2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Development and Applications of a Simulation Framework for a Wall-Climbing Robot

Daniel Schmidt and Karsten Berns

Abstract— **In the range of robotics, simulation frameworks are very common. They are used to perform tests of algorithms, for optimization and to analyze the system behavior in situations, which would be hazardous or difficult for the real robot. This paper addresses a simulation framework related to a wall-climbing robot using negative pressure adhesion in combination with an omnidirectional drive system. The key aspect of this simulation is the adhesion system consisting of simulated pressure sensors, valves between adhesion chambers and vacuum reservoir and a simulated adaptive sealing proofing the vacuum chambers towards ambient air. The interaction of environmental features (e. g. surface characteristics like roughness or special geometries) and the vacuum chambers of the robot is handled by a thermodynamic model providing the basis for airflow simulation between the virtual surface and the robot. These features facilitate the validation of control algorithms and closed-loop controllers in realtime.**

I. INTRODUCTION

Nowadays, robot simulation frameworks are used all over the world to test algorithms and to analyze the system behavior in certain situations without executing them on the real machine. This reduces testing time, avoids unnecessary stress on the hardware and enables developers to analyze a system without danger. Current frameworks facilitate a physically realistic 3D simulation of all kinds of robots from walking machines up to flying and underwater robots. But so far, support for the simulation of negative pressure adhesion in the context of mobile climbing robots is still missing.

This paper presents the simulation of CROMSCI[1] which is a complex wheel-based climbing robot with negative pressure adhesion (figure 1). The simulation of the interaction between robot and environment – especially between adhesion system and surface – allows extensive tests of safety strategies and closed-loop controllers without the use of the real machine. This becomes even more important since real-world experiments come with a large stress on the hardware which is not robust enough for long-term tests.

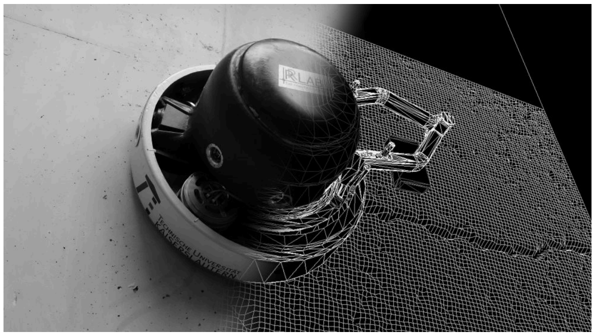


Fig. 1. Composition of real robot CROMSCI (left) and 3D simulation (right).

The prototype CROMSCI has been developed to perform inspection tasks on large concrete buildings area-wide and semi-autonomously [1]. Its key aspects are seven individual adhesion chambers generating downforces via negative pressure, inflatable sliding sealings for leak-tightness, built-in suction engines evacuating a large vacuum reservoir and three steerable driven wheels without suspension. It has a diameter of 80 cm, a height of 40 cm, a weight of about 50 kg, a payload of about 10 kg and a maximal velocity of 9.81 m/min. Its adhesion system generally works in a range of -50 mBar to -100 mBar compared to ambient pressure to generate a downforce of about 2 000 N. All these aspects have been considered for the simulation tool.

Next section II gives a brief overview of existing simulation frameworks. The general concept of the simulation is presented in section III. Section IV describes the SIMVIS3D framework used for 3D simulation and rendering. Afterwards, section V presents the thermodynamic model and leakage simulation. The physical interaction is handled in section VI. Section VII shows some applications, conclusions follow in section VIII.

II. RELATED WORK

In robotics, several different simulation tools exist. Most of these frameworks focus on realistic sensor data and physical interaction between robot and environment. An example is *Gazebo* which supports the well-established robot control framework ROS [2] and uses ODE or *Bullet* physic engine. Also *SimRobot* is based on ODE and focuses on a realistic simulation of camera disturbances and actuator friction [3]. USARSim [4] works on the basis of the *Unreal* game engine and *Karma* physics. *Webots* [5] focuses on research robots (e. g. Aibo, Pioneer or Kuka youBot). V-REP is a feature-rich 3D simulator for wheel-driven, walking and flying robots using either ODE or *Bullet* [6].

Although all these tools are sufficient for collision detection and pick-and-place tasks, they do not support negative pressure adhesion as in the case of CROMSCI. Even the support for particle systems in *Vortex* and V-REP is not suitable to simulate airflow since this is a thermodynamic problem and not related to particle impulses. In the context of climbing robots there only exist individual solutions for computing the adhesion forces: [7] uses *Simulink* to simulate

D. Schmidt and K. Berns are with the Robotics Research Lab, University of Kaiserslautern, 67663 Kaiserslautern, Germany {dschmidt,berns}@cs.uni-kl.de

[1] http://agrosy.cs.uni-kl.de/cromsci/

978-1-4673-6357-0/13/$31.00 ©2013 IEEE

the drive system of the Alicia^3 robot, but only with a simplified model for the wheel-wall contact and none for the suction cups. [8] calculates the contact forces between feet and adhesive objects via continuous mechanics theory as plugin for *Webots*.

The main contribution of this paper is the combination of a thermodynamic model of the negative pressure system, environmentally conditioned sealing leakages and the drive system affected by downforce and wheel slippage.

III. CONCEPT

The complete control and simulation software of CROMSCI has been developed using the MCA framework^2. The embedded simulation itself consists of modular components and connections as depicted in figure 2 an is divided into four logical blocks shown as gray boxes:

- *Actuator simulation* simulates ideal actuators for the omnidirectional drive system, the tool center point, suction engines and chamber valves for adhesion control. The desired commands are executed by the actuators with a certain delay and within individual limits in velocity, acceleration or volume stream, depending on the type of actuator.

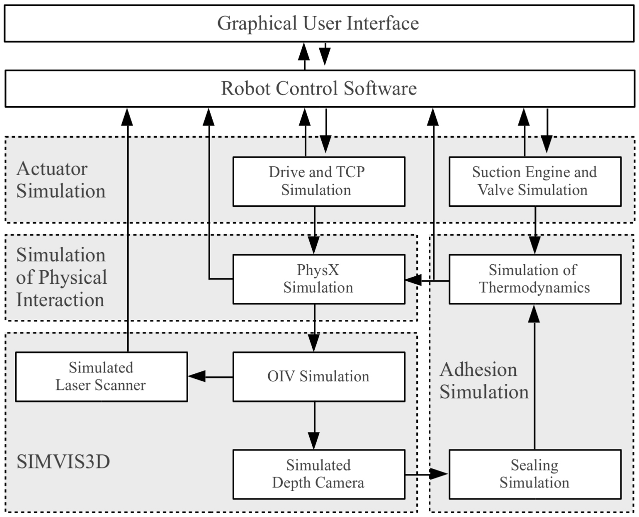


Fig. 2. Components of the simulation framework (gray).

- *Simulation of physical interaction* is optional and can apply physics engines like *PhysX* or *Newton Game Dynamics* with suitable collision and interaction models. It uses the wheel states (steering angle and velocity) to calculate wheel slippage and the resulting robot pose – and finally collisions. Without physics, the odometry of the robot has to be determined by hand for subsequent simulation steps.
- SimVis3D contains the visual representation of the scene. It allows it to simulate environmental sensors (e. g. cameras or laser range sensors) and to visualize the simulated robot scene. Additionally, SimVis3D is used to extract a depth image of the surface below CROMSCI for sealing simulation.
- *Adhesion simulation* contains a thermodynamic model to calculate the progress of airflows and chamber pressures. Therefore, the simulated leakage values as well as the states of suction engines and valves are used to calculate an overall adhesion force (robot downforce) which works contrary to gravity in the physical simulation, if used.

The modular structure of this simulation framework enables the developer to replace single elements by real hardware components. Therefore, it is e. g. possible to test the functionality of the real chamber valves via simulated pressure values. In total, one can enable or disable the real drive, manipulator, valves, pressure sensors and laser ranger.

IV. 3D SIMULATION AND VISUALIZATION

The simulation framework of CROMSCI uses SimVis3D for simulating and visualizing objects in a 3D scene. This tool is based on the well-established *OpenInventor* standard (OIV) using Coin3D as open source implementation, which represents a high level abstraction of *OpenGL*. The core structure is a scene graph which is set up via a XML description and arranges the elements in the simulated environment and manages scene changes like object positions at runtime. Scene elements are visual 3D bodies or modifiers like transformation nodes.

The simulated environment of CROMSCI contains different components for simulation as well as for decoration. Figure 3 shows the basic wireframe model of the robot and its surroundings (trees, bridge elements, ground) and a rendered view including trees, textured surfaces and backgrounds. SimVis3D allows it to move and rotate elements like wheels and manipulator arbitrarily in the scene and visualize the result via a graphical user interface (GUI) in realtime to observe the robot inside the simulation.

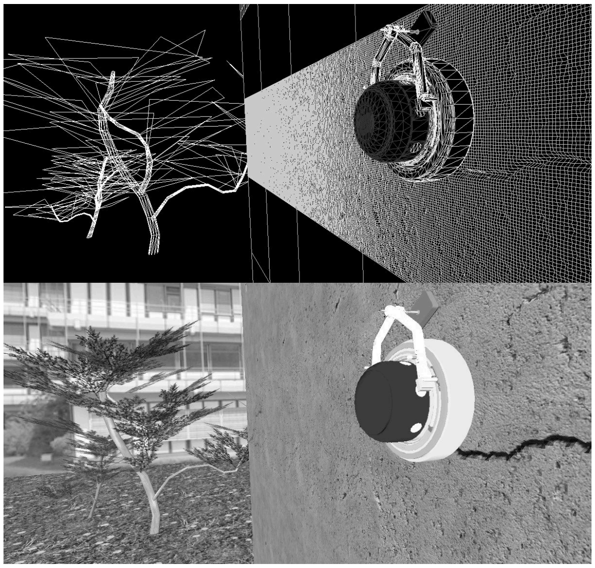


Fig. 3. Wireframe model of CROMSCI and a bridge including a band of rough structured surfaces (top) and rendered visualization (bottom).

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

The simulated bridge pylon consists of single cubes with an edge length of 2 m and one structured face. Therefore, it is possible to insert a cube at any pose via the XML scene file and to create different test conditions easily. Up to now more than 50 different cubes with surface structure, cracks, gaps, holes and protrude structures have been created via *Blender* and integrated into the simulation as *OpenInventor* files.

V. SIMULATION OF NEGATIVE PRESSURE ADHESION

The most significant components of the simulation framework are those which are responsible for the modeling of the adhesion system. This includes a calculation of the airflows between negative pressure chambers and ambient air as well as a determination of chamber leakages based on surface characteristics. Figure 4 gives a structural view on the adhesion system. The top view shows the six outer and one inner adhesion chambers. The side view below depicts the mass flows caused by the suction engines $\dot{m}_A$, valves $\dot{m}_{V_i}$ and sealing leakages $\dot{m}_{L_{ij}}$. Beside basic leakages, the leak-tightness decreases in cases of cracks or surface irregularities.

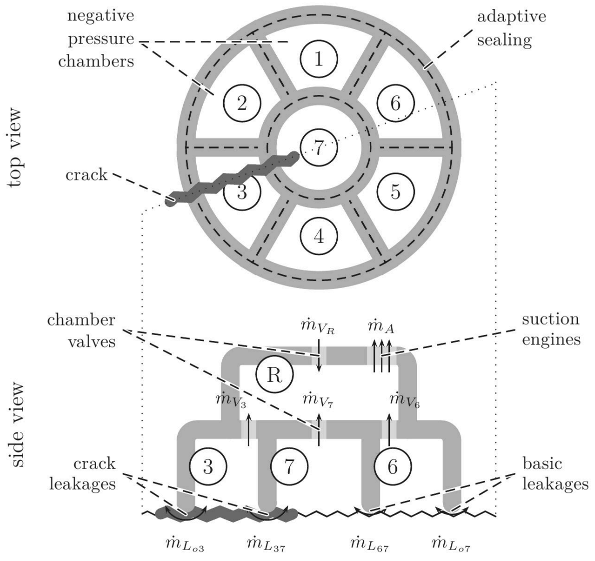


Fig. 4. Structural view of adhesion chambers 1 to 7, reservoir R and an exemplary crack below chambers 3 and 7.

The interaction model between adhesion system and environment is based on the first fundamental theorem of thermodynamics and Bernoulli's equation as described in [9]. Here, a mass flow $\dot{m}_{ij}$ between two volumes i and j depends on pressure values, p_i and p_j, the area of airflow A_{ij} and the density of air ρ_{air} (equation 1):

$$\dot{m}_{ij} = sgn(p_i - p_j) \cdot A_{ij} \cdot \sqrt{2 \cdot \rho_{air} \cdot |p_i - p_j|} \tag{1}$$

Now, the pressure change $\dot{p}_i$ of volume i can be calculated (equation 2) using the adiabatic index of air κ_{air}, the specific gas constant R_{air}, temperature T_{air} and volume V_i. The sum goes over all other volumes k including the ambient air. If there is no direct connection between two volumes, the corresponding massflow is zero.

$$\dot{p}_i = \frac{\kappa_{air} \cdot R_{air} \cdot T_{air}}{V_i} \sum_k \dot{m}_{ik} \tag{2}$$

Given the pressure changes and a starting pressure, the total affecting downforce F and the point of action $\vec{P}_F$ can be calculated (equations 3 and 4). These equations use the ambient air pressure p_o in conjunction with chamber suction area A_i and chamber center point $\vec{P}_i$.

$$F = \sum_{i=1}^{7} F_i = \sum_{i=1}^{7} \left((p_o - p_i) \cdot A_i \right) \tag{3}$$

$$\vec{P}_F = \sum_{i=1}^{7} \left(\vec{P}_i \cdot \frac{(p_o - p_i) \cdot A_i}{F} \right) \tag{4}$$

The last piece of the puzzle is the calculation of sealing leakages [10], which are needed to determine e. g. the airflow between adhesion chamber and ambient air like $\dot{m}_{L_{o3}}$ in figure 4. A simplified sealing model is used which approximates its adaptability to surface irregularities. An orthographic camera (resolution 256×256 pixel) is embedded into simulation via SIMVIS3D to take a picture (size $80 \times 80\,\text{cm}^2$) of the ground below the robot chassis. The z-buffer of this camera is used to extract height information. As depicted in figure 5, the surface height is shown as grayscale image including an overlay of sealing edges (black and white) and wheel units. The observed height range is ±5 cm resulting in a z-resolution of 0.39 mm of the simulated camera.

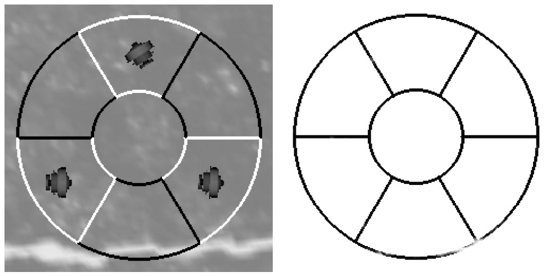


Fig. 5. Depth image, sealing areas (left) and leakages (right).

In a successional step the adaptability of the sealing to this height distribution has to be simulated. But, the real interaction between sealing and surface is to complex to emulate here. Therefore, an approximation is used which applies a maximum height difference between neighbor sealing pixels. Depending on this difference the height values of all sealing pixels are calculated:

```
determine height of all surface pixels
set all sealing pixel heights to surface
  height values below
repeat
  reset changed flag
  for each sealing pixel
    if difference to highest neighbor pixel
      larger than maximum height difference
```

```
        set current sealing pixel height to
        highest one minus maximum difference
        set changed flag
  until not changed
```

The result of this algorithm is illustrated in figure 5 (right). Black pixels indicate leak tight areas whereas gray to white values show permeable areas with a larger gap between ground and sealing. The total amount of indentation and extrusion is limited considering the capabilities of the real robot. In the present case the real sealing can handle structures of up to $\pm 5\,\mathrm{mm}$ (so deeper crack cannot be proofed, higher protrusions cannot be passed). The final leakage L_i (equation 5) of sealing section S_i is based on the length of the sealing segment l_i, the basic leakage L_{basic} caused by the sliding coating and an multiplication factor κ for debugging purposes.

$$L_i = l_i \cdot \left(L_{\text{basic}} + \kappa \cdot \sum_{j \in S_i} \frac{h_s(j) - h_g(j)}{|S_i|} \right) \tag{5}$$

The sum goes over all pixels j lying on sealing segment S_i and calculates the height difference between sealing $h_s(j)$ and ground pixel $h_g(j)$. $|S_i|$ is the total amount of pixels of the sealing segment. Finally, this leakage is used as connection area between the volumes to calculate the pressure change $\dot{p}_i$ according to equation 2.

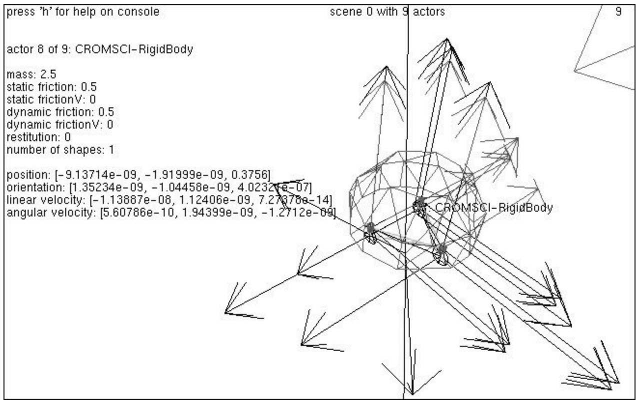
VI. PHYSICAL INTERACTION OF ROBOT AND ENVIRONMENT

For a realistic system behavior, all actuators are simulated with certain limitations of acceleration, velocity or position. These parameters depend on the real hardware settings and cause e. g. a delayed execution of certain motions like drive steering or valve positioning. So far, internal hardware malfunctions (i. e. tolerances, runtime errors or defects) are of no interest in the present case, but they can be added for an analysis of their effect on the system.

To complete the simulation framework, a physics engine can be embedded to simulate the system behavior on affecting forces. Currently, the simulation of CROMSCI uses *Nvidia*'s *PhysX*[3], but it could be replaced by other engines like *Newton Game Dynamics*[4] as it has been done in other research projects before [11]. In fact, each physic engine could be used which considers the following aspects:

- Collision models of robot parts and the environment,
- affecting forces in terms of gravity and adhesion force,
- friction model between wheels and ground surface and
- orientation and velocity of wheels for robot locomotion.

Due to simplicity, the physical environment of CROMSCI does not use the same rough structured mesh as for leakage simulation. Figure 6 shows the physical mesh model of the robot consisting of the convex hull of the basic chassis. Below, the drive system has been emulated via three units, each consisting of one rotational joint for wheel steering and a *PhysX* wheel shape which can be turned for locomotion. Arrows mark the pose of each part which can be annoted with certain characteristics like dynamic and static friction or a mass. Parameters and values of each part can be displayed online, as it is shown on the left side of figure 6.



```
press 'h' for help on console                    scene 0 with 9 actors          9

actor 8 of 9: CROMSCI-RigidBody

mass: 2.5
static friction: 0.5
static frictionV: 0
dynamic friction: 0.5
dynamic frictionV: 0
restitution: 0
number of shapes: 1

position: [-9.13714e-09, -1.91999e-09, 0.3756]
orientation: [1.35234e-09, -1.04458e-09, 4.0232e-07]
linear velocity: [-1.13887e-08, 1.12406e-09, 7.27378e-14]
angular velocity: [5.60786e-10, 1.94399e-09, -1.2712e-09]
```

Fig. 6. Online view on mesh model for physical simulation via *PhysX*.

The definition of the interface between MCA and *PhysX* is done via a configuration class. Here, three different types of interaction exist which have to be set manually inside of the code depending on the type of actuator:

- *Imports* define control values to set e. g. the wheel speed `eIMPORT_WHEEL_SPEED` or the adhesion force `eIMPORT_FORCE` of the chassis, e. g.
  ```
  if (actor == wheel_actor)
    imports |= eIMPORT_WHEEL_SPEED;
  if (actor == cromsci_body_actor)
    imports |= eIMPORT_FORCE;
  ```
- *Exports* are (virtual) sensor values, mainly used to get the pose of all objects (`eEXPORT_LOCAL_POSE`). But it is also possible to read out other values provided by the *PhysX* engine:
  ```
  exports |= eEXPORT_LOCAL_POSE;
  if (actor == cromsci_body_actor)
    exports |= eEXPORT_ROLL_PITCH_YAW;
  ```
- *Parameters* can be set to adjust global values like gravity as well as object settings such as mass, friction or the behavior of the wheels contact point (`ePARAM_WHEEL_TIRE_FUNCTION`), e. g.
  ```
  if (actor == wheel_actor)
    params |= ePARAM_WHEEL_TIRE_FUNCTION;
  ```

The wheel-ground interaction in terms of friction, velocities and affecting forces is simplyfied in two ways. At first, the surface is assumed to be flat, as mentioned before. Furthermore, the friction caused by the inflatable sealing is also not considered here. Nevertheless, one can sufficiently simulate the robot behavior while the system drives. Depending on given parameters like friction values, wheel setup and generated downforce a different robot behavior relative to its motion direction and gravity can be observed. For instance, wheel slip increases due to gravity if the robot drives upwards. In the same way, the physic engine lets the robot fall down if its adhesion force is too low.

[3] http://developer.nvidia.com/physx

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

The current state of the physical simulation can be observed during runtime via a visualization window, as given in figure 6. It is possible to pause the simulation, walk through the scene and visualize current values of all components. Furthermore one can adapt parameters like gravity during runtime – if they have been registered inside of the simulation's description – or apply an impact to a simulated body.

VII. APPLICATION

The presented framework has been created to test algorithms and control strategies which will be applied to the real prototype CROMSCI. The screenshot of CROMSCI's GUI in figure 7 illustrates some parts of the simulation. On the left side control elements like buttons, sliders and joysticks are given to set up the closed-loop controllers and activate robot motion (1). At (2) the chamber states are provided in terms of a circular robot view showing negative pressure values as well as some graphical representations of the estimated leakage or valve opening areas. The current view of the used 3D scene is given in (3) which can be changed during runtime to walk through the scene. Below, a plugin visualizes the current behavior meta values of the control system (4). On the right side the sealing view (5) and some online plots of adhesion and downforce data (6) are shown.

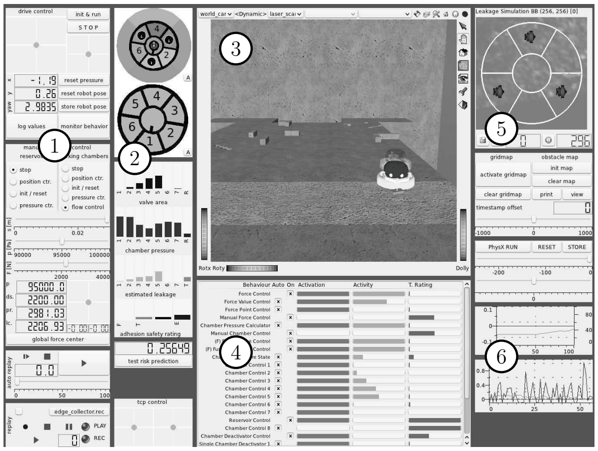


Fig. 7. GUI with visualization plugins for the simulated scene, control elements and state information.

In total, the simulation framework is used to test and to develop a couple of aspects related to climbing robots [1]. This includes software and controllers, but also robot design in terms of different chamber setups, volume size or geometric aspects. Without, it would not be possible to bring CROMSCI this far since its hardware components are still not robust enough for long-term experiments.

A. Closed-loop Adhesion Control

One example of application is the validation of the individual closed-loop adhesion controllers of the chambers and subsequent safety strategies. Figure 8 (top) shows a situation in which the robot drives down a rough wall with a deep crack in front. Here, the left circular view represents the state of the negative pressure of all seven chambers. The darker the color, the lower the pressure (the real GUI as given in figure 7 uses colors changing from green to red). The numbers 2, 4 and 6 indicate the associated adhesion chambers, whereas the dimensions of the black circles in non-labeled chambers 1, 3 and 5 depict the absolute downforce at the corresponding wheel. It can be observed, that these downforces are balanced out (all circles have the same diameter) by a higher negative pressure at the top chambers (darker color of chambers 3, 4 and 5 compared to 1, 2 and 6). This is important to counteract robot tilt caused by gravity and robot's mass distribution.

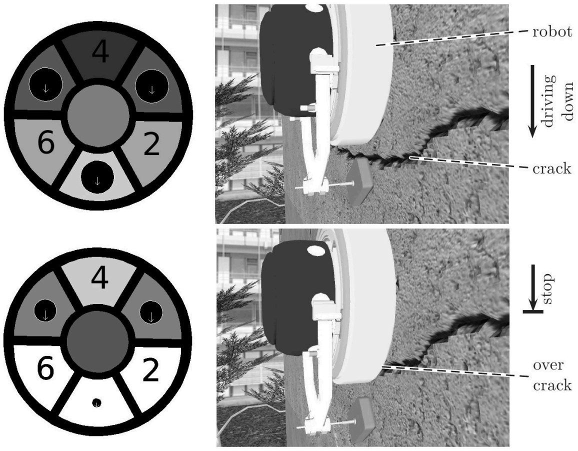


Fig. 8. Screenshots showing the robot driving to a deep crack (top, right) and located on the crack (bottom) with corresponding states of the adhesion system (left).

Later on (figure 8, bottom), the lower chambers are located on the crack and lose adhesion (white chamber color). As a result the downforces at the lower wheel (circle in chamber 1) are small compared to the two wheels on top.

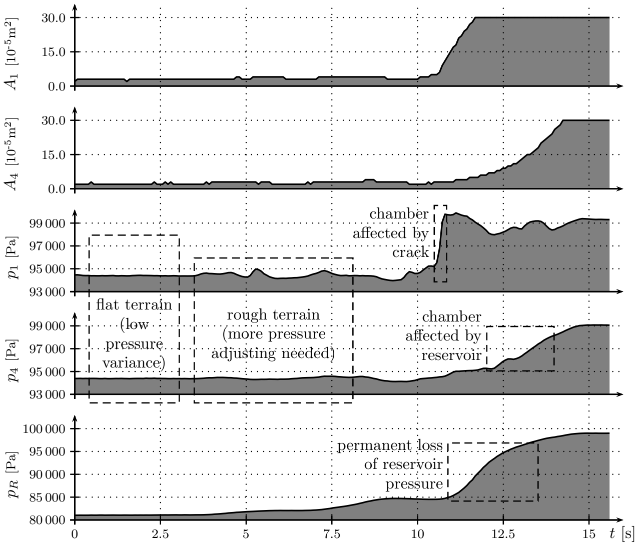


Fig. 9. Simulation experiment of closed-loop pressure controllers in which the robot tries to overcome a deep crack.

Figure 9 presents corresponding pressure p and valve areas A of the frontal chamber 1, the rear chamber 4 and the reservoir pressure p_R. At the beginning, the robot is driven on a very smooth surface with low basic leakages. At about $t = 3.5\,\mathrm{s}$ the frontal chamber reaches a structured patch as given in figure 8. In the following, the chamber controllers have to perform more pressure balancing and the reservoir pressure rises slightly. Then, the first chamber 1 reaches the crack ($t = 11\,\mathrm{s}$) and its pressure increases to ambient pressure very fast. From now on, the controllers try to adjust the pressure by opening the valves, which results in a loss of reservoir pressure. The increasing reservoir pressure also influences those chambers, which are not affected by the crack directly. This is visualized by the pressure value p_4 of the rear chamber, which also starts to increase up to a value close to ambient pressure. The valves of all chambers open to the maximum. In this case the robot has no chance to maintain the desired adhesion forces and drops off, because the adhesion system reaches its limits (valves full opened, suction engines at maximum).

A significant comparison of simulated and real-world experiments is not easy (the general similarity of the air flow has been described in [9]). But, one can state that the simulation is *similar enough* to transfer the control algorithms to the high-dynamic machine easily. In the worst case, some simple parameters need to be adapted.

B. Obstacle Avoidance

Another application of the simulation framework has been the development and optimization of methods for obstacle avoidance based on laser range data [12]. Figure 10 shows a visualization of the simulated scan lines which are emitted from a simulated sensor on top of the manipulator arm.

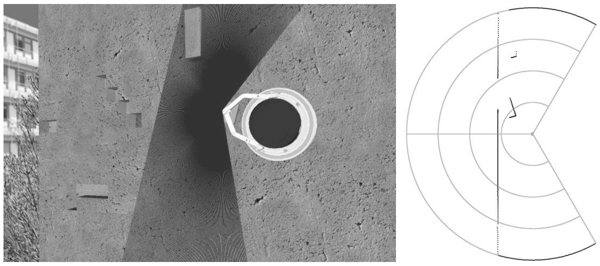


Fig. 10. Simulated scan lines of a laser range sensor (left) and resulting scan points (right) as top-view.

Via parameters this simulated sensor can achieve nearly the same characteristics as the real sensor, such as accuracy, range and angle. The processing and analysis algorithms as well as subsequent behavior-based control components have been tested in the simulation framework first. The simulation was of great benefit to test these safety features without endangering the real system. Since the simulation setup was close enough, the algorithms could be executed on CROMSCI without any problems, as published in [12].

VIII. CONCLUSIONS

Within the scope of this paper a novel framework has been presented to simulate a climbing robot equipped with a negative pressure adhesion system and an omnidirectional drive system. It could be shown that the system can simulate aspects like airflows and pressure changes of the complex vacuum system as well as environmental sensors for obstacle detection. Related to the negative pressure adhesion, the surface structure has a direct impact on the sealing leakages to simulate different ground characteristics like roughness or defects. Hence, this tool supports the development of control algorithms independent of the real robot, enables an offline and realtime validation in arbitrary 3D environments and allows an easy testing of different robot designs and setups.

In future, this simulation will be ported to the new robot control framework FINROC[5]. Further work lies in an easier handling of the *PhysX* integration, since the *Blender* export of collision objects does not include joints (they have to be added manually so far). Finally, the new robot prototype CREA is under construction. Its control software – closed-loop controllers as well as high deliberative functions – will be developed and optimized within the simulation first, before executing it on the real machine.

ACKNOWLEDGEMENTS

This research was funded by the German *Bundesministerium fuer Wirtschaft und Technologie* (BMWi) under the program *Zentrales Innovationsprogramm Mittelstand* (ZIM).

REFERENCES

[1] D. Schmidt, "Safe Navigation of a Wall-Climbing Robot - Risk Assessment and Control Methods," doctoral thesis, University of Kaiserslautern, Germany, 2013.

[2] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *International Conference on Intelligent Robots and Systems (IROS)*, 2004.

[3] T. Laue and T. Röfer, "Simrobot - development and applications," in *International Conference on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR)*, 2008.

[4] J. Wang, M. Lewis, and J. Gennari, "A game engine based simulation of the nist urban search and rescue arenas," in *Winter Simulation Conference*, 2003.

[5] O. Michel, "Webots: Professional mobile robot simulation," *International Journal Of Advanced Robotic Systems*, vol. 1, no. 1, 2004.

[6] M. Freese, S. Singh, F. Ozaki, and N. Matsuhira, "Virtual robot experimentation platform v-rep: A versatile 3d robot simulator." in *2nd International Conference on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR)*, 2010.

[7] D. Longo, G. Muscato, and S. Sessa, "Simulation and locomotion control for the alicia3 climbing robot," in *International Symposium on Automation and Robotics in Construction (ISARC)*, 2005.

[8] I. Pretto, S. Ruffieux, C. Menon, A. Ijspeert, and S. Cocuzza, "A point-wise model of adhesion suitable for real-time applications of bio-inspired climbing robots," *Journal of Bionic Engineering*, vol. 5, 2008.

[9] J. Wettach, C. Hillenbrand, and K. Berns, "Thermodynamical modelling and control of an adhesion system for a climbing robot," in *International Conference on Robotics and Automation (ICRA)*, 2005.

[10] D. Schmidt, J. Wettach, and K. Berns, "3d realtime simulation framework for a wall-climbing robot using negative-pressure adhesion," in *10th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, July 29-31 2013.

[11] J. Wettach, D. Schmidt, and K. Berns, "Simulating vehicle kinematics with simvis3d and newton," in *2nd International Conference on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR)*, 2010.

[12] M. Jung, D. Schmidt, and K. Berns, "Behavior-based obstacle detection and avoidance system for the omnidirectional climbing robot cromsci," in *13th International Conference on Climbing and Walking Robots (CLAWAR)*, August 31 - September 3 2010, pp. 73–80.

[5] http://www.finroc.org