Online Hazard Analysis for Autonomous Robots

Roger Woodman, Alan Winfield, Chris Harper, and Mike Fraser

Bristol Robotics Laboratory,
University of the West of England,
Bristol, England
roger.woodman@brl.ac.uk,alan.winfield@uwe.ac.uk,
cjharper@avian-technologies.co.uk,fraser@compsci.bristol.ac.uk
http://www.brl.ac.uk

Abstract. This paper presents a novel approach for designing robotic systems. The methodology aims to build on traditional functional hazard analysis, with the addition of processes aimed to improve the safety of autonomous personal robots. This will be achieved with the use of a safety protection system, developed during the hazard analysis stage. This protection system will serve dual purposes. Firstly, it will be used to verify that safety constraints, identified during hazard analysis, have been implemented appropriately. Secondly, it will serve as a high-level safety enforcer, by governing the actions of the robot, preventing the control system from performing unsafe operations. This research is particularly focused on the safety of human-robot interaction.

Keywords: robot safety, hazard analysis, safety protection system

1 Introduction

Designers of industrial and commercial robotic systems must consider a wide range of safety risks, for their users, the environment and the robot itself. Robotic systems therefore, as with all safety critical systems, require rigorous analysis at all parts of the design, to ensure the system is safe. As the first large consumer of robotic systems, the manufacturing industry has developed many of the robotic design methods that are used today. These methods were adapted from design principles and practices from other industrial sectors [9]. Incorporated into the design process were proven techniques such as hazard analysis, failure analysis, rigorous design and extensive inspection and testing. In addition to these, a number of safety standards for robotics have been developed; most notably ISO 10218-1 [12] and ANSI /RIA R15.06 [2]. As discussed by Nokata et al. [21] and Desantis et al. [5], the methods currently employed by robotic designers are not appropriate for designing safe robots operating in unstructured environments. This is due to the high complexity associated with a system that must adapt to changes in its environment and perform actions which cannot always be anticipated during development.

In this paper we present a novel approach for designing robotic systems. The described methodology shows how a safety protection system can be developed during the hazard analysis stage. This protection system will serve dual purposes. Initially it will be used to verify that safety constraints realised during hazard analysis, have been implemented appropriately and that no conflicts are present. Subsequently it will serve as a high-level safety enforcer, by governing the actions of the robot, preventing the control system from performing unsafe operations.

1.1 Hazard Analysis

Hazard analysis involves identifying and evaluating potential hazards in a system, which may cause or contribute to an unplanned or undesirable event. Hazards which are directly related to the system are known as functional hazards. Conversely, non-functional hazards relate to everything external to the system, such as the users or the environment. A number of hazard analysis techniques exist, many of which evaluates a system using a methodology appropriate for a particular industry. These techniques are generally considered as specialisations of one of the following [24]:

- Failure Modes and Effects Analysis (FMEA)
- HAZard and OPerability studies (HAZOP)
- Event Tree Analysis (ETA)
- Fault Tree Analysis (FTA)

All of these methods use a systematic approach for analysing hazards. The first step in any hazard analysis technique, according to Bahr [3], is to 'understand the physical and functional characteristics of the system under study'. This involves not only looking at the way the system functions, but also the interrelationship of all subsystems and how they may impact the system as a whole. This, as Bahr states, is often a problem area for engineers, who feel they understand how a system works. This can result in an underestimation of how operating conditions and environment can affect the system.

1.2 Safety of Autonomous Robotic Systems

Autonomous robots are a classification of robot system, which generally have one or more of the following properties: adapt to changes in the environment; plan for future events; learn new tasks; and make informed decisions. Although commercially available autonomous robots are still few, Goodrich and Schultz [7] report that there is an increasing demand for both personal robots for the home and service robots for industry.

At present, much of the research into robotic safety is looking at improving safety by either collision avoidance or failure prevention. Collision avoidance techniques, as the name suggests, aim to prevent robots from coming into contact with surrounding objects. It has been demonstrated that contact avoidance with humans, especially in cooperative situations, requires a higher level of perception compared to other static or dynamic entities [16] [8]. This has lead researchers to suggest that safety of human-robot interaction, requires both high-precision sensory information and fast reaction times, in order to work with and around humans [15] [6]. In addition to collision avoidance, strategies have been developed to integrate post-contact mitigation into the avoidance scheme. Work by Ikuta et al. [11] has shown that in robot development, while designing the control systes, it is important to consider safety implictions involved with moving external parts of the robot.

Among the requirements of autonomous robots, such as that being discussed in this research, is a certain degree of robustness. To achieve this it is important that the system should be able to support changes to its task specification [4]. These changes are necessary, as in a dynamic environment, the robot will frequently find itself in a wide range of previously unseen situations. To date, the majority of research in this area has addressed this issue by using learning algorithms, often implemented as neural networks [19] [17]. However, as Nehmzow et al. [20] identifies, these implementations, although seemingly effective, are difficult to analyse due to the inherent obscurity of connection based algorithms. This means that it is not possible to produce an intelligible model of the system structure, which could be used in safety analysis.

A report by Alami et al. [1], identifies a number of European robotic manufactures that have recently included software modules, to monitor, through external sensing, the space around the robot for any potential dangers. This type of additional monitoring system is traditionally known as a safety protection systems [24]. A practical example of this, for managing a high-powered laser, has been implemented by Wozniak et al. [26]. Their research found that an architecture, which separated safety from control, allowed them to more easily configure and extend the safety parameters, to meet the requirements of future changes.

2 Hazard Analysis Methodology Design

As discussed previously, hazard analysis involves assessing the system requirements, with the aim of identifying potential hazards associated with system operation. Before hazard analysis can take place, the system specification must be produced. This involves first outlining the customer requirements, which leads to task analysis. These complimentary processes result in a document specifying exactly what the system should do and how it will do it. From this document the functional requirements are identified. These requirements relate to the way in which tasks will be performed within the system, and the transformation from system input to system output. Once these processes are complete, hazard analyses can take place, although as with many development methodologies, requirements may be revised at any time.

Our hazard analysis methodology, seeks to bring the development of a safety protection system into the hazard analysis process. This we argue will allow verification that the safety schemes, identified during hazard analysis, have been implemented appropriately. This sentiment is supported by the work of Swarup and Ramaiah [25], who state that the most effective way to ensure a system will operate safely, is to build safety in from the start. The remainder of this paper details the design decisions involved in the development of the safety protection system and the strategy used to integrate it into the hazard analysis process.

2.1 Safety Constraints Verification

An important part of any system development, is to verify that the system implementation meets its specification [3]. Many diverse verification techniques have been established over the last half a century. The ones applicable to robot development are generally associated with mechanical construction, hardware electronics and software code. Mechanical construction will not be considered in this discussion, as we are only concerned with the safety implications inherent in the close interaction between hardware and software.

A popular method used for designing electronic circuits, involves modelling the circuits operations using a hardware description language (HDL). By modelling circuits in this way, designers can use simulation to perform rigorous tests, which can be used in both validation, to ensure the circuit operates how it was designed, and as verification that the design meets its specification [24]. Similar techniques are used for software verification; these include auto generating code, using modelling based techniques such as UML, and a variety of testing strategies. These testing strategies can range from, simply testing a few system inputs, or the boundary and extreme values to complete coverage testing, which means writing test cases that execute every decisional part of the code. The decision on what degree of testing is selected is generally based on the requirements of the safety standard being used [3].

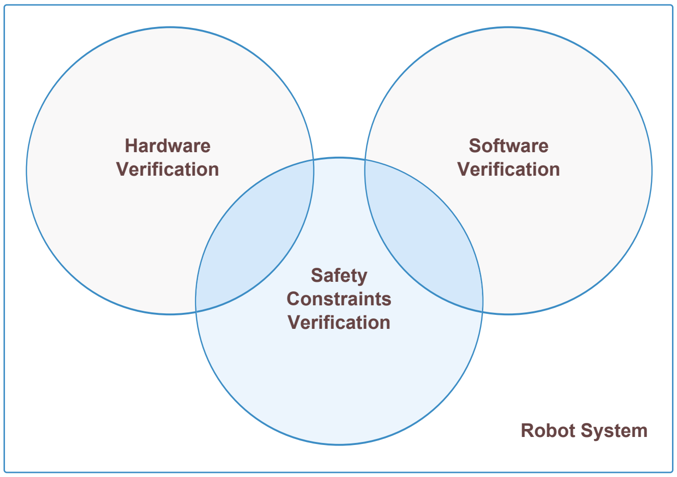

Fig. 1. Venn Diagram Illustrating the Areas of Verification in a Robot System.

The verification techniques discussed thus far are those generally used in the development of safety-critical systems, be it robotic or otherwise. These methods tend to analyse the systems mechanisms and not the behaviour. In this context, the mechanisms can be thought of as the system functions, whereas the behaviour is the ways in which the system functions interact during operation. A study by Swarup and Ramaiah [25], observed that analysis of system behaviour in order to identify violations in safety constraints, will become an increasingly important aspect of safety-critical system verification, as the complexity of systems continue to increase.

The diagram in Figure 1 illustrates how the traditional hardware and software verification methods are generally independent. Our safety constraints verification processes will analyse the system behaviour, with the aim of providing a way to verify safety functions that span the boundary between hardware and software. Examples of these types of safety function will be discussed in section 3.

2.2 Robot Task Example: Part Sorting

To test the viability of the proposed personal robot development techniques, we have devised a robot task, which can be executed with or without human interaction. The idea behind this is that experiments can be performed in an industrial type setting, which almost always requires complete isolation from humans. A human element can then be introduced to the experiment, without any major changes to the setup, allowing for better comparison of results.

Robot Sorting Task – Part 1: Isolation The robot sorting task involves retrieving parts from a collection area and sorting them into good or bad based on quality. Part 1 of the robot sorting task has the following requirements. The robot must select a part from the part dispenser and place it on either a good part or bad part conveyor. The robot should perform an on-board analysis of the part to determine its quality. The distance from the part dispenser to the conveyors is 3 meters. The robot must be completely autonomous and not tethered or fixed in anyway.

To model the task requirements, we used a scenario based technique, currently under development at the Bristol Robotics Laboratory (BRL). This allows us to produce a hierarchical task analysis diagram, as shown in Figure 2. The structure of the diagram is organised into vertical layers, called plans. Each plan further refines the task process, defined by the parent node. The sequence of nodes for each plan, dictate the logical steps by which each task should be performed.

Hierarchical task analysis, developed by Keith Duncan and John Annett in the late 1960s [23], has traditionally been used for time-motion studies. This involves analysing the way in which humans complete tasks, with the aim of improving efficiency. As at present there are no design methodologies available for personal robots, we have adopted this analysis technique. The diagram of

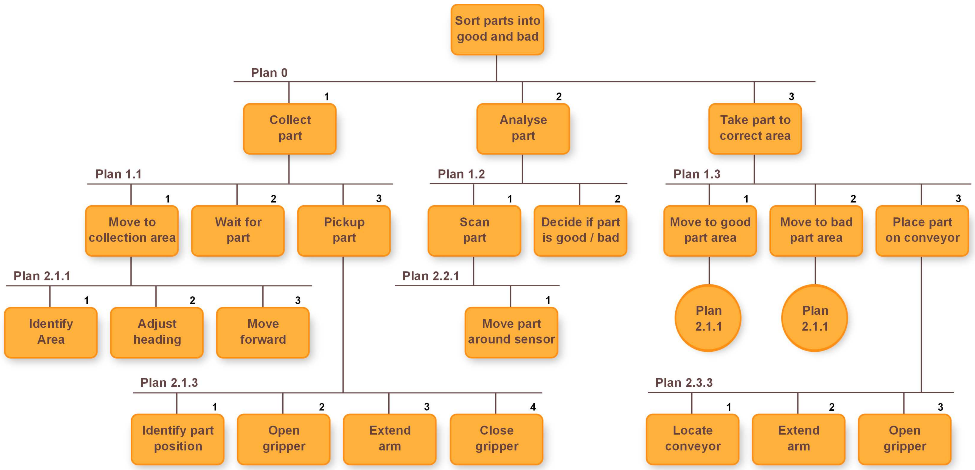

Fig. 2. Hierarchical Task Analysis Diagram of a Part Sorting Robot.

Figure 2 shows it is possible, in principal, to use this technique for modelling customer requirements in a clear and concise manner.

Robot Sorting Task – Part 2: Human-Robot Interaction Building on the requirements set-out in part 1, the task example part 2, requires the robot to operate within close proximity to a human user. The additional requirement is as follows. While the robot is approaching the part collection area, it must be able to identify the location of the human user and stop at a safe distance facing them. The user is then allowed to hold a part for inspection in front of the robot. The robot must negotiate with the human and fully take hold of the part. Finally it should examine the part and take it to the correct good/bad conveyor, all the time maintaining a safe operating speed and distance from the human.

The robot shall maintain a safe separation distance from the user, in accordance with the machinery safety standard ISO 13855 [13]. When in close proximity to the user, the robot shall operate at a reduced speed mode, in accordance with the robot safety standard ISO 10218-1 [12].

3 Designing Safety Policies

In the context of this research paper, a safety policy can be thought of as an interlock implemented in software. These software interlocks, or safety policies, aim to prevent the robot from causing unsafe actions, by means of intervention between the control system and the actuators.

We have chosen to present safety policies as independent rules, which use facts derived from perception data, to impose restrictions on a set of actuators.

This idea is based on principals taken from knowledge based system design [14]. The benefits of this type of design are the inherent parallelism, which treats all rules in the system as separate tasks, all of which are processed simultaneously. Other benefits include, modelling techniques, probabilistic reasoning and priority based inference [10].

The statement which follows reveals the generic structure for our safety policies. The safety policy object (SP) contains a number of variables, which are compared against to determine whether the associated actuators should be restricted or allowed to operate normally. Sensor functions (SF) provide high-level information about sensor readings, as well as a confidence level that quantifies how confident the sensor function is that its output value is correct. For example, the object distance sensor function could output a value of 300mm with a confidence level of 0.9, meaning that it is 90% confident that the nearest object is 300mm away. At present the safety policy required confidence level is chosen arbitrary, however, preliminary experiments show that it may be possible to use the risk rating identified during hazard analysis, at least as a starting value.

```
IF      robot_state = SP.required_state
AND     comparison of SF.value is inside SP.exceptable_bounds
AND     SF.confidence_level >= SP.required_confidence_level
THEN    allow actuators
ELSE    restrict actuators
FINALLY return safety_rating based on SF.confidence_level
```

It is often argued that context awareness in a safety-critical system is a crucial way of maintaining both the availability of the system to complete tasks and its safety [6] [22]. This has been implemented in our rule based approach with the use of a robot state identifier.

An example safety policy is shown below. This policy is based on a requirement from the robot standards guide ISO 10218-1 [12]. The requirement states that while operating at reduced speed, the speed of the robot is limited to 250 mm/s.

```
IF      robot_state = reduced_speed_control
AND     end_effector_speed <= 250mm/s
AND     confidence_level >= 0.95
THEN    allow actuators
ELSE    restrict actuators
FINALLY return safety_rating based on confidence_level
```

The following safety policy example, based on an ISO 13855 requirement [13], is used to maintain a separation distance between the robot and a human user. This example and the one that proceeded, show that it is possible to explicitly represent requirements taken directly from safety standards. This opens up the possibility that safety standards could not only be used as requirement guidelines, but also as specifications for actual safety constraint implementation. However, it must be noted that currently robotic safety standards rarely specify requirements in terms of quantitative values, and instead give general guidelines on the qualities that the final system must possess. It may be that in the future we see a shift in robotic safety standards, from the current safety guidelines, to more prescriptive requirements.

```
IF       robot_state = collaborative_operation_mode
AND      human_distance >= 100mm
AND      robot_speed x robot_stopping_time +
         protective_device_minimum_distance < human_distance
AND      confidence_level >= 0.95
THEN     allow actuators
ELSE     restrict actuators
FINALLY  return safety_rating based on confidence_level
```

When a safety policy is executed, it either allows its set of actuators to operate normally or imposes restrictions. These restrictions can be in the form of limitations on potential output or as suppression, preventing the actuators from operating. In both cases a safety rating is produced, which can be used by the control system to understand the nature of the restrictions, or if none have been imposed, this value is based on the substitution of the sensor function output and the corresponding safety policy comparison values.

4 Safety Protection System Design Principles

As has already been discussed, the safety protection system we are developing is a high-level real-time safety monitor, which can intervene to restrict the control system from activating an actuator in such a way that it could lead to an unsafe event. The design of the protection system is built on the notion of safety policies (see section 3), acting as rules in a type of knowledge based system. The decision was made to represent the system as a traditional graph data structure, which would allow us to make use of tried and tested analysis techniques.

An example of our safety protection system is shown in Figure 3. The model is organised as a series of layers, each layer is composed of nodes of a single type. The top layer represents the robot sensors (S), each of which can be connected to one or more sensor functions (SF). These sensor functions interpret the sensor data and output higher-level information, for example the position of an object. The output of the sensor functions is given as a value (V) and a confidence level (CL). As discussed in section 3, the confidence level is a probabilistic value based on how confident the sensor function is that its output value is correct. The sensor function information is passed to one or more nodes in the safety policy (SP) layer. These safety policies are implemented using the safety policy rule format described in section 3. Finally, the safety policies are connected to one or more actuator (A) nodes, each of which represents a single motion that an actuator is capable of. For example, a drive motor, which is able to operate in forward and reverse, would be modelled as two actuator nodes. This has been done to increase the liveliness of the system, so safety policies only intervene on actions that would put the robot into an unsafe state. An example of this is to

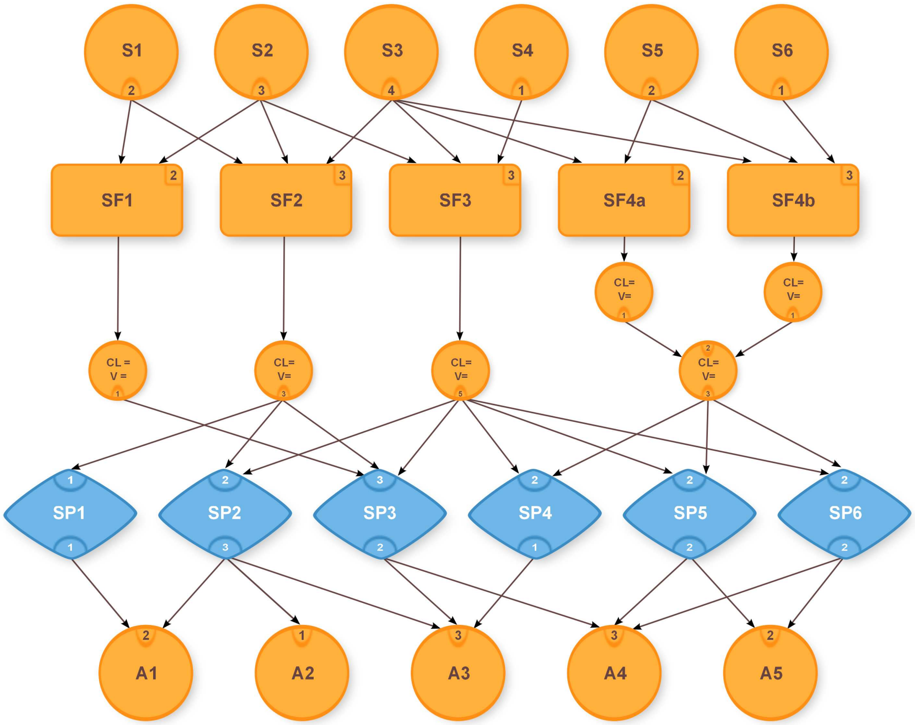

Fig. 3. Example Safety Protection Model.

allow the robot to move away from danger, but prevent it from moving towards it.

From its inception, we made certain to design the safety protection system with all the fundamental requirements that our research had shown a safety system would need. The following sections describe the core design principles that we followed in developing the safety protection system.

4.1 Aid the Hazard Analysis Process

The protection system aids the hazard analysis in two main areas. Firstly by having a collection of safety policies all of the same generic type, it is simple to iterate though them and verify that they all exists as the specification dictated. The second area, and one which partly justifies the creation of a separate safety system, is the metrics that can be taken and used in a quantitate assessment of the system dependencies. In the example model of Figure 3, it can be seen that sensor 4 has one connection to sensor function 3, which in turn connects to five of the six safety policies, which finally connect to all the actuator nodes. This means that if sensor 4 failed, the dependent policies could not be evaluated and the system would halt. This shows how areas of the system which may require added redundancy can be identified. This concept will be explored further in the next section.

The ability to analyse the interdependences within a system is an interesting area of study, not just for aiding the hazard analysis process, but also for identifying areas of high and low activity within the system. This kind of information could be utilised to direct improvement efforts into those parts of the system, which are most used and potentially identify those parts which are not required. This bares some similarity to the work of Nehmzow et al. [20], who demonstrated that it was possible to take an existing robot system, and create a model of that system by learning how it functioned. With this model they were able to show how some of the least important functions could be removed, with little to no effect to the system operation.

4.2 Redundancy

A major concern for robot system designers are sensor malfunctions and failures [25]. Therefore we wanted to address this issue as seamlessly as possible in our safety model. As the example in Figure 3 shows, we have addressed this problem with the use of multiple sensor functions of the same type, SF4a and SF4b. An example of their use could be for monitoring for the presence of a human. These sensor functions could be either identical, using the same sensors and code or diverse, utilising different sensors and processing the data in a different way.

We are currently investigating how best to use the output from multiple sensor functions of the same type. The two main approaches we have identified, involve either using the value from the sensor function with the highest confidence level, or combining the output in some way. Initial trials have suggested that combining output using the confidence levels as a weighting to give more significance to values with higher confidence, is way to both reduce error while preserving the data diversity obtained from multiple sources.

4.3 Flexibility

At the core of the safety protection system are the safety policies. These policies are treated independently and therefore can be amended, added or removed at any time. The design of the safety policies, are such that they are not tied to any specific hardware. This gives designers more flexibility on the construction of the robot, and opens up the possibility that identical safety policies could be deployed on different types of robot.

4.4 Probabilistic Reasoning

Robot sensors and actuators are inherently prone to error, this means that any reasoning about their use, either as data received or actions taken, must incorporate a degree of uncertainty. One of the recent trends in robotics research, for handling uncertain information about the robots environment, is to assign a danger index to objects that the robot perceives could cause a hazard [21] [11] [15]. These indices are continuously updated with the latest sensor readings, with the aim of increasing the robots confidence in its own understanding. As the robots confidence increases, so does the reliance on that knowledge, allowing the robot to continue to operate in the presence of potential hazards.

The amount of processing required to produce an accurate view of the world, necessary for navigating safely, is potentially very large. Some have suggested using probabilistic graphical models in the form of Bayesian networks for handling this complex data set [18] [22]. This involves using Bayes rule to combine all the robot data to produce a reasoned output. The Bayesian method is generally used for statistical analysis of data, which is known to be relatively accurate.

In our safety protection system, we have chosen to use confidence levels (sometimes referred to as confidence factors), which is a method often used in expert systems for dealing with uncertainty [10] [14]. This differs from Bayesian networks as it does not require a priori probability to be assigned to each decisional part of the network. Instead it allows us to assign a value of belief to sensor readings, which can be combined to give an overall confidence level for use in the safety policies.

5 Conclusion

In this paper we have presented a number of development techniques, which aim to improve the safety of personal robots. The approach we have taken has been based on a study of existing techniques, which found that those currently used in robotic development, are not appropriate for designing safe personal robots. This lead us to suggest a process where a high-level safety system is developed during the hazard analysis stage. As it has been shown, this safety protection system is used to both verify the implementation of the safety requirements and to act as a real-time safety monitor, preventing the control system from performing unsafe actions.

To test the viability of the proposed methodology, we have outlined a robotic task, which allows for experiments to be conducted in two stages. The first stage involves a complex robot task, with the robot working in isolation. The second stage introduces a human element and removes inhibitions preventing the robot operating in the presence of a human.

Further experimentation is required in order to determine the effectiveness of the proposed safety techniques. However, preliminary experiments have shown the potential benefits of organising safety constraints as a set of rules which can be easily modelled and modified.

References

1. Alami, R., Albu-Schaeffer, A., Bicchi, A., Bischoff, R., Chatila, R., Luca, A.D., Santis, A.D., Giralt, G., Guiochet, J., Hirzinger, G., Ingrand, F., Lippiello, V., Mattone, R.: Safe and dependable physical human-robot interaction in anthropic domains: State of the art and challenges. Procceedings of IROS'06 (2006)

2. ANSI/RIA-R15.06-1999(R2009): Industrial Robots and Robot Systems - Safety Requirements. American National Standards Institute, New York, USA, (2009)
3. Bahr, N.J.: System Safety Engineering and Risk Assessment: A Practical Approach. Taylor & Francis, Washington, DC, USA, (1997)
4. Bonasso, P., Kortenkamp, D.: Using a layered control architecture to alleviate planning with incomplete information. Proceedings of AAAI pp. 1–4 (1996)
5. Desantis, A., Siciliano, B., Deluca, A., Bicchi, A.: An atlas of physical human robot interaction. Mechanism and Machine Theory pp. 253–270 (2008)
6. Giuliani, M., Lenz, C., Mller, T., Rickert, M., Knoll, A.: Design principles for safety in human-robot interaction. Journal of Social Robotics pp. 253–274 (2010)
7. Goodrich, M.A., Schultz, A.C.: Human-robot interaction: A survey. Foundations and Trends in Human-Computer Interaction pp. 203–275 (2007)
8. Heinzmann, J., Zelinsky, A.: Quantitative safety guarantees for physical human-robot interaction. International Journal of Robotic Research pp. 479–504 (2003)
9. Hgele, M., Nilsson, K., Pires, N.J.: Springer Handbook of Robotics: Industrial Robotics. Springer, Heidelberg, (2008)
10. Hopgood, A.A.: Intelligent systems for engineers and scientists. CRC Press, Florida, USA, (2001)
11. Ikuta, K., Ishii, H., Makoto, N.: Safety evaluation method of design and control for human-care robots. International Journal of Robotic Research pp. 281–298 (2003)
12. ISO-10218-1: Robots for Industrial Environments - Safety Requirements - Part 1: Robot. ISO, Geneva, (2006)
13. ISO-13855: Safety of machinery - Positioning of safeguards with respect to the approach speeds of parts of the human body. ISO, Geneva, (2010)
14. Kendal, S., Creen, M.: An Introduction to Knowledge Engineering. Springer, London, UK, (2007)
15. Kuli, D., Croft, E.: Pre-collision safety strategies for human-robot interaction. Autonomous Robots pp. 149–164 (2007)
16. Kuli, D., Elizabeth, C.: Strategies for safety in human-robot interaction. Proceedings of IEEE International Conference on Advanced Robotics pp. 810–815 (2003)
17. Larsen, T., Hansen, S.: Evolving composite robot behaviour - a modular architecture. Proceedings of RoMoCo'05 pp. 271–276 (2005)
18. Marzwell, N.I., Tso, K.S., Hecht, M.: An integrated fault tolerant robotic control system for high reliability and safety. Proceedings of Technology 2004 (1994)
19. Nehmzow, U.: Flexible control of mobile robots through autonomous competence acquisition. Measurement and Control pp. 48–54 (1995)
20. Nehmzow, U., Kyriacou, T., Iglesias, R., Billings, S.: Robotmodic: Modelling, identification and characterisation of mobile robots. Proceedings of TAROS (2004)
21. Nokata, M., Ikuta, K., Ishii, H.: Safety-optimizing method of human-care robot design and control. Proceedings of IEEE Robotics and Automation (2002)
22. Pipe, A.G., Vaidyanathan, R., Melhuish, C., Bremner, P., Robinson, P., Clark, R., Lenz, A., Eder, K., Hawes, N., Ghahramani, Z., Fraser, M., Mermehdi, M., Healey, P., Skachek, S.: Affective Robotics: Human Motion and Behavioural Inspiration for Cooperation between Humans and Assistive Robots. Taylor & Francis (2011)
23. Shepherd, A.: Hierarchial Task Analysis. Taylor & Francis, London, UK, (2001)
24. Storey, N.R.: Safety Critical Computer Systems. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, (1996)
25. Swarup, B.M., Ramaiah, S.P.: An approach to modeling software safety in safety-critical systems. Journal of Computer Science pp. 311–322 (2009)
26. Wozniak, J., Baggiolini, V., Garcia, D.Q., Wenninger, J.: Software interlocks system. Proceedings of ICALEPCS07 pp. 403–405 (2007)