


Autonomous Free-Form Trenching using a Walking Excavator

Conference Paper**Author(s):**

Jud, Dominic; Leemann, Philipp; Kerscher, Simon; [Hutter, Marco](#) 

Publication date:

2019-10

Permanent link:

<https://doi.org/10.3929/ethz-b-000348588>

Rights / license:

[In Copyright - Non-Commercial Use Permitted](#)

Originally published in:

IEEE Robotics and Automation Letters 4(4), <https://doi.org/10.1109/LRA.2019.2925758>

Autonomous Free-Form Trenching using a Walking Excavator

Dominic Jud¹, Philipp Leemann², Simon Kerscher¹ and Marco Hutter¹

Abstract—This article shows accurate and autonomous creation of free-form trenches using a walking excavator. We present hardware extensions and modifications for full automation, a mapping approach specifically tailored to excavation, environment collision-free trajectory planning on these maps, an arm controller aware of various limits and an improved state machine that enables the execution on real hardware. Furthermore, previous work about excavation planning and the design of a single soil-independent dig cycle is extended and transferred from simulation to hardware. The entire system is tested on a four-segment, piecewise-planar trench and a free-form curved trench. Both shapes were successfully excavated with unprecedented accuracy.

Index Terms—Robotics in Construction, Mining Robotics

I. INTRODUCTION

THE development of robotic systems in architecture had a drastic influence on the design methods architects have at hand [1]. Landscape architecture however, has mostly been unaffected by on-site robots due to the inherent difficulty of automating earthwork. Hurkxkens et al. [2] show a high architectural potential in creating robotic landscapes that are tailored to local conditions, e.g. by using only on-site material, balancing cut and fill or online adaption of the design to the found material.

Many different autonomous excavators have been built over the last two decades. Although none were able to create free-form shapes. Haga et al. [3] created level trenches using an external datum level. Also the Autonomous Loading System (ALS) by Cannon et al. [4] had the objective to leave a level floor after excavation. These two projects further have in common, that they both used pilot stage valves to control the existing main valves. Quang et al. [5] showed that upgrading an excavator with high-performance electrohydraulic servo valves in the main stage will drastically improve control performance and thus achieved accurate bucket edge control while digging. Other hydraulic robotic systems, e.g. the quadrupedal robot HyQ [6], highly rely on the superior performance of servo valves. Also our previous advances on active control of

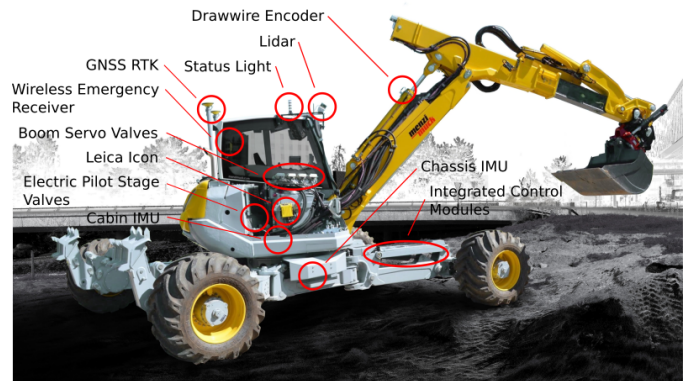


Fig. 1. HEAP is an autonomous walking excavator based on a highly customized Menzi Muck M545.

the excavators chassis [7] leveraged servo valves, motivating the use of these also on the arm automation shown in this work. The creation of more complicated shapes than flat bottom trenches also requires geometric feedback from the manipulated terrain, as inferring the terrain only from the bucket edge motion is not sufficient. ALS [4], as one of the few autonomous excavators with perception, uses lidars to create a map during swing motion. The major challenge for mapping in autonomous excavation is that the machine constantly changes the environment and some important areas might be occluded. General mapping algorithms such as LOAM [8] or ICP [9] will not update at all or fast enough to these self made changes and occlusions are also not handled. Hombberger et al. [10] show fusion of exteroceptive and proprioceptive sensing to successfully estimate the support surface in highly compliant terrain for legged robots. A similar approach of fusing information from the bucket motion (proprioceptive) with lidar scans (exteroceptive) to circumvent the problems of occlusion and dynamic environments is pursued in this work. Regarding control, Maeda et al. [11] used iterative learning control to predict the disturbance in the next dig cycle. No dynamic model in the conventional sense is used by Park et al. [12]. Instead, the dynamics are learned online with an echo state network and used to track a position trajectory. The network can also compensate for changes during execution, e.g. change in fluid temperature. A different approach was taken by Groll et al. [13], where hierarchically organized primitives are used to adapt to a changing environment.

This letter shows highly accurate autonomous landscaping using HEAP, the world's first autonomous walking excavator based on a Menzi Muck M545 shown in Fig. 1. It builds

Manuscript received: February, 24, 2019; Revised May, 17, 2019; Accepted June, 5, 2019.

This paper was recommended for publication by Editor Youngjin Choi upon evaluation of the Associate Editor and Reviewers' comments. This work was supported in part by the Swiss National Science Foundation through the National Centre of Competence in Digital Fabrication (NCCR dfab), Hexagon Geosystems and armasuisse Science and Technology.

¹Authors are with Robotic Systems Lab, ETH Zurich, Switzerland, dominic.jud@mavt.ethz.ch

²Philipp Leemann is with Moog Industrial Group GmbH, Böblingen, Germany

Digital Object Identifier (DOI): see top of this page.

upon our previous work in simulation showing a planning and control approach [14] consisting of force trajectory based dig cycles which reliably fill the bucket without any knowledge about the soil properties. We present the hardware extensions and modifications that are necessary to facilitate these force controlled dig cycles on a real machine, a robust excavation mapping approach based on fusing proprioceptive and exteroceptive mapping, trajectory planning that leverages the excavation map, control for safe and accurate operation of the arm and an improved state machine to adapt to the differences between simulation and real hardware.

II. SYSTEM DESCRIPTION

HEAP was built up from the very beginning for the application of autonomous landscaping. It is based on a 12ton M545 walking excavator commercially available from Menzi Muck, but highly customized with numerous adaptations and additions. Fig. 1 provides an overview of the major changes. To achieve active balancing and adaptation to uneven ground [7], all 14 hydraulic cylinders in the chassis were exchanged with cylinders that include integrated control modules that allow for precise control of leg positions and ground interaction forces.

A. Sensing

A Leica iCON iXE3 with two GNSS antennas and a receiver is used for localization. RTK corrections for the GNSS signals are received over the internet from permanently installed base stations. SBG Ellipse2-A inertial measurement units (IMU) are installed both in the cabin and on the chassis. Sick BCG05-C1QM0199 wire draw encoders measure the piston position and velocity of the arm cylinders at 100Hz. The cylinder force is estimated using pressure sensors integrated in the servo valve control modules. Finally, two Velodyne Puck VLP-16 lidar's are used for terrain mapping. Lidar's were chosen, because they outperform camera based sensors in heavy dust environments and they provide more accurate and dense measurements compared to radar based sensors [15].

B. Arm Actuation

The two most commonly found valve setups for automating an excavator's arm are realized on HEAP. Firstly, electric pilot stage valves are installed in parallel to the hydraulic joysticks, which reflects the most commonly used approach. Secondly, high-performance servo valve prototypes with integrated electronics (microprocessor, current driver and pressure sensors) are installed in the high-pressure stage in parallel to the pilot stage driven main valves. Fig. 2 shows the level of intrusion of these two different principles. The pilot stage valves (blue) require the hydraulic low-pressure circuit and the servo valves (red) the high-pressure circuit to be opened. With this innovative parallel solution, we expect to be able to control large flow rates through the pilot stage setup while achieving very high control bandwidth with the servo valve setup. See Table I for a closer comparison.

1) *Pilot stage*: Hawe PMZ proportional pressure reducing valves are installed in parallel to the joysticks in the pilot

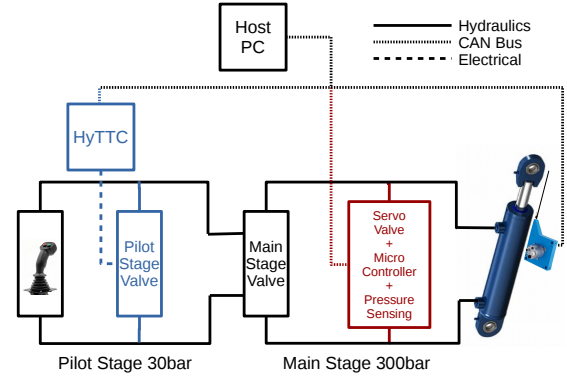


Fig. 2. The two different actuation types are pilot stage valve (blue) and Moog servo valve (red). The black lines and boxes represent the standard components installed by the machine manufacturer.

TABLE I
FREQUENCY SWEEPS FOR VELOCITY AND FORCE CONTROL WERE CONDUCTED ON BOTH PILOT STAGE AND SERVO VALVES IN ORDER TO DETERMINE THE CONTROL BANDWIDTH (BW).

	Max. Flow (dipper)	Velocity Control BW	Force Control BW
Pilot Stage	~ 200l/min	0.5Hz	0.85Hz
Servo Valves	~ 40l/min	> 2Hz	> 20Hz

stage to control the standard main valves installed by the machine manufacturer. The input current of the valve is set by a TTControl HY-TTC 30 control unit connected to the CAN bus. The piston velocity is controlled using a PI-controller and a feed forward lock-up table from desired piston velocities to valve currents running at 100Hz to generate the reference current for the pilot valve. The lookup table is identified over the entire range of applicable piston velocities with 15 value pairs. The piston velocity control bandwidth is 0.5Hz, whereas the piston force can be controlled to up to 0.85Hz.

2) *Servo valve*: An integrated control module (ICM) prototype, developed by Moog Inc., is installed in parallel to the main valves in the high pressure circuit. It consists of a high performance servo valve with a current driver and a check valve for safety. The integrated ARM micro processor can control piston position and velocity at 100Hz and force at 1kHz. The device communicates over CAN at 100Hz to the host computer. The pressure sensors are directly connected to the micro controller. The piston velocity control bandwidth is larger than 2Hz, whereas the piston force can be controlled at more than 20Hz. Due to safety considerations, the bandwidth identification test could not be continued for velocity signals beyond 2Hz, as the excavator starts to shake violently when the signal approaches the eigenfrequency of the entire system. Servo valves will certainly deliver a better performance than the pilot stage valves in all aspects. However, servo valves can cause higher pressure drops that can stall the diesel engine. A hydraulic accumulator (Hydac SBO) is installed on the supply line of the servo valves to support the pump in case of short high flow demands. As these valves were initially designed for smaller leg cylinders, they only allow for slower motion compared to the pilot stage, see first column of Table I. The servo valves provide a clear abstraction of the

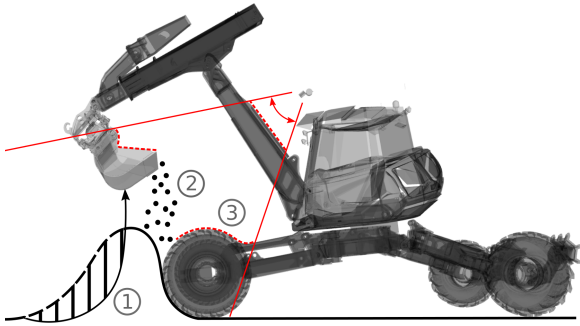


Fig. 3. The three major challenges in mapping for excavation are (1) unperceived changes marked with the striped area, (2) falling soil below the bucket and (3) self perception marked with dotted red lines on the legs and arm.

hydraulic systems dynamics due to no valve overlap. They are fast enough to locally compensate for various disturbances, achieving the required high precision end-effector control for accurate autonomous excavation.

III. ELEVATION MAPPING FOR EXCAVATION

The mapping framework for autonomous excavation is based on robot-centric elevation mapping for mobile robots by Fankhauser *et al.* [16]. The algorithm creates a 2.5 dimensional elevation map from any distance sensor and a corresponding pose estimate. A full 3 dimensional mapping approach is not needed for excavation, as overhanging terrain is not likely. The input to the mapping algorithm is the point cloud of two Velodyne Puck and the transformation of both sensors w.r.t. the world frame extracted from state estimation. The lidars are placed at the front edge of the roof as shown in Fig. 1 for a good vantage point. As these sensors were initially developed for autonomous driving, the vertical field of view is not sufficient for an undulating landscape. In this project, two lidars were placed orthogonally to each other. The vertical lidar is especially valuable as swinging the cabin is an essential part of excavation and the cabin swing will move the vertical sensor in a scanning motion.

The elevation mapping approach is mostly meant for static environments, e.g. a legged robot walking up stairs. However, an autonomous excavator is meant to shape and change its environment. The major issues introduced by this circumstance are illustrated in Fig. 3 and they can be resolved by informing the mapping about the excavation task.

1) *Unperceived changes/occlusions*: The striped area in Fig. 3 is removed through one dig, but the change is occluded from the sensors by the remaining pile. Proprioceptive mapping is added to the otherwise purely exteroceptive mapping. Proprioception is achieved by tracking the bucket edge line as it moves through the soil. Once the bucket is full and the digging process terminates, the erroneous area in the exteroceptive map is replaced with the proprioceptive map. The state of the map after a proprioceptive update can be seen in Fig. 4. The color of the map represents the uncertainty. The proprioceptive update is clearly visible through the high uncertainty (green/blue).

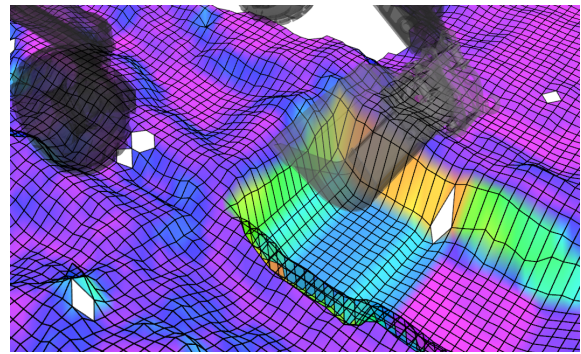


Fig. 4. After a dig cycle, the bucket is lifted from the ground and the proprioceptive map is inserted into the elevation map with a high uncertainty. Regions with high uncertainties are green, whereas low uncertainties are red. Note that the steep side walls always keep a high uncertainty, because near vertical surfaces are inherently hard to represent with a 2.5D map.

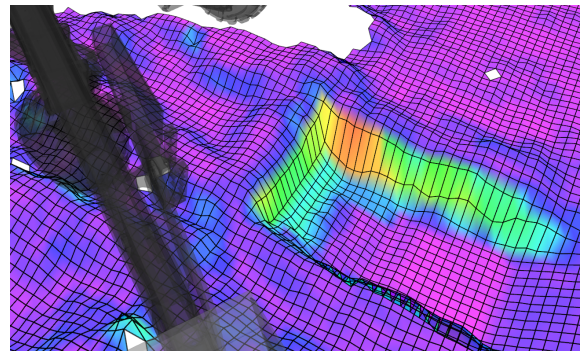


Fig. 5. While the arm swings to the side to dump the soil, the map of the trench is updated with input from the lidars. The uncertainty decreases for those parts represented by the purple color. The occluded areas keep the height values with large uncertainty (green) from the proprioceptive mapping, yielding a complete map.

The proprioceptive map is inserted with a high uncertainty such that these areas are quickly update by the lidars once they are seen. As soon as the arm swings to the side to dump the soil, the trench is updated with lidar information. In Fig. 5, compared to Fig. 4, some areas have a lowered uncertainty (red/purple) due to the lidar updates. Areas in the map that are still occluded, e.g. the front edge of the trench, will keep height values from exteroceptive mapping with a high uncertainty.

2) *Falling soil*: When the bucket is full with soil and it moves through the air, there might be particles falling down from the bucket. These particles will be captured mid-air by the lidars and will falsely create a wall/pile in the map. This issue is negotiated by introducing a falling soil filter whenever the bucket is full. The falling soil filter is a box below the bucket. The box moves along with the bucket. All the points captured inside this box will be removed. The box filter is implemented efficiently using PCL [17].

3) *Self perception*: The arm and the legs move mostly inside the field of view of the sensors. A self filter implemented for PR-2 [18] is used to eliminate points that lie on the robot itself. It was extended in the scope of this work to support multiple meshes per body as demanded by the spatial complexity of the excavator bodies.

The result is a map that is not corrupted by the ongoing ex-

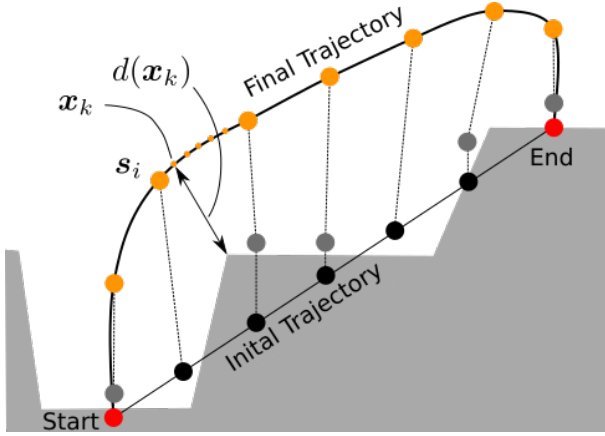


Fig. 6. The SQP optimizer adapts the support points s_i (orange dots) to compute a feasible 3D path from start to end position (red dots) using concatenated cubic Hermite splines. The optimizer is initialized with collision-free support points (grey dots) from a straight line solution (black dots). The SDF is not illustrated for the sake of simplicity, but an example distance $d(\mathbf{x}_k)$ from sampled position \mathbf{x}_k to the closest point on the surface is given.

cavation process, but still adapts quickly to self-made changes. The map is created with a resolution of 0.1m, which is detailed enough as the bucket width is still 15 times wider. Moreover, the map is georeferenced through GNSS and can thus be easily positioned in a construction site environment.

IV. SQP-BASED SPLINE TRAJECTORY OPTIMIZATION

An elementary operation in excavation is moving the bucket through the air from the current position after digging to the desired dump position. During this operation, collisions with the environment should be avoided by planning a collision-free 3D trajectory for the bucket. Note that only collisions with the environment are handled in the planner. Self-collisions are avoided locally in the controller, see Section V-A.

The path is parametrized by cubic Hermite splines [19], which have the advantage that they are continuous and easily differentiable. They are defined by start and end positions as well as the derivatives at these positions. Multiple splines are concatenated by support points to form the final trajectory.

A 3D signed distance field (SDF) can be computed from the 2.5D elevation map provided by the mapping framework. An SDF stores the distance and gradient to the closest point on the surface from any point in the robot workspace. The sign of the distance indicates if the point is above or below the surface. This property is especially important in excavation, where the start of the trajectory might lie below the surface after digging. A Sequential Quadratic Program (SQP) is used to optimize the position of the support points. The optimization objective/costs c consists of a length, collision and regularization cost. It is computed by sampling the spline with discrete time steps yielding the sampled positions \mathbf{x}_k . The length cost c_l for N samples is

$$c_l = \sum_{k=1}^{N-1} \|\mathbf{x}_k - \mathbf{x}_{k+1}\|. \quad (1)$$

The collision cost c_c , inspired by CHOMP [20], is

$$c_c = \sum_{k=1}^N c_{c,k} \quad (2)$$

$$\text{with } c_{c,k} = \begin{cases} -d(\mathbf{x}_k) + \frac{\epsilon}{2}, & \text{if } d(\mathbf{x}_k) < 0 \\ \frac{1}{2\epsilon}(d(\mathbf{x}_k) - \epsilon)^2, & \text{if } 0 < d(\mathbf{x}_k) \leq \epsilon \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

The shortest distance $d(\mathbf{x}_k)$ from position \mathbf{x}_k to the map surface is extracted from the SDF. The collision cost is zero if the point is further away than the distance threshold ϵ . The regularization cost c_r keeps the support points equally spread along the spline by penalizing a high standard deviation σ of the distances between the support points s_i and is defined as

$$c_r = \sigma(\|s_i - s_{i+1}\|). \quad (4)$$

The final cost c is defined as the weighted sum of the individual costs:

$$c = w_l c_l + w_c c_c + w_r c_r \quad (5)$$

The weights w_l , w_c and w_r control the behaviour of the optimization. Fig. 6 shows the initialization and the final trajectory for a simple 2D example. The trajectory is initialized by a straight line between start and end position (red dots) with intermediate support points (black dots). If any of the support points are in collision, they are moved up until they are out of collision using the SDF, yielding the grey dots in Fig. 6. Connecting the grey dots with splines will not yet yield an entire collision-free trajectory. The optimizer will move the support points such that the entire trajectory is collision-free and optimize the length simultaneously, resulting in the final trajectory through the orange support points. Note that the ratio between the weight for length and collision controls the approach angle of the trajectory to the surface. If avoiding collisions has a much higher weight than minimizing path length, the final trajectory leaves/approaches the start/end point almost orthogonal to the surface. This minimizes the risk of the bucket colliding with the environment while leaving/approaching start/end position. On the other hand, if the collision avoidance weight is too low, the trajectory will pass by objects too closely.

V. ARM CONTROL

The definition of generalized joint positions \mathbf{q} (a subset of the generalized coordinates of the excavator) and the generalized joint velocities \mathbf{u} are

$$\mathbf{q} = [\psi \quad \theta_1 \quad \theta_2 \quad x_{tele} \quad \theta_3]^T \in \mathbb{R}^n \quad (6)$$

$$\mathbf{u} = \dot{\mathbf{q}} \in \mathbb{R}^n, n = 5. \quad (7)$$

These coordinates include the rotation of the cabin ψ , the angle θ_1 between boom and cabin, the angle θ_2 between boom and dipper, the extension of the telescope x_{tele} and the angle θ_3 between telescope and shovel.

The generalized forces $\boldsymbol{\tau} \in \mathbb{R}^n$ are related to the actuator forces $\boldsymbol{\tau}_p$ through the diagonal force mapping matrix $E(\mathbf{q}) \in \mathbb{R}^{n \times n}$ as

$$\boldsymbol{\tau} = E(\mathbf{q})\boldsymbol{\tau}_p \quad (8)$$

$$\text{with } \boldsymbol{\tau} = [\tau_{\text{turn}} \quad \tau_{\text{boom}} \quad \tau_{\text{dipper}} \quad F_{\text{tele}} \quad \tau_{\text{shovel}}]^T \quad (9)$$

$$\text{and } \boldsymbol{\tau}_p = [\tau_{\text{turn}} \quad F_{\text{boom}} \quad F_{\text{dipper}} \quad F_{\text{tele}} \quad F_{\text{shovel}}]^T. \quad (10)$$

The exertable task-space force $\mathbf{f}_t \in \mathbb{R}^m$ with $m = 4$ consists of forces in x,y and z and the shovel torque. Other task-space forces and moments can not be actuated and are dropped for the sake of simplicity. The 5 degree of freedom excavator arm has one redundant degree as $n - m = 1$.

A. Hierarchical Optimization Inverse Kinematic Arm Controller

The arm controller is based on the hierarchical optimization by Bellicoso et al. [21], which solves the control problem as a constrained quadratic optimization. The desired joint motion \mathbf{u}_d is found through a number of least square optimization problems that include equalities as well as inequalities in the form of $A\mathbf{x} = \mathbf{b}$ and $A\mathbf{x} \leq \mathbf{b}$, respectively. Due to the redundancy of the arm, we can not only optimize for the task space motion, but additionally include other objectives. Moreover, this formulation allows us to handle collisions and hardware limits. The set of prioritized tasks are in decreasing priority:

1) *Pump Flow Limit*: The hydraulic pump of the excavator has a maximum oil flow Q_{max} that it can deliver. The flow per cylinder is calculated from the cylinder areas A_A and A_B with $A_A > A_B$. The vector of relevant cylinder areas $A \in \mathbb{R}^{1 \times n}$ produces the summed up flow of all cylinders when multiplied with piston velocities and is derived per joint i as

$$A(i) = \begin{cases} A_A, & \text{if } \dot{q}(i) > 0 \\ -A_B, & \text{otherwise.} \end{cases} \quad (11)$$

The inequality constraint to avoid exceeding the maximum flow is

$$AE(\mathbf{q})\mathbf{u}_d \leq Q_{\text{max}} \quad (12)$$

with $E(\mathbf{q})$ being the joint position dependent transformation from joint velocities to piston velocities.

2) *Cylinder Velocity Limits*: The maximum flow of oil through a valve limits the maximum velocity a hydraulic cylinder can achieve. The corresponding piston velocity limits $[\mathbf{v}_{\text{min}}, \mathbf{v}_{\text{max}}]$ are transformed to joint velocity limits using the inverse of $E(\mathbf{q})$ and recomputed every time step. The inequality constraint is

$$\begin{bmatrix} \mathbb{I}_{n \times n} \\ -\mathbb{I}_{n \times n} \end{bmatrix} \mathbf{u}_d \leq \begin{bmatrix} E(\mathbf{q})^{-1}\mathbf{v}_{\text{max}} \\ -(E(\mathbf{q})^{-1}\mathbf{v}_{\text{min}}) \end{bmatrix} \quad (13)$$

3) *Cylinder Position Limits*: The hydraulic cylinders also have a limited stroke and therefore the joint positions are limited by incorporating

$$\begin{bmatrix} \delta t \mathbb{I}_{n \times n} \\ -\delta t \mathbb{I}_{n \times n} \end{bmatrix} \mathbf{u}_d \leq \begin{bmatrix} \mathbf{q}_{\text{max}} - \mathbf{q} - \mathbf{u} \delta t \\ -(\mathbf{q}_{\text{min}} - \mathbf{q} - \mathbf{u} \delta t) \end{bmatrix} \quad (14)$$

where \mathbf{q}_{min} and \mathbf{q}_{max} are the joint position limits computed from the actuator position limits.

4) *Self-Collision Avoidance*: Local self collision avoidance for convex objects is implemented as proposed by Faverjon and Tournassoud [22]. The approach limits the velocity of the bodies in direction of the collision. The links are represented with collision primitives, i.e. boxes, and checked for collision using the Bullet physics engine by Coumans et al. [23]. Bullet will provide the two closest points \mathbf{p}_1 and \mathbf{p}_2 between two bodies. The normalized direction of collision is

$$\mathbf{n} = (\mathbf{p}_2 - \mathbf{p}_1)/d \quad \text{with} \quad d = \|\mathbf{p}_2 - \mathbf{p}_1\|. \quad (15)$$

The distance where the constraint will have an influence is d_i and the minimum safety distance is d_s . The constraint, that limits the end effector velocity in direction of the collision, is then

$$\mathbf{n}^T J_{\mathbf{p}_1} \mathbf{u}_d \leq \xi \frac{d - d_s}{d_i - d_s}. \quad (16)$$

5) *End-Effector Orientation*: A desired end-effector rotational motion $\dot{\mathbf{r}}_{R,\text{des}}$ can be incorporated with the equality

$$J_R \mathbf{u}_d = \dot{\mathbf{r}}_{R,\text{des}} \quad (17)$$

using the rotational Jacobian J_R .

6) *End-Effector Position*: A desired end-effector translational motion $\dot{\mathbf{r}}_{T,\text{des}}$ can be incorporated with the equality

$$J_T \mathbf{u}_d = \dot{\mathbf{r}}_{T,\text{des}} \quad (18)$$

using the translational Jacobian J_T .

7) *Minimize Cylinder Velocities*: If there is still a nullspace left, the controller minimizes the joint velocities

$$\min \|\mathbf{u}_d\|_2. \quad (19)$$

The desired end-effector velocity $\dot{\mathbf{r}}_{\text{des}}$ is computed from the desired end effector trajectory defined by a velocity $\dot{\mathbf{r}}_{\text{traj}}$ and a position \mathbf{r}_{traj} , as

$$\dot{\mathbf{r}}_{\text{des}} = \dot{\mathbf{r}}_{\text{traj}} + \text{PID}(\mathbf{r}, \mathbf{r}_{\text{traj}}) \quad (20)$$

with position feedback through a PID controller. Both $\dot{\mathbf{r}}_{\text{traj}}$ and \mathbf{r}_{traj} are extracted from the cubic Hermite splines. Note that the end-effector motion task is split into two tasks, i.e. orientation and position task of the end-effector. This allows for the orientation task to be of higher priority than the position task. This setup guarantees that if the collision avoidance already restricts the end-effector motion, the bucket orientation is treated with higher priority than the translation. Thus, avoiding a collision cannot change the orientation but only the translation of the shovel and there will be no soil spillage.

B. Digging Force Controller

Soil independent excavation is achieved through force controlled digging. The desired joint torques $\boldsymbol{\tau}_d$ are computed using the equation of motion and a viscous and static friction compensation as

$$\boldsymbol{\tau}_d = J^T \mathbf{f}_t^{\text{des}} + \mathbf{b} + \mathbf{g} + F_v \mathbf{u} + F_s \text{sign}(\mathbf{u}) \quad (21)$$

$$\text{with } \boldsymbol{\tau}_d \leq E(\mathbf{q})\boldsymbol{\tau}_p^{\text{max}}, \boldsymbol{\tau}_d \geq E(\mathbf{q})\boldsymbol{\tau}_p^{\text{min}} \quad (22)$$

where $\mathbf{b}(\mathbf{q}, \mathbf{u}) \in \mathbb{R}^n$ is the vector of centrifugal and Coriolis terms, $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^n$ is the vector of gravity terms, $F_v \in \mathbb{R}^{n \times n}$

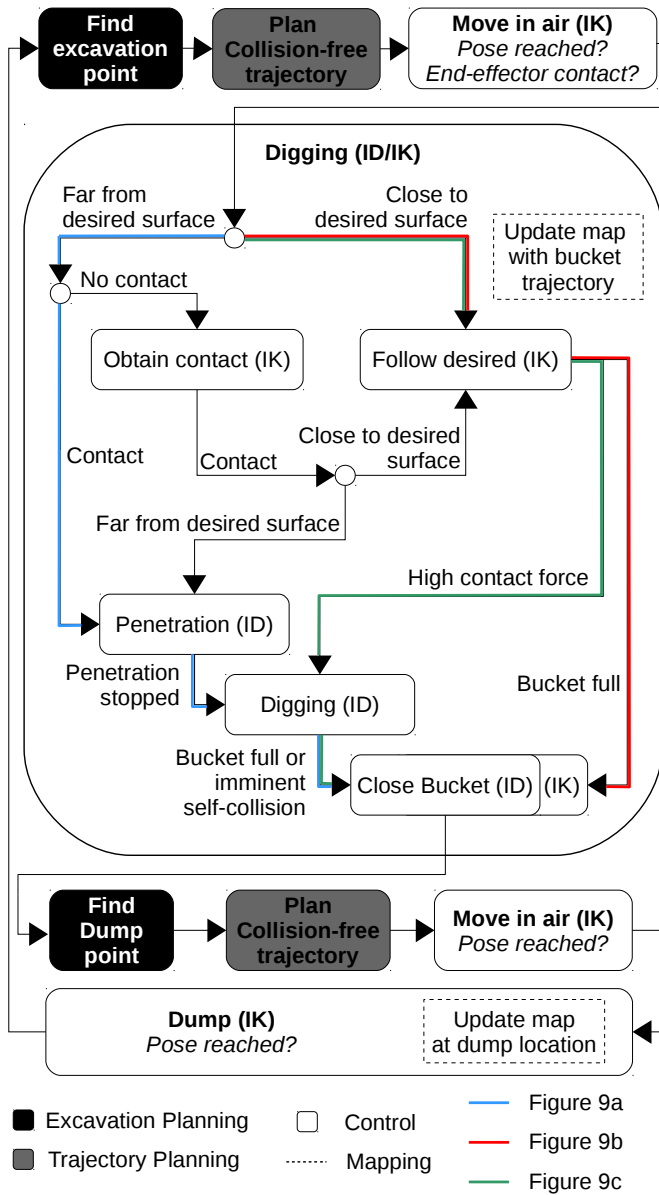


Fig. 7. The building blocks for autonomous execution of one excavation cycle are excavation planning (black), trajectory planning (gray), mapping and control (white). IK (inverse kinematics) and ID (inverse dynamics) describe which arm control strategy is used. Three different dig cycles are shown with blue, red and green lines. The respective plots are in Fig. 9a, 9b and 9c.

is the diagonal viscous friction matrix, $F_s \in \mathbb{R}^{n \times n}$ is the diagonal static friction matrix and $[\tau_p^{\min}, \tau_p^{\max}]$ the maximum cylinder forces/torques. The desired end-effector force in task space f_t^{des} is mapped to generalized forces in joint space by the spatial shovel origin Jacobian $J \in \mathbb{R}^{m \times n}$. RBDL, the Rigid Body Dynamics Library [24], was used for implementation.

VI. AUTONOMOUS EXCAVATION

Autonomous excavation is achieved with the state machine shown in Fig. 7. First, a suitable point of attack for digging, using the excavation planner proposed in previous work [14], is found. The trajectory planner, as shown in Section IV, subsequently plans an environment collision-free trajectory from the current location of the bucket to the excavation point.

The inverse kinematic controller, see Section V-A, is used to track the trajectory reference in the air. The controller avoids running into any limits and locally avoids self-collisions. The pilot stage driven main valves are used for motions in the air, because they allow for more flow compared to the servo valves. Additionally, the superior control performance of the servo valves is not needed. The excavator controls the distance to the excavation point by driving forward and backwards. The active chassis [7] adapts to the ground while driving and guarantees a stable stance for excavation.

The digging process changes depending on the contact state of the bucket and/or the distance of the start position to the desired surface. This change is necessary to avoid executing purely force controlled digging without the bucket being in contact. The contact force of the bucket with the soil is estimated with a second order residual wrench approach similar to Magrini et al. [25], but extended to include friction terms. Whenever the start position of the digging cycle is already close to the desired surface, the controller tries to track the desired surface instead of executing a force controlled dig cycle (red path in Fig. 7). Closeness is defined as the vertical distance to the desired surface. However, following the desired surface might drive the bucket deep into the ground or stones and roots might be in its way. The inverse kinematic controller will not be able to cope with these large interaction forces. If this estimated interaction force is too high, the controller stops trying to track the desired surface with the inverse kinematic controller and switches to force controlled digging (green path in Fig. 7). In case the starting point of a dig cycle is not close to the desired shape, a purely force controlled dig cycle is executed (blue path). To guarantee contact with the ground before switching to force control, an "obtain contact" procedure is executed where the bucket moves down vertically until contact is detected. The digging process is stopped either when the shovel is full or the bucket is getting too close to the legs or chassis. The soil volume in the bucket is estimated using the map prior to the dig cycle and the trajectory of the bucket during the dig cycle. Once the digging is terminated, the elevation map is updated with the measured bucket edge trajectory as shown in Section III.

Dumping follows a similar strategy. The dump point is found by excavation planning and a trajectory to this dump point is planned. The trajectory, as well as the actual dumping of the soil, is done with the inverse kinematic controller. An update of the map at the dump location is then forced to quickly adapt to the height changes.

In our previous work in simulation [14], the first step was to roughly excavate the entire trench and, in a last step, to refine it to achieve a precise shape. Due to splitting these actions into two separate parts, the excavator had to go to the start of the trench after finishing rough excavation. This is clearly sub optimal and might lead to damage of the excavated contour in case the excavator has to drive over it. In this work, the planner can switch according to the needs between refinement ("follow desired" state) and rough excavation ("digging" state) as often as necessary.

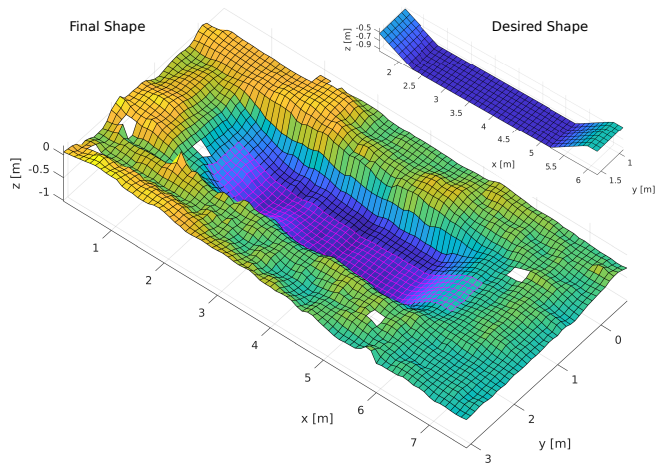
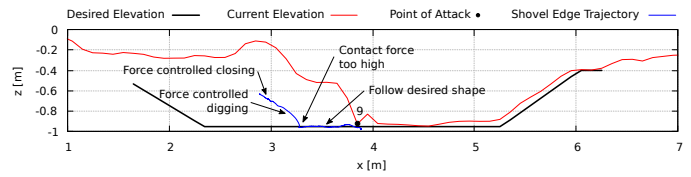


Fig. 8. The desired shape is a four-segment piecewise-planar surface depicted in the top right corner. The area in the final shape corresponding to the desired plan is marked with magenta grid lines. The coloring represents the elevation of the cells.

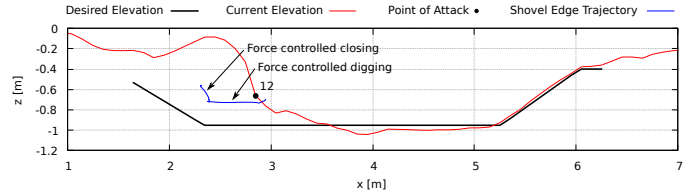
VII. RESULTS

The desired trench geometry is created in a CAD program and converted into a 2.5D elevation map by ray-tracing the surface from above. Two different trench shapes were used for evaluation of the system in real world scenarios. First, a desired shape consisting of four piecewise-planar surfaces is excavated.¹ It is shown in the top right corner of Fig. 8. Secondly, a free-form curved shape was also used for testing. The autonomous walking excavator HEAP with an attached grading bucket (clean edge, no teeth, 1.5m width) recreated the desired shapes in one go without any user intervention. Fig. 8 shows the elevation map created with the onboard lidars after successful completion of the linear segments shape. The area in the final shape that corresponds to the desired plan is marked with magenta grid lines.

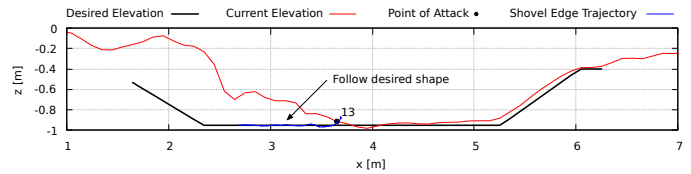
Fig. 9 explains in detail the process of autonomous excavation on the example of the 9th, 12th and 13th dig cycle. Note that there are two dig cycles between Fig. 9a and 9b that are not shown in this sequence. The corresponding decisions taken in the state machine for these three individual dig cycles is illustrated in Fig. 7 with colored lines. The excavator is positioned to the left of the plots and works from right to left. First, in Fig. 9a, the planned point of attack is already close to the desired surface. Thus, the bucket tries to track the desired surface as close as possible with the servo valves and the inverse kinematic controller until the estimated contact force exceeds a threshold. The state machine then switches to force controlled digging to fill the bucket despite large interaction forces. The excavation planner chooses the point of attack for the excavation such that it does not excavate at the very bottom of a steep wall. The arm could collide with the top edge of the trench and the forces at the bottom of such a steep wall would prevent successful excavation. Fig. 9b shows such a dig cycle where the planner chooses a point not at the bottom of the trench to first remove the soil on top. The point of attack



(a) The 9th dig cycle starts with the point of attack close to the desired surface. Thus, the bucket follows the desired shape until, in this case, the contact force exceeds the limit and the excavator continues the dig cycle with force controlled digging to fill the bucket.



(b) The 12th dig cycle is purely force controlled, because the start is not close to the desired surface. The excavation planner chooses this start location to avoid creating steep walls that are difficult to excavate.



(c) The 13th dig cycle in this excavation only uses the "follow desired" strategy to fill the bucket, because only little soil is left in that region.

Fig. 9. The three different dig cycles correspond to the three colored paths through the state machine in Fig. 7, i.e. a) is the blue, b) is the red and c) is the green path. The current elevation shown as a red line is the elevation along the center line of the trench.

is not close to the desired shape, thus the digging cycle is fully force controlled without forcing the bucket to stay on a certain trajectory. The bucket trajectory results from applying the force trajectory and how the soil reacts to this applied force. Thus, it is only a coincidence that the bucket trajectory in Fig. 9b is a straight and horizontal line. With the pile of soil removed from the previous dig cycle, the bucket is filled in the cycle in Fig. 9c only by tracking the desired shape as there is only very little soil left in that part of the trench.

The shape error of the finished trench is shown in Fig. 10 before and after autonomous excavation. It shows a cut through the trench in longitudinal (x-)direction. The error bars represent the average error in lateral (y-)direction. The four-segment piecewise-planar shape was excavated with a mean error of 0.027m and a standard deviation of 0.035m. Fig. 10 shows additionally the σ -error bound. Note that this approach shows the same accuracy for horizontal planes as well as angled slopes. Additionally, the approach was also tested on a free-form curved shape shown in Fig. 11. The shape was successfully excavated with a mean error of 0.024m and a standard deviation of 0.032m. Fig. 10 and 11 in combination show that free-form shapes can be excavated with this approach with unprecedented high accuracy.

The results shown in this article are compared to our previous work in simulation [14], Shao *et al.* [26] and Maeda *et al.* [11]. In our simulation work, a mean error of 0.035m and 0.044m for two different shapes was achieved. With the

¹Video Reference: <https://youtu.be/3ZV78hA-HNA>

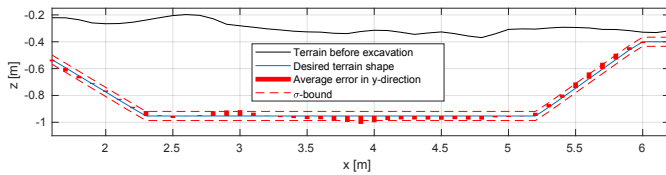


Fig. 10. The terrain, before the autonomous excavation starts and traced by the black line, shows an error of up to 0.7m at certain locations and a total error volume of $\sim 2.1\text{m}^3$ of soil that has to be removed. The desired shape is shown with the blue line and the error with red bars. The error bars show the error averaged in y-direction. The final state after the autonomous excavation shows a mean error of 0.027m with a standard deviation of 0.034m. Additionally, the red dotted line represents the σ -bound of the error.

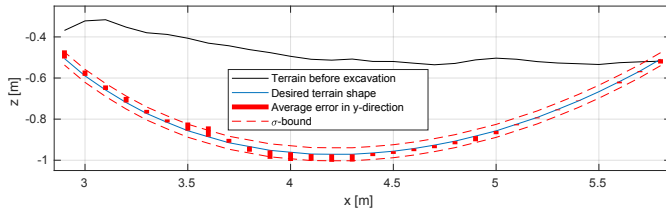


Fig. 11. The initial terrain (black line) was excavated to achieve the desired shape (blue line). The remaining average error is 0.024m shown with red bar plots and a standard deviation of 0.032m illustrated by the red dotted σ -bound lines.

advances shown in this article, the same or even better results could be achieved in real world experiments. As there is no other existing work on accurately excavating free-form shapes, we compare our approach to work on simple flat bottom trenches. Shao et al. [26] achieved an error of "less than 0.1m" whereas Maeda et al. [11] achieved a trench with 0.025m accuracy. However, the latter approach does not accurately capture the actual terrain shape with exteroceptive sensors. Instead, they only use the bucket edge motion to infer the actual elevation.

VIII. CONCLUSION

We have presented the necessary building blocks to autonomously and accurately excavate free-form shapes using a walking excavator. It is built on top of our previous work on soil independent excavation planning carried out in simulation. The necessary changes to a standard walking excavator to fully automate the machine for autonomous landscaping tasks are shown. Furthermore, robust elevation mapping fusing proprioceptive and exteroceptive sensing tailored to the challenges in excavation is introduced. A terrain collision free trajectory is planned on this map. The arm controller is aware of various machine limits and includes self collision avoidance for safe operation. The improved state machine enables the execution of autonomous excavation on real hardware.

Further development of the current framework will enable the creation of extended robotically fabricated landscapes. However, improving the excavation accuracy will most likely not be part of future work.

REFERENCES

[1] F. Gramazio and M. Kohler, *Procedural Landscapes. In The Robotic Touch: How Robots Change Architecture*. Zurich: Park Books, 2014.

[2] I. Hurkxkens, C. Girot, and M. Hutter, "Robotic landscapes. developing computational design tools towards autonomous terrain modeling," in *DISCIPLINES & DISRUPTION Proceedings Catalog of the 37th Annual Conference of ACADIA*, 2017, pp. 292–297.

[3] M. Haga, W. Hiroshi, and K. Fujishima, "Digging control system for hydraulic excavator," *Mechatronics*, vol. 11, no. 6, pp. 665–676, 2001.

[4] H. Cannon, "Extended Earthmoving with an Autonomous Excavator," *The Robotics Institute*, vol. PhD, 1999.

[5] Q. Ha, M. Santos, Q. Nguyen, D. Rye, and H. Durrant-Whyte, "Robotic excavation in construction automation," *IEEE Robotics Automation Magazine*, vol. 9, no. 1, pp. 20–28, Mar 2002.

[6] C. Semini, N. G. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella, and D. G. Caldwell, "Design of hyq—a hydraulically and electrically actuated quadruped robot," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 2011.

[7] M. Hutter, P. Leemann, G. Hottiger, R. Figi, S. Tagmann, G. Rey, and G. Small, "Force control for active chassis balancing," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 2, pp. 613–622, April 2017.

[8] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," in *Robotics: Science and Systems*, vol. 2, 2014, p. 9.

[9] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," in *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*. IEEE, 2001, pp. 145–152.

[10] P. F. Timon Homberger, Lorenz Wellhausen and M. Hutter, "Support surface estimation for legged robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[11] G. J. Maeda, I. R. Manchester, and D. C. Rye, "Combined ilc and disturbance observer for the rejection of near-repetitive disturbances, with application to excavation," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 5, pp. 1754–1769, Sep. 2015.

[12] J. Park, B. Lee, S. Kang, P. Y. Kim, and H. J. Kim, "Online Learning Control of Hydraulic Excavators Based on Echo-State Networks," *IEEE Transactions on Automation Science and Engineering*, jan 2017.

[13] T. Groll, S. Hemer, T. Ropertz, and K. Berns, "Autonomous trenching with hierarchically organized primitives," *Automation in Construction*, vol. 98, pp. 214 – 224, 2019.

[14] D. Jud, G. Hottiger, P. Leemann, and M. Hutter, "Planning and control for autonomous excavation," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2151–2158, Oct 2017.

[15] T. G. Phillips, N. Guenther, and P. R. McAree, "When the dust settles: The four behaviors of lidar in the presence of fine airborne particulates," *Journal of Field Robotics*, vol. 34, no. 5, 2017.

[16] P. Fankhauser, M. Bloesch, and M. Hutter, "Probabilistic terrain mapping for mobile robots with uncertain localization," *IEEE Robotics and Automation Letters (RA-L)*, vol. 3, no. 4, pp. 3019–3026, 2018.

[17] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[18] "Robot self filter," https://github.com/PR2/robot_self_filter, accessed: 2018-10-19.

[19] M.-J. Kim, M.-S. Kim, and S. Yong Shin, "General construction scheme for unit quaternion curves with simple high order derivatives," in *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, 01 1995, pp. 369–376.

[20] M. Zucker, N. Ratliff, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa, "Chomp: Covariant hamiltonian optimization for motion planning," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1164–1193, 2013.

[21] C. D. Bellicoso, C. Gehring, J. Hwangbo, P. Fankhauser, and M. Hutter, "Perception-less terrain adaptation through whole body control and hierarchical optimization," in *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, Nov 2016, pp. 558–564.

[22] B. Faverjon and P. Tournassoud, "A Local Based Approach for Path Planning of Manipulators With a High Number of Degrees of Freedom," in *IEEE International Conference on Robotics and Automation (ICRA)*, 1987, pp. 1152–1159.

[23] E. Coumans et al., "Bullet physics library," *Open source: bulletphysics.org*, vol. 15, no. 49, p. 5, 2013.

[24] M. L. Felis, "Rbd: an efficient rigid-body dynamics library using recursive algorithms," *Autonomous Robots*, pp. 1–17, 2016.

[25] E. Magrini, F. Flacco, and A. D. Luca, "Control of generalized contact motion and force in physical human-robot interaction," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[26] H. Shao, H. Yamamoto, Y. Sakaida, T. Yamaguchi, Y. Yanagisawa, and A. Nozue, "Automatic excavation planning of hydraulic excavator," in *Intelligent Robotics and Applications*, C. Xiong, H. Liu, Y. Huang, and Y. Xiong, Eds. Berlin, Heidelberg: Springer, 2008, pp. 1201–1211.