This is the submitted version of a paper published in *Robotics and Autonomous Systems*.

Citation for the original published paper (version of record):

Computational exploration of robotic rock loading
*Robotics and Autonomous Systems*
https://doi.org/10.1016/j.robot.2018.04.010

Access to the published version may require subscription.

N.B. When citing this work, cite the original published paper.

Permanent link to this version:
http://urn.kb.se/resolve?urn=urn:nbn:se:umu:diva-147237
Computational exploration of robotic rock loading

Daniel M. Lindmark\textsuperscript{a}, Martin Servin\textsuperscript{a}

\textsuperscript{a}Umeå University, Umeå, Sweden

Abstract

A method for simulation-based development of robotic rock loading systems is described and tested. The idea is to first formulate a generic loading strategy as a function of the shape of the rock pile, the kinematics of the machine and a set of motion design variables that will be used by the autonomous control system. The relation between the loading strategy and resulting performance is then explored systematically using contacting multibody dynamics simulation, multiobjective optimisation and surrogate modelling. With the surrogate model it is possible to find Pareto optimal loading strategies for dig plans that are adapted to the current shape of the pile. The method is tested on a load-haul-dump machine loading from a large muck pile in an underground mine, with the loading performance measured by productivity, machine wear and rock debris spill that cause interruptions.

Keywords: robotic excavation, autonomous loading, rock pile, multibody dynamics, discrete element, surrogate modelling

2010 MSC: 00-01, 99-00

1. Introduction

Loading piles of fragmented rock is a challenging task, for a machine operator and even more for an autonomous system. For this reason mining vehicles for loading, hauling and dumping (LHD) are still not fully automated but rely on a human operator controlling the loading task, by remote or from within the machine. With fully autonomous control, productivity is lower and maintenance needs are higher than with operators \cite{1}. The difficulty lies in perceiving the state of the pile, planning and controlling a movement that quickly fills the bucket, avoiding excessive wear on the machine, and leaving the pile in a state well-suited for continued loading. Interruptions in the loading occur when rock debris are left in front of the pile or if an overhang is created. Improper handling, like ramming into a large boulder or filling the bucket unevenly, causes damage to the machine.

This paper describes and tests a method for simulation-based development of systems for robotic loading of fragmented rock piles. The idea is to formulate a generic loading strategy as a function of the observed shape of the pile and the kinematics of the machine, and to use multiobjective optimisation and contacting multibody dynamics to determine the strategy that maximises performance from a large set of realistic simulation experiments. Surrogate modelling is used to explore the loading strategy design space and to identify the Pareto optimal strategies. The aim is to show that this is a feasible and useful approach for developing and testing automation strategies for robotic rock loading. The method is tested on a specific LHD system for underground mining. Statistically significant relations between the loading strategy and performance are found, and the computational complexity of the method is analysed.

Robotic loading can be divided into pile analysis, high-level dig planning and low-level loading control. In this paper, we apply the method to the dig planning task only. The bucket motion is given a generic representation depending on some design variables that parametrise the entry point in the pile, the dig depth, bucket curl and target fill ratio. The dig motion planner takes the vehicle kinematics, observed pile geometry and the design variables as input and outputs a specific bucket trajectory and corresponding motion plan for the machine. Objective functions are introduced for the productivity, joint damage and amount of rock debris in front of the pile causing interruptions. Together, the objective functions determine the loading performance. The simulations are based on a detailed 3D multibody dynamics model of the vehicle and the rock pile. The machine model includes actuator models and a control algorithm to execute the motion plan. The actuators have limited force, and the machine may lose ground traction if the pile resistance is large enough to cause frictional slippage. Consequently, the simulated bucket trajectory may deviate from the plan. This provides realism that is not available when using a kinematic model of the vehicle or bucket. For each simulation, the forces and moments in the joints of the machine are measured, as well as the loading cycle time, final bucket fill ratio as well as the resulting state of the pile. Both the actuating forces and the constraint forces keeping the joints in place are measured, and may be analysed for power consumption and mechanical wear, respectively.

A surrogate model is constructed for the relationships between the design variables and the objective functions, i.e., between the loading strategy and loading performance. This enables systematic exploration of the design space and the possibility to steer the computational resources to the most interesting parts of the design space. Using the surrogate model, it is easy to answer specific questions regarding loading strate-
gies, for instance, whether it is beneficial to dig deep or shallow, attempt to overfill the bucket or not, and whether to jerk the bucket at pile breakout.

Recent literature overviews of robotic loading of piled and fragmented rock material can be found in [1][2][3]. Hemami introduced kinematic modelling of an LHD as a robotic manipulator and suggested an end-effector force model to study bucket trajectories that minimise energy consumption [4]. Singh [5] presented a planning algorithm that, through exhaustive search in the action space, finds the best plan that maximises the swept volume and minimises the maximum and accumulated torque for a given machine and pile geometry in 2D. Many methods for automated loading are based on scripted bucket trajectories generated from 3D pile shape measurements [6][7]. In order to adjust the motion to the actual and varying conditions in a pile, this is combined with a method for shifting between scripted trajectories, based on the observed resistance force [8]. An alternative approach, better suited for dense inhomogeneous material, such as fragmented rock, is compliance control, where the bucket motion is modified continuously based on the observed compliance of the pile. One instance of this is impedance control [9], where the applied force depends on the bucket velocity. Another instance is admittance control [1], where the bucket target velocity depends on the sensed force resistance.

Computer simulation is an attractive complement to physical experiments. It allows for controlled and repeatable experiments of many scenarios that can occur in rock excavation. The net loading performance depends both on how the machine reacts to the pile and how the pile is affected by the machine. The resulting pile state should be suitable for continued loading. Filla [10] used simulations based on the discrete element method (DEM) to validate algorithms for parametric generation of trajectories for automated loading. Like previous DEM based studies of the interaction between a bucket and granular material [11][12][13], it uses a kinematic model for the bucket motion. Kinematics has the drawback of allowing bucket trajectories that are not realisable with the vehicle’s limited power, force transmission and ground traction. Several examples can be found [14][15] of multibody dynamics simulation of automatic loading with empirical models for the reaction force from the excavated media, represented by its surface, and for estimating the bucket filling and volumetric displacement of the material. However, efficient, this approach is limited to homogeneous and fine-grained materials, compared to the length-scale of the bucket, and the model does not capture the, often erratic, flow behaviour of the material that lead to the new shape of the pile and spillage and rock debris around the pile from the loading motion. To capture this requires dynamic models of both the vehicle and the rock pile. To the best of our knowledge, there are no previously published scientific papers on this. The reason, presumably, is that few multibody simulation software supports both granular materials, articulated mechanisms and their control and actuation. The alternative is to do co-simulation with distinct software [16], but this is both demanding to construct and associated with computational overhead [17]. The work in this paper is based on a nonsmooth multibody dynamics formulation, which applies both to machinery and granular material [18][19], time-integrator and hybrid direct-iterative split-solver that enables fast and stable simulation of the full system [20][21].

1.1. Main contribution of the paper

The main contribution of the paper is a methodology for exploring how the loading performance depend on a set of dig plan variables, by generating and analysing a surrogate model from a large amount of simulated (or real) loading cycles. This can be used to construct an automatic dig planner that take measurements of the observed pile surface as input, through the Pareto optimal design variables, compute and outputs a motion plan for highest loading performance. Having a systematic methodology become increasingly important as the size and dimensionality of the design space increase with the computational power. This may be considered a first step in developing autonomous loading by applying machine learning on large sets of synthetic training data.

The rigid multibody simulation model is novel in including both the dynamics of the machine and the rock pile, and is therefore described in some detail. An important effect of including a dynamic model of the pile and having granular level of detail is that loading motions that results in poor pile states can be detected and penalized in the search of optimal loading strategies. Furthermore, the presence of large boulders, potentially damaging to the machine, is automatically included in the simulation analysis. The precise dig plan variables and loading performance functions used in this paper serves as examples. The method applies just as well for other choices.

2. System model

2.1. Non-smooth multibody dynamics

The machine and the rock pile are modelled as a 3D contacting rigid multibody system on descriptor form [22]. This means that all bodies are represented explicitly with their six degrees of freedom. The bodies are coupled by kinematic constraints for modelling of mechanical joints and actuators. This extends the Newton-Euler equations of rigid body motion to a set of coupled differential algebraic equations (DAEs). An LHD interacting with a rock pile also involve impacts, resting and sliding frictional contacts, limits on joints and motor torque that cause sudden changes in the number of system variables, their bound and connectivity. To enable fast simulations, with large time-step implicit integration, one must treat the dynamics that occur on short timescales compared to the time-step, as non-smooth [23][24][18]. This means that velocities may change discontinuously in accordance with some contact and impact law, expressed in terms of inequality and complementarity conditions, in addition to the equations of motion and DAEs. The alternative is to use fine enough temporal resolution for all the dynamics to appear smooth and be modellled in terms of DAEs or ordinary differential equations alone. In general, that is an intractable approach for full-system simulations of mechatronics system with any substantial complexity.
Following \( [20, 25] \), we assume the following equations for modelling the machine and the granular material as a nonsmooth multibody dynamics system:

\[
\begin{align*}
\text{M} \dot{\text{v}} - \text{G}^{\text{J}}_j \lambda_j - \text{G}^{\text{I}}_j \dot{\lambda}_j - \text{G}_j \dot{\lambda}_j &= f_{\text{ext}}, \\
0 \leq \varepsilon_n \lambda_n + g_n + \tau_n \text{G}_n \text{v} &\perp \lambda_n \geq 0, \\
\gamma_t \dot{\lambda}_j + \text{G}_j \text{v} &= 0, |\dot{\lambda}_j| \leq \mu_t |\lambda_j|, \\
\varepsilon_i \dot{\lambda}_j + \eta_i g_i + \tau_i \text{G}_i \text{v} &= \omega_j, \quad \lambda_j \leq \lambda_j^t,
\end{align*}
\]

where \( \text{v} = \text{x} \) is the system velocity vector, \( \text{M} \) is the mass and inertia tensor matrix and the position vector \( \text{x}(t) \in \mathbb{R}^6 \) represents both translational and rotational degrees of freedom of the \( N_p \) rigid bodies. The first equation is the Newton-Euler equation of motion for rigid bodies with external (smooth) forces \( f_{\text{ext}} \) and constraint force \( \text{G}^{\text{J}} \lambda \) with Lagrange multiplier \( \lambda \) and Jacobian \( \text{G} \), divided into normal (n), tangential (t) and articulation (a) constraints. Equations (2)-(3) are the Signorini-Coulomb conditions with constraint regularisation and stabilisation terms \( \varepsilon_n, \tau_n \) and \( \gamma_t \). With \( \varepsilon_n = \tau_n = 0 \), Eq. (2) state that bodies should be separated or have zero overlap, \( \text{G}(\text{x}) \geq 0 \), and if so the normal force should be non-cohesive, \( \lambda_n \geq 0 \). With \( \gamma_t = 0 \), Eq. (3) state that contacts should have zero relative slide velocity, \( \text{G}_j \text{v} = 0 \), provided that the friction force remains bounded by the Coulomb friction law with friction coefficient \( \mu_t \). The constraint force, \( \text{G}^{\text{I}}_j \dot{\lambda}_j \), arise for articulated rigid bodies jointed with kinematic links and motors represented with the generic constraint equation (4). With \( \varepsilon_i, \tau_i, \omega_i = 0 \) and \( \eta_i = 1 \), it becomes an ideal holonomic constraint \( \text{G}(\text{x}) = 0 \) that may represent a hinge. For \( \varepsilon, \eta \neq 0 \) and \( \tau = 1 \), it becomes an ideal Pfaffian constraint \( \text{G} = \omega_j \) that may emulate a motor striving to achieve a target joint velocity \( \omega_j \) with limits \( \lambda_j^t \) on the motor constraint force. With \( \varepsilon, \eta, \tau \neq 0 \) it can represent a generic constraint with compliance and damping. An important feature of using descriptor form (no coordinate reduction) and joint compliance is that the constraint reaction forces and moments that keep the kinematic joints in place are computed and known explicitly. Details of the numerical time-integration scheme, solver and the mapping from physical parameters to simulation parameters are given in Appendix A.

### 2.2. Machine model

We choose to consider a load-haul-dump (LHD) machine, designed to operate in underground mines along narrow paths. An LHD is a four-wheeled vehicle with an articulation steering joint that separate the rear driving unit from the front loading unit that is equipped with a hydraulic actuated boom and bucket. The machine is modelled as a rigid multibody system consisting of 18 bodies and 24 joints. For actuation, the wheel, boom and bucket joints are equipped with hinge motor constraints that apply torque to achieve a target joint velocity set manually or by an automatic control system. For the dimensions, mass and powertrain characteristics we choose a specific LHD, namely, a Sandvik LH621 with specifications listed in \( [26] \) and CAD model \( [27] \) illustrated in Fig. 1. The machine has an operating weight of 56.4 tonne and is 12 m long and 3.8 m wide. The wheel radius is \( r_w = 1.0 \) m. The standard bucket volume capacity is \( V_b = 8.0 \) m³, weight capacity 21 tonne and maximum breakout force 378 kN. The key body and joint data are listed in Table 1 and 2. The front loading unit is depicted in Fig. 2. To make the powertrain less ideal and match the maximum breakout force, the motor constraints in Eq. (4) are given maximal torque limits listed in Table 2 and a compliance of 10⁻⁸ m/N. These numbers were identified in numerical experiments where the breakout force was measured. The Z-bar linkage mechanism connecting the boom and bucket is modelled explicitly with joint limits on the cylinder extension. Together this hinders the simulated machine from performing a motion that is not physically viable.

The velocity at the tip of the bucket in world coordinates is denoted \( \text{s}_b(t) \in \mathbb{R}^6 \). It is related to the joint velocities, \( \text{q}(t) \), by the relation \( \text{s}_b = \text{J} \text{q} \) with Jacobian \( \text{J} \equiv \partial \text{s}_b / \partial \text{q} \). For a scooping motion with no sideways movement the Jacobian becomes

\[
\text{J} = \begin{bmatrix}
0 & -r_{b1} \sin(\phi_1 + \phi_2) + r_{b2} \sin(\phi_1) & -r_{b3} \sin(\phi_1 + \phi_2) \\
0 & 0 & 0 \\
1 & r_{b1} \cos(\phi_1 + \phi_2) + r_{b2} \cos(\phi_1) & -r_{b3} \cos(\phi_1 + \phi_2) \\
0 & 0 & 0 \\
0 & -1 & -1 \\
0 & 0 & 0
\end{bmatrix}
\]

where the joint velocity vector for the wheel, boom link and bucket is \( \text{q} = [\dot{\phi}_w, \dot{\phi}_1, \dot{\phi}_b]^T \) and the joint angles are defined as in \( [22] \).

### 2.3. Rock pile and tunnel model

We consider the loading of piled fragmented rock, known as muck, at the end of a narrow tunnel, known as a drift. The

<table>
<thead>
<tr>
<th>Body</th>
<th>Mass [ton]</th>
<th>Size [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>wheel</td>
<td>1.9</td>
<td>1.0</td>
</tr>
<tr>
<td>rear unit</td>
<td>27</td>
<td>5.9 × 3.0 × 2.9</td>
</tr>
<tr>
<td>front unit</td>
<td>15</td>
<td>4.0 × 3.0 × 2.9</td>
</tr>
<tr>
<td>boom</td>
<td>3.7</td>
<td>2.5</td>
</tr>
<tr>
<td>Z-bar</td>
<td>0.88</td>
<td>1.6</td>
</tr>
<tr>
<td>bucket</td>
<td>6.4</td>
<td>2.0 m, ( V_b = 8.0 ) m³</td>
</tr>
</tbody>
</table>

Table 1: The machine’s main bodies and their key properties.

Figure 1: The 3D model of the LHD machine.
muck pile is assumed to have been formed by blasting tall slices of solid rock at the end of the tunnel roof, a technique called sublevel caving [28]. The front of the pile is known as the draw point. As muck is extracted from the draw point, new muck will flow from above. The tunnel is modelled as a static geometry, shown in Fig. 3 and 4. It is 6.5 m wide and 5 m tall with a slightly curved roof. At the drawpoint, the drift opens up in an vertical cone-shaped void that is filled with blasted muck. The inclination of the void wall facing the tunnel, where the last blasting occurred, is 10°.

The muck model consists of rocks of six different convex shapes, shown in Fig. 5 and size distribution ranging between 0.3 and 0.8 m. The assumed material properties are characteristic for rock material and are listed in Table 3. The rocks are treated as rigid bodies with frictional, non-cohesive contacts and uniform mass density obeying the multibody dynamics equations of motion Eq. (1)-(4) with numerical integrator described in Appendix A. The friction coefficient between the rock and tunnel, rock and bucket, as well as tire and tunnel is set to 1.0. In reality the particle size ranges down to millimetre. Up to 20 – 50 % of the muck mass may consist of particles smaller than 0.1 while the largest rocks can be wider than 0.8 m [29].

The simulated muck pile is created by emitting rocks from above into the vertical void above the draw point and simulating until the system is fully relaxed. First, a rear boundary of rocks is created in the back of the void using a confining plane. This 1 m thick wall is then made static and the confining plane is removed. Next, the remaining void is filled with rocks until the level reaches up to the emitter that is placed at the height of 8 m, which is 3 m above the tunnel roof. This form a natural pile state with an angle of repose that is estimated to 35°. During loading with the LHD, the rock level drops and new material is emitted to maintain the level. This is illustrated in Fig. 4. In average the simulations involve 5500 rocks, not including the static rocks forming the rear boundary. In reality, the blasted rock mass stretches many times the height of the tunnel.

3. Loading operation model

The robotic loading problem can be divided into pile analysis, high-level dig planning and low-level loading control. In this article, we focus on the dig planning task. To execute the dig plan, a velocity feedforward motion control is used for the
target joint velocities. The loading operation model is summarised as an algorithm in Sec. 3.2.

We use a Cartesian coordinate system with \( x \) axis in the tunnel direction, \( y \) in the perpendicular horizontal direction and \( z \) for the vertical distance from the tunnel ground. The gravity is antiparallel with the \( z \) axis.

### 3.1. Pile analysis

Before making a dig plan and starting a loading cycle, the pile is measured with a simulated sensor device that outputs a 3D point cloud that can be obtained using a LIDAR or time-of-flight 3D camera directed at the pile surface. A pile front curve, \( x_f(y) \), is obtained from \( 0 \approx h(x_f, y) \). The pile volume bound by the floor and the height field surface is discretized in vertical slices, stretching from the front to the rear of the tunnel. The centre of mass position of each slice is computed assuming a spatially uniform mass distribution under the height-field surface. A centre of mass curve \( x_m(y) \) is thus obtained, and will be used to determine a suitable attack point. See Fig. 6 and 8 for illustrations. The local slope angle of the pile is computed as \( \xi(y) = \arctan\left[ h(x_f, x_m(y))/(x_m(y) - x_f) \right] \), using the pile height at the centre off mass point.

### 3.2. Dig plan

A dig plan is a bucket path, \( x_b(s) : \mathbb{R} \to \mathbb{R}^6 \), that we parametrise by the path-length, \( s \in [s_0, s_{\text{end}}] \). The bucket path defines a swept volume that intersects the undisturbed pile geometry to form a nominal dig volume, \( V_d \). The relation between the bucket velocity, \( \dot{x}_b = \left[ \dot{x}_b^T, \dot{\alpha}_b^T \right]^T \), and the actuator joint velocity, \( \dot{q}(s) = \left[ \dot{\varphi}_m(s), \dot{\varphi}_l(s), \dot{\varphi}_b(s) \right] \) is \( \dot{x}_b = J\dot{q} \) with the Jacobian given in Eq. 3. This assumes zero slip, in which case the motion is kinematically nonredundant. It is furthermore assumed that the LHD runs parallel to tunnel direction and have access to sensor data about the muck pile surface facing the vehicle, at least the front curve, \( x_f(y) \), local angle of repose, \( \xi(y) \), and centre of mass curve, \( x_m(y) \). A generic dig path that mimics the planning of human operators on flat ground is adopted from [10]. It is illustrated in Fig. 7 with waypoints \( s_0, ..., s_4 \). It starts by entering the pile \( (s_0) \) with the bucket lowered and aligned with the ground while moving horizontally into the pile to a given depth \( (s_1) \). The bucket tip then moves diagonally, carving a slice of constant depth by lifting, curling and running forward simultaneously. This occurs until breakout initiation \( (s_2) \), where the vehicle is halted and the bucket is rotated any remaining angle to the maximally tilted state \( \varphi_{b-m} \) and waypoint \( s_3 \). Finally, the bucket moves horizontally backwards from the pile to the endpoint \( s_4 \). The bucket curling between segment \( s_1 \) and \( s_3 \) is defined by the initial and final bucket rotation angles, \( \varphi_b(s_1) = \varphi_{b-o} \) and \( \varphi_b(s_2) = \varphi_{b-o} + (\varphi_{b-m} - \varphi_{b-o}) \kappa \), respectively.

We use the bucket curl, \( \kappa \in [0, 1] \), between \( s_1 \) and \( s_3 \) as one of the dig strategy design variables. The remaining freedom for the dig plan is used to introduce additional design variables, \( \alpha, \beta \) and \( \gamma \), explained below. We use \( \gamma \in [0, 1] \) to control the entry point that we denote \( y_e \). The centre of mass positions \( x_m(y) \) are ordered by size and mapped to \( y \) such that \( y = 1 \) corresponds to the front-most centre of mass point and \( y = 0 \) to the rear-most point. The target dig volume is chosen as some \( \alpha \in [0, 2] \) of the maximum capacity, and we denote it by \( V_d^\alpha = \alpha V_b \). A bucket path can be shallow or deep relative to the pile surface. We use \( \beta \in [0, 1] \) to distinguish between deep \( (\beta \geq 0) \) and shallow \( (\beta \leq 1) \) digging. The bucket lift height

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of rocks</td>
<td>( \sim 5.500 )</td>
</tr>
<tr>
<td>size distribution</td>
<td>([0.3, 0.4, 0.6, 0.8] ) m</td>
</tr>
<tr>
<td>mass density</td>
<td>( 2.500 ) kg/m(^3)</td>
</tr>
<tr>
<td>elastic modulus</td>
<td>( 1.0 ) GPa</td>
</tr>
<tr>
<td>Poisson ratio</td>
<td>( 0.3 )</td>
</tr>
<tr>
<td>rock-rock restitution</td>
<td>( 0.0 )</td>
</tr>
<tr>
<td>machine-rock restitution</td>
<td>( 0.1 )</td>
</tr>
<tr>
<td>rock-rock friction coeff.</td>
<td>( 0.9 )</td>
</tr>
<tr>
<td>other friction coeff.</td>
<td>( 1.0 )</td>
</tr>
</tbody>
</table>

Table 3: Rock pile properties.

![Fig. 6](image1.png)  
Top view illustration of the bucket approaching the pile that is analysed in a discrete grid and segmented vertical slices. The slices with the front-most and rear-most center of mass are indicated with pink.

![Fig. 7](image2.png)  
Sideview illustration of the pile with the bucket approaching. A generic dig plan with waypoints \( s_0, ..., s_4 \) is illustrated by the dashed purple line. The pile slope is illustrated by the blue line. A maximally deep dig plan \( (\beta = 0) \) is also shown with the dashed-dotted green line, in which case the waypoints \( s_1 \) and \( s_2 \) coincide. The bucket is curled between \( s_1 \) and \( s_3 \).
Figure 8: Sideview illustration of the pile during excavation along a dig plan illustrated by the dashed purple line. The bucket tip move parallel to the estimated slope of the pile between the waypoints $s_1$ to $s_2$.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Range</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>[0.2, 2]</td>
<td>bucket fill ratio $V_d/V_b$</td>
</tr>
<tr>
<td>$\beta$</td>
<td>[0, 1]</td>
<td>dig height $h_z/h_p$</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>[0, 1]</td>
<td>entry point mass proximity</td>
</tr>
<tr>
<td>$\kappa$</td>
<td>[0, 1]</td>
<td>bucket curl in segment $s_1$ to $s_2$</td>
</tr>
</tbody>
</table>

Table 4: Dig plan design variables

$h_{ez} = z_0(s_2)$ at waypoint $s_2$ is then set to $h_{ez} = \beta h_p$, with the pile height $h_p = r_0 \sin(\xi)$ at a distance $r_0 = 4.24$ m from the pile front, which is the furthest point on the pile surface that the bucket tip can reach before the front wheels touch the pile front.

The dig depth, $x_{d1} = x_{d2}$, then follows from a geometric analysis of the swept volume that integrates to $V_d^s$. The full set of design variables, $x = [\alpha, \beta, \gamma, \kappa]$, their ranges and meaning are listed in Table 4.

The bucket velocities in world coordinates, $v_x(s), v_y(s)$ and $\omega_z(s)$, along the dig plan are chosen as in Fig. 9 with $v_{max} = 1.3$ m/s. $v_x(s) = \tan(\xi) \cdot v_y(s)$ for $s \in [s_1, s_2]$, and $\omega_z(s) = \omega_0(s_2) \cdot \frac{v_x(s)}{h_{z2}}$. The actuator joint velocities, $\dot{q} = J^{-1}\dot{s}_b$, thus become

$$\dot{\varphi}_u(s) = r_u v_x(s) + f_{ul}(s) \varphi_1(s) + f_{ub}(s) \varphi_2(s),$$

$$\dot{\varphi}_1(s) = \min\left[f_{ul}(s) \omega_1(s) + f_{ib}(s) v_x(s), 0\right],$$

$$\dot{\varphi}_2(s) = -\varphi_1(s) - \omega_2(s),$$

with

$$f_{ul}(s) = -\frac{1}{r_u^2} \left[r_u \cos(\varphi_1 + \varphi_2) + r_h \cos(\varphi_1)\right],$$

$$f_{ib}(s) = -\frac{b_u}{r_u^2} \cos(\varphi_1 + \varphi_0),$$

$$f_{ul}(s) = \frac{b_u}{r_u} \sin(\varphi_1 + \varphi_0) / \sin(\varphi),$$

$$f_{ib}(s) = -\frac{1}{r_u^2} \sin(\varphi).$$

and $\varphi_0(s) = \varphi_0(b) + \int_{s_0}^{s} \dot{\varphi}_0(b) \, dt$. Clamping of $\varphi_1(s) \geq 0$ prevents pushing the bucket downwards into the ground.

3.3. Loading control

A velocity feedforward motion controller is used to execute the dig plan. The bucket tip position, $x_b(t)$, is tracked to obtain the corresponding closest point, $s$, on the dig path. The corresponding target joint velocities are obtained and fed as input signal to the joint motors that deliver a torque computed by the numerical solver. Due to the vehicle inertia and pile resistance, the torque required to reach the target velocity may exceed the maximum torque of the actuator model. The vehicle may also lose traction and start slipping. Consequently, the dig trajectory will deviate somewhat from the plan. The deviation may be large if the swept volume largely exceeds the maximum capacity or if the bucket encounters large resistance. We do not apply a feedback control to return the bucket to the dig plan. One measure that is taken to reduce the effect of a potential deviation is to split the dig path parameter into translational and rotational parameters, $s \rightarrow (s_x, s_y, s_0)$ for the parametrization of the bucket velocities profiles $v_x(s_x), v_y(s_y)$ and $\omega_0(s_0)$ in Fig. 9.

Also, the controller automatically jumps to the next segment in the dig plan if the time-duration in the current segment exceeds what is expected by more than 5 s.

3.4. Loading algorithm

The loading operation model can be summarised with the following algorithm.

1. Analyse the current state of the pile.
   (a) Detect the pile surface height field $h(x, y)$.
   (b) Compute the front $x_f(y)$, centre of mass $x_m(y)$ and slope $\xi(y)$.

2. Compute a dig plan given $x = [\alpha, \beta, \gamma, \kappa]$.
   (a) Determine the entry point $y_f$ from $x_m(y)$ and $\gamma$.
   (b) Compute slope average $\xi_f$ around $y_f$.
   (c) Generate the bucket path $x_b(s), x_f(s)$ and $\omega_0(x)$.
   (d) Compute the bucket velocity $v_b(s)$ along the path.
   (e) Compute the actuator joint velocities $\dot{q}(s)$.

3. Perform loading.
   (a) Enter the pile at given position and velocity.
   (b) Track the bucket and set target joint velocity to $\dot{q}(s)$.
   (c) Control the actuator torque $\tau_l \in [\tau^u, \tau^d]$ for target joint velocity.
3.5. Loading performance

The loading performance is defined by three objective functions: the production, $\phi_1$; the structural damage, $\phi_2$; and the amount of rock debris, $\phi_3$, in front of the excavated pile. The objective functions are defined

$$\phi_1 = -\frac{m_{\text{bucket}}}{t_{0-4}},$$

$$\phi_2 = \frac{1}{\rho_{\text{char}} \sqrt{\gamma_s}} \int_0^t \max[|\tau(t)| - \tau_c, 0] \, dt,$$

$$\phi_3 = m_{\text{debris}}.$$  

The production is the excavated rock mass carried in the bucket, $m_{\text{bucket}}$, divided by the measured loading time, $t_{0-4} = t_4 - t_0$. The damage model assumes that damage occur if the magnitude of the bending or twisting joint moment, $\tau$, exceed a critical value, $\tau_c$. Observe that we distinguish between bending moment, that acts to maintain a mechanical joint in place, and actuator torque, $\tau$, that acts to drive a joint motor at a target velocity. The net damage is assumed proportional to how much the critical bending moment is exceeded and for how long time. We normalise this with the characteristic loading time $t_{0-4} = 12$ s. For simplicity, we only compute the damage in the hinge joint connecting the bucket to the boom link. As a critical bending or twisting moment we choose $\tau_c = 145$ kNm. This value was measured in a simulation where the bucket was pushed towards a static obstacle at the corner of the bucket by driving the vehicle forward with maximum engine torque. The damage model emulates an unevenly loaded bucket or ramming a heavy object. The amount of rock debris is the total mass of the rocks, $m_{\text{debris}}$, that is left on the ground but no longer in contact with the pile after loading.

4. Simulation

For each choice of design variable, $x = [x, y, \beta, \gamma, \kappa]$, the corresponding loading performance, $\phi(x) = [\phi_1, \phi_2, \phi_3]$, is computed from a sequence of 20 rock loading simulations. At the end of each simulation the rock mass in the bucket, $m_{\text{bucket}}$, is measured and any rock debris, $m_{\text{debris}}$, from the previous excavation is removed. New rocks are emitted from atop the void until the nominal material height is restored, see Fig. 3. The pile is then left to relax into a resting state. The next simulation begins with analysis of the new pile state and generation of a dig plan, as described in Sec. 3.1 and 3.2. The LHD is placed in a position aligned with tunnel and heading straight towards the entry point, $\gamma_y$. Each sequence of 20 loading simulations start from the same original pile, but it evolves differently depending on the motion design variable. This way any loading strategy that systematically results in poor pile states will automatically lead to a poor loading performance. An image sequence from simulations is shown in Fig. 10 and 11. Videos from simulations is available as supplementary material at http://unit.cs.umu.se/loading/

The system and loading operation models, described in Sec. 2 and 3, are implemented in the simulation engine AGX Dynamics [30] and run with the settings listed in Table 5. The adaptive model reduction technique [25] merges regions of co-moving or stationary rocks into rigid aggregates to reduce the computational time considerably. To avoid the reduced rock pile from being artificially rigid at first impact with the machine, a model-refinement-sensor of length $r_b$ is attached in front of the bucket. It was verified that this did not affect the dynamics of the rock pile and vehicle by any significance. The average time for a loading cycle was 12 s. The computational time for one simulation was approximately 500 s while running four threads in parallel on a desktop computer with Intel(R) Core(TM) Xeon X5690, 3.46 GHz, 48 GB RAM on a Windows 64 bit system.

5. Surrogate modelling

A surrogate model [31], also known as a response surface model, is an approximation of an original unknown function. The original function, $y = \phi(x)$ with $x \in \mathbb{R}^n$, is usually very expensive to evaluate, restricting the maximum number of evaluations, $N$, while the surrogate model, $\hat{\phi}(x)$, is cheap to evaluate but adds some error to the result, $\phi(x) = \hat{\phi}(x) + \varepsilon$. When building a surrogate model, the $N$ sample points are chosen cleverly and included in a sample plan, $X = [x_1, \ldots, x_N]^T$. The surrogate model, $\hat{\phi}(x)$, is fitted to the response, $Y = [y_1, \ldots, y_N]^T$, from evaluating $\phi(x)$ on the sample plan. The surrogate model is then used in algorithms that require many function evaluations, e.g., for optimisation or design space exploration.

We use the regression Kriging method. The surrogate model is represented as

$$\hat{\phi}(x) = w^T \Psi = \sum_{i=1}^N w_i \psi_i(|x - x_i^0|),$$

with one Kriging basis function for each sample point $(i)$

$$\psi_i(|x - x_i^0|) = \exp\left(-\sum_{j=1}^n \theta_i |x_j - x_j^0|^{p}\right).$$

The model parameters are the weight function $w$, radial width $\theta$ and exponent $p$. To construct the model is to compute the model parameters that best fit the sample data according to

$$[\Psi + \lambda I] w = Y$$


| Time-step | 0.01 s |
| Impact threshold | 0.1 m/s |
| Solver | hybrid direct-iterative split |

Table 5: The AGX Dynamics simulation settings that were used in the simulations.
Figure 10: Side view image sequence from a loading simulation with design variables $x = [1.3, 0.85, 1.0, 0.6]$.

Figure 11: Top view image sequence from a loading simulation with design variables $x = [1.3, 0.85, 1.0, 0.6]$. 
where the Gram matrix is $\Psi^{(T)} = \psi(x^{(i)}) - x^{(i)}$ and a regularisation parameter $\lambda$ has been added to avoid overtraining the surrogate model to noisy sample data. The basis parameters $\theta$ are identified using Maximum Likelihood Estimation (MLE) while $p$ is held fix, and the weights $w$ are computed as the least squares solution to Eq. (18).

To test how well the model approximates the data, a portion of the observation data is left out from the sample points and form a validation set $\phi(k)$. The model accuracy is then computed as the normalized root mean square error (NRMSE)

$$E_j \equiv \frac{1}{y_j^{\text{range}}} \frac{1}{N_j} \sum_{k} \left( \phi_j^{(k)} - \tilde{\phi}_j^{(k)} \right)^2,$$

where $y_j^{\text{range}} = \max(Y_j) - \min(Y_j)$ and $N_j$ is the number of sample points in the validation set. In the present work, we use $N_j/N = 0.2$, chosen randomly in the design space. No results from the validation set are used to fit the surrogate model.

We use the Matlab toolbox SUrrogate MOdelling (SUMO)\textsuperscript{12} \textsuperscript{33} to build the surrogate model. It supports many different models and provides useful features, e.g., methods for initial and sequential sampling of the design space, plotting, profiling and validation of the surrogate model. Finding Pareto optimal solutions are done using the genetic multiobjective optimisation algorithm NSGA-II, implemented in Matlab’s optimisation algorithm gamultiobj, with population size 300 and every other setting default. Table 6 lists the SUMO features and settings that are used.

### 6. Computational exploration

Designing a well-performing robotic rock loading system involves meeting multiple competing objectives $\psi_k$, $k = 1, \ldots, 3 \equiv N_0$ introduced in Sec. 3.5. Therefore it is suitable to approach this as a multiobjective optimisation problem

$$\begin{align*}
\text{minimise} & \quad [\psi_1(x), \ldots, \psi_{N_0}(x)] \\
\text{subject to} & \quad [x_j^i \leq x_j \leq x_j^r, j = 1, \ldots, n_d] \geq S \\
& \quad n\geq S
\end{align*}$$

and systematically explore the behaviour of $\psi_k$ over the design space, $S$, to identify and visualize the trade-off surfaces, known as the Pareto front, $P \subset S$. A Pareto optimal solution, $x^* \in P$, is a solution that cannot be improved in any objective without worsening it in at least one of the other objectives, i.e., there is no $x \in S$ such that $\psi_k(x) < \psi_k(x^*)$ for all $k = 1, \ldots, N_0$. The set $P$ of Pareto points form the Pareto front $\psi(x^*)$. This form the basis for deciding on a loading strategy that is optimal for a given trade-off between the objectives. Computing these Pareto front require many evaluations of the objective functions and building accurate surrogate models from a limited set of simulations is essential.

We propose the following procedure for computational exploration of robotic rock loading system:

1. Construct a system model and a loading operation model that depends on a number of design variables, $x \in S$.

### Table 6: SUMO features that are used.

<table>
<thead>
<tr>
<th>Model type</th>
<th>Regression Kriging</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial design</td>
<td>Latin hypercube</td>
</tr>
<tr>
<td>Sequential design</td>
<td>Lola-Voronoi and error</td>
</tr>
<tr>
<td>Target</td>
<td>$N = 200$ sample points</td>
</tr>
<tr>
<td>Number of validation points</td>
<td>20%</td>
</tr>
</tbody>
</table>

(a) Rigid multibody vehicle with actuator models.
(b) Rigid multibody rock pile.
(c) Dig plan and loading control as a function of the machine and pile.
(d) Choose loading design variables and their range.

2. Introduce objective functions, $\psi$, that define the loading performance.
(a) Production.
(b) Damaging forces causing machine wear.
(c) Poor pile state that cause interruption.

3. Build surrogate models, $\phi$, from simulations.
(a) Add sample points to sample plan $X$ covering the design space $S$.
(b) Simulate $N_1$ loadings for each new sample point $x^{(i)}$ and receive the response $y^{(i)} = \psi(x^{(i)})$.
(c) Update $w$, $\theta$ and $\phi$, leaving out sample points for validation.
(d) Compute the error $x^*$ to validate the surrogate model.
(e) If target is reached, continue, otherwise go to 3.(a).

4. Exploration and optimisation using the surrogate model.
(a) Visualise the objective functions $\psi(x)$, $x \in S$.
(b) Compute the Pareto fronts $\phi(x)$, $x \in P$.
(c) Identify optimal loading strategy for given performance trade-off.

Since we use a fixed number of sample points instead of a given target error we only need to evaluate the model error once, i.e., we switch order between 3.(d) and 3.(e).

### 7. Results and discussion

A surrogate model of the loading performance is constructed from a set of $N = 204$ sample points covering the design space. This mean 4080 simulations of the loading operation. Another 40 sample points are gathered for validation of the surrogate model and used for computing the model accuracy. The design space coverage of the training and validation sets can be found in Appendix B. The found model parameters, $\theta^*$, are presented in Table 7 with the corresponding regularization and model accuracy. The loading performance dependency on the design variables can be understood from the contour plots of the objective functions in Fig. 12-14. The Pareto front is visualized in Fig. 15-17.

From the values of the surrogate model parameters in Table 7 we observe that $\alpha$, the bucket fill ratio target, is the design variable that most strongly influence the loading performance, i.e., $\theta^* > \theta^*$ for $j = \beta, \gamma, \kappa$. The second most important design variable is $\beta$, that controls the target dig depth. The entry point variable, $\gamma$, affect only the rock debris objective, $\psi_3$, by any significance. The bucket curl variable, $\kappa$, has a small effect on the production, $\psi_1$, and lesser so on the other objectives.

The three objective functions are visualized as nested contour plots in Fig. 12-14. From Fig. 12 we conclude that high production is promoted by dig plan with a slight bucket overfilling, $\alpha = 1.3$, shallow digging and maximal entry point mass proximity and large bucket curl. According to Fig. 13 structural damage is avoided by deep digging and small bucket fill.
The found surrogate model parameters $\theta$, regularization $\lambda$ and model accuracy $\varepsilon$ for the three objective functions.

<table>
<thead>
<tr>
<th>$\phi_1$</th>
<th>$\theta_9$</th>
<th>$\theta_{10}$</th>
<th>$\theta_{11}$</th>
<th>$\lambda$</th>
<th>$\varepsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.677</td>
<td>0.040</td>
<td>0.003</td>
<td>0.019</td>
<td>0.0002</td>
<td>0.05</td>
</tr>
<tr>
<td>0.726</td>
<td>0.404</td>
<td>0.003</td>
<td>0.001</td>
<td>0.002</td>
<td>0.17</td>
</tr>
<tr>
<td>1.552</td>
<td>0.353</td>
<td>0.194</td>
<td>0.001</td>
<td>0.008</td>
<td>0.18</td>
</tr>
</tbody>
</table>

Table 7: The found surrogate model parameters $\theta$, regularization $\lambda$ and model accuracy $\varepsilon$ for the three objective functions.

Three sample points are selected to exemplify how to interpret the plots. The example points have the following design variables

- $x_A = [1.3, 0.85, 1.0, 0.6]$,
- $x_s = [0.3, 0.6, 0.6, 0.2]$,
- $x_B = [1.8, 0.1, 0.2, 1.0]$.

The points represent contrasting trajectories with different loading performance $\phi(x)$ that are indicated in Fig. 12 and 14. The first sample point, $x_A$, is a high and shallow trajectory that attempts to overfill the bucket by 30%. It attacks the pile at the front most centre of mass point and curls the bucket moderately between waypoint $s_1$ and $s_2$. The second point, $x_s$, is a medium high trajectory that is satisfied with filling the bucket only to 30% of the capacity. It attacks the pile at a middle centre of mass point and applies

- $x_B$, is a low and deep trajectory that tries to overfill the bucket a lot. It attacks the pile at a rear centre of mass point and applies

Figure 12: Contour plot of the production objective function, $\hat{\phi}_1$, in units [kg/s].

Figure 13: Contour plot of the structural damage objective function, $\hat{\phi}_2$. 

Figure 14: Contour plot of the debris objective function, $\hat{\phi}_3$, in units [kg].
maximum curl of the bucket. Of the three points, \( \mathbf{x}_{\Phi} \) achieves the largest production, see Fig. [12]. It is a good combination of bucket filling and short time to execute the trajectory. The sample point \( \mathbf{x}_{\Phi} \) has the second highest production. The extra time spent on trying to overfill the bucket is affecting the production negatively. The sample point \( \mathbf{x}_{\Phi} \) has by far the lowest production. There is no gain in production by under-filling the bucket although it results in faster loading time. For the structural damage, \( \Phi_1(x) \), and the debris, \( \Phi_2(x) \), objective functions, the relationship is the reverse. The sample point \( \mathbf{x}_{\Phi} \) has the lowest structural damage and the lowest amount of debris. In the middle lies the sample point \( \mathbf{x}_{\Phi} \) and the worst is \( \mathbf{x}_{\Phi} \).

The loading performance measures are also objective functions to the multiobjective optimisation problem stated in Eq. (20). The Pareto front is visualized in Fig. [15-17] by projection to a two-dimensional space and color coded in the remaining dimension. The three example points are also included for reference, but they are not Pareto optimal solutions. It is clear, and expected, that \( \Phi_1 \) competes with \( \Phi_2 \) and \( \Phi_3 \). A higher production is associated with more debris in front of the pile and larger bending moments in the joints. The design variables for the Pareto solutions with the highest production are approximately, \( x = [1.3, 0.8, 0.9, 0.8] \). That is when the vehicle attack the pile at a point with center of mass closer to the front, with a high and shallow trajectory that tries to slightly overfill the bucket and applies large bucket curl. The objectives \( \Phi_2 \) and \( \Phi_3 \) do not compete as seen in Fig. [17]. It is possible to achieve a low structural damage and low amount of debris at the same time.

The accuracy of the surrogate model is measured with the NRMSE in Eq. [19] and is listed in Table 7. The error varies between 5% (for production) and 17 - 18% (for damage and debris). This is deemed acceptable for practical application. It should be noted that NRMSE is a global error since all points are weighted equal. Once a region of particular interest have been identified in the design space the local error can be investigated and improved by adding weight to the sample points in that region or adding additional points. A test was done with 500 sample points and a slightly different control strategy. The accuracy of that model was the same. This indicate that the response surface has a noise level that is not meaningful to resolve any further.

A number of assumptions and simplifications have been made that limit the model validity for a real LHD. The most severe simplification is probably that the particle size distribution of the muck pile is truncated at 0.3 m. The contact model parameters are representative but have not been validated in experiments. It is likely that cohesion should also be included as a contact parameter when considering materials with large fraction of fine particles and presence of moisture. It should be pointed out that including the effect fine particles can be achieved by mapping the contact parameters to measured bulk behavior [34] and do not necessarily rely on representing the true particle size distribution. The maximum forces of the actuators have been estimated from the maximal breakout force specified in data sheets from the manufacturer. There is a risk that the machine is too weak in certain configurations and too strong in others. This would be resolved by replacing the actuator models with a more faithful powertrain model. The damage model only considers the bending moment in one selected joint and one mode of deformation. This is easily remedied by extending the damage objective function. The current loading operation model does not apply any gradual raising of the boom arm to increase the ground grip and avoid losing traction while digging into the pile. The next step, before applying the proposed technique to a specific machine and rock material, is to validate the dynamic models, investigate the sensitivity to the model parameters and remedy any severe short-comings that are found. The machine dynamics model parameters, and the rock material model should be identified and calibrated using independent experiments, e.g., tracking the loading motion with given payloads and performing triaxial test [34]. Finally, the combined model is validated by executing a sample of dig plans on measured piles and comparing with simulations of the same.

In this paper, the method for computational exploration is tested on the dig planning problem rather than on the loading control. The method is just as applicable to the control problem, e.g., for exploring the promising admittance control [1] and finding the control parameters maximize the performance for a given vehicle design and rock material properties.

It is also interesting to reflect on the computational complexity of the proposed method. The presented methodology for computational exploration involve three objective functions for measuring the performance and four design variables that determine the loading strategy. A total of 4880 multibody dynamics simulations of rock loading were performed, covering the four-dimensional design space with 244 sample points and 20 loadings in each point. The computational cost of simulating one loading cycle is 500 CPU seconds with the specified hardware. This amount to almost 3 CPU hours per sample point and 700 CPU hours in total. The computational time can be assumed to grow at least linearly with the number of bodies in the rock pile, depending on the required time-step, PGS iterations and on the parallel scalability on the given hardware. The computational time for obtaining the surrogate model from the response data and Eq. (18) is about 60 s. A single evaluation of the surrogate model takes only 0.1 ms. The Pareto set was determined using a genetic optimization algorithm (gamultiobj) in MATLAB that required approximately 40 · 10^3 evaluations. The net computational time for this was also close to 60 s. Both these steps are clearly negligible compared to the simulation time. To be precise, the surrogate model is roughly a factor 10^5 more efficient than the original simulations on which it is based. Computing the Pareto front using the multibody simulation would take about 10^5 CPU hours, or 300 years with a single CPU. There is no cost in adding additional objective functions unless the response is sensitive and need to be resolved using many more sample points. Introducing additional design variables, on the other hand, is more costly since the required number of sample points grows exponentially to maintain a constant sampling density in the design space. Hence, an increase in computational performance by a factor 10 can at best be converted into 10 times more particles in the rock pile or 10 times more sample points, which would allow one more design variable in the
A method for simulation-based development of robotic rock loading is described and tested. It is shown feasible to use non-smooth multibody dynamics simulation to explore the dependency between different dig planning strategies and the resulting loading performance, measured by productivity, structural damage to the machine and rock debris that cause interruptions. A total of 4880 loading cycles are simulated and a surrogate model for the relation between the motion design variables and loading performance is built. The surrogate model is a faceted triangulation on a semi-implicit discretization. This generates a time-stepping scheme for the position and velocity, \((x_t, v_t, \lambda_t)\) in Eq. (1)-(4) and applying a semi-implicit discretization. This generates a time-stepping scheme for the position and velocity, \((x_t, v_t) \rightarrow (x_{t+1}, v_{t+1})\),

The accuracy of the surrogate model was found to be between 5 - 18% for the different performance measures. By exploring the surrogate model it is found that the highest productivity is achieved with a high and shallow trajectory, attempting to overfill the bucket by 30%, applying a large bucket curl and entering the pile at a point with center of mass closer to the front. The proposed simulation-based methodology should apply equally well to determine optimal loading control parameters as well.

Acknowledgements

The project was funded in part by the VINNOVA grants 2014-01901, Umeå University and supported by Algoryx Simulation. The authors are thankful for fruitful discussions with experts at LKAB.

9. Appendix

A. Numerical method

The regularisation and stabilisation terms, \(\varepsilon\) and \(\gamma\), introduce compliance and dissipation in motion orthogonal to the constraint manifold. In the absence of the inequality and complementarity conditions, the regularised constraints may be viewed as Legendre transforms of a potential and Rayleigh dissipation function of the form \(U_p(x) = \frac{1}{2} g^T g\) and \(R_p(x, v) = \frac{1}{2} (Gv)^T (Gv)\) [20]. This enables modelling of arbitrarily stiff elastic and viscous interactions in terms of constraint forces with a direct mapping between the regularisation and stabilisation terms to material parameters such as the elasticity modulus. The diagonal matrices \(\Sigma_n, \Sigma_s, \Sigma_i\) and \(\Sigma_h\) contain the contact material parameters and can be found in [23].

The numerical time integration scheme is based on the SPOOK stepper [20] derived from discrete variational principle for the augmented system \((x, v, \lambda, \lambda)\) in Eq. (1)-(4) and applying a semi-implicit discretization. This generates a time-stepping scheme for the position and velocity, \((x_t, v_t) \rightarrow (x_{t+1}, v_{t+1})\),

![Figure 15: The Pareto set projected onto production \(\phi_1\) and damage \(\phi_2\), and color coded by debris \(\phi_3\).](image1)

![Figure 16: The Pareto set projected onto production \(\phi_1\) and debris \(\phi_3\), and color coded by production \(\phi_1\).](image2)

![Figure 17: The Pareto set projected onto damage \(\phi_2\) and debris \(\phi_3\), and color coded by production \(\phi_1\).](image3)
from time \(t_n\) to \(t_{n+1} = t_n + \Delta t\) that is linearly stable and \(O(\Delta t^2)\) accurate for constraint violations \([20]\) and involve solving the following mixed complementarity problem (MCP) \([35]\).

\[
Hz + b = w_i - w_u \quad 0 \leq z - 1 \wedge w_i \geq 0 \quad 0 \leq u - z \wedge w_c \geq 0
\]

where

\[
H = \begin{bmatrix}
M & -G_i^T & -G_j^T \\
G_i & \Sigma_g^0 & 0 \\
G_j & 0 & \Sigma_l \\
\end{bmatrix},
\]

\[
z = \begin{bmatrix}
v_{n+1} \\
\lambda_{u,n+1} \\
\lambda_{l,n+1}
\end{bmatrix}, \quad b = \begin{bmatrix}
-Mv_n - \Delta t M^{-1} f_{ext} \\
\frac{1}{2} \gamma_n g_n - \gamma_l G_n v_n \\
-\omega_j + \frac{1}{2} \gamma_j G_j v_j
\end{bmatrix}.
\]

The solution vector \(z\) contains the new velocities and the Lagrange multipliers \(\lambda_u\), \(\lambda_l\) and \(\lambda_j\). The position update is simply \(x_{n+1} = x_n + \Delta t v_{n+1}\). For notational convenience, a factor \(\Delta t\) has been absorbed in the multipliers such that the constraint force reads \(G^T \lambda / \Delta t\). The upper and lower limits, \(u\) and \(l\), in Eq. \((22)\) follow from the contact law and joint and motor limits. Since the limits depend on the solution, this is a partially nonlinear complementarity problem. \(w_i\) and \(w_u\) are temporary slack variables used internally by the MCP solver only. In the present paper the full MCP is solved with a hybrid direct-iterative split solver using the simulation engine AGX Dynamics \([30]\). The articulated machine and the contact normal forces between the machine and rocks are thus solved using a sparse direct block pivoting LDM solver \([21]\). The rock pile contact network and the friction forces between the machine and the rocks are solved to lower precision using a projected Gauss-Seidel (PGS) solver \([18]\). To accelerate the PGS solver computations, we employ domain decomposition for parallel processing, warmstarting \([56]\) and model reduction \([25]\).

The contacts are divided into impacts and continuous contacts, depending on the magnitude of the incoming relative normal velocities \(G_n^T v_n\). The impulse transfer through the system is assumed to satisfy the Newton impact law, \(G_n^T v_n = -e G_n^T v_n\), with coefficient of restitution \(e\) for the impacts \((i)\), as well as preserve all remaining constraints \((j)\) on velocity level, \(G^T \lambda_j = 0\).

**B. Design space coverage**

It is important that the sample points cover the design space well. The distribution of sample points and validation points in this study is shown in Fig. \[18\] and \[19\].

**References**


