2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Simulation and Control of an Autonomous Bucket Excavator for Landscaping Tasks

Daniel Schmidt, Martin Proetzsch, and Karsten Berns

Abstract— **Introducing autonomous machines to construction areas can improve many of the occurring processes. Therefore, this paper deals with problems in the field of creating an autonomous bucket excavator. A novel behavior-based approach for motion control is presented which allows natural boom trajectories, achieves good environment disturbance compensation behavior, guarantees safety of the movements concerning human beings and structures, and keeps a high extensibility of the system for future improvements. Furthermore, a simulation of soil deals as the basis for safe tests of the excavator's behavior in a simulated dynamic environment.**

I. INTRODUCTION

Although the mechanical performance of construction equipment could be continuously increased over the past decades, a further efficiency enhancement is often limited by the operator's skills. A promising approach is the use of autonomous machines as this avoids dangerous and exhausting situations for a human driver . The area of building appears suitable for this, as even today large excavation pits are precisely planned and provide a rather controlled environment. Excavators and trucks working almost 24 hours a day would lead to shorter construction times and to lower building costs and will therefore have a high market potential in the future.

As these machines can harm human beings and can damage buildings or themselves, a simulation environment has to be created in which sensor data evaluation and control approaches can be safely tested. On the one hand, a dynamic simulation of the construction equipment within the environment is required. On the other hand, a physically adequate and real-time capable simulation of soil must be implemented. Two differently motivated approaches exist for the latter one. The first one is used in the area of material research and needs high computational effort to simulate effects like soil-tool interaction [1] or rock-flows and mud-flows [2]. The Discrete Element Method (DEM) used is designed for high physical precision, is computationally complex, and cannot fulfill real-time requirements. The second approach can be found in computer games. Particle systems, which are often GPU (Graphics Processing Unit) powered, simulate particulate material in real-time by omitting physical correctness [3], [4]. As a consequence, the approach presented in the work at hand fulfills real time requirements by reducing precise calculations to the necessary regions.

Related work concerning the control of autonomous excavators [5] uses task centered-goal oriented planning structures which define goals for a specific trench excavation task based on real operator behavior [6]. These goals are divided into different activities fulfilling them. Such a planner can create smooth movements concerning position and velocity changes of the bucket. An additional strategy is implemented for the removal of obstacles inside the excavation space. As no complex sensor evaluation took place, the strategies are not working for obstacles wider than the trench. Others [7], [8], [9] can be described as parametrized scripted joint control based on existing knowledge about excavation strategies. Here, a central planner creates the control values from excavation until dumping using a set of parametrized scripts. In the case of unexpected disturbances, however, script parameters have to be recalculated, which may take some time and can lead to unwanted freezing of the movement. Furthermore, these scripts cannot be paused at arbitrary times, which leads to undesired repetitions of scripted steps or a complete maneuver.

This paper presents the project AMoBa (Autonomer Mobilbagger) dealing with control approaches for an autonomous mobile excavator. After an introduction to the excavator system (Sect. II), the physics based simulation environment is presented in Sect. III. It is used to evaluate a novel control approach for excavator control which is based on the iB2C (integrated Behavior-Based Control) architecture [10] and allows an incremental implementation of a robust excavation system (see Sect. IV and V).

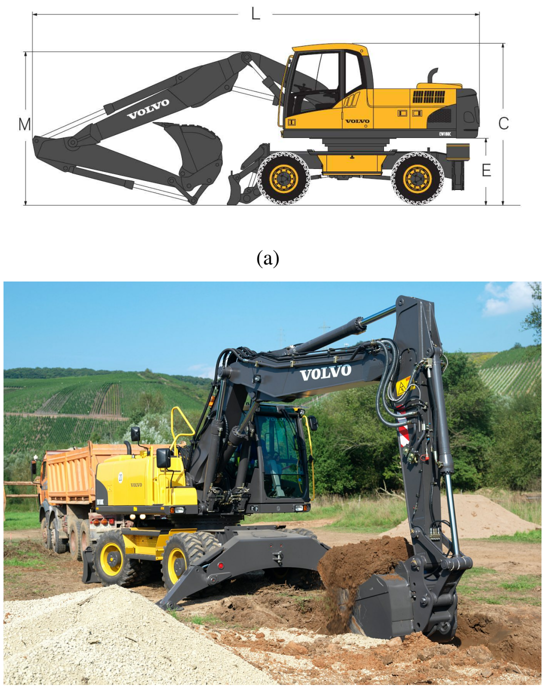
II. EXCAVATOR SYSTEM

The work at hand is based on the wheeled 20 ton Volvo EW/180B bucket excavator [11], see Fig. 1. It can produce lifting forces around 100 kN. The project's long-term objective is to create a completely autonomous excavator system which should be able to perform landscaping tasks on a planned excavation site. The desired 3D surface should be modeled by a landscape architect in an appropriate CAD (computer-aided design) tool. The actual environment was modeled in 3DStudioMax[1].

The software system needs to gather information about the internal excavator state and its environment and should be able control the machine. Therefore, the bucket excavator

The authors are with the Robotics Research Lab at the Department of Computer Sciences, University of Kaiserslautern, P.O. Box 3049, 67653 Kaiserslautern, Germany http://agrosy.informatik.uni-kl.de
d_smith@cs.uni-kl.de, proetzsch@cs.uni-kl.de, berns@cs.uni-kl.de

[1] Autodesk 3ds Max is a 3D construction toolkit. More information can be found under http://www.autodesk.com



(a)

(b)

Fig. 1. Schematic side view of the bucket excavator as given in [11] (a) and a picture of the real one (b). The depicted dimensions are $M = 2.92\,\mathrm{m}$, $L = 8.72\,\mathrm{m}$, $C = 3.17\,\mathrm{m}$, $E = 1.29\,\mathrm{m}$.

is currently modified and equipped with joint encoders, pressure sensors, load cells, electronic valves, DSPs (digital signal processors) and Embedded PCs (personal computers). High level sensors like laser scanners will be used to detect surface changes and objects around the excavator. Until this process is finished, an appropriate test environment is used for evaluating functionality already implemented.

III. Simulation System

The existing software system can be divided into a lower simulation part which creates an appropriate representation of the reality and its behavior inside the software system and an upper control part. It contains knowledge about the direct and inverse kinematics of the excavator to control its movements, environment perception algorithms, and high level planners to identify excavation positions.

The lower part needs to be sufficiently exact to allow for directly transferring the control part knowledge onto the real equipped excavator. As it is a highly dynamic system, a physics engine (Newton Game Dynamics[2]) is used which allows to simulate the excavator masses and joints and produces a quite realistic behavior of the whole excavator. Interaction with other dynamic elements like walls or other machines is modeled, too. Due the fact that particulate material cannot be realized within the physics engine, an external soil simulation has been constructed and connected to make excavation and landscaping possible.

As realistic soil behavior of inhomogeneous material cannot be simulated with the required precision under the given real time requirements, the presented approach deals as a validation technique for the basic functionality of the control system and its containing algorithms for environment change detection from sensor data.

In contrast to describing soil with a continuum approach, the simulation is based on single particles and their physical constraints. As the granularity of the material is implicitly considered, the explicit modeling of soil-bucket interaction or composition is not required. Continuum models would need many physical rules to correctly handle these cases.

A particle can be described as a point in the three dimensional space $\vec{p}$ with a velocity vector $\vec{v}$ driving it from one position to another over the course of time. At the beginning of each time step Δt, the particle is moved according to equation 1. During these fixed calculation steps, the movement of each particle is taken to be uniform due to Newton's first law of motion.

$$\vec{p}_{\text{new}} = \vec{p}_{\text{old}} + \vec{v} \cdot \Delta t \tag{1}$$

As all other abilities are specifically added depending on the physical requirements of the simulation, a force vector $\vec{f}$ is introduced. This allows for summing up all forces acting on the particle and the determination of the movement via the second Newtonian law shown in equation 2.

$$\vec{f} = m \cdot \vec{v} \cdot \Delta t \tag{2}$$

The forces acting on each particle during each time step can be divided into such which externally act on each particle independently from its position, while more complex ones act between multiple particles.

The first types implemented here are gravity $\vec{g}$, velocity $\vec{v}$, and global damping or friction d. Their relating forces $\vec{f}_g$ and $\vec{f}_d$ are calculated by $\vec{f}_g = m \cdot \vec{g}$ and $\vec{f}_d = \vec{v}^2 \cdot d$.

The basic principle for the second ones is modeling the particles as smooth uniform spheres with a radius r, which preserve their distance to other objects via spring-damper systems. To simulate glued material, an attraction force is produced in the opposite direction. A graphical representation of all forces acting on the particles is shown in Fig. 2.

During each calculation step all colliding neighbors are found for each particle and the interaction forces are calculated. A 3D cube presorting structure, in which each particle is inserted depending on its position, immensely reduces the search effort, as only neighboring cubes have to be evaluated.

The soil conditions can be changed by adjusting the parameters per time step or adding more complex interaction laws which allows to extended the simulation to more complex conditions. As the modeling just deals as a rough validation mechanism and real-world conditions can never be exactly reproduced, the presented approach suffices. The particle simulation allows to move soil with the bucket inside the simulation environment which produces the desired detectable surface changes. Figure 3 shows a screen shot taken from the external simulation.

[2] http://www.newtondynamics.com/

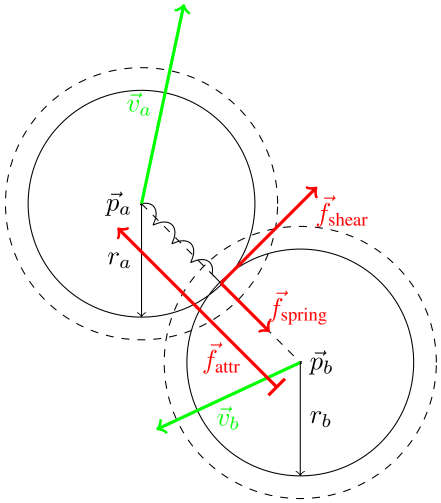


Fig. 2. Graphical representation of all forces acting on a particle a at position p_a with radius r_a from particle b (position p_b, radius r_b). The forces are the longitudinal spring force f_{spring}, the transversal shear force f_{shear}, and the attracting force f_{attr}.

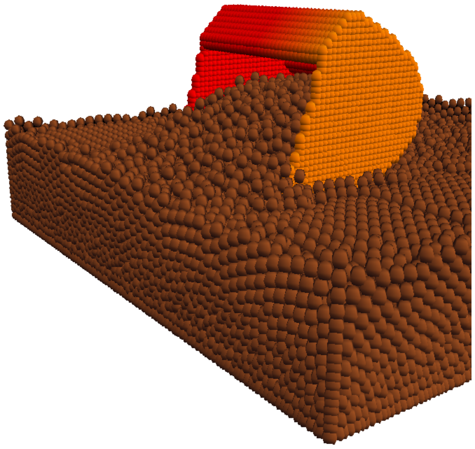


Fig. 3. Visualization of the particle soil simulation including a bucket consisting of particles.

IV. Behavior-Based Excavation Control Approach

Due to the multitude of possible disturbances in the area of outdoor robotics, an appropriate solution for controlling the excavation process is required. As behavior-based approaches have shown their suitability for several applications [12], the same can be expected for the field of construction equipment. Here, the iB2C architecture [10] is used which is implemented in the Kaiserslautern branch of the Modular Controller Architecture.[3] Here behaviors are implemented as modules providing a standardized interface for building up large-scale behavior networks. The coordination of competing behaviors is accomplished by so-called fusion behaviors having the same interface as basic behaviors. A further hierarchical level can be introduced by arranging several behaviors in a behavioral group which again has the standard behavior interface.

The development process for iB2C starts with the top-down design stage where the given task is decomposed into sub tasks which in combination lead to the desired goals. The behavior-based driving modules inside the RAVON project [13] proved to safely operate an autonomous four-wheeled outdoor robot and can be easily adopted for this project. Therefore the main task in the work at hand was to create a central structure which is able to control the bucket excavator at a fixed position inside the simulated system environment to perform autonomous surface shaping. The desired surface, a trench in this case, is given by the operator. The excavator should take the current sensor information about joint angles, the TCP and its global position and provide the desired TCP pose of the bucket. The autonomous excavation and landscaping task can be decomposed into the following parts:

1) Make a surface scan with the (simulated) laser scanners and identify a target excavation position.
2) Approach the target position with the bucket.
3) Scratch the surface deep enough to dig out the desired amount of soil.
4) Move the bucket to a target dumping position.
5) Dump the soil onto a pile at a given position.

Each of them can be decomposed according to the degrees of freedom they have to influence. These are:

- Turn the torso angle (e. g. initial surface scan)
- Adjust the boom length (e. g. approaching the excavation point)
- Adjust the TCP height (e. g. digging deep)
- Change the bucket pitch angle (e. g. dumping of soil)

Besides the *Conversion Layer* containing data conversion modules from delta (`DeltaToAbsoluteValues`) to absolute cylinder or Cartesian coordinates and vice versa (`CylinderCartesianConverter`), the implementation in iB2C proceeds bottom-up by adding a fusion behavior for each motion direction (see Fig. 4, *Behavior Fusion Layer*).

On top, each of the DOFs (degrees of freedom) is split up into a positive and a negative fusion direction (e. g. `(F) pitch right` and `(F) pitch left`) represented by the `mbbBasicDriveBehavior` modules (marked as BDB). Furthermore, a fusion behavior for each direction coordinates the concurrent access of more than one behavior (e. g. `(F) pitch angle`). Within this structure, each higher behavior can influence its desired direction command, e. g. increasing or decreasing the bucket angle, by submitting commands to the respective fusion behavior.

The next stage deals with behaviors implementing the required sub tasks for the excavation process. The *Excavation Process Behavior Layer* contains three behavioral groups which are used during the five excavation states. Each of them consists of modules which influence one or more of the motion directions provided by the *Behavior Fusion Layer*.

The `InitialScanning` group uses the (simulated) laser scanners to scan the actual surface and store the

[3] MCA2-KL `http://rrlib.cs.uni-kl.de`

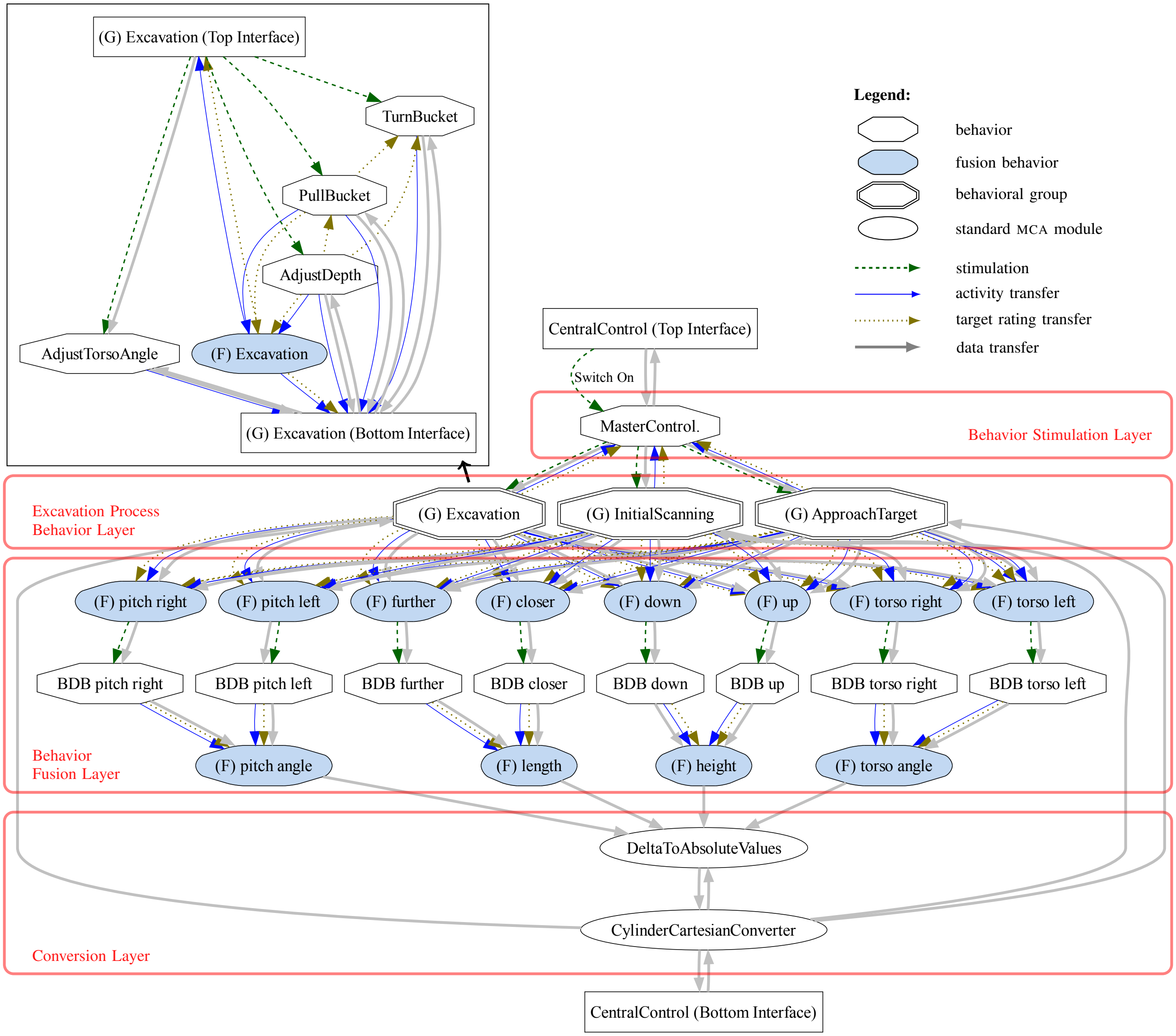


Fig. 4. Behavior network of the central control component as well as the contents of one of the contained groups (`(G) Excavation`, top left).

gathered 3D point cloud (see Fig. 5). It is used before each excavation step and determines areas to excavate in the scanned environment based on the actual scanned distance data and the desired one. The algorithm uses one two-dimensional grid for each point cloud and evaluates the average value of scanned points per cell. If no point in the cell can be found the surrounding cells are used to build an average value where empty cells without surrounding filled ones will get a height in z direction of zero. Afterwards, the two evaluated grids for the actual and the desired surface are compared. A third height difference grid is built by subtracting the desired surface grid from the actual surface grid. To find possible positions for excavations areas in the size of the bucket, around $1.5\,\mathrm{m}^2$ in the actual case, with a minimum excavation depth of $20\,\mathrm{cm}$, are searched. That one with the most space around and deepest excavation depth is chosen as inaccuracies will not lead to excavation of too much material. Once the different kinds of grids are evaluated the module can additionally deliver the actual distance of the TCP from the actual or the desired surface. The whole process is graphically presented in Fig. 6. Although extensions like obstacle detection or object recognition have to be implemented, the used algorithms can be transferred directly to the real excavator as the simulated laser scanners deliver data realistic enough for the application. An overview about the simulation framework SimVis3D[4] and its sensor data approximation compared to reality is given in [14].

The group `ApproachTarget` is used three times during the excavation procedure and contains behaviors for reaching

[4] http://rrlib.cs.uni-kl.de

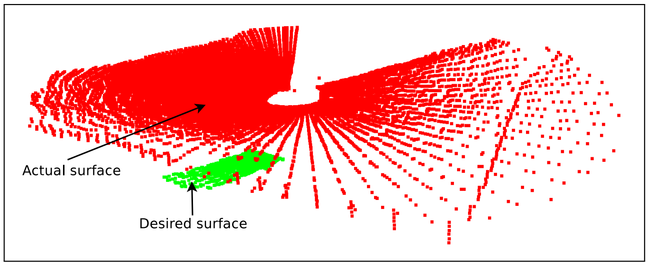


Fig. 5. Example laser scanning point cloud of the actual surface and the generated desired surface.

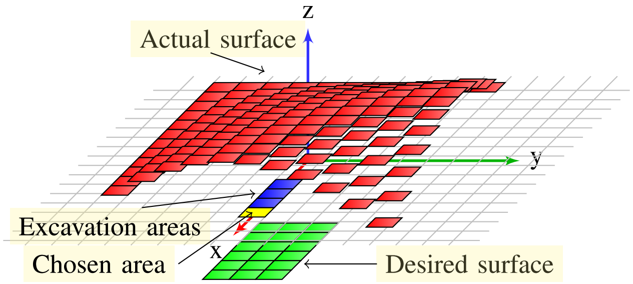


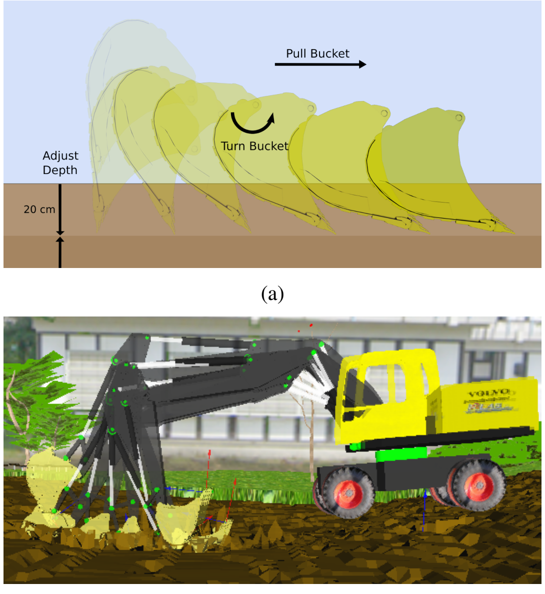
Fig. 6. Graphical representation of the surface distance evaluation algorithm using two grid structures for the actual surface and the desired surface. The four possible excavation areas are identified and the first area is chosen as it is surrounded by eight green fields and has the highest excavation depth, i.e. the red area at this position is high.

a specific target TCP pose. It is used to approach the excavation area (Approach Excavation Position), to move to the dump position (Approach Dumping Position), and to dump the soil (Dump). Included safety behaviors can be enabled to keep a safety distance during the movement to prevent the excavator from touching objects next to its trajectory.

Removing soil from the surface is performed by the Excavation group, see the detailed view in Fig. 4. Similar to the strategy of a human driver, the bucket vertically (i. e. pitch angle equals $-90°$) penetrates the surface until it reaches a depth of around 20 cm. Then, a scraping behavior is achieved by a combination of adjusting the boom elongation and the bucket angle while the bucket height and the torso angle are kept, see Fig. 7. The complete process is performed by four behaviors influencing the required degrees of freedom (DOF) of the bucket.

Each of these basic abilities can be evaluated during tests before the development proceeds. This way, the developer benefits from an iterative implementation with incremental steps.

In order to execute these stages in the correct order, a switching mechanism for the excavation states is contained in the *Behavior Stimulation Layer*. The MasterControl behavior implements a state machine as depicted in Fig. 8. The transition between states is done according to the behavior signals (activity and target rating) of the stimulated behavior groups of the next layer. Furthermore, additional error and pause states are also included. Simulation tests in the next section prove that this approach is able to create the desired trajectories and is stable in relation to occurring disturbances.



(a)

(b)

Fig. 7. Idealized desired trajectory of the bucket during the excavation process (a) and time lapse screen capture from the simulation (b). First the AdjustDepth behavior is active until the desired value of 20 cm is reached. Then PullBucket and TurnBucket become active at the same time to constantly decrease the boom length and adjust the bucket angle.

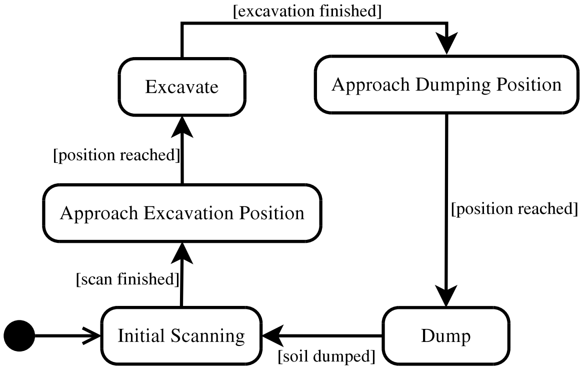


Fig. 8. State machine of the central MasterControl module.

V. Excavation Process Evaluation

To show the ability of the behavior-based control system to produce natural trajectories and the error compensation efficiency during the excavation process, the TCP position is recorded and visualized in the following experiments.

Figure 9 shows the initial scanning procedure. The task is to first turn the torso to an initial angle and rotate until a maximum scanning angle is reached. Thereby the bucket itself is kept at a specific height and distance from the turning center. The movement starts at some point P with an almost direct approach of the minimum scanning position

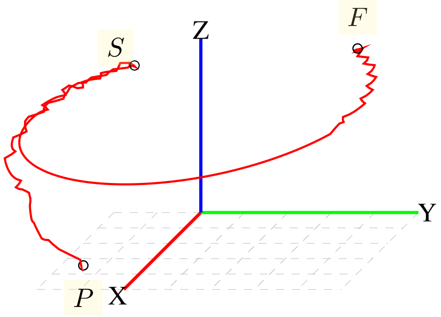
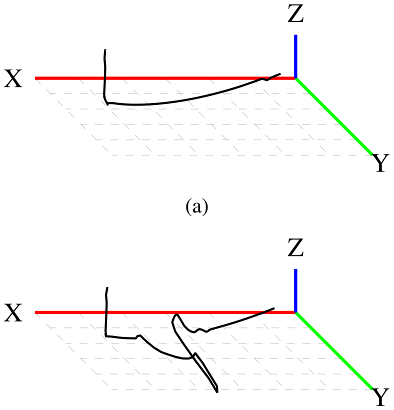


Fig. 9. Recorded trajectory of the initial scanning procedure. The apparently shaking zigzag movement is in the range of 15 cm and mostly depends on unadjusted closed loop controller parameters.



(a)

(b)

Fig. 10. Undisturbed (a) and disturbed (b) trajectory of the excavation process

S. Once it is reached, the scan starts and a rather ideal semi-circle proves the smooth movement during turning until this procedure is finished (F).

An undisturbed and a disturbed excavation process is shown in Fig. 10(a) and 10(b). In the undisturbed case, the trajectory is smooth and shows the desired behavior as it should remain 20 cm below the curved surface shown in Fig. 7. The other trajectory, in which a strong disturbance occurs while the bucket is pulled near, contains a loop at the relating position. A force acting for a short period on the bucket in Y-Z direction which may be produced by an obstacle inside the earth (e. g. a stone), pushes the bucket in the relating direction. As the object is passed, the force vanishes and the bucket returns directly to the desired trajectory and correctly completes the process. This example shows the inherent ability of the behavior-based system to intelligently handle disturbances directly without canceling the actual process or recalculating the trajectory.

VI. Conclusion

The soil particle simulation allows to move soil from one position to another and produces desired surface changes. The long-term goal is to decrease the particle size and add more complex interaction rules as the informational value of the simulated tests will directly benefit from an increased realism. As this will lead to higher computational effort, an already started porting to the graphics card via the OpenCL GPU interface[5]shows impressive results and should overcome occurring problems fulfilling real-time requirements. Nevertheless, the existing physical simulation of the excavator and the environment allows testing implemented excavation strategies and adjusting parameters.

Also the behavior-based control approach proved to create natural trajectories even in the case of disturbances. Future work includes the addition of a safety layer which prevents the excavator from undesired and maybe destructive movements. It will depend on the correct analysis of high level sensor data taken from the environment. Finally the application on the real excavator is outstanding.

Acknowledgments

The research work presented in this paper is sponsored by the Stiftung Rheinland-Pfalz für Innovation. Thanks to the company Volvo Construction Equipment (Konz) for providing the excavator and technical information and to the Department of Mechanical and Process Engineering chaired by Prof. Dr.-Ing. Schindler for multi-body system simulation of the excavator.

References

[1] A. R. Carillo, D. A. Horner, J. F. Peters, and J. E. West, "Design of a large scale discrete element soil model for high performance computing systems," in *ACM/IEEE Conference on Supercomputing*, 1996.

[2] J. Gascuel, M. Cani, M. Desbrun, E. Leroy, and C. Mirgon, "Simulating landslides for natural disaster prevention," in *9th Eurographics Workshop on Computer Animation and Simulation (EGCAS98)*, 1998.

[3] D. McAllister, "The design of an api for particle systems," *UNC Computer Science Tech Report*, 2000.

[4] S. Green, "Cuda particles," NVIDIA Corporation, Tech. Rep., September 19 2007.

[5] D. A. Bradley and D. W. Seward, "The development, control and operation of an autonomous robotic excavator," *Journal of Intelligent and Robotic Systems*, vol. 21, no. 1, pp. 73–97, November 2 2004.

[6] Y. Sakaida, D. Chugo, K. Kawabata, H. Kaetsu, and H. Asama, "The analysis of excavator operation by skillful operator," in *Proc. of 23rd International Symposium on Automation and Robotics in Construction*, 2006, pp. 543–547.

[7] H. Cannon, "Extended earthmoving with an autonomous excavator," Master's thesis, Carnegie Mellon Robotics Institute, 1999.

[8] P. Rowe and A. Stentz, "Parameterized scripts for motion planning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 97*, vol. 2, 1997.

[9] A. Stentz, J. Bares, S. Singh, and P. Rowe, "A robotic excavator for autonomous truck loading," *Autonomous Robots*, vol. 7, no. 2, pp. 175–186, September 1999.

[10] M. Proetzsch, T. Luksch, and K. Berns, "Development of complex robotic systems using the behavior-based control architecture iB2C," *Robotics and Autonomous Systems*, vol. 58, no. 1, 2010.

[11] K. Volvo, "Volvo wheeled excavator ew180c," March 3 2007.

[12] R. Arkin, *Behaviour-Based Robotics.* MIT Press, 1998.

[13] C. Armbrust, T. Braun, T. Foehst, M. Proetzsch, A. Renner, H. Schaefer, and K. Berns, "Ravon — the robust autonomous vehicle for off-road navigation," in *Proceedings of the IARP International Workshop on Robotics for Risky Interventions and Environmental Surveillance 2009 (RISE 2009).* Brussels, Belgium: IARP, January 12–14 2009.

[14] T. Braun, J. Wettach, and K. Berns, "A customizable, multi-host simulation and visualization framework for robot applications," in *13th International Conference on Advanced Robotics (ICAR07)*, Jeju, Korea, August 21-24 2007, pp. 1105–1110.

[5] http://www.khronos.org/opencl/