Safety for an Autonomous Bucket Excavator During Typical Landscaping Tasks

Gregor Zolynski, Daniel Schmidt and Karsten Berns

Abstract The Robotics Research Lab in Kaiserserlautern, Germany, pursues the goal of automating a mobile bucket excavator for excavation and loading tasks. This document contains a short introduction to the autonomous bucket excavator THOR, a concept for low-level safety using laser scanners, and a high-level collision avoidance system using behavior-based control.

1 Introduction

Today's research in the domain of construction machines moves into the field of intelligent robotics. In the future fully autonomous machines will take over more and more complex construction tasks. To reach this goal, small steps need to be taken. The Robotics Research Lab in Kaiserserlautern, Germany, pursues the goal of automating a mobile bucket excavator for excavation and loading tasks. During this process many different problems need to be solved. One of these problems is the safety of the vehicle and of course the safety of humans. This document contains a short introduction to the autonomous bucket excavator THOR, a concept for low-level safety using laser scanners, and a high-level collision avoidance system using behavior-based control.

G. Zolynski (✉) · D. Schmidt · K. Berns
University of Kaiserslautern, Kaiserslautern, Germany
e-mail: zolynski@cs.uni-kl.de

D. Schmidt
e-mail: d_smith@cs.uni-kl.de

K. Berns
e-mail: berns@cs.uni-kl.de

A. Rodić et al. (eds.), *New Trends in Medical and Service Robots*,
Mechanisms and Machine Science 20, DOI: 10.1007/978-3-319-05431-5_24,

Fig. 1 The autonomous bucket excavator THOR (*Terraforming Heavy Outdoor Robot*)

2 Autonomous Bucket Excavator THOR

The test platform at the Robotics Research Lab is the robot THOR (*Terraforming Heavy Outdoor Robot*) which is based on a VOLVO EW 180B excavator (Fig. 1). The long-term goal of the project is to develop an excavator which is capable to perform material movement and landscaping tasks on a construction site in a fully autonomous manner.

Apart from developing the complex software control system which generates autonomous decisions to solve the problems, the standard human-driven 18 ton wheeled excavator had to be converted into a PC-controllable robot.

Therefore, electrohydraulic control valves (Fig. 2) for steering the machine, length measurement sensors (Fig. 3) for indirectly measuring the joint angles from cylinder lengths, angular joint encoders (Fig. 4) for measuring joint angles and velocities, electric circuit boards (Fig. 5) for closed-loop control of the joints and a powerful embedded PC (Fig. 6) were installed on the machine. For environment perception the machine uses multiple outdoor laser scanners from SICK (Fig. 7) which deliver precise distance data around the machine and can store multiple hardware safety fields for low-level safety (explained in Sect. 4).

As the decisions of the machine have to be taken in a highly dynamic environment full of disturbances no “classical” preliminary planning and trajectory generation system could be chosen.

Instead, a reactive behavior-based control approach was chosen which already proved to deliver good results for complex offroad-navigation in unstructured

Fig. 2 Electrohydraulic control valves for steering the machine

Fig. 3 Length measurement sensors for indirectly measuring the joint angles from cylinder lengths

Fig. 4 Wheel encoders for measuring joint angles and velocities

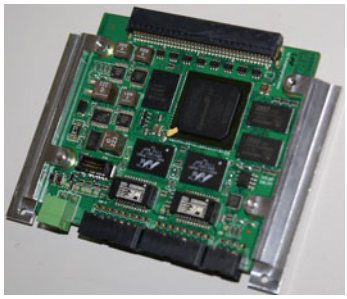

Fig. 5 Electronic circuit boards for closed-loop control of the joints

Fig. 6 Powerful linux PC running the complex software system

Fig. 7 Laser scanner for gathering exact distance information for the machine

terrain (Armbrust et al. 2011). After a brief description of the used behavior framework and the existing behavior network, Sect. 5 presents the integrated obstacle avoidance layer to guarantee safety during trajectory generation.

3 Low-Level and High-Level Safety Layers

In this work two different systems for safety are presented, which shall be clarified in this section. The low-level safety is a very basic system which is meant to prevent harm in the case of catastrophic control failure. It is designed as a standalone system which can bring the machine to an emergency stop if an imminent collision is detected. In this work, only the arm of the excavator is considered, but the system can be extended to also provide safety e.g. for the rear side.

The high-level safety is a concept which is embedded into the control architecture of the excavator. It provides implicit safety against collision with detected, known obstacles and also against self-collisions of the machine (the construction of the arm allows configurations which can damage the driver's cabin). In theory, the high-level safety should always be able to avoid any collisions so that the

low-level system will never need to become active. Also, the high-level system is designed to work around a collision while the low-level system's only ability is the emergency halt.

4 Safety Fields for Collision Avoidance

To ensure a safe emergency stop of the bucket excavator during panning even in the event of non-responding control software or a disrupted messaging system, a "last line of defense", realized in hardware, has to be present. This solution is thought to react only if all other safety measures have failed, i.e. because of software or hardware malfunction. To this end, a specialized secondary safety system has to be used. The function of this approach is to switch on a collision-preventing subsystem before panning the excavator's superstructure.

Industrial grade laser scanners like the SICK S3xxx[1] family of sensors contain a feature called Protective Fields to switch a special output whenever a predefined safety field is violated. In this work we use a SICK LMS 151 which is not as rigorously certified as the S series but contains a Field Evaluation functionality which can be used as demonstrator of the safety system.

The working principle of these laser scanners is to emit laser pulses through a rotating mirror and to measure the phase shift of the returned signal. The measurements of each revolution are combined into a vector of distances. A safety field (or safety envelope) is a vector containing the minimum distances for each of distance vector's elements. If any of the measurements falls below its respective minimum a field violation is triggered.

The most problematic part of hardware-supported safety fields is the limited amount of available safety field definitions. The top-of-the-range models can store up to 42 fields, which must be defined offline and can be switched during operation. The LMS 151 offers 10 evaluation fields.

The approach presented here consists of three steps: (1) generating a representative set of potential arm poses, (2) clustering of similar arm poses to minimize the number of required safety envelopes, and (3) the computation of safety envelopes which both fit tightly around the associated arm poses and do not leave any portion of the arm uncovered.

4.1 Preparing a Set of Arm Poses

The generation of safety field descriptions begins with a set of arm poses which are likely to occur during normal operation. These poses are calculated in a nested loop and with distinct ranges and step sizes for each arm joint angle α_i. The step

[1] www.mysick.com

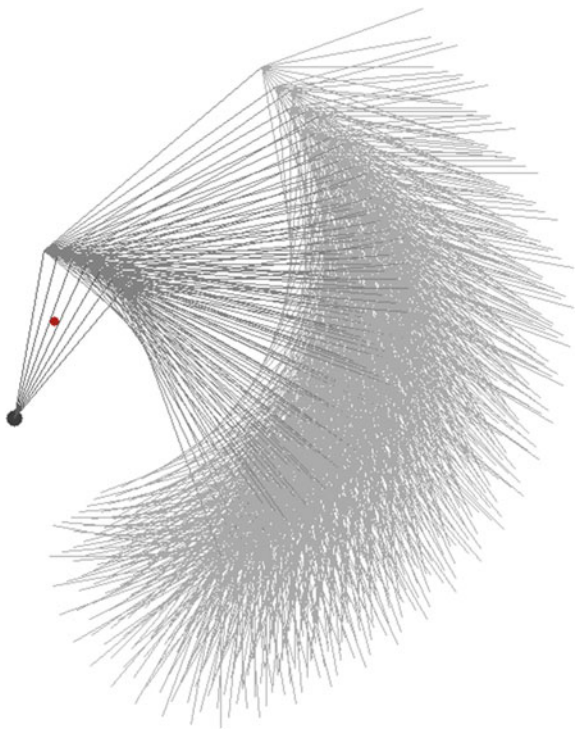

Fig. 8 Example of possible arm poses for a 3-joint-arm (*side view*). The *black dot* represents the first arm joint (the origin), the *red dot* is the position of a laser scanner

size of joints closer to the base needs to be chosen smaller, because of the bigger impact caused by a longer lever arm. The position J_i of each joint can be calculated using forward kinematics:

$$J_i := J_{i-1} + l_i \begin{pmatrix} \cos \sum_{k=0}^{i} \alpha_k \\ \sin \sum_{k=0}^{i} \alpha_k \end{pmatrix}, \quad J_0 := \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{1}$$

where l_i is the length of the ith arm segment, and α_k are the joint angles. Figure 8 shows an example of several thousand arm poses.

4.2 Clustering of Arm Poses

The most important and interesting part is the grouping of the arm poses into a predefined number of clusters. As already mentioned, a safety scanner is only able to store a very small amount of safety field descriptions, in our case 10. To group the generated poses into a given number of groups a clustering technique needs to be employed. Since the number of clusters is known beforehand, the well-known k-means clustering has been chosen. This clustering method needs a distance function which is used as measure for intra-group similarity and inter-group dissimilarity.

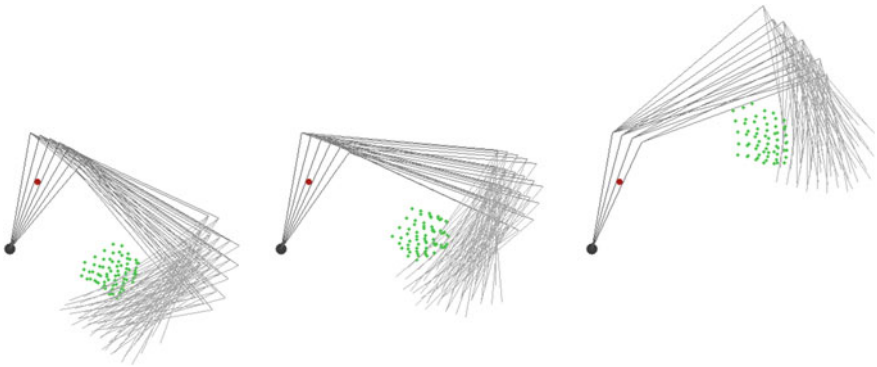

Fig. 9 Example of clustered arm poses using the center of mass (*green dots*)

Finding the right distance metric is the most crucial part of the clustering. The function should yield smoothly varying results when the arm pose is changed by small amounts, it should describe the arm pose in a coherent way, and it should have a small number of parameters (low dimensionality).

One such metric, which has also been found to give very good clustering results, is the Euclidean distance of the geometrical average of each arm pose. The geometrical average (or "center of mass") is easily calculated as

$$M = \frac{1}{N+1} \sum_{i=0}^{N} J_i \tag{2}$$

where $J_0 \ldots J_{N-1}$ are joint positions and J_N is the position of the tool center point. The Euclidean distance of average joint positions (and TCP) fulfills all the requirements listed above for a good metric for arm pose clustering. Figure 9 shows an example of three arm pose groups.

4.3 Calculation of Safety Fields

After having grouped similar arm poses into clusters a safety field has to be defined for each group. The definition of safety fields is done using a simple geometrical calculation. The sensor position P is defined relative to the first arm joint, which is positioned at the origin of the coordinate system. The safety envelopes are generated using ray intersection. From the position of the scanner rays are cast at a predefined angular resolution.

$$R_i := P + \begin{pmatrix} \cos \beta_i \\ \sin \beta_i \end{pmatrix}. \tag{3}$$

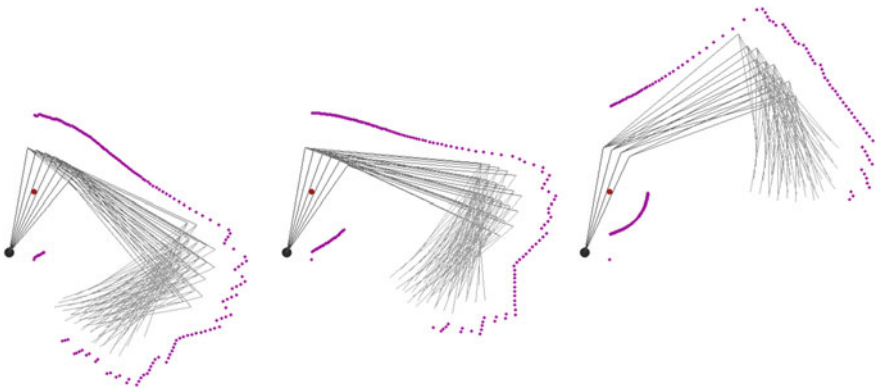

Fig. 10 Example of clustered arm poses with associated safety fields

The rays are then intersected with the segments of each arm pose. The longest distance (plus an additional safety margin) between an intersection point and the ray is stored as minimum needed distance for coverage. Figure 10 shows a few examples of arm pose groups and their respective safety fields.

5 Behavior-Based Obstacle Avoidance

As the previously presented safety layer guarantees that the machine does not hit obstacles, even in case of wrong high-level software decisions the behavior-based obstacle avoidance tries to guarantee the generation of collision-free trajectories which prevent from activation of low-level safety.

To make this possible, the control structure of the excavator has to be briefly explained first. The classic approach for control systems for autonomous mobile robots uses global knowledge to solve the task like in the sense-plan-paradigm (Brooks 1991). It uses all sensor information from the bottom layer to calculate the solution on the highest layer. This top-level solver gets very complex and difficult to implement. In behavior-based control systems like *iB2C*[2] the task is split into relatively simple behaviors following the behavior-based paradigm (Arkin 1998).

There is a set of decentralized behaviors, which get the information they need to perform their task. These behaviors have defined ports to connect them to a network and organize them in groups. One single behavior is usually easy to implement and can only solve relatively simple problems in most cases. More detailed information can be found in (Albiez et al. 2003) or (Proetzsch et al. 2007).

The behavior module is the result of the developments in (Albiez 2007) and (Proetzsch et al. 2005). Each behavior is wrapped into a uniform module (as shown

[2] Integrated behavior-based control.

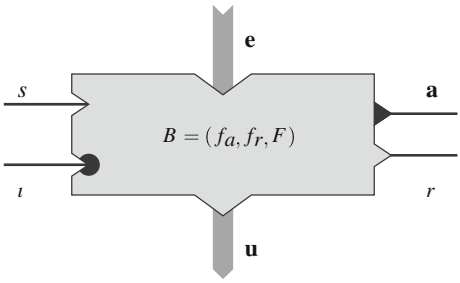

Fig. 11 Standard *iB2C* behavior module: the input ports are on the *left* (stimulation *s* and inhibition *i*) and output ports on the *right* (activity **a** and target rating *r*). With input **e** and output **u** arbitrary ports are provided

in Fig. 11). Standardized ports allow the reuse of behavior modules in a different context (Kiekbusch and Proetzsch 2011). The stimulation s stands for the relevance of the behavior ($s = 0$ means no stimulation, $s = 1$ full stimulation, values between 0 and 1 mean a partial stimulation). Inhibition i is used to suppress the behavior action from the outside and is the important port used for the obstacle avoidance. The activity a represents the current amount of influence of the behavior ($a = 1$ all output values have the highest impact and $a = 0$ means the behavior is inactive). The target rating signalizes how content the behavior is ($r = 1$ means maximum dissatisfaction and $r = 0$ total contentment).

One of the most difficult parts in behavior based controls is the fusion of the single behaviors. If there are different behaviors, with different goals that want to dictate the same output control value, a clever decision must be made. Derived from the context of the behaviors, different strategies are necessary. The *iB2C* offers three types of fusion behaviors (*maximum fusion*, *weighted average*, *weighted sum*), see (Proetzsch et al. 2010) for further details.

5.1 Behavior-Based Inverse Kinematics Solver

As it is an important issue of this document how the excavator arm can be inhibited, a short overview of the arm control is given here. It is realized with a behavior-based inverse kinematic solver, which works within the *iB2C* Framework (Pluzhnikov et al. 2012). The solver gets the desired position and orientation of the tool center point as input and generates six values for the different joint angles (shown in Fig. 12) as result.

In this control the distributed principle as mentioned above has been applied. There is no global control that evaluates a centralized solution for the task. Instead each joint has an own control to decide in which direction it rotates to solve the task. If a new desired pose is given to the system, every joint control requests its current angle and calculates only a small step (about 1°–4°) towards the given pose. The movement of all joints is then internally simulated and used as the origin for the next step. If the calculated pose is close enough to the desired pose the movement process is stopped by a supervisor module. This design has been

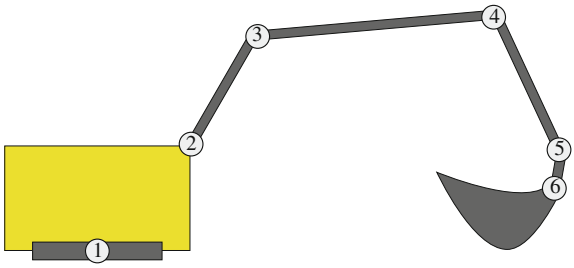

Fig. 12 Schematic view of the excavator arm. *1* Torso joint, *2* boom joint, *3* dipper arm joint, *4* stick joint, *5* bucket pitch joint, *6* bucket roll joint

realized with *iB2C* behavior modules. Each joints has two control parts, one to reach the desired position and one to adjust the orientation which are combined with a *weighted sum* fusion behavior. Each of these control parts consist of two behaviors to turn the joint in a positive or a negative direction, which then is combined by a *maximum fusion* behavior. These four behaviors per joint are also partially connected interconnected by inhibition ports, to avoid a too large steps of joints that lie in the same rotational plane.

5.2 Obstacle Avoidance

The newly developed behavior-based obstacle avoidance is then also connected to the inhibition ports for each joint. It uses the information gathered from internal models about the machine itself and objects in the environment to calculate the different workspace views for every component of the excavators digging arm. It has six parallel ports to request the nearest obstacles for every joint. Also the perpendicular foot for every component is calculated.

It is first checked if the obstacle distance is smaller than the component length. In this case a collision with the component is possible. The next check tests if the obstacle is lying inside of the inhibition range of the other dimensions. This has the following reasons: It is only necessary to consider an obstacle for the height limitation, if azimuth and length are lying inside a given range. Otherwise a obstacle that may be very near in height but with a totally different azimuth will cause an inhibition. But this would lead to unwanted effects, as it restricts the joint movements for no reason. And because we are interested in a collision with the component and not only with the joints, we use the perpendicular foot, that gives us the shortest distance to the component itself. Afterwards we differentiate between the case a minimum or a maximum has to be defined, based on the obstacles position. Only one of the limitations is set, because only one obstacle is considered, what prevents a inflation of limitations. The structure of the length and azimuth workspace calculations is equal, only with the difference that the azimuth limitations for all joints (except the last—bucket roll) are only calculated and then the maximum/minimum is published, because the azimuth can only be inhibited at

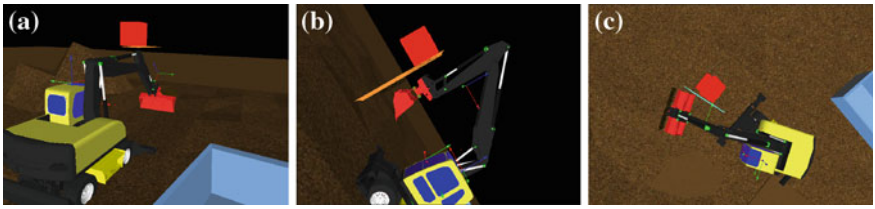

Fig. 13 Demonstration of the *WorkspaceLimiter*. **a** (*Left*) shows a height-limitation with a maximum, **b** (*center*) a length-limitation also maximum and **c** (*right*) shows a limitation for the azimuth

Table 1 The inhibition relation between joints and coordinate system axis

	Torso	Boom	Dipper arm	Stick	Bucket pitch	Bucket roll
Azimuth	•					•
Length		•	•	•	•	•
Height		•	•	•	•	•

the boom and bucket roll joint. For all vacant limitations a standard parameter is published, to reset a possible older value.

To validate the results and visualize the process a complex debug output has been implemented. If activated multiple planes are drawn directly in the simulation to represent the single limitations. Different colors symbolize minimums or maximums for all dimensions and joints. The series of images in Fig. 13 shows various settings.

There are three modules with similar functions to restrict the behaviors: One for height, length and for azimuth restrictions. They affect the joints as depicted in Table 1.

The modules calculate the restrictions for all joints they are responsible for in a parallel job. Therefore, the following tasks are performed:

- Fetch limitation values and current values for all elements
- Calculate the inhibition values for all joints
- Charge the values against each other and limit them to the boundings
- Publish the calculated values.

After all inhibition values have been calculated they are charged against each other. Because of the structure of the digging arm, for example an inhibition of the bucket means also an inhibition for the stick, dipper arm and the boom. Due to that fact it is necessary to add up all inhibitions along the element chain. But because the inhibition value has to lie between 0 and 1, it has to be limited afterwards.

6 Conclusion

In this document we have introduced the autonomous bucket excavator project and the demonstrator vehicle THOR. We presented a short overview of the Robotics Research Lab's behavior-based control system *iB2C* as well as its application for the calculation of inverse kinematics.

The novel parts of this document presented two separate research topics. First, a concept for a hardware-supported "last line of defense" collision prevention system during panning motions of our excavator. This system was designed to allow for automatic generation of a very limited amount of safety fields for any sensor position using only the kinematic description of the excavator.

Second, the utilization of *iB2C* for workspace limitation and (self-)collision avoidance. An existing behavior-based control structure was extended to generate smooth trajectories while avoiding known obstacles as well as configurations which could damage the vehicle itself.

References

Jan Albiez. *Verhaltensnetzwerke zur adaptiven Steuerung biologisch motivierter Laufmaschinen*. GCA Verlag, 2007.

Jan Albiez, Tobias Luksch, Karsten Berns, and Rüdiger Dillmann. An activation-based behavior control architecture for walking machines. *The International Journal on Robotics Research, Sage Publications*, 22:203–211, 2003.

R. Arkin. *Behaviour-Based Robotics*. MIT Press, 1998. ISBN-10: 0-262-01165-4; ISBN-13: 978-0-262-01165-5.

Christopher Armbrust, Martin Proetzsch, and Karsten Berns. Behaviour-based off-road robot navigation. *KI - Künstliche Intelligenz*, 25(2):155–160, May 2011. http://dx.doi.org/10.1007/s13218-011-0090-2.

R. A. Brooks. Intelligence without reason. A.i. memo no. 1293, Massachusetts Institute of Technology (MIT), Artificial Intelligence Laboratory, 1991.

Lisa Kiekbusch and Martin Proetzsch. Tutorial: Control system development using ib2c. Technical report, Robotics Research Lab, University of Kaiserslautern, 2011. unpublished.

Sergey Pluzhnikov, Daniel Schmidt, Jochen Hirth, and Karsten Berns. Behavior-based arm control for an autonomous bucket excavator. In *Commercial Vehicle Technology 2012 – Proceedings of the Commercial Vehicle Technology Symposium (CVT)*, pages 251–261, Kaiserslautern, Germany, March 14–15 2012.

M. Proetzsch, T. Luksch, and K. Berns. Fault-tolerant behavior-based motion control for offroad navigation. In *Proceedings of the 20th IEEE International Conference on Robotics and Automation (ICRA)*, pages 4697–4702, Barcelona, Spain, April 18-22 2005.

M. Proetzsch, T. Luksch, and K. Berns. The behaviour-based control architecture ib2c for complex robotic systems (extended version). unpublished, 2007.

Martin Proetzsch, Tobias Luksch, and Karsten Berns. Development of complex robotic systems using the behavior-based control architecture iB2C. *Robotics and Autonomous Systems*, 58(1):46–67, January 2010. doi:10.1016/j.robot.2009.07.027.