of a force/torque sensor mounted on the robot's wrist, which measures the force applied by the human operator on the tool or on the workpiece gripped by the robot. Generally, we assume that both amount and direction of the operator's input force are measured online and serve as input for controlling the guided robot motion.

Since large movements should be performed with little effort during guided gross motions, zero-gravity robot behavior is advantageous, whereby the robot behaves like a damped inertial object in zero-gravity. In this case, we have motion generation where the acceleration of the robot movement depends on the force. This behavior is represented by the *force/acceleration function* within the state Guided/Gross.

In the state Guided/Fine, this behavior is less useful since it makes the execution of accurate movements more difficult. Here, movement generation with velocity directly dependent on force is more suitable. This behavior is represented by the *force/velocity function* within the state Guided/Fine. In both states, the actual velocity of the robot is

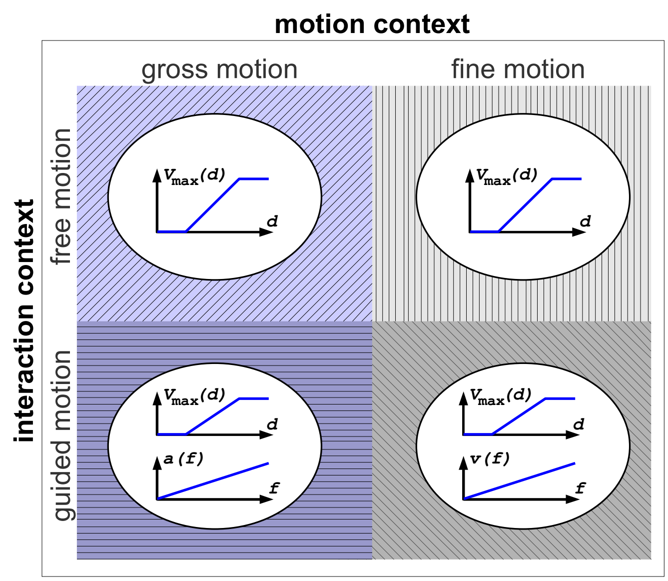

Fig. 4: The four major cooperation states with the modelled intuitive behavior. The models are based on the shortest distance d between the surveillance area and the nearest obstacle, the maximum allowed robot speed v_{max}, the guiding force f, the robot's acceleration a and speed v.

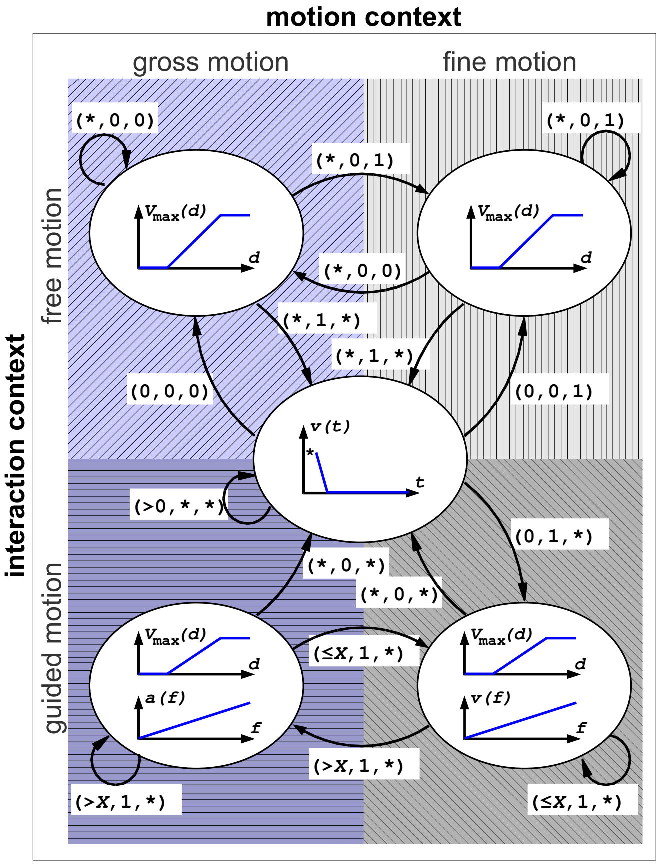

Fig. 5: The final state transition diagram including the four major states and an intermediate state as well as the transitions between the states. The tripel (*velocity of the robot*, *guided robot motion*, *program-controlled fine robot motion*) triggers the transitions between the states.

again limited by the *distance/velocity function*.

IV. Transitions between the states

In the preceding section, the concrete behavior of the robot was described within the individual states. This section treats the transitions between these cooperation states.

To enable the robot to distinguish between the four major cooperation states, three internal parameters are used, which are responsible for switching between these orthogonal contexts. One parameter indicates whether a free or a guided robot motion should be performed and the other two parameters determines whether a gross or a fine robot motion is to be performed.

In general, if a human performs a large free motion, for example transporting a bottle, he can do it more quickly than he could a fine task, such as threading a needle for example. Thus, velocity is a simple and natural way to detect whether a gross or fine motion is being performed. Since a guided robot motion is performed by a human operator, the robot velocity can be used directly as first parameter to switch between guided gross and guided fine robot motions. During free robot motion this parameter is inapplicable because the robot may also execute gross motions slowly. Therefore, it is more reasonable to let the the robot control program decide whether a fine or a gross motion is needed, by setting the second parameter.

In order to decide whether free or guided robot motion should be executed, we need a third parameter. Since this decision has to be done only by the human operator, he has to indicate his decision to the robot system. In practice this could be realized via a hardware or software button, a human/robot contact detection (e.g. via tremor detection [11]) or an other comfortable and intuitive way.

Transitions between free gross and free fine robot motion as well as transitions between guided gross and guided fine robot motion are relatively simple to handle because the origin of the control input does not change.

Transitions between free and guided robot motion are more difficult. For example, if a guided motion is performed, how should the current robot motion be dealt with if the human operator suddenly causes a switch to the free motions? This may be when using force following in combination with human-robot contact detection and the human accidentally looses the contact to the robot. On the other hand, if a free motion is performed and the human operator wants to guide the robot – how should the current robot motion controlled by the robot program switch to a robot motion controlled by the human operator? Thus, the question is how to handle velocity and the associated direction of a robot movement during a transition and which behavior appears intuitive to the human operator. An easy and reasonable way to achieve this is to perform the transitions only in combination with a stop of the robot.

Other solutions like synchronizing speed with the interacting human operator are conflicting with the safety requirements. In the case of using force following in order to control the robot in the guided interaction context, a synchronization would be impossible anyway since we follow the principle distance-controlled velocity. Weaking or modifying this principle would compromise safety. Another problem would be the switching between the areas of surveillance. Should the switching be done before or after the synchronization? If we would switch after synchronization in the case of using force following a collision with the human operator would be detected so we need to switch before synchronization can occur. The consequence would be that the robot would move without collision avoidance for the forearm while synchronizing with the human operator. This behavior would be inacceptable for safety reasons.

Now, we can complete the state transition diagram (Fig. 5). In order to bring the robot to a halt, it is necessary to introduce an intermediate state. This state is used whenever a transition between guided and free robot motion (and vice versa) is to be performed. The behavior within this state only serves to decelerate the robot as fast as possible starting from the current velocity v (represented in the diagram by '*').

Based on the above considerations, we are now able to specify the transitions between all reasonable cooperation states more formally using the three input parameters:

- v is the current robot motion velocity. It determines the transition between guided fine ($v \leq X$) and guided gross ($v > X$) motion. The threshold X needs to be measured experimentally. In the case of a free robot motion the robot program controls the velocity v, otherwise it is specified by the human operator. In both cases it is bounded by the *distance/velocity function*.

- M is the parameter indicating whether free (M=0) or guided ($M=1$) motion should be performed. The parameter M can be set in different ways, for example by pressing a button, or by human/robot contact detection.
- F indicates whether fine (F=1) or gross (F=0) motion is desirable. The parameter F only is relevant during free robot motion and it is explicitly set by the robot program.

In the state transition diagram, we use the triple (v, M, F) as the inputs. In the following, we describe all transitions starting from each state:

A. *Transitions starting from the state Free/Gross*

- (*,0,0): As long as there is no need for program-controlled fine robot motion and the human operator does not desire manual control of the robot, the robot remains in the state Free/Gross.
- (*,0,1): If program-controlled fine robot motion is required, the robot switches to the state Free/Fine.
- (*,1,*): If guided robot motion is desired, the robot switches to the intermediate state.
- All these transitions are independent of the velocity v, since the robot application program provides this velocity (limited to v_{max}) during free motion.

B. *Transitions starting from the state Free/Fine*

- (*,0,1): If program-controlled fine robot motion is required and the human operator does not want to control the robot manually, the robot remains in the state Free/Fine.
- (*,0,0): If program-controlled fine robot motion is no longer required, the robot switches to the state Free/Gross.
- (*,1,*): As with the state Free/Gross, the robot switches to the intermediate state when a guided robot motion is desired.
- Equivalent to the case of the state Free/Gross, the transitions are independent of the velocity v here too.

C. *Transitions starting from the state Guided/Gross*

- (>X,1,*): As long as the velocity is greater than a specific value X and there is still guided robot motion required, the robot remains in the state Guided/Gross.
- (≤X,1,*): If the velocity falls below the specific value X, the robot switches to the state Guided/Fine.
- (*,0,*): If manual control of the robot is not required anymore, the robot switches to the intermediate state.
- All transitions are independent of the variable F, since in guided cooperation states the user guidance dominates the robot application program.

D. *Transitions starting from the state Guided/Fine*

- (≤X,1,*): As long as the velocity is less than the specific value X and there is still guided robot motion required, the robot remains in the state Guided/Fine.
- (>X,1,*): If the velocity exceeds the specific value X, the robot switches to the state Guided/Gross.
- (*,0,*): If manual control of the robot is not required anymore, the robot switches to the intermediate state.
- As with the state Guided/Gross, all transitions here are independent of the variable F.

E. *Transitions starting from the intermediate state*

- (>0,*,*): Until the robot has stopped, there is no change to another state possible.
- If the robot has stopped and we do not want to perform a guided robot motion, the robot switches to a state in the free robot motion section, dependent on F. If program-controlled fine robot motion is activated (0,0,1), the robot switches to the state Free/Fine, otherwise (0,0,0) to the state Free/Gross.
- (0,1,*): If the robot has stopped and we want to perform a guided robot motion, then the robot switches to Guided/Fine.
- There is no transition from the intermediate state to the state Guided/Gross, since the velocity is zero and the robot would immediately switch to the state Guided/Fine.

Up until now, the guided fine and gross robot motions were regarded as two discrete states within the guided behavior part. The abrupt transition between those two states may lead to an undesirable behavior, because in case of guided fine robot motions the input force causes a change of the position of the robot and in case of guided gross robot motions it causes a change of velocity. In other words, when the system switches between the two guided motion states, this input is suddenly interpreted in a different way. Therefore, a smooth transition between the two states is favorable in practice, which can be expressed by a blending function depend-

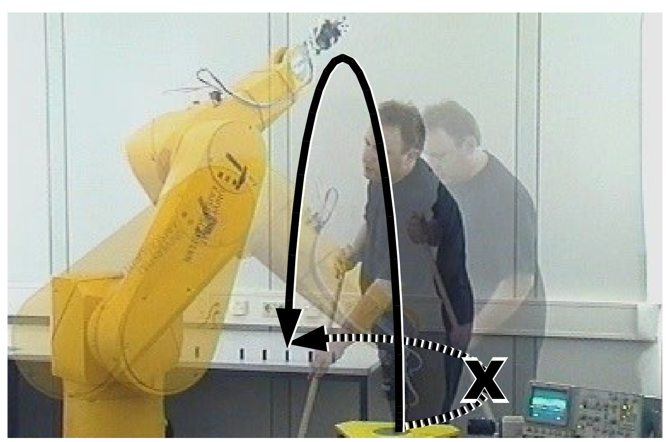

Fig. 6: Illustration of the already implemented free gross motion: Whole arm collision detection in conjunction with evasive movement based on the difference image method implemented by [4] (but without distance-controlled velocity).

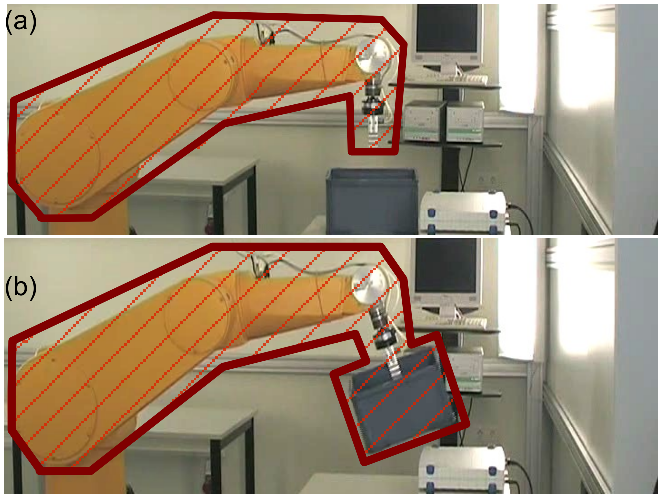

Fig. 7: Illustration of the implementation of free fine motion: Adaptable area of surveillance using the example of a workpiece admission (but still lacking distance-controlled velocity).

ing on the velocity [11]. Dependent on the velocity, the behavior for the gross robot motion is more weighted than the behavior for the fine robot motion and vice versa. Different shaped blending functions have to be evaluated in practice.

V. Conclusions

We presented a model for an intuitive and safe behavior which enables safe human/robot coexistence and in particular cooperation. For this purpose, all aspects of the robot behavior are divided into four parts with no conflicting demands regarding the robot behavior. The two main principles *adapted area of surveillance* and *distance-controlled velocity* provide a basis for a safe robot behavior (i.e., safe coexistence and cooperation between the human operator and the robot). Based on this, we defined the concrete robot behavior within each of the derived cooperation states as well as the transitions between the cooperation states.

Current research includes the implementation of a system based on the presented state transition diagram. Whole-arm collision detection with evasive movement based on the difference image method has already been implemented [4] and can be seen as a part of the category Free/Gross (Fig. 6). The human operator is detected by the difference image method so that an evasive movement can be planned. Only one area of surveillance is used in this specific case. An adaptable version has been presented recently [8] but does not use distance-controlled velocity (Fig. 7). The next step in our research will be the implementation of this feature, followed by intergration of the guided robot motions from [11].

Acknowledgements

This work has been supported by the German Research Foundation (DFG) under the project name „Sicherheitsstrategien für die Mensch/Roboter-Koexistenz und -Kooperation" (SIMERO).

References

[1] Bearveldt, A.J.: "A Safety System for Close Interaction between Man and Robot". Proceedings of IFAC Conference on Safety Security Reliability SAFECOMP `92, Zuerich, 1992.

[2] Bearveldt, A.J.: "Cooperation between Man and Robot: Interface and Safety". Proceedings of IEEE International Workshop on Robot and Human Communication, tokyo, Sept. 1992.

[3] Burghart, C., Yigit, S., Kerpa, O., Osswald, D., Woern, H.: "Concept for Human Robot Co-operation Integrating Artificial Haptic Perception". Proceedings of IAS-7, Marina del Rey, USA, 03. 2002, p. 38-45.

[4] Ebert, D.: "Bildbasierte Erzeugung kollisionsfreier Transferbewegungen für Industrieroboter". Dissertation am Fachbereich Informatik der Universität Kaiserslautern.

[5] Ebert, D., Henrich, D.: "Safe Human-Robot-Cooperation: Problem Analysis, System Concept and Fast Sensor Fusion". In: IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems, pp. 239-244, Baden-Baden, Germany, August 20-22, 2001.

[6] Dhillon, B. S.: "Robot Safety: A Challenging Issue for the 21st Century". Proceedings of the IASTED International Conference; Automation, Control, and Information Technology, June 10-13, 2002, Novosibirsk, Russia.

[7] Feddema J.T., Novak J.L.: "Whole Arm Obstacle Avoidance for Teleoperated Robots". In: IEEE Robotics and Automation Proceedings, pp.3303 – 3309, 1994.

[8] Gecks, T., Henrich, D.: "Human-robot cooperation: Safe Pick-and-Place Operations". Proceedings of 14th IEEE International Workshop on Robot and Human Interactive Communication (RO-MAN), 2005.

[9] Lumelsky V., Cheung E.: "Real-Time Collision Avoidance in Teleoperated Whole-Sensitive Robot Arm Manipulators". In: IEEE Transactions on Systems, Man and Cybernetics, Vol.23 No.1, pp.194-203,1993.

[10] Novak J.L., Feddema J.T.: "A Capacitance-Based Proximity Sensor for Whole Arm Obstacle Avoidance". In: IEEE Proceedings of the Intl. Conf. on Robotics and Automation, pp. 1307-1314, 1992.

[11] Stolka, Ph., Henrich, D.: "A Hybrid Force Following Controller For Multi-Scale Motions", Accepted for SYROCO 2003 - 7th International Symposium on Robot Control, Sept 1-3, 2003 – Wroclaw/Poland.

[12] Yu Y., Gupta K.: "Sensor-Based Roadmaps for Motion-Planning for Articulated Robots in Unknown Environment: Some Experiments with an Eye-in-hand System". In: IEEE International Conference on Intelligent Robots and Systems, pp.1707-1714, 1999.