

A Behaviour-based Integration of Fully Autonomous, Semi-autonomous, and Tele-operated Control Modes for an Off-road Robot

Christopher Armbrust* Martin Proetzsch*
Bernd-Helge Schäfer* Karsten Berns*

* Robotics Research Lab, Department of Computer Sciences,
University of Kaiserslautern
P.O. Box 3049, 67653 Kaiserslautern, Germany
Email: {armbrust, proetzsch, b_schaef, berns}@cs.uni-kl.de

Abstract: This paper describes a concept for a behaviour-based integration of tele-operator commands into the control system of a mobile robot. The control system presented features different control modes which allow for pure tele-operation, assisted tele-operation, and fully autonomous navigation. It is explained how the tele-operator can influence the robot's motion at any time and how he can simply and continuously change the degree of his influence.

Keywords: Robotics; Teleoperation; Robot control; Autonomous mobile robots.

1. INTRODUCTION

In various application domains, the use of tele-operated vehicles can help to fulfil tasks in areas that humans cannot enter at all or only at a high risk for their lives. Examples include space robotics (e.g. Bejczy (1994)), underwater robotics (e.g. Lin and Kuo (2001)), or the use of robots for humanitarian demining (e.g. Trevelyan et al. (2008)).

Unfortunately, the tele-operation of a complex robotic vehicle is often difficult and requires long and expensive training. Even in the best circumstances, the tele-operator is not at the robot's place, which results in the robot having a better view on its direct environment and a higher chance of detecting potential dangers. The concept of employing fully autonomous robots would solve these problems. However, even after many years of research in the field of autonomous robots, the developed vehicles cannot be employed in all environments and for all tasks without a human operator. Furthermore, with a tele-operator supervising and correcting an autonomous vehicle's operation the performance of the system can be increased.



Fig. 1. The autonomous off-road robot RAVON

Therefore, the best approach for risky missions in dangerous environments at the moment seems to be a sophisticated combination of tele-operation with semi-autonomous capabilities like collision avoidance. The author of Whitcomb (2000) gives an overview of remotely operated underwater vehicles as well as autonomous underwater vehicles.

The focus of the paper at hand is to describe a concept for a behaviour-based integration of several control modes that provide different degrees of tele-operation for the off-road robot RAVON (see Armbrust et al. (2010) and Figure 1).

2. STATE OF THE ART

The simplest way of combining tele-operation and autonomous capabilities on one vehicle is to allow for switching between different modes. The authors of Lingemann et al. (2005) provide three alternatives for calculating their robot's desired velocity and angular velocity: tele-operation and two different autonomous modes. When the robot is directly controlled by an operator, no collision avoidance is available. During autonomous motion, a laser range finder is used to detect obstacles, but no direct control via a joystick is available. The disadvantage of this approach is that the human operator has to look for obstacles himself when he decides to directly control the robot.

In the system described in Doroftei et al. (2009), the outputs of three different behaviours (one for driving towards a target, one for collision avoidance, and one for handling direct user commands) are combined by a special fusion component. This component employs a combination of two fusion methods that the authors call robot-centred and human-centred, respectively. The former gives more influence to more reliable behaviours.

The reliability of a behaviour is calculated automatically. The latter calculates the influence of a set of behaviours on the fusion by calculating the minimum combined deviation from human-defined goals. A drawback of this approach is that goals as well as the configuration parameter have to be chosen by a human operator.

In Eck et al. (2007), two different types of support for a tele-operator are identified, called passive and active support, respectively. Passive support is a combining term for all methods with which a system consults or advises the operator, e. g. by displaying status messages at the operator's console. Mechanisms for active support, by contrast, are realised on the robot with the purpose to adapt the operator's commands to the robot's current environment. For example, a safety component could provide active support by stopping the robot if it gets too close to an obstacle. The robot control system presented there features different operation modes, among them two in which the user can drive the robot manually, being supported by several assistance systems if desired. An autonomous mode is also available. The separation into active and passive support is taken up in Section 4.2 of this paper.

3. BEHAVIOUR-BASED APPROACH

In the work at hand, behaviour-based concepts are used to combine different modes of control in one robot control system. The authors make use of iB2C¹, a behaviour-based architecture that has been implemented using the robot control software framework MCA2². In the following, basic information about iB2C is provided. For details, the reader is referred to Proetzsch et al. (2010).

All of the behaviours in iB2C have a common interface for transferring so-called behaviour signals between behaviours (see Figure 2). The stimulation s is used to gradually enable a behaviour, whereas the vector $\boldsymbol{\iota}$ of inhibitory inputs is used to gradually disable it. The combined value $\iota = a \cdot (1 - i)$, with $i = \max\{\boldsymbol{\iota}\}$ is called activation. Activity a and target rating r provide information about the degree of influence a behaviour wants to have in a behaviour network and its satisfaction with the current situation, respectively. The values of the behaviour signals are limited to $[0, 1]$, which allows for an easy connection of several behaviours to build a behaviour-based network.

In addition to the standardised ports, a behaviour can have an arbitrary number of ports for control data. The output vector \boldsymbol{u} is calculated as $\boldsymbol{u} = F(\boldsymbol{e}, \iota)$ with \boldsymbol{e} being a vector of control inputs and F the behaviour's transfer function.

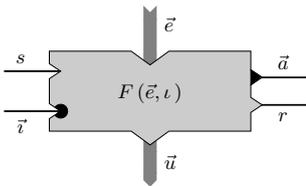


Fig. 2. A basic behaviour

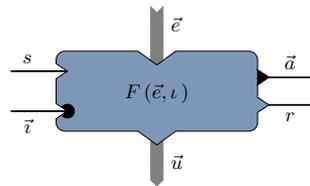


Fig. 3. A fusion behaviour

¹ iB2C: integrated Behaviour-based Control

² MCA2: Modular Controller Architecture (see <http://rrlib.cs.uni-kl.de/>)

A special type of behaviour, so-called fusion behaviours, are used to combine the outputs of p other behaviours B_c . A fusion behaviour has a normal behaviour interface. Its input vector \boldsymbol{e} is composed of the activities a_c , target ratings r_c , and output vectors \boldsymbol{u}_c . Two fusion types are available, which determine how the fusion behaviour's activity, target rating and control vector are calculated. In the case of a weighted fusion, the following formulas are used:

$$a = \frac{\sum_{j=0}^{p-1} a_j^2}{\sum_{k=0}^{p-1} a_k} \cdot \iota; \quad r = \frac{\sum_{j=0}^{p-1} a_j \cdot r_j}{\sum_{k=0}^{p-1} a_k}; \quad \boldsymbol{u} = \frac{\sum_{j=0}^{p-1} a_j \cdot \boldsymbol{u}_j}{\sum_{k=0}^{p-1} a_k} \quad (1)$$

4. SUPPORTING DIFFERENT TYPES OF CONTROL

This section starts by giving a brief overview of the control system of the mobile robot RAVON. The focus lies on the elements used for tele-operating the robot and the corresponding control flows. The different types of control that the system offers to an operator are then introduced, followed by a detailed explanation of how they are realised.

4.1 System Overview

Figure 4 depicts the main components of the control system as well as the control flow between these components that is relevant for the work at hand. As can be seen, the human operator can input commands into the global navigation system or into the behaviour-based drive system. Combined drive commands from the latter are then sent to a safety system realising collision avoidance. The output of this system is then used as input for the hardware abstraction layer (HAL). Alternatively, the pure operator commands can be used.

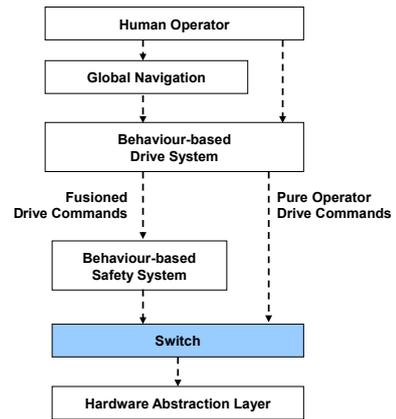


Fig. 4. The relevant control flow of the complete system

4.2 Types of Control

RAVON features three different control modes offering different degrees of active support:

- (1) Pure tele-operation: The operator explicitly bypasses the safety layer and thus has full control over the robot. The robot does not provide active support, hence colliding with objects is possible.

- (2) Assisted tele-operation: The operator controls the robot, but the robot’s safety system avoids collisions.
- (3) Full autonomy: The operator only provides high-level goals, e.g. a target position. The robot tries to get there without colliding with obstacles. The operator is able to help the robot at any time.

This classification is similar to three of the four modes of remote intervention presented in Bruemmer et al. (2002). Its authors use the term “safe mode” for what is called “assisted tele-operation” here. Furthermore, their vehicle supports a “shared control” mode in which the robot handles low-level navigation and collision avoidance, but is supplied with intermittent commands by the operator.

In the following, RAVON’s different modes and their integration are presented in detail.

4.3 Pure Tele-operation

The pure tele-operation mode is realised by sending the commands of the operator’s input devices directly (i.e. without altering by the anti-collision system) to the HAL. Two direct control input devices can be used:

- (1) the MCAGUI, which is a special GUI developed using MCA2 (It can be connected to the robot using standard WLAN or a special long-distance link.)
- (2) an off-the-shelf joystick (At the moment, a Logitech® Cordless Rumblepad™ 2 is employed.)

By simply pressing a button, the user can initiate a switching between the outputs of the safety system and the direct operator commands at any time. The reason for using a switch at this low level is that the user shall be able to very fast gain full control over the vehicle in case manual steering is necessary. This can happen in situations in which the collision avoidance might interfere with the user commands (e.g. when driving the robot through a narrow door into the lab) or in which the autonomous part of the control system fails.³ It shall be mentioned here that the autonomous part of the system is *not* turned off if the user switches to pure tele-operation. The algorithms for sensor processing and obstacle avoidance are still executed. Thus when switching to one of the other two modes, the robot can instantaneously avoid collisions with obstacles.

Figure 5 depicts a window of the GUI used for controlling RAVON. The user gets extensive data about RAVON’s state and the world around it. For example, a message from the long-range navigation is displayed in the GUI when the robot has reached a navigation point. This passive support can significantly improve the operator’s assessment of the situation and thus help him in controlling the robot.

4.4 Assisted Tele-operation

If the operator wants to control the robot, but also let it try to protect itself against collisions, he can use the assisted tele-operation mode. In this mode, the human’s inputs are filtered by the underlying safety system whenever required. Instead of the pure operator commands, the moderated outputs are sent to the HAL.

³ There is also a wireless switch that can be used to open the robot’s safety chain and thus stop it instantaneously in case of an emergency.

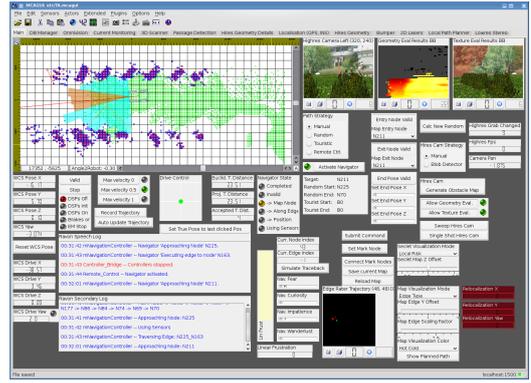


Fig. 5. A window of the GUI that is used for controlling RAVON. Here, it is connected to the RAVON simulation. Thus simulated sensor data is visualised in the GUI widgets.

Figure 6 shows an overview of the relevant parts of the control flow in the drive system. Note that this is a very high-level view on the system. The real network contains far above 100 single components.

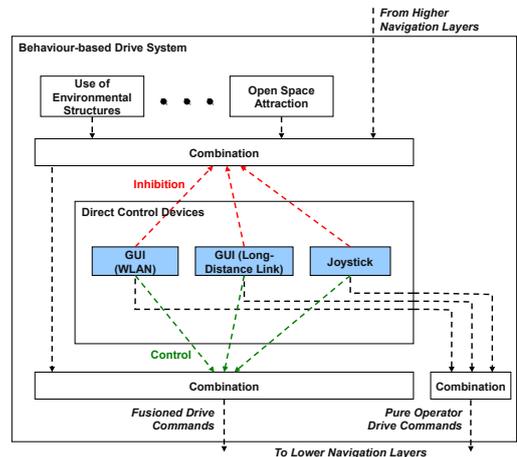


Fig. 6. The relevant control flow of the drive system

Each of the tele-operator’s input devices is handled by three components which are connected to the lower layers of the control system. Within RAVON’s control system, the commands for the six motion directions (forward/backward, rotation to the left/right, sideward left/right) are processed in six different control streams. Thus, it is possible to influence e.g. only the robot’s rotation to the left, while leaving the other motion commands unchanged. For a detailed insight into the underlying so-called degree of freedom (DOF) access pattern, the reader is referred to Proetzsch et al. (2010).

For each degree of freedom, an input device yields a motion command $c_{dev} \in [-1, 1]$. Positive values are attributed to forward, rotation to the left and sideward motion, negative ones to the opposite directions. There are six behaviours B_{OC} encapsulating the operator’s commands for the different directions and generating an activity a_{OC} , a target rating r_{OC} , and a desired control value $c_{OC} \in [0, 1]$ for the corresponding motion direction. The control values c_{OC} of these behaviours are calculated from the outputs of the input devices as follows:

$$c_{OC} = \begin{cases} \max\{0, \min\{1, c_{dev}\}\} & \text{forward, rot. left,} \\ & \text{sideward left} \\ \max\{0, \min\{1, -c_{dev}\}\} & \text{backward, rot. right,} \\ & \text{sideward right} \end{cases}$$

Activity and target rating are calculated according to the following two formulas:

$$a_{OC} = \iota \cdot c_{OC} \quad r_{OC} = \frac{1}{2} \cdot (c_{dev} - c_{actual}),$$

where c_{actual} is the actual value delivered by the sensor processing. So by changing a control value, the operator also changes the behaviour signals of these behaviours.

The outputs of the behaviour processing operator commands are sent to the safety system, which alters them in order to ensure collision free operation. So the user has full control over the robot's motion as long as the robot does not get too close to obstacles.

4.5 Full Autonomy

This control mode is the most advanced one, as it features the highest degree of autonomy. Although the robot shall navigate fully autonomously in this mode, help by an operator may be needed in difficult or dangerous situations with which the robot cannot deal on its own. In this case, the operator can help the vehicle by intervening into the motion control. Furthermore, even an autonomous robot needs an operator to provide it with (high-level) tasks.

In the fully autonomous mode of RAVON's control system, the operator can exercise control in two different ways:

- (1) by providing goals for the high-level navigation, e.g. target coordinates to which the robot shall drive
- (2) by directly influencing the robot's motion using a control input device

Controlling the high-level navigation is mainly done using the GUI, in which the operator can set target points, alter the parameters of the navigation components, etc.

If the operator chooses to directly influence the robot's motion, the components introduced in Section 4.4 are employed. Furthermore, the operator's commands are combined with the commands of the high-level navigation components in a sophisticated way that allows for a seamless, gradual transition from using *only* the autonomous components over using a *combination* of autonomous components and operator commands to using *only* the operator's commands as input for the safety system. In the following, this transition mechanism is explained in detail.

Each of the six behaviours introduced in Section 4.4 is not only connected to the lower layer, but also to the navigation layers higher up in the hierarchy. The basic idea behind this double connection is to let the operator's inputs overwrite the commands of higher layers and send its own commands to lower layers. In order to allow for a gradual transition between commands of the operator and the higher layers, behaviour-based techniques are employed.

Figure 7 depicts how the behaviour and control signals of one of these six behaviours are connected to the higher and lower layers. Assuming that p behaviours B_0, \dots, B_{p-1} are connected to the fusion behaviour at layer 1 (B_{FB1}). The activity output of the behaviour encapsulating the operator commands (B_{OC}) is connected to the inhibitory input of the fusion behaviour. The outputs of B_{FB1} and B_{OC} are coordinated by a fusion behaviour at layer 0 (B_{FB0}). The following formulas describe the calculation of the activity, target rating, and control vector of B_{FB0} .

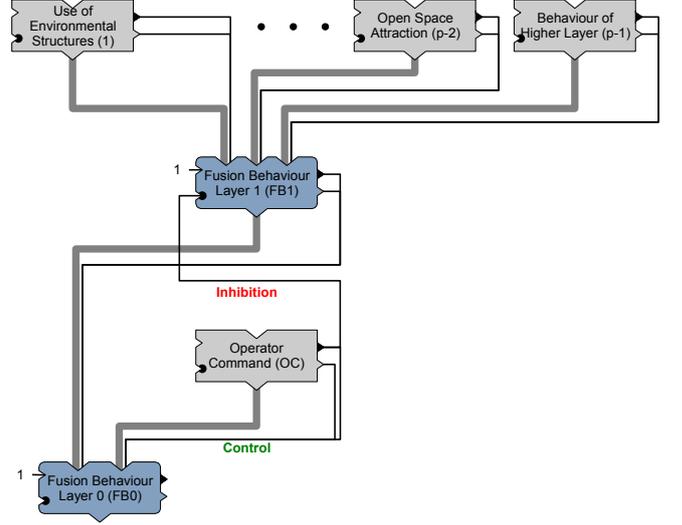


Fig. 7. The operator's commands inhibit commands of higher layers and are sent to lower layers.

According to Equations 1, the following holds true for B_{FB1} :

$$a_{FB1} = \frac{\sum_{j=0}^{p-1} a_j^2}{\sum_{k=0}^{p-1} a_k} \cdot \iota_{FB1} = \frac{\sum_{j=0}^{p-1} a_j^2}{\sum_{k=0}^{p-1} a_k} \cdot (1 - a_{OC}) \quad (2)$$

$$r_{FB1} = \frac{\sum_{j=0}^{p-1} a_j \cdot r_j}{\sum_{k=0}^{p-1} a_k}; \quad c_{FB1} = \frac{\sum_{j=0}^{p-1} a_j \cdot c_j}{\sum_{k=0}^{p-1} a_k} \quad (3)$$

The outputs of B_{FB0} can then be calculated as follows:

$$a_{FB0} = \frac{a_{FB1}^2 + a_{OC}^2}{a_{FB1} + a_{OC}} \cdot \iota_{FB0} \quad (4)$$

$$= \frac{\left(\frac{\sum_{j=0}^{p-1} a_j^2}{\sum_{k=0}^{p-1} a_k} \cdot (1 - a_{OC}) \right)^2 + a_{OC}^2}{\frac{\sum_{j=0}^{p-1} a_j^2}{\sum_{k=0}^{p-1} a_k} \cdot (1 - a_{OC}) + a_{OC}} \cdot \iota_{FB0} \quad (5)$$

$$r_{FB0} = \frac{\frac{\sum_{j=0}^{p-1} a_j^2}{\sum_{k=0}^{p-1} a_k} \cdot (1 - a_{OC}) \cdot \frac{\sum_{j=0}^{p-1} a_j \cdot r_j}{\sum_{k=0}^{p-1} a_k} + a_{OC} \cdot r_{OC}}{\frac{\sum_{j=0}^{p-1} a_j^2}{\sum_{k=0}^{p-1} a_k} \cdot (1 - a_{OC}) + a_{OC}} \quad (6)$$

$$c_{FB0} = \frac{\frac{\sum_{j=0}^{p-1} a_j^2}{\sum_{k=0}^{p-1} a_k} \cdot (1 - a_{OC}) \cdot \frac{\sum_{j=0}^{p-1} a_j \cdot c_j}{\sum_{k=0}^{p-1} a_k} + a_{OC} \cdot c_{OC}}{\frac{\sum_{j=0}^{p-1} a_j^2}{\sum_{k=0}^{p-1} a_k} \cdot (1 - a_{OC}) + a_{OC}} \quad (7)$$

Assuming that the tele-operator inputs a control value of 0, then $c_{OC} = 0$ and $a_{OC} = 0$. In this case, B_{FB1} is not inhibited and B_{OC} does not have any influence at B_{FB0} , so only the outputs of the higher navigation layers are sent to the lower layers. In case the operator issues a command with the maximum value of 1, the activity of B_{OC} will go up to 1, fully inhibiting B_{FB1} and gaining maximum influence in the weighted fusion at B_{FB0} . Figure 8 provides a diagram showing the gradual change from no operator control to maximum operator control.

If none of the high-level behaviours is active (i. e. $a_{FB1} = 0$), only the operator's commands have an influence at B_{FB0} , setting the system into the assisted tele-operation mode.

5. EXAMPLE

An example shall demonstrate how the influence of the operator in the assisted tele-operation or fully autonomous mode works. The control flow in the behaviour network depicted in Figure 7 shall be examined in detail for the two behaviour subnets dealing with forward and backward motion.

Assuming that the combined desired velocity of the higher navigation behaviours is 0.5, i. e. the behaviours want to drive the robot in forward direction with half speed and a desired activity of 1.0. The operator intervenes by moving the joystick from "full speed backward" (-1.0) to "full speed forward" (1.0). In order to properly inhibit the higher layers, the activity outputs of $B_{OC_{FW}}$ and $B_{OC_{BW}}$ are both connected to the inhibitory inputs of $B_{FB1_{FW}}$ and $B_{FB1_{BW}}$, thus their inhibition is $\max\{a_{OC_{FW}}, a_{OC_{BW}}\} =: a_{OC}$.

For the subnet dealing with forward motion, the following formulas result:

$$c_{FB1_{FW}} = 0.5 \quad a_{FB1_{FW}} = (1 - a_{OC}) \quad (8)$$

Two cases have to be distinguished for the influence of the tele-operator:

$$c_{OC_{FW}} = \begin{cases} 0 & \text{if } c_{dev} = -1 \rightarrow 0 \text{ (I)} \\ 0 \rightarrow 1 & \text{if } c_{dev} = 0 \rightarrow 1 \text{ (II)} \end{cases} \quad (9)$$

$$a_{OC_{FW}} = \begin{cases} 0 & \text{if (I)} \\ 0 \rightarrow 1 & \text{if (II)} \end{cases} \quad (10)$$

Thus the fusion of the higher navigation behaviours and the operator's input yields for the forward subnet:

$$c_{FB0_{FW}} = \begin{cases} 0.5 & \text{if (I)} \\ (1 - a_{OC_{FW}}) \cdot 0.5 + a_{OC_{FW}} \cdot c_{OC_{FW}} & \text{if (II)} \end{cases} \quad (11)$$

$$a_{OC_{FW}} = \begin{cases} (1 - a_{OC}) \cdot \iota_{FB0_{FW}} & \text{if (I)} \\ \left((1 - a_{OC_{FW}})^2 + a_{OC_{FW}}^2 \right) \cdot \iota_{FB0_{FW}} & \text{if (II)} \end{cases} \quad (12)$$

The corresponding formulas for the subnet dealing with backward motion are:

$$c_{FB1_{BW}} = 0 \quad a_{FB1_{BW}} = 0 \quad (13)$$

Again, two cases are to be distinguished for the tele-operator's influence:

$$c_{OC_{BW}} = \begin{cases} 1 \rightarrow 0 & \text{if (I)} \\ 0 & \text{if (II)} \end{cases}; a_{OC_{BW}} = \begin{cases} 1 \rightarrow 0 & \text{if (I)} \\ 0 & \text{if (II)} \end{cases} \quad (14)$$

And the fusion of the two inputs yields:

$$c_{FB0_{BW}} = \begin{cases} c_{OC_{BW}} & \text{if (I)} \\ 0 & \text{if (II)} \end{cases} \quad (15)$$

$$a_{OC_{BW}} = \begin{cases} a_{OC_{BW}} \cdot \iota_{FB0_{BW}} & \text{if (I)} \\ 0 & \text{if (II)} \end{cases} \quad (16)$$

In order to control the robot, the outputs of the subnets for the two directions have to be coordinated. For this purpose, the control value c_{FB0} of the backward subnet is negated. Again, a weighted fusion is employed, yielding:

$$c_{combined} = \begin{cases} \frac{(1 - a_{OC}) \cdot \iota_{FB0_{FW}} \cdot 0.5 + a_{OC_{BW}} \cdot \iota_{FB0_{BW}} \cdot (-c_{OC_{BW}})}{(1 - a_{OC}) \cdot \iota_{FB0_{FW}} + a_{OC_{BW}} \cdot \iota_{FB0_{BW}}} & \text{if (I)} \\ (1 - a_{OC_{FW}}) \cdot 0.5 + a_{OC_{FW}} \cdot c_{OC_{FW}} & \text{if (II)} \end{cases} \quad (17)$$

$$a_{combined} = \begin{cases} \frac{\left((1 - a_{OC}) \cdot \iota_{FB0_{FW}} \right)^2 + \left(a_{OC_{BW}} \cdot \iota_{FB0_{BW}} \right)^2}{\left((1 - a_{OC}) \cdot \iota_{FB0_{FW}} \right) + \left(a_{OC_{BW}} \cdot \iota_{FB0_{BW}} \right)} \cdot \iota_{combined} & \text{if (I)} \\ \left((1 - a_{OC_{FW}})^2 + a_{OC_{FW}}^2 \right) \cdot \iota_{FB0_{FW}} \cdot \iota_{combined} & \text{if (II)} \end{cases} \quad (18)$$

Figure 8 shows the values of some of the different velocities calculated in the behaviour-based network against the operator's control input c_{dev} .

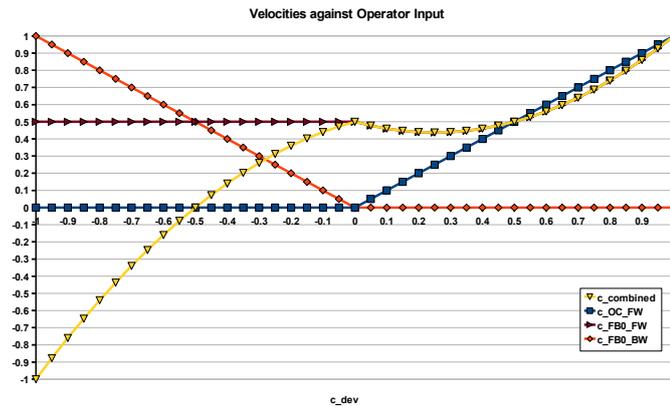


Fig. 8. Different velocities calculated within the behaviour-based network printed against the tele-operator's input

Looking at the curve of $c_{combined}$, it can be seen that the operator is able to increase the robot's velocity continuously from the lowest to the highest value against the desire of the higher navigation behaviours to move the robot forward with a constant velocity. For $-1 \leq c_{dev} < -0.5$, the operator's input lets the robot move backwards. At $c_{dev} = -0.5$, the combination of operator and higher behaviour inputs results in the robot not moving at all. If the operator inputs a control value $c_{dev} = 0$ then the higher navigation behaviours have full control over the robot. For $c_{dev} \geq 0$, the activity of the backward subnet is 0. Hence it does not have any influence, resulting in $c_{combined} = c_{FB0_FW}$.

6. CONCLUSION AND FUTURE WORK

The paper at hand presented the integration of different control modes with different levels of operator influence within one robot control system. It described how an operator can easily choose the control mode fitting best to the current situation. Using behaviour fusion, the control system allows an operator to continuously increase or decrease his influence on the robot's motion.

In the future, the currently used GUI shall be replaced by a Java-based application (see Koch et al. (2008)). A widget in the Java-GUI shall display imagery provided by GoogleEarth[®]. The operator will then be able to use that widget to provide the robot with paths and targets, which will significantly improve the high-level control of the robot.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the funding of Christopher Armbrust and Bernd-Helge Schäfer by the PhD Programme of the University of Kaiserslautern's Department of Computer Sciences.

Furthermore, Team RAVON thanks the following companies for their technical and financial support: IK elektronik, Mayser, Hankook, MiniTec, SICK, DSM Com-

puter, Hübner Giessen, John Deere, Optima, ITT Cannon, MOBOTIX, and Unitek Industrie Elektronik.

REFERENCES

- C. Armbrust, T. Braun, T. Föhst, M. Proetzsch, A. Renner, H. Schäfer, and K. Berns. *Using robots in hazardous environments: Landmine detection, de-mining and other applications*, chapter RAVON – The Robust Autonomous Vehicle for Off-road Navigation. Woodhead Publishing Limited, 2010. to be published.
- Antal K. Bejczy. Toward advanced teleoperation in space. In Steven B. Skaar and Carl F. Ruoff, editors, *Teleoperation and Robotics in Space*, Progress in Astronautics and Aeronautics Series. AIAA, 1994.
- David J. Bruemmer, Donald D. Dudenhoeffer, and Julie L. Marble. Dynamic-autonomy for urban search and rescue. In *AAAI Mobile Robot Competition 2002*, AAAI Technical Report, pages 33–37, Edmonton, Alberta, Canada, 28 July - 1 August 2002. AAAI Press.
- Daniela Doroftei, Geert De Cubber, Eric Colon, and Yvan Baudoin. Behavior based control for an outdoor crisis management robot. In *Proceedings of the IARP International Workshop on Robotics for Risky Interventions and Environmental Surveillance 2009 (RISE 2009)*, Brussels, Belgium, January 12–14 2009. IARP.
- Daniel Eck, Manuel Stahl, and Klaus Schilling. The small outdoor rover merlin and its assistance system for tele-operations. In *6th International Conference on Field and Service Robotics (FSR 2007)*, pages 277–286, Chamonix, France, July 9–12 2007.
- J. Koch, M. Reichardt, and K. Berns. Universal web interfaces for robot control frameworks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nice, France, September 22–26 2008.
- Qingping Lin and Chengji Kuo. On applying virtual reality to underwater robot tele-operation and pilot training. *The International Journal of Virtual Reality*, 5(1), 2001.
- Kai Lingemann, Andreas Nüchter, Joachim Hertzberg, and Hartmut Surmann. About the control of high speed mobile indoor robots. In *Proceedings of the Second European Conference on Mobile Robotics (ECMR '05)*, pages 218–223, Ancona, Italy, September 7–10 2005.
- M. Proetzsch, T. Luksch, and K. Berns. Development of complex robotic systems using the behavior-based control architecture iB2C. *Robotics and Autonomous Systems*, 58(1):46–67, 2010.
- James P. Trevelyan, Sung-Chul Kang, and William R. Hamel. Robotics in hazardous applications. In Bruno Siciliano and Oussama Khatib, editors, *Springer Handbook of Robotics*, chapter 48, pages 1101–1126. Springer Berlin Heidelberg, 2008.
- Louis L. Whitcomb. Underwater robotics: Out of the research laboratory and into the field. In *Proceedings of the 2000 IEEE International Conference on Robotics & Automation (ICRA 2000)*, pages 709–716, San Francisco, CA, USA, April 24–28 2000.