Interconnection of the behavior-based control architecture and a detailed mechatronic machine model for realistic behavior verification of the Thor project

Daniel Schmidt[1], Dr. Tamas Juhasz[2], Prof. Karsten Berns[1], Prof. Ulrich Schmucker[2]

Robotics Research Lab[1]
Department of Computer Science
University of Kaiserslautern
Gottlieb-Daimler-Strasse, Building 48
D-67653 Kaiserslautern
d_smith@cs.uni-kl.de
berns@cs.uni-kl.de

Fraunhofer Institute for
Factory Operation and Automation IFF[2]
Sandtorstrasse 22
D-39106 Magdeburg
Tamas.Juhasz@iff.fraunhofer.de
Ulrich.Schmucker@iff.fraunhofer.de

Abstract: The enclosing Thor (Terraforming Heavy Outdoor Robot) project's goal is to perform typical tasks of a bucket excavator autonomously, like landscaping, mass excavation, or material transport on construction or mining sites. The paper at hand presents an approach for close interoperateration between its reactive behavior-based control approach for highly dynamic environments with a lot of disturbances and a realistic dynamic simulation of the robot for safe parameter evaluation under given time boundaries.

1 Introduction

In these days, construction machine manufacturers become aware of the fact that automation of trucks, excavators, caterpillars and similar machines could improve work processes and lead to increased safety and productivity. Additionally, such machines can also operate in areas which are polluted with toxic or nuclear waste without harming human beings. Together with Volvo Construction Equipment, the goal of the Thor (Terraforming Heavy Outdoor Robot) project is to develop an excavator test platform which is capable of performing tasks from simple mass excavation to full landscaping autonomously on a construction site [SPB10]. Therefore, a wheeled 18 ton excavator Volvo Ew 180B (see figure 1) was equipped with electro-hydraulic control valves (figure 2), length (figure 3) and angular sensors for measuring joint angles and velocities (figure 4), electronic circuit boards for performing closed-loop control (figure 5), laser scanners for environment perception (figure 6), and powerful PC hardware (figure 7) to perform high level autonomous robot control inside the complex robotics framework Finroc [RFB13, HSB+10].

Figure 1: Picture of the 18 ton mobile excavator Volvo EW180B.

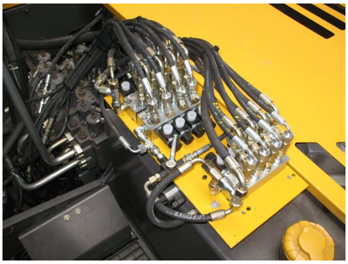

Figure 2: Electrohydraulic control valves from HYDAC.

Figure 3: Highly precise length measurement sensors from MTS-Sensors.

Figure 4: Absolute multi-turn angular encoders from Kuebler.

Figure 5: Modular rack with electronic circuit boards for closed-loop control from Robot Makers.

Figure 6: Planar Laser-scanner LMS-151 from SICK.

Figure 7: Powerful Core-i7 Control PC running Linux and the robot control framework Finroc.

In comparison to rather distinct and fixed working conditions of industry robots this machine should operate under highly dynamic and disturbed conditions on a real construction site together with human operated machines and construction workers. Therefore, a highly reactive control structure is needed, which is capable of taking correct decisions in case of disturbances and uncertainty. Here the behavior-based control (iB2C) [Pro10] architecture is used to build up an adaptive control structure which is capable of performing cyclic tasks such as material transport and rather specific landscaping operations [ASB12].

As the machine can tremendously harm workers or objects around the machine, the safety of its operation plays an important role inside the whole project. In order to perform safe and cost-efficient tests of the control system, a basic simulated test environment was built (see Figure 8). The test environment is based on the

Figure 8: The 3D SimVis3D simulation framework containing the machine and its environment

SimVis3D framework [BWB07] that utilizes the Newton physics engine [WSB10]. This allows 3D rendering of shapes, gathering data from complex virtual cameras and laser scanners, and simulate basic physical interactions between rigid bodies. This is enough for simulating most of the smaller robots which should avoid collisions at all and perform no interaction with the environment. In the Thor project the dynamics of the machine will play an important role and a simulated machine model which does not contain these elements will not deliver reliable test results which could be transferred onto a real machine.

In order to increase the realism of the virtual excavator, we decided to model its mechanical and hydraulic parts completely inside a mechatronic simulation environment. This work has been done within the ViERforES-II project (Virtual and augmented reality for highest safety and reliability of embedded systems) and was funded by the German Federal Ministry of Education and Research. The Modelica language has been chosen to describe the physical behavior of the excavator. Modelica is an object-oriented, declarative, multi-domain modeling language for component-oriented modeling of complex systems, e.g., systems containing mechanical, electrical, electronic, hydraulic, thermal, control, electric power or process-oriented subcomponents[1]. The mechatronic model is simulated in the Dymola environment[2] and is connected to the SimVis3D framework as an alternative physics simulation for the machine itself.

[1]Modelica: `http://www.modelica.org`.

[2]Dymola: `http://www.dynasim.se`.

The next section will give an overview about the behavior-based control approach and its implementation for cyclic excavation inside the project. Section 3 presents the elements of the realistic dynamic simulation of the excavator hydraulics system. In section 4 the system coupling and operation results from both parts are presented. The whole work is finally concluded in section 5.

2 Behavior-based control approach

As disturbances are sure to appear in the area of outdoor robotics, a highly reactive behavior-based approach to create its control commands and implicitly keep parameter bounds during operation was chosen. Furthermore, this method is very adaptive as it includes fusion of different "desires" influenced by safety goals or target achievement ratings, similar to a human being and can therefore produce comprehensible results.

The behavior-based architecture iB2C extends the modular Finroc structure. Such modules and groups contain already existing edges and internal methods which allow for specific connection of the modules. Behavior-based groups can subsume other modules and groups as the usual groups do. As a single behavior-based element (module or group) is usually simulating a small part of the system's behavior, a lot of "intelligence" is implicitly defined by its structure. A single behavior is shown in figure 9.

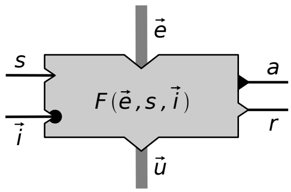

Figure 9: Behavior module in iB2C [PLB07] with its defined edges for stimulation (s), inhibition ($\vec{i}$), activity (a), target rating (r), and in- and output vectors ($\vec{e}, \vec{u}$).

The task in the work at hand was to create a central structure which is able to control the autonomous bucket excavator at a fixed position during the excavation process. It should take the actual sensor information about joint angles and TCP position and deliver the desired TCP pose of the bucket. The idea was to identify the basic movement maneuvers during the excavation process and let each of these low level behaviors control one of these sub-steps per time — *Turn the torso angle (e.g. during an initial surface scan), Adjust the boom length (e.g. for approaching the excavation point), Adjust the TCP height (e.g. make digging deep possible), Turn the bucket pitch angle (e.g. while dumping soil into the truck).*

In Cartesian coordinates the different behaviors would disturb each others' axes.

Therefore, the cylinder coordinate system was chosen. Here the position is identified via turning angle, elongation, height and the orientation vector.

The abstract view of the group central control is shown in figure 10 containing different layers. The highest control layer (*behavior stimulation layer*) contains the `Master Control State-Machine` which is a central behavior module controlling the other behavior parts. It shall be mentioned that control values are normalized

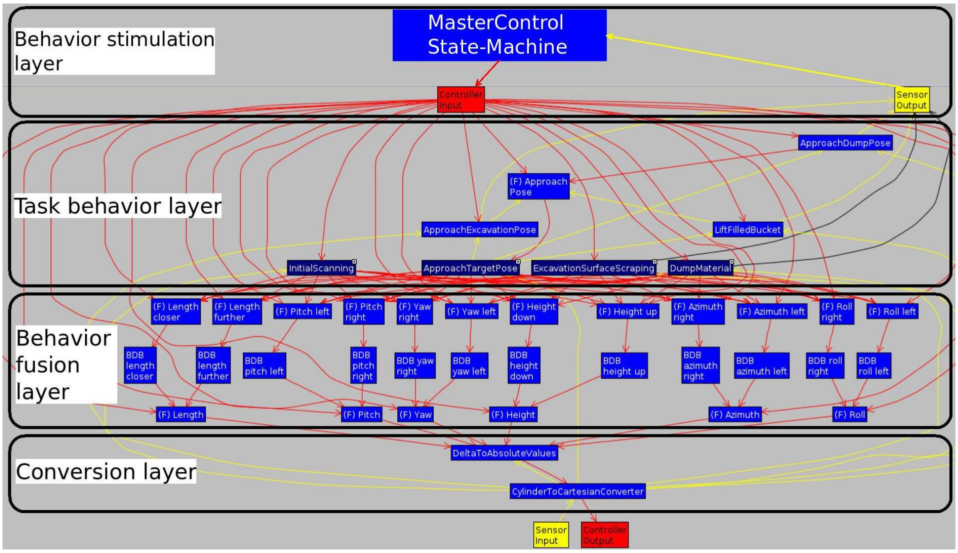

Figure 10: Modules of the central control group which can be divided into four layers.

between ± 1 and represent variations from the actual value (i.e. delta changes). Therefore, the rather straightforward *conversion layer* contains two non-behavior-based modules. The `DeltaToAbsoluteValues` converter creates the relating output value ($value_{new}$) by adding the delta value ($\Delta value$) multiplied by a parameter factor (scalefactor) scaling the delta value to the actual value ($value_{actual}$). The `CylinderToCartesianConverter` performs the translation from cylinder coordinates to Cartesian coordinates and vice versa.

The behavior-based layers of the central control from top to bottom are the *behavior stimulation layer*, the *task behavior layer* and the *behavior fusion layer*. The first one contains central knowledge about the usage of the lower layers and controls their stimulation to create the overall system behavior. The three layers are described in the following sections.

2.1 Behavior stimulation layer

It contains a state machine which controls the system shown in figure 11. Usually one state is kept until the target rating and activity input of the actually stimulated behavior group reaches zero or a "finishing signal" from below is received. Then the next state with its relating outputs is activated. Additional error and pause states are also included.

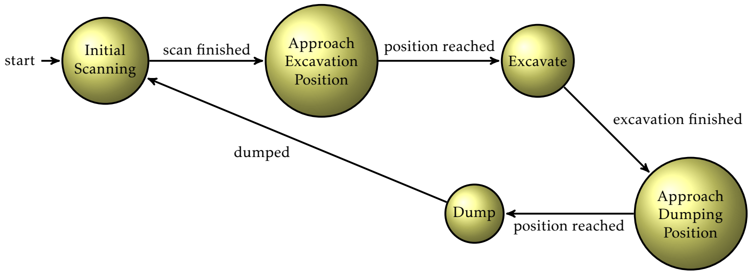

Figure 11: State machine of the central MasterControl module.

2.2 Task behavior layer

The *Task behavior layer* contains the task specific behavior groups which are used during state execution.

Initial scanning of the surface The `InitialScanning` group uses the simulated laser scanners to scan the actual surface and is only used during the correspondent state. It contains different smaller behaviors and one central behavior `StateMachine` which controls the whole scanning procedure. The group is parametrisable with minimum and maximum initial turning angles $\alpha_{\text{initial turn}_{\text{min}}}$ and $\alpha_{\text{initial turn}_{\text{max}}}$ and initial turning velocity $v_{\text{initial turn}}$.

Approach a specific target pose This group `ApproachTargetPose` is used multiple times during the excavation procedure and contains behaviors for reaching a specific target *TCP* pose. It is used in the first time to approach the excavation area (`ApproachExcavationPose`), secondly to move to the dump position (`ApproachDumpPose`), and finally to dump the soil (`DumpMaterial`).

Shaping the surface (excavation) The group mostly influencing the way the surface is shaped is the `ExcavationSurfaceScraping` group. To achieve the goal of successively removing the surface, the bucket vertically (i.e. pitch angle equals

−90°) penetrates the surface until it reaches a depth of around 20 cm. Then the boom and stick are adjusted to reduce the elongation. Therefore, the boom angle is increased and the stick angle decreased. During this process the bucket angle is permanently reduced to achieve a scraping behavior. The whole process is shown in figure 12.

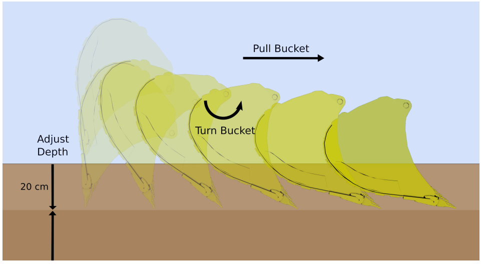

Figure 12: Idealised desired trajectory of the bucket during the excavation process. First the `AdjustDepth` behavior is active until the desired value of 20 cm is reached. Then `PullBucket` and `TurnBucket` become active at the same time to constantly decrease the boom length and decrease the bucket angle.

A standard approach for this task would have been to define important sampling points for the process, use a path planner to create the desired trajectory, and precisely follow it afterwards. This would have been acceptable in an almost disturbance free system, but not in this case where unexpected rocks can be under the surface or slack ground can lead to shaking of the excavator including its boom. Therefore, the error tolerant and adopting structure of the behavior-based control architecture is used for implicit path planning. Here multiple competing behaviors produce, if correctly coordinated, the desired excavation trajectory. The first active module is the `AdjustDepth` behavior. It suppresses the other two modules until it has reached the target depth. As long as the depth can be kept the other ones, `PullBucket` and `TurnBucket` remain active. They permanently lower the respective values they influence (length and bucket pitch angle) until a specific minimum is reached.

2.3 Behavior fusion layer

The last layer building the bridge between the system's decision layers and the lower execution layers is the *behavior fusion layer*. It is mostly influenced by its structure which is shown in figure 10. The basic idea behind this rather complex structure is that every behavior has a desired influence direction. For example, the `PullBucket` behavior wants to decrease the value. Its “positive” desire is to make the value smaller. For each dimension of the cylinder coordinate system a positive and a negative value linked to a fusion behavior exists (up, down, forward, backward, left, right) to which groups and modules can connect.

Generally can be stated that the implemented behavior structure works well for the task of shaping the environment and is capable of handling disturbances quite well considering the behavior parameters have been correctly adjusted. The next section describes the dynamic simulation of the machine which is needed to safely tune parameter values under realistic conditions.

3 Realistic dynamic simulation

A realistic mechatronic model is necessary to verify various control algorithms in advance prior their usage on a real excavator. In this work it is assumed that the chassis does not move during the excavation process. As the real drive axles are usually locked in such cases because of stability reasons, we could eliminate the modeling of suspension elements and the tires, as well. Therefore, the modeled chassis has no degrees of freedom relative to the ground, which increases simulation performance of course.

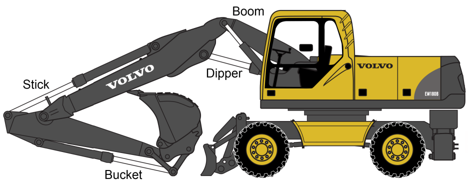

Figure 13: Schematic structure of the physical model of the excavator.

Figure 13 shows the schematic image of the modeled multi-body system having four hydraulic actuators (boom, dipper, stick and bucket) and a cabin that can be rotated along its vertical axis relative to the fixed chassis. The complex kinematic dependencies between adjacent arm segments are included in the mechanical subsystem of the model. The arm includes multiple passive revolute joints, which build four kinematic loops together with their respective hydraulic cylinders. Care had been taken to model the arising hydraulic forces in a right way, omitting algebraic loops in the mathematical formulation.

3.1 The Modelica language

In the last decades vehicle systems are subjected to even shorter development phases in order to reduce time-to-market. The Modelica language[3] and a new,

[3]Modelica: http://www.modelica.org

object-oriented modeling paradigm and its simulator tools have emerged due to this situation at the end of the 1990s. The Modelica language has been established as an intuitive way to describe physical systems containing mechanical, electrical, electronic, hydraulic, thermal, control, electric power or process-oriented subcomponents in a very effective, comfortable way. Modelica simulation environments are available commercially and free of charge, such as CATIA Systems, CyModelica, Dymola, LMS AMESim, JModelica.org, MapleSim, OpenModelica, SCICOS, SimulationX,Vertex and Wolfram SystemModeler. Because of its robust and fast solver, our simulator choice is Dymola from Dassault Systems [4].

3.2 Semi-automated modeling workflow in Modelica

The VINCENT tool has been developed to utilize the creation of multi-body kinematic structures based on hierarchical CAD assemblies [5]. By means of the STEP exchange format, more or less any CAD systems can be used as the source of the geometry. The individual CAD parts and assemblies can be assigned to separate bodies in a fast and intuitive way. A multi-body skeleton model is created by connecting these bodies with various joint definitions. There is no restriction of the kinematic structure: it can also include branches and loops, as well. Based on the CAD assembly the virtual Volvo EW180B excavator has been assembled into a multi-body structure in VINCENT. By assigning material information, the mechanical Modelica (v3.2) model of the excavator could be created in an automated workflow [JS08].

This model has twofold usage: on the one hand it has the normal role as the dynamic model of the vehicle. On the other hand - because Modelica is based on bidirectional mathematical equations - the solver can easily create an inverse model that is used for inverse kinematic (IK) calculations. This IK model is needed, because the top-level controller sees only the revolute joints, though the excavator arm is actuated using hydraulic cylinders. Figure 14 shows an important area of the whole mechanical model that has been generated. The highlighted joint-assembly in the middle is part of a new development in the aforementioned CAD2SIM workflow. This component is required to resolve each kinematic loop in an analytic way. Instead of using primitive joints that produce nonlinearities, this block formulates a better set of differential equations to be solved. The other highlighted component is the interface to the hydraulic cylinder model, which is the source of a 1D force of the internal prismatic joint. This new part of the generated model - together with the aforementioned joint-assembly component - increases the performance of the dynamic simulation.

[4] Dymola: www.dymola.com

[5] VINCENT: http://www.iff.fraunhofer.de/de/geschaeftsbereiche/virtual-engineering/vincent.html

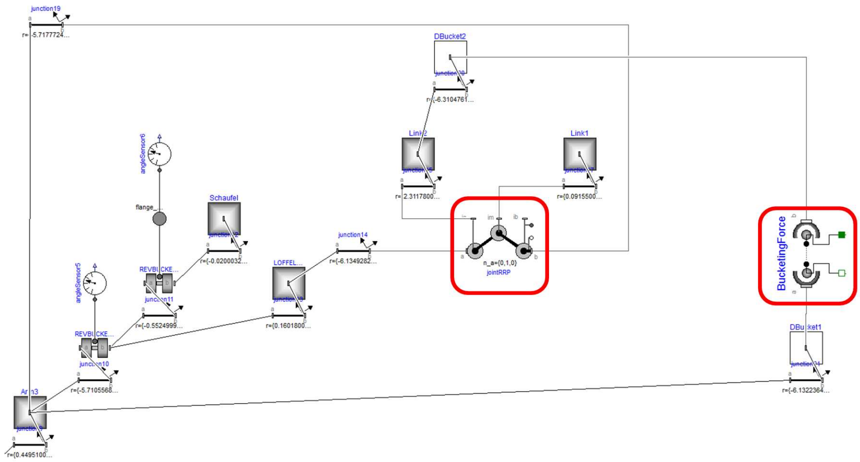

Figure 14: Part of the generated mechanical model.

3.3 Hydraulic library in Modelica

The parts of the hydraulic subsystem were modeled with elements of the Modelica Hydraulics library from Modelon[6]. This is a collection of high-performance hydraulic system components - such as pumps, motors and cylinders, restrictions and valves, hydraulic lines, lumped volumes and sensors - described in pure Modelica code. The modeling concept allows hydraulic components to be connected in an arbitrary way by drawing connection lines, no special components for splits or mergers are required.

3.4 Model of a position-controlled hydraulic actuator

This important subsystem relies heavily on the component-oriented architecture of Modelica models. It is also a good example of encapsulation of interdisciplinary interfaces towards mechanical and control-related components. Figure 15 shows the top-level diagram of this subsystem including its icon representation.

The position control of the dual-acting hydraulic cylinder is realized by means of a proportional valve. This magnetic-actuated valve has second order spool dynamics and includes nonlinearities such as magnetic hysteresis. The laminar/turbulent flow through the valve is modeled as a flow through orifices without cavitation. However, due to performance reasons, no flow forces are modeled here. The cylinder model includes two active chambers on both sides of the piston

[6]Modelon Hydraulics: `http://www.modelon.com/products/modelica-libraries/hydraulics-library`

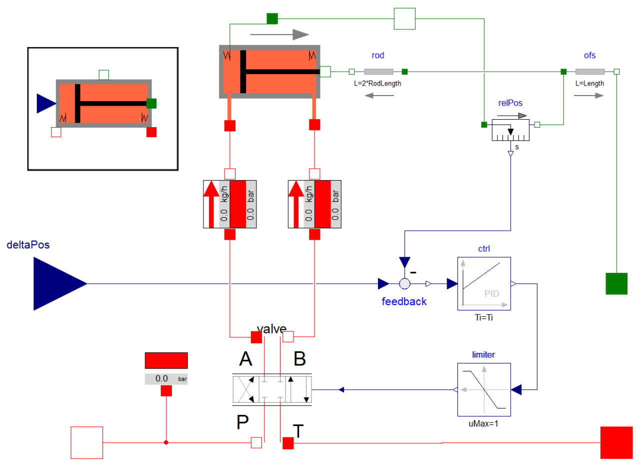

Figure 15: Modelica model of a linear hydraulic actuator

and can take external leakage on the rod side also into account. The bulk modulus (pressure-dependent compressibility) of the hydraulic oil inside the chambers uses a simplified mathematical formulation [Bea99]. The pressure difference of the chambers acts as a force on the connected mechanical parts described earlier. The mechanical losses due to friction between the cylinder wall and the piston are described by using the parametric Stribeck model [SS03]. Components of two elastic stops with stiff springs and viscous dampers are also included in each cylinder subsystem. The position control uses the feedback from the 1D mechanical position sensor and influences the servo valve over a PID block.

3.5 Integrated mechatronic model of the excavator

The complete model of the machine is depicted on Figure 16. Due to previously mentioned reasons, the behavior-based controller outputs only the reference angles between adjacent arm segments. Based on these angles the inverse-kinematic module (IK) computes the reference positions of the cylinder pistons implicitly. Due to the fact that Modelica relies on bidirectional mathematical equations, the IK subsystem can easily be derived from the semi-automatically generated mechanical subsystem. Only the causalities of the input and output connectors had to be turned around manually, thus the solver can implicitly compute the inverse model.

Similarly to the real excavator, there is a central pump model that produces the controlled flow amount of the hydraulic fluid, according to the actually measured pressure situations in the chambers of each cylinder. The great advantage of the

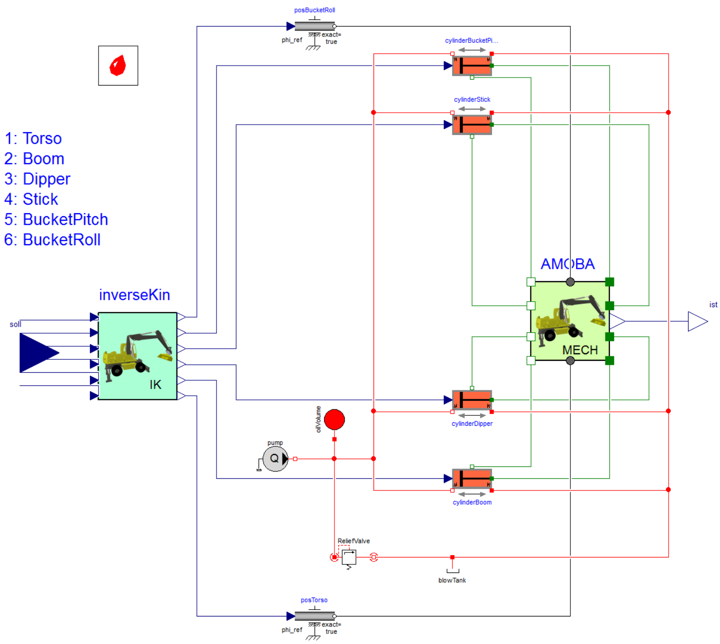

Figure 16: Modelica-based complete physical model of the excavator

Hydraulic Library in Modelica is that one central, parametric oil model is propagated implicitly to each hydraulic part of the system.

In order to stabilize the hydraulic model right at the start of the simulation, some initial pressure conditions are determined. Table 1 summarizes the computed initial states of the pressurized chambers.

CYLINDER	Fstatic [kN]	pStatic [bar]
Boom	330	134
Dipper	199	88
Stick	-58	73
Bucket	-25	49

Table 1: Preload chamber pressures of the hydraulic cylinders

A negative force means that the respective piston is initially being pulled instead of pushed.

3.6 Modelica simulation parameters

The pressure conditions within the hydraulic cylinder chambers can change very fast, because the hydraulic fluid is known to be almost incompressible. The equations of the complex bulk modulus model of the oil would imply a time constant that lays way below 1 msec. However, during a normal excavation process the mechanical parts represent much slower dynamics. These facts cause a stiff DAE, the solution of which is very intensive to compute. Therefore, for deterministic real time simulations within the Dymola environment the robust fixed-step RK solvers must always be used. We could run the experiments successfully in real-time by setting a step size of 0.5 msec using the 3rd order Runge-Kutta solver. In comparison with the behavior of the old physics simulation, we could achieve a way more realistic behavior in case of disturbances.

4 Coupling and results

After the Modelica model has been tested against robustness and accuracy, the behavior-based control system had to be co-simulated in a common time frame. The Real-Time-Interface (RTI [BKSW09]) of the Fraunhofer IFF realizes a platform independent mirrored shared-memory interface, and thereby supports the inter-process communication among software modules running in different clients.

The virtual shared memory is being transmitted over a TCP channel between the master RTI module and its registered remote slaves. The slaves use a standard operating system mechanism to access their local shared memory areas. At the moment there are Windows and Linux implementations of the RTI available.

The behavior-based controller writes the reference angles of the arm into the virtual shared memory. Besides the actual ground-interaction forces, these angles are mirrored periodically to the computer that runs the Dymola simulation. The RTI signals are exchanged at a rate of 500 Hz over a reliable TCP channel. The Modelica model is able to include C++ code (over a C wrapper), thus the shared memory signals can be read and continuously interpolated for the sake of the physical model. The Dymola simulator executes the Modelica code and computes the hydraulic forces that are acting in each time slice. The actual positions of the Modelica excavator's joints are always reflected back into the shared memory. New joint values became available for the other modules after each RTI cycle.

With the coupled systems an appropriate set of behavior parameters could be found which allow for cyclic excavation execution within the simulated environment. The same were then used to perform an automated excavation and truck loading procedure with the real machine and a fixed truck position shown in picture 17.

Figure 17: Automated test excavator Thor continuously loading a truck.

5 Conclusion and Outlook

Although the behavior-based control approach has proven to produce good control results in high dynamic environments, a problem can be the fine-tuning of its behavior parameters. Using a realistic test environment can support the verification and improvement of the autonomous excavation algorithms and their parameter sets. Therefore, the removal of large and fatal errors are cheap and safe compared to tests with the real machine. Furthermore, the results show that a more realistic behavior of the simulated machine can decrease the amount of adjustment needed when switching to the real platform.

The next steps of the project include extending the behavior network to perform more complex landscaping operations like trench excavation or material transport including autonomous driving. As those operations are closely linked to the environment, an additional behavior safety layer has to be installed which protects humans, machines, and structures from being hit by the machine. So far the dumping position inside the truck is fixed. Developing more complex algorithms for environment perception including dynamic obstacle detection and classification will increase safety and finally close the gap to a fully autonomous construction machine.

Acknowledgment The work at hand was funded by the Federal Ministry of Research and Education within the ViERforES project and supported by Volvo Construction Equipment in Konz, Germany and the Robot Makers GmbH.

References

[ASB12] Christopher Armbrust, Daniel Schmidt, and Karsten Berns. Generating Behaviour Networks from Finite-State Machines. In *Proceedings of the German Conference on Robotics (Robotik)*, May 22–25 2012.

[Bea99] Peter Beater. *Entwurf hydraulischer Maschinen: Modellbildung, Stabilitätsanalyse und Simulation hydrostatischer Antriebe und Steuerungen*. Springer Verlag, 1999.

[BKSW09] T Böhme, M Kennel, M Schumann, and A Winge. Automatisierte Erstellung domänenübergreifender Modelle und echtzeitfähige Kopplung von Simulation, Visualisierung und realen Steuerungen. In *Paderborner Workshop in der Produktentstehung*, number 252, pages 155–170, 2009.

[BWB07] Tim Braun, Jens Wettach, and Karsten Berns. A Customizable, Multi-Host Simulation and Visualization Framework for Robot Applications. In *13th International Conference on Advanced Robotics (ICAR07)*, pages 1105–1110, Jeju, Korea, August 21-24 2007.

[HSB+10] Carsten Hillenbrand, Daniel Schmidt, Nureddin Bennett, Peter Bach, Karsten Berns, and Christian Schindler. Feasibility Study for the Automation of Commercial Vehicles on the Example of a Mobile Excavator. In Karsten Berns, Christian Schindler, Klaus Dreßler, Barbara Jörg, Ralf Kalmar, and Jochen Hirth, editors, *Commercial Vehicle Technology 2010 - Proceedings of the 1st Commercial Vehicle Technology Symposium (CVT 2010)*, pages 22–31, Kaiserslautern, Germany, March 16-18 2010. Shaker Verlag. ISBN 978-3-8322-9040-5.

[JS08] Tamas Juhasz and Ulrich Schmucker. CAD to SIM: CAD model conversion for Dymola-based mechatronic simulation. In *Computer Modeling and Simulation, 2008. UKSIM 2008. Tenth International Conference on*, pages 289–294. IEEE, 2008.

[PLB07] Martin Proetzsch, Tobias Luksch, and Karsten Berns. The Behaviour-Based Control Architecture iB2C for Complex Robotic Systems. In *Proceedings of the 30th Annual German Conference on Artificial Intelligence (KI)*, pages 494–497, Osnabrück, Germany, September 10-13 2007.

[Pro10] Martin Proetzsch. *Development Process for Complex Behavior-Based Robot Control Systems*. RRLab Dissertations. Verlag Dr. Hut, 2010. ISBN: 978-3-86853-626-3.

[RFB13] Max Reichardt, Tobias Föhst, and Karsten Berns. On Software Quality-motivated Design of a Real-time Framework for Complex Robot Control Systems. In *Proceedings of the 7th International Workshop on Software Quality and Maintainability (SQM), in conjunction with the 17th European Conference on Software Maintenance and Reengineering (CSMR)*, Genoa, Italy, March 5 2013.

[SPB10] Daniel Schmidt, Martin Proetzsch, and Karsten Berns. Simulation and Control of an Autonomous Bucket Excavator for Landscaping Tasks. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5108–5113, Anchorage, Alaska, USA, May 3-8 2010.

[SS03] Richard Stribeck and M Schröter. *Die wesentlichen Eigenschaften der Gleit-und Rollenlager*. Springer, 1903.

[WSB10] Jens Wettach, Daniel Schmidt, and Karsten Berns. Simulating Vehicle Kinematics with SimVis3D and Newton. In *2nd International Conference on Simulation, Modeling and Programming for Autonomous Robots*, Darmstadt, Germany, November 15-18 2010.