Model-based Development of an Automated and Remotely-Controlled Demolition Excavator

Christian Richter, Masoud Shakoorianfard, Volker Waurich, Frank Will

Professur für Baumaschinen, TU Dresden, Münchner Platz 3, 01069 Dresden, Germany
{christian.richter1, masoud.shakoorianfard, volker.waurich, frank.will}@tu-dresden.de

SNE 33(3), 2023, 111-116, DOI: 10.11128/sne.33.tn.10652
Selected ASIM SST 2022 Postconf. Publication: 2023-08-15;
Received Revised Improved: 2023-10-18; Accepted: 2023-10-22
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. This paper describes the application of a 3D-simulation model to develop the control system of a demolition excavator. In order to simplify the system development and provide means for early prototype validation, a simulation-based development workflow is presented. A 3D model of the excavator, a demolishable wall, a virtual LiDAR sensor as well as CAN and Ethernet communication interfaces are used to support the development and testing of different software components as well as a test setup for automation strategies.

Introduction

In the case of nearby settlements and infrastructure, the demolition of natural draft cooling towers is restricted to non-explosive demolition techniques. One innovative approach is the usage of a modified excavator that sits on the wall structure and demolishes the wall piece by piece with a pulverizer.Figure 1 shows the excavator during operation. The excavator drives around the wall to demolish a 2 m high ring segment during each turn and reduce the height of the cooling tower up to a certain height. After this, the cooling tower can be collapsed in a controlled fashion by weakening the structure.

Figure 1: Demolition of the cooling tower in Mühlheim-Kährlich with the demolition excavator.

The excavator is currently operated from a platform on top of the cooling tower with a handheld remote control by sight, as depicted in Figure 2. Former attempts of controlling the excavator by video streams have been neglected because of image latency and poor depth perception. Due to these inefficient and dangerous operation conditions, an improved approach for a tele-remote workplace must be designed and implemented. In order to further facilitate the operation, suitable sub-tasks (e.g., the demolition of a wall segment) must be automated. The authors developed a simulation model of the machine and the process in order to support the product development even in the early stages.

This chapter gives an introduction to the basic machine setup and the corresponding simulation model. Chapter 1 describes the virtual LiDAR sensor and the point cloud processing.

Chapter 2 presents the abilities to use the simulation model to validate CAN-communication and to act as a Hardware-in-the-Loop (HiL)-simulation. Chapter 3 outlines the automation strategy and how this can be tested with the simulation model. Chapter 4 describes the developed approach for digital twin-based teleoperation. Chapter 5 closes the paper with a conclusion.

Figure 2: Conventional operation of the excavator with a handheld remote control from a platform on top of the cooling tower.

1 Machine and Simulation Setup

1.1 Equipment and Sensor Setup

The excavator consists of a frame structure including two driven wheel flanges that drive the whole machine on the edge of the cooling tower. On the frame, the working attachment of a conventional excavator is installed. During demolition work, the frame is clamped on the wall by two runners on both sides. The excavator has an adjustable boom and has four cylinders to adjust the height and range of the tool. The working attachment can be rotated by a slewing cylinder. The demolition tool is a pulverizer e.g., a massive gripper that cuts out a segment of the reinforced concrete. The tool itself can be rotated, opened, and closed.

In order to sense the working attachment, the main boom, the adjustable boom, the arm, and the tool coupler are equipped with 1D inclination sensors in order to compute the joint angles. The length of the slewing cylinder is sensed with a draw-wire sensor.

An encoder inside the rotary feedthrough of the pulverizer captures the rotation angle. Pressure sensors determine the opening and closing state of the tool. The pitch and roll angle of the machine is captured by a 2D-inclination sensor. With this sensor setup, a complete kinematic representation is possible by calculating the forward kinematics.

A 3D-LiDAR sensor is used to capture the working process, i.e. the wall during demolition. A LiDAR is a laser-based distance sensor that is able to record a 3D point cloud of the surrounding. The LiDAR is attached to the frame and faces the inner face of the nearby wall that is in reach of the excavator. Figure 3 illustrates the sensors to sense the working attachment and the wall surface.

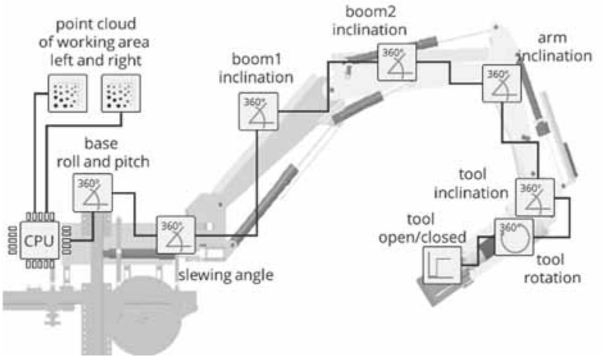

Figure 3: Sensor setup of the demolishion excavator.

1.2 Simulation Setup

At the beginning of the project, the excavator was modeled as a CAD-Model. The excavator working attachment itself had to be reverse-engineered whereas the manufacturer has provided a CAD-Model of the frame. Based on the geometric information, a simulation model has been developed that serves the following purposes:

- Kinematic simulation to test reachability
- 3D-collision detection
- Simulation of a destructible wall volume
- Emulation of LiDAR sensors to retrieve virtual point cloud data of the machine and the variable wall geometry
- Software and hardware interfaces to enable HiL, SiL, and Co-simulation
- An appealing 3D visualization to reuse the model for a tele-remote operation
- Realtime Feedback
- Scriptable simulations for automated testing

As a simulation framework, the Unity Engine [1] was chosen since it provides an efficient 3D environment with a built-in physics engine, which is optimized for real-time applications. Compared to domain-specific simulation environments, it is relatively easy to implement special-purpose features such as specific virtual sensors or variable 3D geometries.

The serial kinematic of the machine is modeled as a kinematic tree that is visualized with reduced CAD models, as can be seen in Figure 4. The inputs to drive the model are the angles of the revolute joints that connect the rigid bodies of the working attachment. The visualization of hydraulic cylinders and rods moves accordingly. A kinetic multi-body simulation or hydraulic system simulation is not implemented as this is unnecessary for the application. A Co-Simulation to e.g., a Modelica-based model via an FMU or interprocess communication would be possible as done by the authors in previous projects [2].

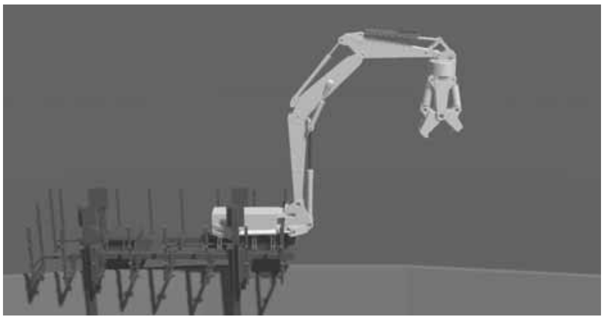

Figure 4: Visualization of the kinematic excavator model.

2 Virtual LiDAR and Wall Sensing

To detect the wall segment in front of the excavator, a Livox Mid-70 LiDAR is used [4]. In general, a LiDAR is an array of laser-based distance measurement sensors that either rotate around an axis (rotary LiDAR) or are arranged in a matrix (flash LiDAR).

The used LiDAR sensor is somehow unique in its scanning pattern. It features a non-repetitive scanning technology in the shape of a hypotrochoid (i.e., a small circle rolls inside a bigger one). Therefore, the coverage inside the field of view (FOV) increases over time.

In the simulated sensor model, the FOV, rotation speed, number of rays per shot, ratio of the circle radiuses are adjustable. In addition to the simulated LiDAR, a point cloud accumulator is implemented, that aggregates the measured point clouds over time.

Figure 5 depicts the scanning patterns of the rotating hypotrochoid in the simulation and the accumulated point clouds over time.

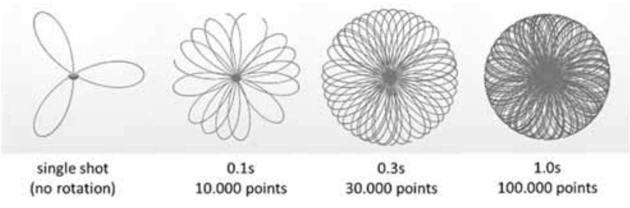

Figure 5: Scanning Pattern and accumulated point cloud over time.

The simulated sensor is able to provide the same point rate of 100.000 points/s as the real sensor on an i7-3930K @3.20GHz.

The simulation of the LiDAR utilizes the built-in physics engine in the Unity software. Each distance measurement is done by calling the built-in function:

```
bool UnityEngine.Physics.Raycast(
Vector3 origin,
Vector3 direction,
out RaycastHit hit,
float maxDistance);
```

A Raycast outputs a RaycastHit object if the ray hits a so-called GameObject that has an instance of a collider-class. Therefore, every object that must be detected by the LiDAR needs to provide this ability.

There are different collider classes available, both for geometric primitives (cubes, spheres, to name but two) and mesh-based colliders. For the detection of the deforming wall, an adaptive mesh-collider is needed.

For the detection of the working attachment of the excavator, geometric primitives as bounding boxes of the complex geometry are sufficient. The base machine and the cooling tower do not feature a Collider object because these areas will be cropped out of the point clouds.

Moreover, considering the interaction between the wall and the tool as the most important part of the simulation, it is required to differentiate the points that corresponded with the wall and tool from the other objects in the scene. This step, which is called Segmentation, is done by taking advantage of the kinematic information provided by sensors and accordingly moving colliders of the working attachment of the excavator, resulting in removing the points that corresponded with the machine objects except for the tool. Figure 6 shows the simulated point cloud on the wall segment and the tool.

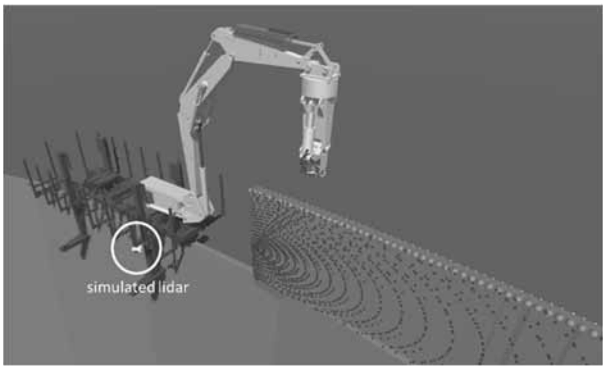

Figure 6: Simulated LiDAR with a wall segment that implements a mesh collider.

In order to simulate the demolition process of the wall, parts of the wall have to be removed. Therefore, a procedural and time-varying mesh of the wall segment is implemented. The same mesh is used, both for visualization and to serve as a mesh collider for the simulated LiDAR.

The procedural wall model is initialized with a fixed height and depth at a specified position. At the position of the excavator tool, a cubic collider is attached, that has the size of the pulverizer chamber. If a wall vertex is hit by the tool collider, the top vertices of the mesh are shifted vertically to the bounding box of the collider. Figure 7 illustrates the deformation of the wall mesh by shifting the top vertices.

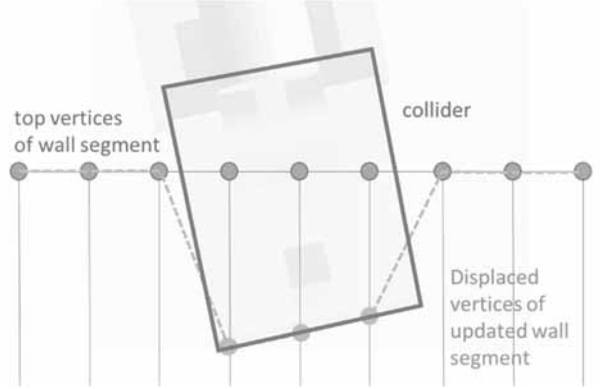

Figure 7: Wall deformation by the excavator tool.

The result of a deformed wall segment reflecting the raycasts by the emulated LiDAR can be seen in Figure 8.

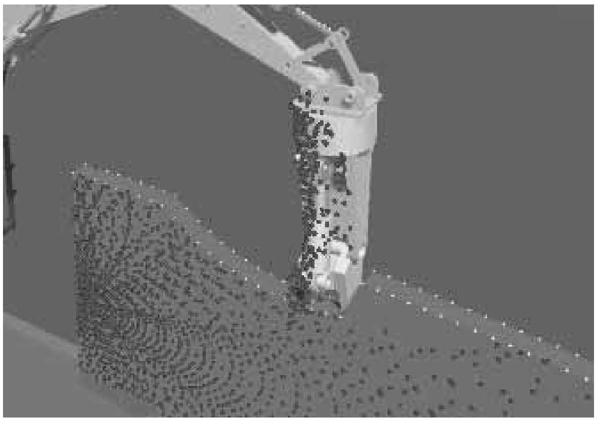

Figure 8: Deformed wall segment with updated mesh collider to reflect the raycasts of the LiDAR.

The resulting point cloud data of the emulated LiDAR sensor is provided as a simple array of Cartesian coordinates referenced in the sensor coordinate system to be used for interprocess communication via shared memory. Additionally, a network interface is provided as well, featuring the basic parts of the official Livox-communication protocol, resulting in the possibility to establish an emulated network stream to communicate with the commonly used tools officially provided by the manufacturer, including the official Livox-SDK, livox-ros-driver and livox-ros2-driver ROS packages, and the Livox Viewer software.

The point cloud data provided by the simulation must be processed as if it is provided by the real sensor installed on the machine. To do so, an industrial PC (IPC) and the ROS2 framework are utilized as the processing unit.

This unit provides the required data for the visualization unit in the tele-remote-operation mode and implements the overall automation strategy in automation mode, both with the help of the processed data.

In order to have a more realistic data communication in the simulation of the process, the simulation PC and the IPC are networked, and the point cloud data is sent and received over the implemented emulated network stream. An instance of the simulation in Unity, ROS2, and Livox Viewer can be seen in Figure 9 , which shows that the point cloud data is successfully sent from the Unity (simulation PC), received by ROS2 (IPC), and optionally visualized by Livox Viewer (IPC) which is the official software provided by the manufacturer to use with real sensors.

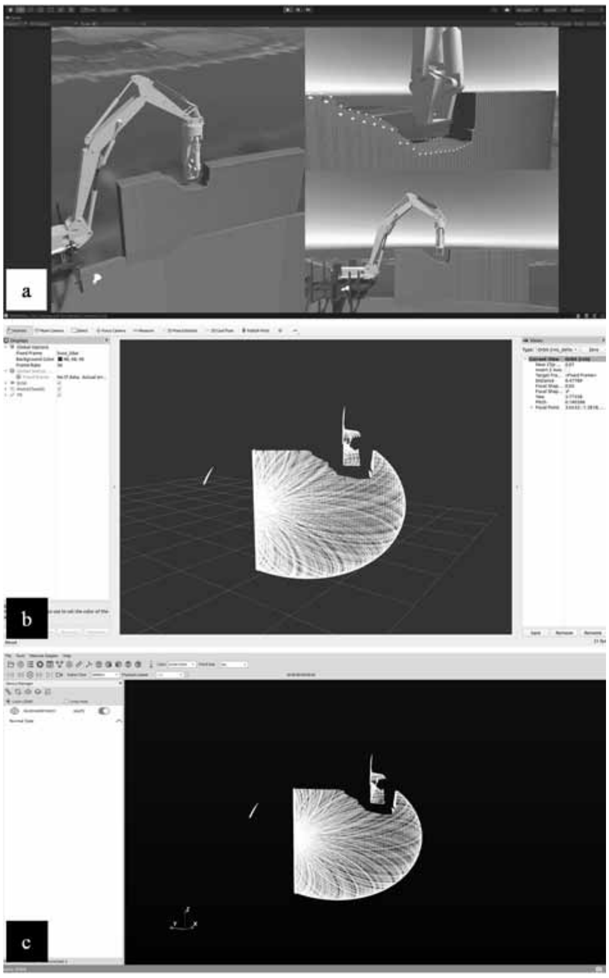

Figure 9: An instance of the simulation environment in: a) Unity software on the simulation PC, b) ROS2 framework on IPC, c) Livox Viewer on IPC. The point cloud data is accumulated for 3 seconds.

3 CAN-Communciation

The real machine is equipped with two CAN bus systems. The first CAN bus features the sensor data of the angle sensors. All the sensors attached to the working equipment are CANOpen based [5].

With the help of a USB-to-CAN interface and the corresponding API, the simulation model feeds its sensor data with a real CAN bus. A subset of the CANOpen-slave functionality is implemented in order to provide the same CAN-bus protocol as the real sensors in the expected operation mode.

The second CAN bus serves as an interface between the ECU to control the hydraulic system and the IPC that processes the point-cloud data and implements the overall automation strategy. A custom CAN protocol has been developed to send the data of the desired trajectory from the IPC to the ECU and to transmit status and control messages between the automation authority and the control ECU.

In order to test the CAN communication on the sensor bus as well as to test and modify the custom CAN protocol, the simulation model which implements the emulated CAN interface is a helpful tool.

The emulation of the CAN communication which is fed by a comprehensive system simulation provides several advantages:

- Software development and interface specification can be done without the need of the real system.
- The simulation can be distributed several times whereas the real machine exists only once.
- Simulation-based but realistic data is a feasible way to provide a vast amount of test-cases for test-driven development.
- A simulator can facilitate the reconstruction of dangerous or complex situations.

4 Tele-Remote Operation

The machine is supposed to be controlled remotely. In previous applications, the machine was controlled with a handheld device with a direct line of sight. A camera-based solution to control the excavator from a control cabin has been rejected by the operators because of the high delay of the video streaming which makes the tele-operation inefficient.

Besides the latency issue, camera-based tele-remote control solutions have several other drawbacks:

- The 2D image of a video hinders the depth perception of the operation which is important particularly when moving the working attachment of an excavator
- The camera position is fixed and does not allow a moving point of view.
- Optical cameras only work in good light conditions and in the absence of fog and dust.
- Compression and Decompression of digital video signals introduces additional delay and requires dedicated hardware
- Data throughput increases with resolution and number of cameras

In order to provide an efficient solution for tele-remote operation, a digital twin approach is implemented. All sensor signals are used to visualize a 3D model of the machine. The amount of data to represent all degrees of freedom of the excavator is comparatively low (9 × Int16 at 50Hz). As the raw point cloud data stream requires a performant wireless link, a post-processing algorithm is implemented on the machine’s IPC. This reduces the throughput to transmit the point data drastically (300 × Int16 at 10Hz).

The basic idea to reduce the point cloud data is to extract only the relevant information to control the demolishing process, i.e. the top edge of the remaining wall. Different filter algorithms are applied to the point cloud to compute an array of points in a discrete spacing to indicate the top-right edge of the wall. Figure 10 shows the result of the edge detection algorithm based on the simulated LiDAR data.

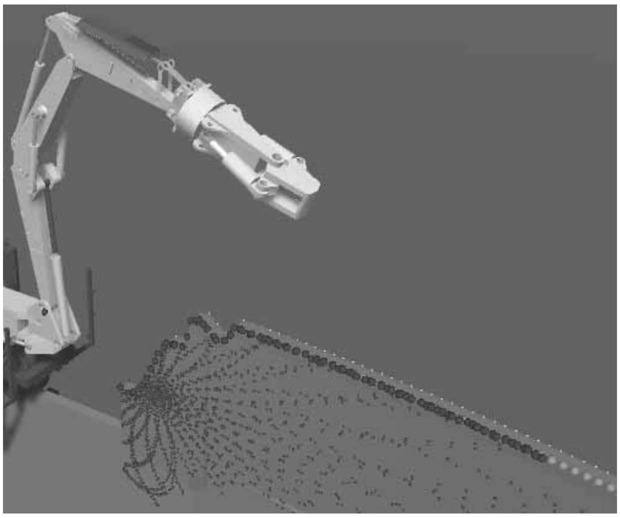

Figure 10: Edge detection for the simulated wall under demolition. The red spheres indicate the detected points of the top-right wall edge.

To provide a real-time tele-remote visualization, the 3D visualization of the simulator is reused to visualize the machine and point cloud data. Based on the detected points on the top-right edge of the wall, a mesh is created that resembles the real wall. Considering the fixed wall thickness and using radial extrapolation of the given points, a textured volume can be constructed to visualize the wall segment within reach of the excavator.

This approach to visualize this particular machine-process interaction has several advantages:

- The overall amount of data to visualize all relevant information is relatively low compared to multiple high-definition video streams. This reduces delay.
- The virtual camera that is used to render the scene can be moved by the user during operation or several virtual cameras can be used to visualize the scene from different locations.
- Additional information can be computed and displayed in the scene, of which position coordinates of the end effector, limits of the working area, recommendations to align the tool efficiently, to name but a few, are of great importance.
- To enhance depth perception of the scene, the visualization could be rendered for a VR-Headset. As this is not feasible for long operations, a head tracking-based camera movement has been tested to align the view with the head movement of the operator and to create motion parallax.

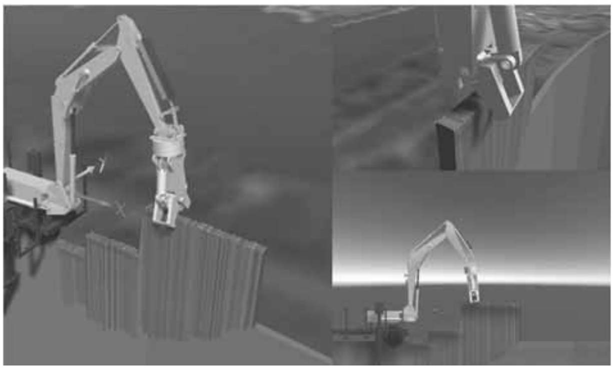

Figure 11: Visualization of real machine data and real sensor data for tele-remote operation with 3 different camera positions. The left bottom image shows a side view with an orthographic projection.

5 Conclusion

This paper describes the components of a 3D-simulation environment for a demolition excavator, which can also serve as a visualization tool for a real-time tele-remote operation. The simulation comprises a 3D kinematic model for the excavator, an emulated Livox LiDAR, CAN bus-based interfaces for Hardware-in-the-loop applications, and a deformable wall segment that resembles a demolishable wall. The simulation is a helpful tool to develop and test the communication protocol, the software for automation and machine control as well as the point cloud processing. The majority of the simulation environment has been reused to serve as a real-time visualization tool for a tele-remote operation that has several advantages compared to a conventional camera-based view of the process.

In the ongoing project, a partly automation of the demolition process has to be developed. Therefore, a trajectory planner and a process strategy have to be implemented and tested. As the real demolition of a wall is a laborious and costly procedure, the simulation-based development of the process routine is an invaluable method.

Acknowledgement

This Project is supported by the Federal Ministry for Economic Affairs and Energy (BMWi) on the basis of a decision by the German Bundestag.

References

[1] Unity Technologies, 14 10 2021. [Online]. Available: www.unity.com

[2] Waurich V, Weber J. Interactive FMU-Based Visualization for an Early Design Experience. In *Proceedings of the 12th international Modelica Conference*, Prague, 2017.

[3] Livox Technology, [Online]. Available: https://www.livoxtech.com/de/mid-70. [Access am 15 10 2021]

[4] CiA, „www.can-cia.org," [Online]. Available: https://www.can-cia.org/canopen/. [Access 26 10 2021]

[5] V. Waurich V. Simulation der automatisierten Schaufelbefüllung eines Radladers mit Modelica. In *Tagungsband ASim Workshop STS/GMMS*, Heilbronn, 2018