Haptic Control for the Interactive Behavior Operated Shopping Trolley InBOT

Michael Göller and **Thilo Kerscher** and **Marco Ziegenmeyer** and **Arne Rönnau** and **J.Marius Zöllner** and **Rüdiger Dillmann** [1]

Abstract. An important challenge in service robotics is to design user interfaces that can be used intuitively to enable the potential users to benefit from the robot's functionalities. These interfaces have to be easy to understand on the one hand but have to be able to access the various features that are provided by robotics. This document presents the haptic-based control of an Interactive Behavior Operated shopping Trolley (InBOT). The holonomic robot InBOT is an approach to transfer state of the art robotics technology into all-day environments to make their benefits available for the populace. The haptic-based mode of control enables the user to steer InBOT just like an ordinary shopping cart while being released from the burden to push the cart with muscle power. Additionally haptic commands are introduced which the user can give the robot using the haptic handle. These commands are classified by a vector support machine and activate corresponding behaviors of the robot's advanced behavior repertoire. Finally the user is supported continuously by features from the robot's basic behavior repertoire like the avoidance of obstacles.

1 Introduction

The Interactive Behavior Operated Trolley (InBOT, fig.1) is an approach to transfer state of the art robotics technology into human everyday environments. The benefits of the actual technology shall become available for the populace. InBOT addresses several everyday problems and its main idea is to ease the daily shopping. Among other possibilities this means to help the customer to find the desired products without extensive search in big supermarkets or to relieve the customer from the burden to push the shopping cart from his own force all the time, especially if the cart is heavily loaded or the customer is an elderly person.

InBOT provides four different modes of operation. Their interaction with the remaining system is roughly sketched in fig.2. The first one is the *haptic steering mode*. This is the mode with the closest coupling between user and robot. Here InBOT is pushed like an ordinary shopping cart but supports the user with several features like the full motor power of the robot, an obstacle avoidance assistant or local haptic commands. The transition into this mode is special. While for the other modes an explicit command of the user is necessary InBOT automatically switches to the *haptic steering mode* as soon as a force is applied to the handle. The second one is the *following mode*. Here the robot follows the user wherever he goes in a well-defined distance. The third one is the *guiding mode*. In this mode the robot guides the user to a desired target or product or even along a whole shopping list step by step while continuously adapting to the user's movements. The last mode is the *autonomous mode* with the weakest coupling between user and robot. Here InBOT acts only if explicitly commanded by the user and once commanded the robot performs the task independently. As baseline of all these modes acts a *behavior-based control* ([1]) that provides a repertoire of behaviors to be used in the course of the four modes. Among the provided behaviors is for example the capability to localize itself and to find a way to a desired product or the avoidance of obstacles or dynamic objects. This document describes the haptic control of InBOT which is the most crucial part of the *haptic steering mode*.

Figure 1. The Interactive Behavior Operated shopping Trolley (InBOT). It is equipped with a mecanum drive for full holonomic movements and a haptic handle to enable the user to steer it just like an ordinary trolley.

The most intuitive method of using the robot trolley is probably the interaction based on physical contact by using a haptic handle (fig.3). The user doesn't need extensive training and doesn't have to learn voice or gesture commands. Additionally this kind of interaction has a high degree of reliability. The close coupling of user and robot in the control task allows the user to perceive the robot's actions while allowing him to contribute his own intentions simultaneously. Thus he preserves a feeling of control and responsibility ([3]). Due to these advantages the *haptic steering mode* might be the most important mode of controlling InBOT. The user is able to apply ex-

[1] FZI Research Center for Information Technology, Karlsruhe, Germany, email: [goeller, kerscher, ziegenmeyer, roennau, zoellner, dillmann]@fzi.de

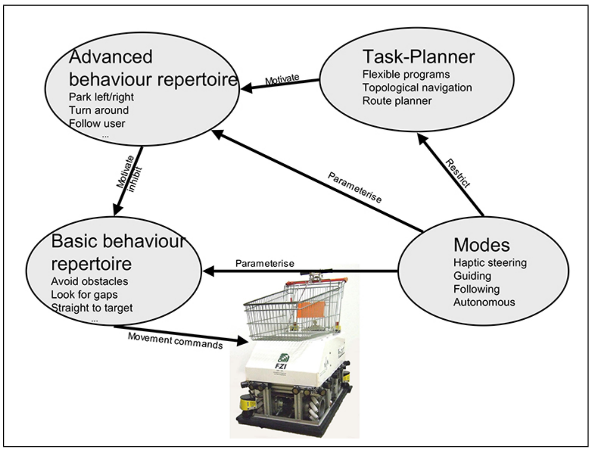

Figure 2. The abilities of InBOT are grouped in 2 repertoires of behaviors with a flexible task-planner on top. The different modes of operation influence the work of the different behaviors and of the task planner([2]).

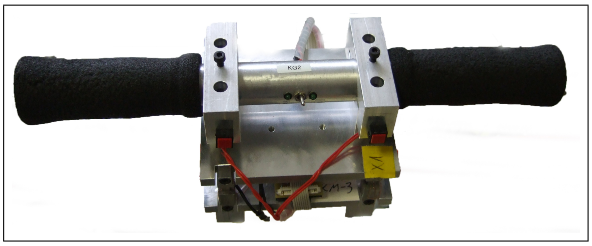

Figure 3. Close-up of the haptic handle with the two red dead man's switches.

periences from a common, manual trolley to the robot trolley but is supported by several additional functionalities like motor power, obstacle avoidance or local maneuvers controlled by haptic commands.

2 Control chain for the haptic handle

To achieve a high level of usability the trolley's handle must be able to detect the applied forces and torques, gather them and transfer them to a classification mechanism that is based on a *support vector machine*(SVM). After the classification motion commands are generated and transferred to the behavior network ([1]) implementing the *basic behavior repertoire*. After the movement commands have been merged with the commands from other behaviors the final movement command is handed down to the motion control in the hardware abstraction layer.

In detail (see fig.4) this means the following: The forces that the user exerts on the bars of the haptic handle are measured by strain gauges. The values are digitalized and transferred to a UCOM-board ([4]). Here the several measured forces are transformed into two forces and one torque, which are the most significant information used in the haptic mode. This procedure of data acquisition is described in detail in the next section. Additional signals are provided by two buttons on the handle that are used as dead man's switches. The UCOM-board transfers the processed forces to the navigation system only if at least one of the buttons is pressed. The second functionality of the buttons is to trigger the classification of haptic commands. At the moment when both buttons are released the re-

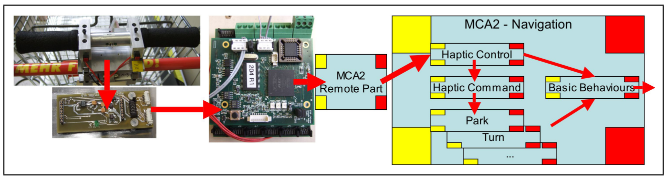

Figure 4. Control chain of the haptic mode: Haptic handle - ACAM-Board - UCOM-Board - MCA2 Remote part - Module Haptic Control - Module Haptic Command Classification (based on SVM) - Advanced Behaviors - Basic Behaviors

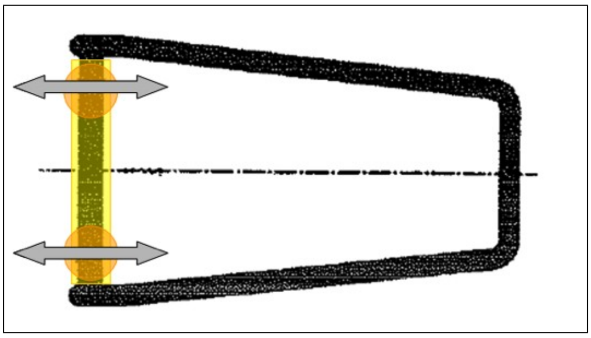

Figure 5. Typical forces applied by human users.

sponsible *support vector machine* processes the forces measured in the last second and classifies the given command (see section 2.3 for a detailed description). Each identified command motivates one of the behaviors of the *advanced behavior repertoire*. While at least one of the buttons is pressed the actual command is classified as *haptic steering* and the processed forces are transformed into movement commands (section 2.2). Based upon the classification and the calculated information the different behaviors of the behavior repertoires can be activated.

2.1 Haptic-based detection of commands

To sense and detect the user-intention using the trolley's handle a physical contact of user and trolley is necessary. The user grabs the handle at least with one hand. The possible area for the physical contact of the user is illustrated in fig.5 by the yellow rectangle. The most typical areas for the contact are illustrated by the orange circles. The arrows demonstrate the most typical direction of the forces applied by the user.

The user is able to apply forces as well as torques to the trolley's handle one or both hands. During contact of both hands the rigid body of the handle will combine the different forces and torques to a resulting force and torque (see fig.6). For the steering, but also for the classification of the intention, only the projection of the force on the horizontal plan (fig.7) as well as the torque around an axis parallel to the normal vector of the horizontal plan is used.

The sensorized trolley handle must detect three values which are the absolute value of the force, the direction of the force, and the torque around the normal vector of the horizontal plane. The two values for the force should be orthogonal.

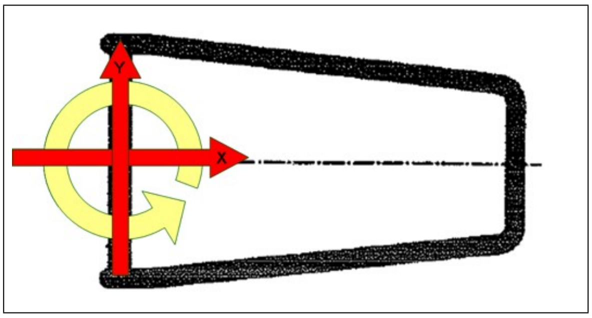

Figure 6. Forces (red arrows) and torque (yellow circle) that are finally provided by the haptic handle.

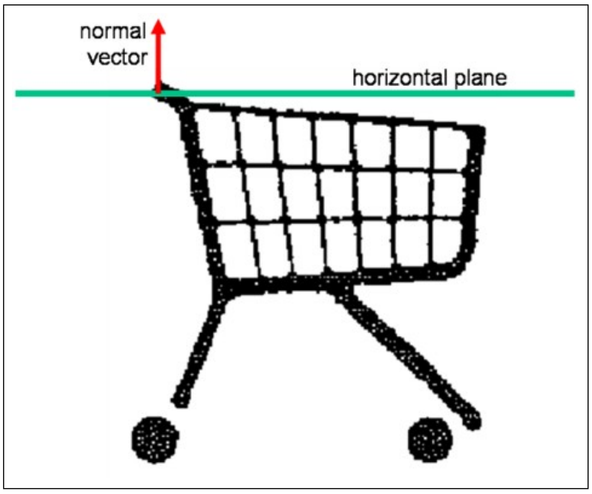

Figure 7. Horizontal plane and normal vector for the force and torque direction

2.1.1 *Mechanical design of the haptic handle*

The current setup of the haptic handle focuses on the elementary requirements which have been described above. This setup can be used as a haptic handle for the robot trolley as well as a sensor of a common trolley to gather sensor information for example for the user-centered prototyping evaluation. The setup of the haptic handle is shown in fig.3.

The main part which hosts the force and torque sensor consists of two planes, which are connected by four strain gauges beams. One plane is directly connected to the trolley. The second plane carries the handle, which will be used by the user to steer the trolley. Due to this setup the flow of force must go from the handle through the second plane, the strain gauges beams and finally the first plane to the trolley.

The strain gauges for the detection of the forces and the torques are placed on three of the four strain gauges beams. Figure 8 illustrates the positions of the different strain gauges and the force direction they sense. The beams B1 and B2 are equipped with strain gauges to sense the forces in the x-direction. The beams B1 and B3 are equipped with strain gauges to sense the forces in the y-direction. The number of four strain gauges is sufficient because in general three forces (with independent force directions) will be enough to calculate the forces and the torque of the haptic handle.

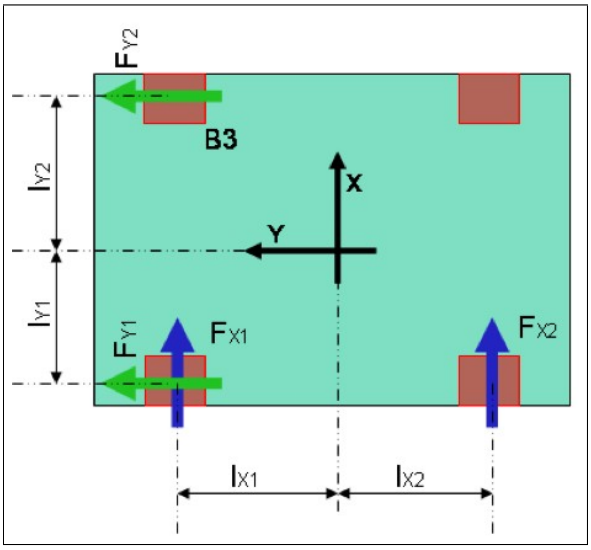

Figure 8. Base of the haptic handle with indicated resulting force directions (black arrows). The green arrows indicate the two different forces measured in Y-direction, the two blue arrows indicate the forces measured in X-direction.

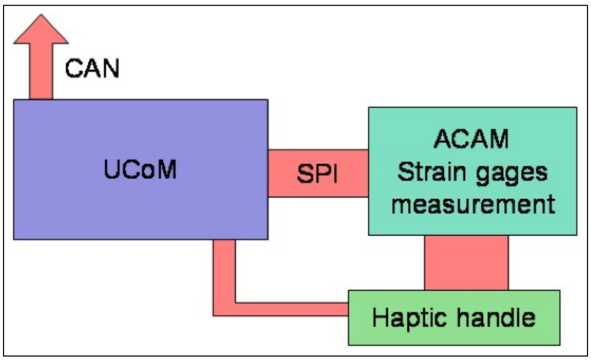

Figure 9. Schematic diagram on the electronic components of the haptic handle

2.1.2 *Electronic components of the haptic handle*

The electronic components of the haptic handle are responsible for the measurement of the strain gauges, the measurement of the human contact, the pre-processing of the gathered sensor information as well as the communication with the overall system. Figure 9 illustrates the used components and their connections.

For the measurement of the strain gauges the specialized component "PS021 - Digital Amplifier for Strain Gauges" from acam-messelectronic GmbH (www.acam.de) is used. The main advantage of this component is that the integrated circuits are realized without any analogue component but an approved Time-to-Digital Converter (TDC) technology. This technology has a lot of advantages like there is no need for a full-bridge, 2 resistors (half-bridge) are sufficient or no reference voltage is needed. The PS021is connected to the UCoM over SPI-BUS.

The major part of the components is the Universal Controller Module (UCoM), which is responsible for the gathering as well for the pre-processing of the sensor information. This board has been developed at the FZI as a modular hardware component for example for lower level control tasks (motor control) as well as sensor preprocessing. A detailed description of the UCoM can be found in [4]. The UCoM is encapsulated in a network-transparent remote part using the MCA2 framework ([8], [9]) and communicates with the overall system over CAN-BUS. The UCoM is directly connected to the sensors for the contact of the user.

2.1.3 Calculation of the applied forces and torques at the haptic handle

The major part of the software component is the calculation of the applied forces and torques at the haptic handle. The positions of the measured forces are illustrated in fig.8. The different measured forces are not independent enough to use them as raw data for the calculation. This is caused by manufacturing tolerances, varying material properties and the fact that the manual placed strain gauges are not total identically positioned. So a calibration of the total assembled system is necessary. The calibration is done by linear calibration using a 4x4 calibration matrix **A** which transfers the raw sensor values $\vec{RS} = (F_{XR1}F_{XR2}F_{YR1}F_{YR2})^T$ to the calibrated ones $\vec{CS} = (F_{X1}F_{X2}F_{Y1}F_{Y2})^T$ using the relation:

$$\vec{CS} = \mathbf{A} \cdot \vec{RS} \tag{1}$$

The calibration matrix transfers the four raw values to more independent values. So afterwards the force values can be treated as independent. The calibration matrix has been determined by applying the least square method to calculate the necessary elements of the calibration matrix. For this purpose a defined set of independent test forces has been applied on the haptic handle.

Now, the resulting force and torques can be calculated with the following equations:

$$F_X = F_{X1} + F_{X2} \tag{2}$$

$$F_Y = F_{Y1} + F_{Y2} \tag{3}$$

$$T = -F_{X1} \cdot l_{X1} + F_{X2} \cdot l_{X2} - F_{Y1} \cdot l_{Y1} + F_{Y2} \cdot l_{Y2} \tag{4}$$

2.2 Generation of haptic-based movement

If the module responsible for the haptic control receives forces from the UCOM-board it demands a mode change to the haptic mode at once. This request is processed in a higher layer. If operating in the haptic mode the module transforms the measured forces into a velocity vector using a dynamic approach. Formula 5 describes how the set point vector for the velocity (v_t) is calculated. It depends on the past velocity, the proportion of the measured force (F) compared to the maximum accepted force (F_{max}), an amplification factor (f_a) and a damping factor (f_d). The state variable v represents the set point vector for the desired velocity. It is amplified by the measured forces and damped by its actual value. This velocity vector is then passed to the behavior-based control. Here it is merged with velocity vectors of other behaviors like *obstacle avoidance* and finally passed to the hardware abstraction layer. Here the resulting desired velocities are transformed in engine speed values for the mecanum drive.

$$v_t = v_{t-1} + min\left(\frac{|F|}{F_{max}}, 1\right) \cdot sign(F) \cdot f_a \cdot \Delta t - f_d \cdot v_{t-1} \cdot \Delta t \tag{5}$$

2.3 SVM-based classification of haptic commands

Another alternative to generating velocity vectors out of the measured forces directly is to analyze the measured forces over a period of time and to try to identify haptic commands. If identified these commands overrule the direct generation of velocity vectors. An identified command activates the corresponding behaviors from the *advanced behavior repertoire*. The activated advanced behaviors are responsible for the generation of the velocity vectors themselves. Usually they again activate behaviors from the *basic behavior repertoire* (see fig.4) .

The discrimination between these two approaches is realized by using the haptic handle's contact sensors. If a user has contact with the haptic handle the first approach is active. If the user releases the handle the past temporal progression of the applied forces and torques is used to classify the different intentions of the user.

2.3.1 Classes of haptic commands

The definition of the classes is motivated by the different user-intentions which should be distinguisted by the classificatory. Each of this intentions is connected to a more haptic handle specific description, which is the basis of the command classes. The following intentions of the user are supported: ”Wait exactly here”, ”Stay near the actual position”, ”Rotate to support loading”, ”Park at side”, ”Drive a little forward”.

- ”Steer”
  - Continuous contact between user and handle
  - → Class name: STEER
- ”Wait exactly here”
  - Handle has been released
  - During release a high force is applied on the handle in backward direction
  - → Class name: PULL
- ”Stay near the actual position”
  - Handle has been released
  - At time of release no force and torque has been applied on the handle
  - → Class name: WAIT
- ”Rotate to support loading”
  - Handle has been released
  - During release a high torque is applied on the handle
  - → Class name: ROTATE LEFT
  - → Class name: ROTATE RIGHT
- ”Park at side”
  - Handle has been released
  - During release a high force is applied on the handle
  - → Class name: PUSH LEFT
  - → Class name: PUSH LEFT
- ”Drive a little forward”
  - Handle has been released
  - During release a high force is applied on the handle
  - → Class name: PUSH

The haptic commands which result from the classification of the user intention are used to command the trolley. The resulting classes of commands are summarized in table 1.

Table 1. Supported classes, the corresponding force characteristics and the corresponding advanced behavior that is triggered after the classification.

Class name	R	FX	FY	T	AB
STEER	No	-	-	-	Steer
PULL	Yes	H(-)	S	S	Stay here
WAIT	Yes	S	S	S	Stay in area
ROTATE L/R	Yes	S	S	H(+/-)	Support loading
PUSH L/R	Yes	S	H(+/-)	S	Park
PUSH FRONT	Yes	H(+)	S	Sl	Drive forward

R: Release of handle
F: Value of force in [X,Y]-direction. Values: High(sign) or Small
T: Value of torque. Values: High(sign) or Small
AB: Corresponding advanced behavior

The relevant information is the state of the dead man's switches (pressed/released) and the two measured forces and the torque. The last column names the corresponding advanced behaviors that are activated after identifying a command.

2.3.2 Definition of feature classes as basis for the classification

For the quality of the classification the definition of significant feature classes is mandatory. For example taking only the raw force and torque progressions of different measurements with different strength would not be generalized to one class even if the intentions was identical.

To gain the best result 23 feature classes have been defined. Some of them have been preliminary defined like the average value for the different progressions. Other feature classes have been defined during the learning and evaluation to improve the quality of the classification like for example the amount of positive force values or the dominant force direction: If the absolute value of force x is bigger than the absolute value of force y then $dom(F_X) = \pm 1$. The algebraic sign is determined by comparing the value of min and max. If min is larger then the sign is negative. Finally the used feature classes are reported in the table 2. For the usability with the SVM it is necessary that all input data is standardized.

Most of the features cannot clearly be assigned to one of the classes directly. Some features are irrelevant for some classes. The method of how the class "ambiguous" is assigned is very important for the classification system in general. Naturally for all implementations of artificial intelligence there are ambiguous decisions, just like in real life. For SVM this ambiguity is manifested in different ways. If the learning has been incomplete no activations for any class are present. In this case as well as for activations that are below a specified threshold the class "ambiguous" is assigned. These assignments represent incomplete training, which is of course also possible for rather matured learning machines. If, to the contrary, the learning has advanced very far, multiple activations are possible. In such a case a rule can be conceived of that would tell what class activations overrule what other class activation. It was deemed clearer and for the beginning to also classify these features as ambiguous.

2.3.3 Support vector machine for the haptic-based classification of user intention

As input for the classification the different sensor signals coming from the haptic handle are used. These are the forces in X- and Y-direction, the torque and the contact between user and trolley. These information are accumulated over a period of time. The time progression of these sensor values is analyzed. Crucial for the success of the learning and later classification of the different classes is the calculation of distinctive features which was described in the prior section. These features make up the vectors for the *Support Vector Machine* (SVM).

Table 2. Defined features for the classification

Feature class	Description	Range
$avr(F_X)$ $avr(F_Y)$ $avr(F_T)$	Average value for force/torque in the period	[-1,...,1]
$min(F_X)$ $min(F_Y)$ $min(F_T)$	Minimum value for force/torque in the period	[-1,...,1]
$max(F_X)$ $max(F_Y)$ $max(F_T)$	Maximum value for force/torque in the period	[-1,...,1]
$\#neg(F_X)$ $\#neg(F_Y)$ $\#neg(F_T)$	Number of negative values for force/torque in the period	[0,...,1]
$\#pos(F_X)$ $\#pos(F_Y)$ $\#pos(F_T)$	Number of positive values for force/torque in the period	[0,...,1]
$max(diff\|F_X\|)$ $max(diff\|F_Y\|)$ $max(diff\|F_T\|)$	Maximum value of the absolute value of the first derivation	[-1,...,1]
$dev(F_X)$ $dev(F_Y)$ $dev(F_T)$	Standard deviation for force/torque in the period	[0,...,1]
$dom(F_X)$ $dom(F_Y)$	Investigation of dominant force direction	[-1,0,1]

$F_{...}$: Value of force in [X,Y]-direction
F_T: Value of torque.

SVMs belong to the relatively new family of Kernel Methods, that combine the simplicity and computational efficiency of linear algorithms, such as the perceptron algorithm, with the flexibility of non-linear systems, such as for example neural networks, and the rigor of statistical approaches such as regularization methods in multivariate statistics ([5]). By reducing the learning step to a convex optimization problem, which can always be solved in polynomial time, the problem of local minima typical of neural networks, decision trees and other non-linear approaches is avoided. Therefore the training of support vector machines is deterministic and retraining is faster and easier.

Moreover, due to their foundation in the principles of Statistical Learning Theory ([6]) they are remarkably resistant to over fitting especially in circumstances where other methods are affected by dimensionality.

The absolute value of the resulting decision function is called activation. It corresponds to the distance between the projected data input and the separating hyper plane in the vector space. Therefore it indicates the classification quality for a given feature: a low activation stands for an unconfident classification and a high activation for a confident classification ([7]).

3 Experimental results

Several experiments were performed to evaluate the *haptic steering* mode for InBOT. Figure 10 shows a user pushing InBOT like an ordinary shopping cart. He is supported by various features. First of all no muscle power is required to move the robot even if heavily loaded. Second InBOT automatically avoids collisions with obstacles. The user is even able to grab the handle with one hand only like depicted in figure 11. Even if the user doesn't recognize the obstacle himself InBOT moves around it safely. The effect of the responsible behaviors from the *basic behavior repertoire* is depicted in figure 12. Additionally the user is able to control InBOT by using haptic commands that are classified by a SVM.

Figure 10. The user is steering InBOT like an ordinary shopping cart using the handle. He is supported by InBOT's motors.

Figure 11. The user is steering InBOT like an ordinary shopping cart using the handle. Because he needs his left hand otherwise he grabs the handle with his right hand only. This makes it difficult for him to maneuver dexterously. So the user benefits from being supported by by the *obstacle avoidance* of the *basic bahavior repertoire*.

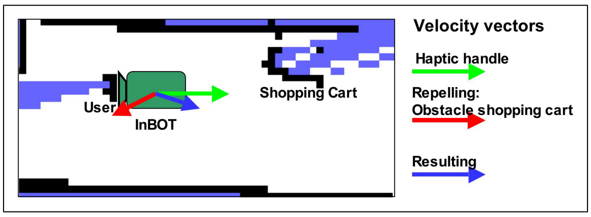

Figure 12. Occupancy grid of the scene from figure 11. The influence of the supporting *avoid obstacle* behaviors is illustrated by the different arrows ([1]).

To maximize the outcome and the quality of the classification it is important to have significant and independent data for each class. For the final test on the learning of the developed classification 4 test persons developed learning data. Each test person has contributed ten measurements per class. In the end a total number of 280 tests were collected. Each class has 40 measurements. For the verification the data of one test person has been used. Here each class has 20 measurements, in total there have been 140 tests for evaluation.

Figures 13 to 18 show representative measurements together with the identified class to get an impression about the overall outcome of the sensor data. Figure 19 shows InBOT reacting the command ROTATELEFT.

Using overall 280 tests of several different users the SVM has been trained. All data has been verified as valid for testing. As result of the learning 86 support vectors have been used to generalize the collected data. For verification of the learned classes a set of 140 tests has been used. 136 measurements have been classified correct. The classification of 4 measurements has not been successful. So the error rate is 2,8 %. Looking at the confusion matrix one can see that the classes WAIT, PUSH, PUSH LEFT, PUSH RIGHT and PULL have been classified correct (100%). For that set of data only for the two classes ROTATE LEFT and ROTATE RIGHT a full correct classification was not possible. For the class ROTATE LEFT one measurement has been classified as PUSH LEFT. For the class ROTATERIGHT three measurements have been not correctly classified, one of them has been classified as PUSH RIGHT the others have not been classified at all. The misclassification of the rotational classes and the sideward-push classes is due to the similar forces in the time progression. The reason for this is that it is difficult for the human user to push or rotate the trolley without implicitly applying additional forces or torques. The details of the confusion matrix are shown in table 3.

Table 3. Confusion matrix: percentage of correctly classified commands. The rows represent the commands given by the user, the columns the result of the classification. The values represent the percentage of how often one command was assigned to a specific class by the SVM.

	WAIT	PUSHF	PUSHL	PUSHR	PULL	ROTL	ROTR	NONE
WAIT	1,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
PUSHFRONT	0,0	1,0	0,0	0,0	0,0	0,0	0,0	0,0
PUSHLEFT	0,0	0,0	1,0	0,0	0,0	0,0	0,0	0,0
PUSHRIRGHT	0,0	0,0	0,0	1,0	0,0	0,0	0,0	0,0
PULL	0,0	0,0	0,0	0,0	1,0	0,0	0,0	0,0
ROTATELEFT	0,0	0,0	0,0	0,05	0,0	0,95	0,0	0,0
ROTATERIGHT	0,0	0,0	0,05	0,0	0,0	0,0	0,85	0,1

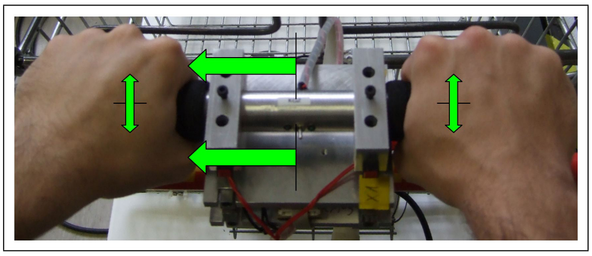

Figure 13. The haptic handle in use with the command PUSHLEFT indicated by the green arrows. Large forces are exerted in Y-direction to the left, small forces are exerted in X-direction.

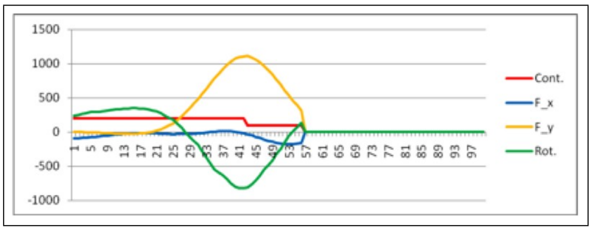

Figure 14. Example for measured forces that were classified as PUSHLEFT.

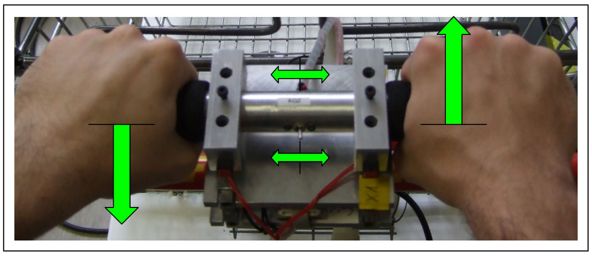

Figure 15. The haptic handle in use with the command ROTATELEFT indicated by the green arrows. Large forces are exerted in positive (right side) and negative (left side) X-direction, hardly and forces are exerted in Y-direction.

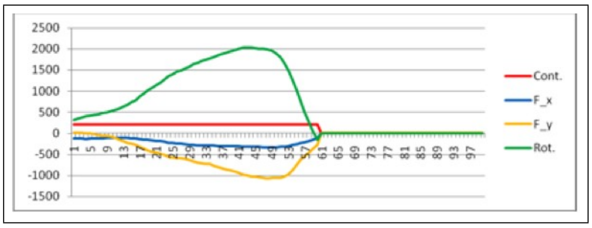

Figure 16. Example for measured forces that were classified as ROTATELEFT.

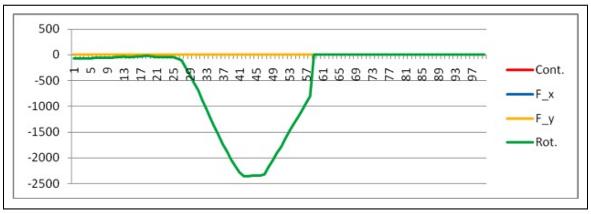

Figure 17. Example for measured forces that were classified as ROTATERIGHT.

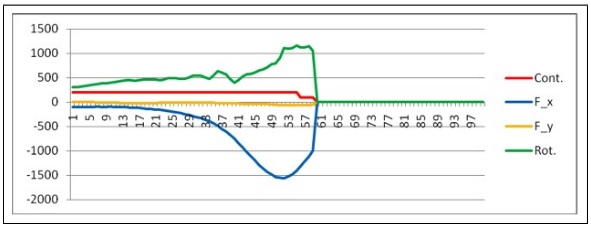

Figure 18. Example for measured forces that were classified as PULL.

Figure 19. Example of ROTATELEFT: The user pushes InBOT to a heavy crate and gives the command ROTATELEFT to ease the loading (left). InBOT turns around and stops with the side to the user (right).

4 Conclusion

For the intuitive use of the trolley a haptic interface has been developed. Based on the physical contact of user and trolley different user intention have been derived such as a "push" command or a "steering" command where user and robot move together using a version of zero-force-control. This haptic interface is realized using a sensorized trolley handle. This handle is equipped with different sensors for detecting the forces and torques applied by the user.

Based on the gathered sensor data a classification using *support vector machines* has been developed, implemented and tested. For the classification the following classes have been defined: "Steer", "Wait here", "push left", "push right", "push forward", "push backward" and "Turn around". A major concern was the discrimination of steering and the other commands to avoid possible confusions of the user. If user and trolley have contact always the steering command is active. Therefore a direct control of the robot InBOT has been developed. Only if the user releases the handle the classificator for the other command classes is used. This classificator has been trained using collected sensor data of various people (280 measurements) and then verified (140 measurements) with a success rate of 97,1%. The identified classes are then used to activate advanced behaviors. Further user studies will be performed in the future to determine the necessity of each different command from the user's point of view. A possible reduction of the number of commands will probably lower the confusion even more. This is very important even if the classification performs very well because each false-positive results in an unpredictable movement. On the other hand the user studies could reveal the need of one or more additional commands.

From the author's point of view the haptic control of a robot shopping cart is very crucial because it is very intuitive. This way of control most untrained users will try first simply because they are used to use a shopping cart this way. Additionally the user has the feeling of really controlling the robot, not only of being a spectator. Therefore the authors think that it was important to develop such an intuitive possibility of robot-control to raise the level of acceptance for this new technology. The main idea of InBOT was to make robotic technology available in all-day situations, so it was obvious not only to develop the haptic steering of the robot trolley, but also to enhance it with augmentations known from robotics like the obstacle avoidance or to give the user additional haptic-based tools to control the robot. Some of these are even known to customers, for example pushing the (empty) conventional trolley sideways to park it at a side of the corridor, others on the other hand are hopefully a new enrichment of the shopping experience.

ACKNOWLEDGEMENTS

This research has been carried out in the CommRob project (www.commrob.eu) and is partially funded by the EU (contract number IST-045441 under 6th framework programme).

REFERENCES

[1] M.Göller, T.Kerscher, J.M.Zöllner, R.Dillmann: Behavior Network Control for a Holonomic Mobile Robot in Realistic Environments, CLAWAR 2008

[2] M.Göller, F.Steinhardt, T.Kerscher, J.M.Zöllner, R.Dillmann: RFID Transponder Barriers as Artificial Landmarks for the Semantic Navigation of Autonomous Robots, CLAWAR 2008

[3] Paul Griffiths and R. Brent Gillespie: Sharing Control Between Human and Automation Using Haptic Interface: Primary and Secondary Task Performance Benefits, Human Factors: The Journal of the Human Factors and Ergonomics Society, Vol. 47, No. 3, Fall 2005, pp. 574-590.

[4] K. Regenstein, T. Kerscher, C. Birkenhofer, T. Asfour, J.M. Zöllner, R. Dillmann: Universal Controller Module (UCoM) - component of a modular concept in robotic systems. IEEE International Symposium on Industrial Electronics, 2007.

[5] N. Cristianini and J. Shawe-Taylor, 2003, "Support Vector and Kernel Methods", In: Berthold, Michael and David J. Hands (Eds.): Intelligent Data Analysis. Springer-Verlag Berlin Heidelberg New York.

[6] V.N. Vapnik, "Statistical Learning Theory", 1998, John Wiley & Sons.

[7] H. Willems, K. Reber, J. Zöllner, M. Ziegenmeyer: New Approaches to the automated analysis of ultrasonic in-line inspection data. ASME IPC 2004, Calgary, Canada.

[8] K. U. Scholl, J. Albiez, and B. Gassmann, "MCA - An Expandable Modular Controller Architecture," in 3rd Real-Time Linux Workshop, 2001, Milano, Italy, 2001.

[9] FZI, "Modular Controller Architecture Version 2." [Online]. Available: http://www.mca2.org/