Physics-based virtual environments for autonomous earthmoving and mining machinery

Martin Servin[1,2] and Michael Brandl[1]

[1] Algoryx Simulation AB, Kuratorvägen 2, SE-90736 Umeå, Sweden
[2] Umeå University, SE-90187 Umeå Sweden
martin.servin@umu.se

Abstract. The scientific foundation for constructing virtual environments (VE) that support the development of earthmoving and mining machinery with autonomous capabilities is summarized. It is explained how the physics simulation engine AGX Dynamics supports this. Finally, a methodology for computational design exploration of an autonomous load-haul-dump machine in a physics-based VE is described.

Keywords: Physics Engine, Autonomous Mining, Multibody, Granular, LHD.

1 Introduction

Increased automation in the field of earthmoving and mining is important for economic, environmental and safety reasons. It is however challenging to find solutions that are both efficient and robust. The machines move in unstructured and dynamic environment, and handle fragmented material whose behavior is difficult to predict. In a physics-based virtual environment it is possible to explore many alternative solutions in different scenarios and generate sufficiently large and varied data sets to train autonomous systems. Such a virtual environment requires reliable physics integrated with computational algorithms for the many subsystems: mechanics, materials, actuators and sensors. The performance and precision must be sufficient for the intended use. Real-time performance is often required when the simulation involves human- or hardware in-the-loop. The virtual environment needs also be modifiable with interoperability between the software and engineering tools used to compose it, so that validated models are easy to share and reuse within and between organizations.

2 Scientific foundation

A *virtual environment* (VE) is a visual interactive computer-simulated world whose elements have geometric and physical attributes, and evolve according to a given set of physical laws. To the virtual world there are interfaces in the form of 3D visualization, signal inputs and outputs for sensors and actuators that are used for control - that may be manual, semiautonomous or fully autonomous.

The hardware for running VEs range from an ordinary laptop to distributed over a cluster of powerful computers. It may include advanced VR systems such as motion platforms and haptic feedback.

VEs are designed to be very general. All possible states and courses of events that may occur with the given elements can be simulated as long as they are consistent with the physics laws that are implemented. The core of a VE is a so-called *physics engine*. That is a software library with dedicated data structures and algorithms for representing physical objects, managing their interaction and solving the dynamic system's equations of motion to evolve the VE forward in time. The scientific foundation of modern physics engines for industrial and scientific applications is described below, with the design and control of earth-moving and mining machinery particularly in mind.

2.1 Multi-domain nonsmooth multibody dynamics

A physics engine must support faithful multi-domain and multibody dynamics, while ensuring a fast and stable simulation. The presence of contacts, controllers and human or hardware in-the-loop means that the connectivity between the system variables varies frequently and unpredictably during simulation. Articulated rigid and flexible bodies are therefore modeled on *descriptor form*[3] and Lagrangian formulation in terms of differential algebraic equations (DAEs) [15]. This can be extended to include dynamic models of other physics domains - such as hydraulics, electronics and drivetrain - and multidomain couplings in terms of effort and flow variables [19]. Stable numerical steppers, designed to preserve the fundamental physical invariants, are constructed with discrete-time mechanics and variational integrators [21]. The integration of the DAEs of motions requires solving large block-sparse linear systems of equations. This can be done with high performance codes for direct factorization that exploit parallelism using graph partitioning and block operations with optimized BLAS3 kernels [25].

It has been realized that contacts, dry friction, limits on joint and actuators are best treated in terms of inequality constraints and complementarity problems [28] and considering the time-discrete system as a *nonsmooth dynamics* system [1]. This means that velocities may change discontinuously in accordance with some contact and impact law giving rise to impulses that propagate *instantaneously* throughout the system. This enables fixed-step time integration with large time-step. Many VEs with heavy machines run with time-step size $1 - 20$ ms. The alternative is to do exact event detection (in time) or to use fine enough temporal resolution for all the dynamics to appear smooth and modeled in terms of continuous DAEs and ordinary differential equations alone. That is an intractable approach for full-system simulations of mechatronics system with any substantial complexity.

The time-stepping subproblem in the nonsmooth dynamics approach becomes a nonlinear complementarity problem (NCP) or quadratic programming problem (QP) in place of nonlinear system of equations. This is often linearized

[3] All the degrees of freedom are represented and no coordinate reduction is done.

into a mixed complementarity problem (MCP) [12]. The strict computational budget of most VEs means that one-stage methods of first or second order are favored. Popular methods include reformulating the NCP constraint equations as Fischer-Burmeister or Fischer-Newton NCP function [10], Alart-Curnier function [2] or proximal (prox) function. The PATH solver [9] has been widely used in solving MLCPs. Constraint regularization and stabilization are important to reduce errors that may occur from high-velocity impacts and to improve numerical solvability [4]. The SPOOK stepper is derived from a variational formulation of nonholonomic and compliant constraints, and has been proven to be linearly stable [17]. The regularized constraints may be viewed as Legendre transforms of a potential or a Rayleigh dissipation function. This enables modeling of arbitrarily stiff elastic and viscous interactions in terms of constraint forces with direct mapping between the regularization and stabilization terms to physical material parameters. The physics engine *AGX Dynamics* is one instance of the described computational technology, summarized in Fig. 1.

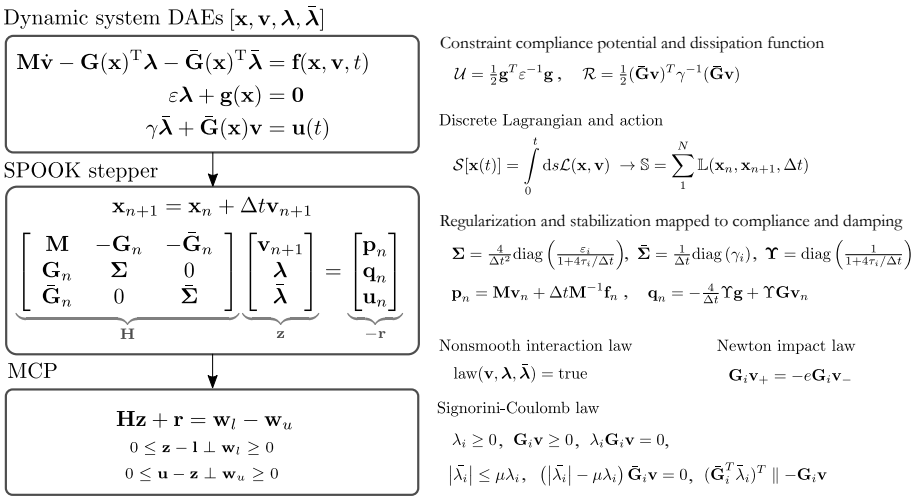


Fig. 1: The mathematical building blocks of the physics engine AGX Dynamics - from equations of motion to the discrete-time integrator and its mixed complementarity problem following the contact laws and joint and actuator limits.

2.2 Computational granular dynamics

Rocks and soil are granular matter, which is defined as a large collection of discrete macroscopic particles interacting locally by interfacial contact forces that are highly dissipative. There is a multitude of methods for simulating these materials [22]. The widely used *discrete element method* (DEM) is popular for its

versatility. It can be used for simulating granular matter in all its three phases: gaseous, liquid as well as solid. With DEM the bulk dynamics emerge naturally from the collective behavior of the particles with no need for constitutive modeling other than the local contact law and particle properties such as shape, viscoelasticity and surface friction. When the dynamics of a machine and granular material are strongly coupled, it is advantageous to treat them both as nonsmooth dynamics systems to achieve a fast and stable simulation by means of time-implicit integration. Several versions of *nonsmooth DEM* (NDEM) have been provided [24,27] that enable time-step up to 1000 times larger than for conventional DEM. Each solve is, however, computationally intense, why iterative solvers that may be truncated at a desired precision are preferred. Many techniques for computational acceleration have been developed [23,29,26,30].

2.3 Computational robotics and machine vision

There is a growing interest in using virtual environments as complement to physical experiments for machine learning and intelligent task and motion planning for machines handling objects in a cluttered or rough terrain environments [5]. There are also promising results of using physics driven virtual environments in-the-loop for object motion tracking [7] and path planning of terrain vehicles [8]. These motivate integration with software libraries for robotic control and machine vision, e.g., Robotic Operating System (ROS), OpenCV, Point Cloud Library (PCL) or VANE [13]. The open source robotics simulation software Gazebo [16] supports this integration and has a generic physics API.

2.4 Co-simulation

The term co-simulation describes a setup where the full system is composed of two or more subsystems that share dynamic variables, yet are simulated as black boxes in a distributed manner. The main reasons for doing co-simulation are that the subsystems rely on distinct simulation software and solvers, or in order to avoid disclosing confidential models. The subsystems are coupled either by sharing physical forces at the boundaries or by algebraic constraints. The Functional Mockup Interface (FMI) [3] provides a tool independent standard for creating and interconnecting subsystem software components, and several runtime environments with master steppers are available [11,18].

3 AGX Dynamics

AGX Dynamics from Algoryx Simulation AB is a multi-purpose physics engine for simulators, VR applications, engineering and scientific simulation of mechatronics, materials and industrial processes. Typical systems include robots, vehicles, cranes and other complex mechanisms found in manufacturing, transportation, construction and bulk material processing. The customer's purpose of use

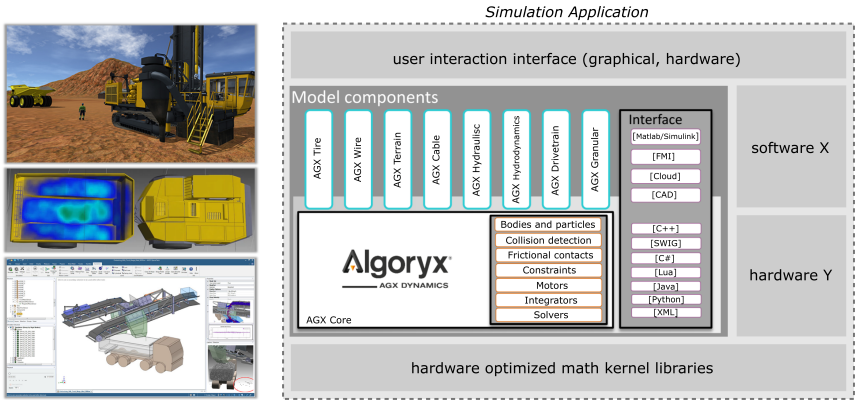


Fig. 2: AGX Dynamics is a multi-purpose physics engine for simulators, VR applications, engineering and scientific simulation enabling VE:s as stand-alone simulation application or integration into other software and hardware.

ranges from simulation based design, control and optimization to testing with human- or hardware-in-the-loop, operator training, sales and marketing.

The software is architectured as a Software Development Kit (SDK), see Fig. 2. It consists of hundreds of C++ classes of highly optimized and portable code for integration with simulator software and hardware, or for stand-alone simulation applications. The core library is oriented around rigid multibody systems with nonsmooth dynamics, numerical time integration based on SPOOK [17] and high-performance parallel equation solvers that provide robust and valid simulations even for systems with extreme loads and mass ratios, kinematic loops, redundant constraints, frictional contacts and impulsive dynamics. A number of library modules support high-level modeling and simulation of cable and wire, drivetrain, hydraulics, hydrodynamics, granular materials, terrain and tires. User code can be injected for event handling, virtual sensors, actuators and controllers. Simulations can be authored and managed either directly through the C++ API, through high-level scripting via Python, C# or Lua, or through the graphical user interface in the 3D modeling environments *AGX Momentum* and the VR and simulator development platform *Unity*. AGX supports FMI export and co-simulation with other software tools that support the FMI standard [3,18]. It also supports direct coupling with Matlab/Simulink.

AGX has both direct and iterative MCP solvers that are tailored for the sparse linear algebra operations of contacting multibody systems. A hybrid direct-iterative solver and smart splitting scheme supports the simultaneous use of both direct and iterative solver kernels for solving different parts of the system with different precision. This enables simulations that combine high performance, accuracy and scalability. This is essential for efficient and reliable simulation of heavy machinery excavating or transporting granular materials. The direct solver is applied to the vehicle mechanics and hydraulics and is ex-

act to machine precision, while the scalable iterative solver provides a fast and approximate solution of the granular dynamics. The coupling can be solved either using a direct or iterative method, or a combination. With the NDEM approach to granular materials, the viscoelastic material parameters can be mapped to the constraint compliance and damping parameters of the SPOOK stepper [27]. For small relative contact velocities, or in the limit of small time-steps, the normal constraint force coincides with the Hertz contact law, i.e., $\mathbf{G}^{\mathrm{T}}\boldsymbol{\lambda} \rightarrow k[\delta^{3/2} + c\delta^{1/2}\dot{\delta}]\mathbf{n}$, with overlap δ, normal stiffness $k = E^*\sqrt{2d}/3$ in terms of the effective Young's modulus E^*, the particle diameter d, and the viscocity c. NDEM granular sub-systems are solved using a parallel projected Gauss-Seidel algorithm with domain decomposition. The computational speed is further accelerated using adaptive model reduction [26] and warmstarting [30].

4 Computational exploration of an autonomous LHD

For demonstration purpose, we describe a method for computational exploration of an autonomous load-haul-dump (LHD) machine using a virtual environment. An LHD is a four-wheeled vehicle designed to operate in underground mines along narrow paths. It has an articulation steering joint that separates the rear driving unit from the front loading unit, which is equipped with a loading boom arm and bucket. LHDs are still not fully automated but rely on a human operator for controlling the loading task, normally by remote. The haul and dump tasks have been successfully automated, but loading of fragmented rock remains a challenge [6]. The difficulty lies in perceiving the state of the pile, making a dig plan and controlling the machine movement to fill the bucket in short time, avoiding damage on the vehicle, and leaving the pile in a state well-suited for continued loading.

Autonomous loading can be divided into pile analysis, dig planning and motion control during approach, entry, digging and breakout. We propose using VEs for systematic exploration of the design space for dig planning and loading control. The method is outlined in Fig. 3. The idea is to first construct a detailed simulation model for the rock pile and the machine equipped with sensors, powertrain and actuators, using the method in Fig. 1. The next step is to provide the system with a generic dig plan and loading control algorithm that are functions of a set of design variables, $\boldsymbol{x}$, as well as of the sensor data and the kinematics of the machine, see Fig. 4. The planned bucket motion depends on four design variables, $\boldsymbol{x} = [\alpha, \beta, \gamma, \kappa]$, that parametrize the target fill ratio (α), the dig depth (β), entry point (γ) and bucket curl during filling (κ). One shallow and one deep dig trajectory, relative to the estimated pile surface, are indicated in Fig. 4. A large set of simulated loading operations are carried out with different values on the design variable and different rock piles. A loading performance, $\phi(\boldsymbol{x})$, is measured in each simulation according to a specified set of objective functions for the productivity, machine damage and resulting pile state. A surrogate model is constructed, using the SUMO Toolbox [14], for how the loading performance depends on the design variables for dig planning and

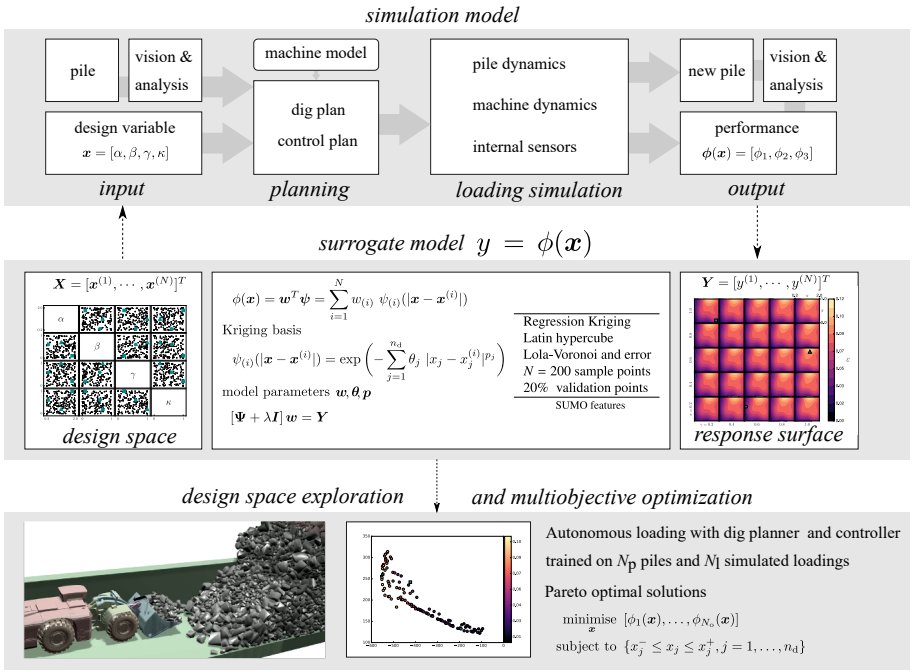
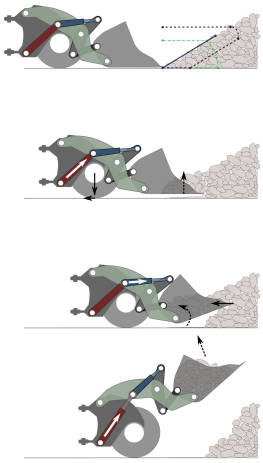


Fig. 3: Illustration of the proposed method for computational exploration of dig planning and loading control with sample results from [20].

loading control. The evaluation time for the surrogate model is negligible compared to the time for physics-based simulation. The surrogate model enables systematic exploration of the design space and visualization of the response surfaces. A set of Pareto optimal solutions can be determined using multiobjective optimization, for which the genetic algorithm NSGA-II in Matlab's *gamultiobj* is used.

The described method for computational exploration was applied in a recent study [20] considering the dig planning task for loading at the draw point in a narrow tunnel (Scenario 1). Nearly 5000 loading cycles where simulated with a muck pile consisting of approximately 5000 rocks. A surrogate model with accuracy of $5-18\%$ was built, analysed and a set of Pareto optimal solutions for the dig plan design variables were found. Sample results are shown in Fig. 3 and videos are found at `http://umit.cs.umu.se/loading`. The objective function for damage was based on critical twisting forces in the joint connecting the boom and the bucket. The objective function for the resulting pile state was based on the amount of rock spill in front of the pile. A set of dig planning design variables where identified that combine high productivity with low damage and rock spill. These variables correspond to shallow dig trajectories, attempting to overfill the bucket by 30 %, applying a large bucket curl and entering the pile at a point



Algorithm 1 Automated loading

Input: Dig planning and control variables $\boldsymbol{x}$
Output: Loading performance $\boldsymbol{y}$

pile measurement and analysis $\rightarrow \mathcal{P}$
generate a dig plan $\mathcal{D} \leftarrow [\mathcal{P}, \boldsymbol{x}]$
enter pile at given point and pose $\leftarrow \mathcal{D}$
while breakout condition $(\boldsymbol{x})$ = false **do**
    drive at target speed v_e
    measure wheel slip and boom cylinder force
    if wheel slip tendency **then**
        increase boom lift force $\leftarrow \boldsymbol{x}$
    end if
    if pile resistance $(\boldsymbol{x})$ **then**
        increase bucket curl $\leftarrow \boldsymbol{x}$
    end if
end while
raise boom and bucket maximally if not there
retract from and measure the pile
compute loading perforance $\rightarrow \boldsymbol{y}$

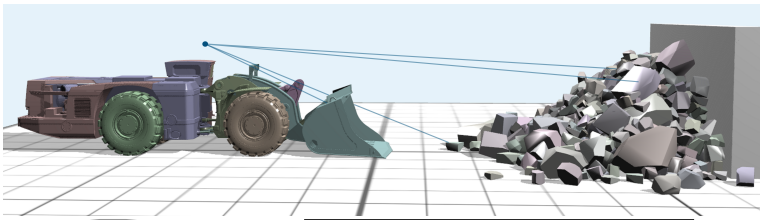
Fig. 4: Illustration of automated loading and algorithm for force-based control. The operations that depend on the design variable are colour highlighted.

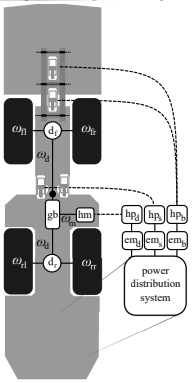
with center of mass closer to the front. Force-based loading control was not used in [20] and the present powertrain model is more evolved.

The LHD is modelled as a rigid multibody system consisting of 18 bodies and 24 joints, primarily hinge and prismatic constraints with joint limits on the extensions of hydraulic cylinders. The geometry, mass and powertrain characteristics are those of a Sandvik LH621[4], which has an operating weight of 56 tonnes, is 12 m long and 3 m wide. The key bodies are listed in Fig. 5, where the LHD and the powertrain system are also illustrated.

The rock pile is modeled as contacting rigid bodies of different convex shapes and size. The material parameters are listed in Fig. 5. Two scenarios are created. Scenario 1 represents loading at the draw point in the end of a narrow tunnel where the pile stretches further back and up [20]. Scenario 2 is a pile in front of a static wall. For vision, a virtual 3D imaging device is assumed mounted on top of the machine. The 3D imaging sensor consists of a simulated camera and a laser scanner modeled as a point-based source of rays with angular density of about 1 K rays per steradian and a point-based receiving unit placed 100 mm from the source. The sensor outputs 2D images from the simulated camera as well as a 3D point cloud in the coordinate system of the mine. Sample data is displayed in Fig. 6. A pile analysis algorithm transforms the points representing a pile surface into a mass distribution from which the center of mass can be estimated and projected to the pile front on the ground to obtain a pile entry point to which the LHD can be steered.

[4] Sandvik Technical Specification LH621-18,`http://unitedminingrentals.com/pdf/trucks/LH621.pdf` and 3D model from `https://www.cgtrader.com/3d-models/vehicle/industrial/sandvik-underground-loader-toro-0011`.





Body	Mass [ton]	Size [m]
wheel	1.9	$r_{\mathrm{w}} = 1.0$
rear unit	27	$5.9 \times 3.0 \times 2.9$
front unit	15	$4.0 \times 3.0 \times 2.9$
boom	3.7	$r_{\mathrm{lb}} = 2.5$
Z-bar	0.88	1.6
bucket	6.4	$r_{\mathrm{bt}} = 2.0$ m, $V_{\mathrm{b}} = 8.0$ m^3

Property	Value
number of rocks in tunnel	~ 5.500
size distribution	$[0.3, 0.4, 0.6, 0.8]$ m
	$[20,\ 30,\ 30,\ 20]$ % of mass
mass density	2.500 kg/m^3
elastic modulus	1.0 GPa
Poisson ratio	0.3
rock-rock restitution	0.0
machine-rock restitution	0.1
rock friction coefficient	0.9
other friction coefficients	1.0

Fig. 5: Illustration of the LHD, equipped with a 3D vision sensor (center point and view frame illustrated) and electric-hydraulic power distribution system.

The described method for computational exploration can also be applied to the loading control only. For this purpose a simple force-based loading control algorithm, mimicking the more sophisticated admittance control [6], was developed and examined. The idea, illustrated in Fig. 4, is to control the bucket motion based on the wheel slip and pile resistance, measured as the moving average $[f_{\mathrm{l}}(t)]_\tau$ of the lift cylinder force over a time window $t - \tau$ to t. The pile resistance increases as the LHD enters the pile. This may cause loss of traction. If any wheel starts spinning, $[\omega_{ij}]_\tau > \omega_{\mathrm{spin}}$, the boom lifting is activated to increase the normal and traction force until the wheel spinning ceases. The bucket tilting is activated when the force in the lift cylinder exceeds a threshold. This is to prevent excessive slip, bucket under- or overfilling and prolonged loading time. The bucket tilting is held constant while $[f_{\mathrm{l}}(t)]_\tau > f_{\mathrm{min}} \equiv c_{\mathrm{bc}} M_{\mathrm{bc}} g_{\mathrm{acc}}$ and otherwise $\omega_{bt}(t) = \omega_{bt}^{\mathrm{max}} \left(1 - f_{\mathrm{min}}/[f_{\mathrm{l}}(t)]_\tau\right)$, where M_{bc} is the loading capacity of the bucket and c_{bc} is determined experimentally and considered as a control design variable in addition to $\omega_{\mathrm{bt}}^{\mathrm{max}}$.

A simplified electric-hydraulic powertrain model is assumed to distribute the total available power, $P_{\mathrm{max}} = P_{\mathrm{d}}^{\mathrm{max}}(t) + P_{\mathrm{s}}^{\mathrm{max}}(t) + P_{\mathrm{b}}^{\mathrm{max}}(t)$, to the subsystems for drive (d), articulated steering (s) and for lifting and tilting the bucket (b). The power limits may be distributed differently during entry of the pile, bucket filling and breakout. Each actuator is modeled by an effort constraint, e.g., $\varepsilon_{\mathrm{m}} \lambda_{\mathrm{m}} + \omega_{\mathrm{m}} = \omega_{\mathrm{m}}^{\mathrm{target}}(t)$ for the hydraulic motor driving the

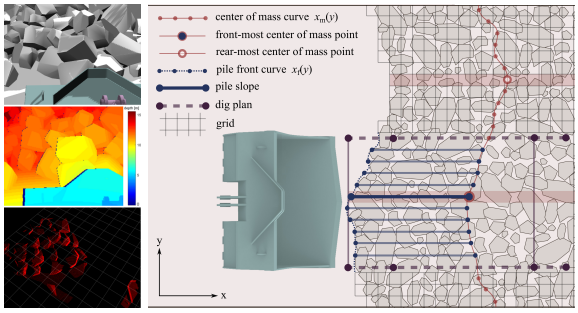


Fig. 6: The imaging sensor produce a 2D camera image (top left) and a 3D point cloud (middle and lower left). A pile analysis algorithm transform the surface points into a mass distribution that is input to a dig planning algorithm.

motor shaft at target target speed $\omega_{\mathrm{m}}^{\mathrm{target}}(t)$ that is set by the control system (or operator). The applied torque and power, $\tau_{\mathrm{m}} = \lambda_{\mathrm{m}}$ and $P_{\mathrm{m}} = \tau_{\mathrm{m}}\omega_{\mathrm{m}}$, depend on the dynamics of the full system and follow from the numerical computations carried out by the physics engine, as described in Fig. 1. The hydraulic motor is assumed to be fed by hydrostatic transmission, with hydraulic pump ($\mathrm{hp_d}$), and an electric motor ($\mathrm{em_d}$) such that the total power for actuating the motor shaft is $P_{\mathrm{d}} = \eta_{\mathrm{depm}} P_{\mathrm{m}}$ where η_{depm} is the combined energy efficiency of the power transmission between the electric motor and hydrostatic transmission. A gearbox (gb) couples the motor shaft to the main drive shaft and provides it with velocity and torque pair $[\omega_{\mathrm{d}}, \tau_{\mathrm{d}}]$ and with gear ratio constraint $\omega_{\mathrm{d}} = c_{\mathrm{gb}}\omega_{\mathrm{m}}$. The drive shaft is then connected to the four wheel axes with angular velocity $[\omega_{fl}, \omega_{fr}, \omega_{rl}, \omega_{rr}]$ via a front and rear differential ($\mathrm{d_d}$ and $\mathrm{d_r}$) modeled by the constraints $\omega_{\mathrm{fl}} + \omega_{\mathrm{fr}} = \omega_{\mathrm{d}}$ and $\omega_{\mathrm{rl}} + \omega_{\mathrm{rr}} = \omega_{\mathrm{d}}$. A power limit, $P_{\mathrm{d}}^{\mathrm{max}}$, on the drive system thus translates to a constraint limit $|\lambda_{\mathrm{m}}| \leq P_{\mathrm{d}}^{\mathrm{max}}/\eta_{\mathrm{depm}}\omega_{\mathrm{m}}$. The articulated steering is similarly modeled as an actuator that attempts to drive the articulation joint at a desired rotational speed (ω_{s}) given a limit on the constraint multiplier, $|\lambda_{\mathrm{s}}| \leq P_{\mathrm{s}}^{max}/\eta_{\mathrm{sepc}}\omega_{\mathrm{s}}$, set by available power for steering and efficiency η_{sepc} of the electric motor ($\mathrm{em_s}$) and hydraulic pump ($\mathrm{hp_s}$). The loading unit connects the boom arm and bucket through a Z-bar linkage mechanism with one hydraulic cylinder for lifting (l) the boom and one for tilting (t) the bucket. The two cylinders share the power supplied by a hydraulic pump ($\mathrm{hp_b}$) and electric motor ($\mathrm{em_b}$). This is modeled as two linear actuators with effort constraint $\varepsilon_{\mathrm{b}i}\lambda_{\mathrm{b}i} + v_{\mathrm{b}i} = v_{\mathrm{b}i}^{\mathrm{target}}(t)$ with $i = \mathrm{l}, \mathrm{t}$. Since the two actuators share a common power source, the constraint limits are $|\lambda_{\mathrm{bi}}| \leq (P_{\mathrm{b}}^{\mathrm{max}} - P_{\mathrm{bj}})/\eta_{\mathrm{bepm}}v_{\mathrm{bi}}$ for $j \neq i$ and actuator power $P_{\mathrm{bi}} = \lambda_{\mathrm{b}i}v_{\mathrm{b}i}$. The joints and actuators are equipped with force sensors and motion encoders producing signals for controlling the machine.

Three samples of simulations for Scenario 2 with different control design variables, $\boldsymbol{x} = [c_{\mathrm{bc}}, \omega_{\mathrm{bt}}^{\mathrm{max}}]$ are presented in Fig. 7. The bucket filling was measured to $8.7, 11.0$ and 12.7 tonnes, the respective loading time $10, 11$ and 12 s. The lift

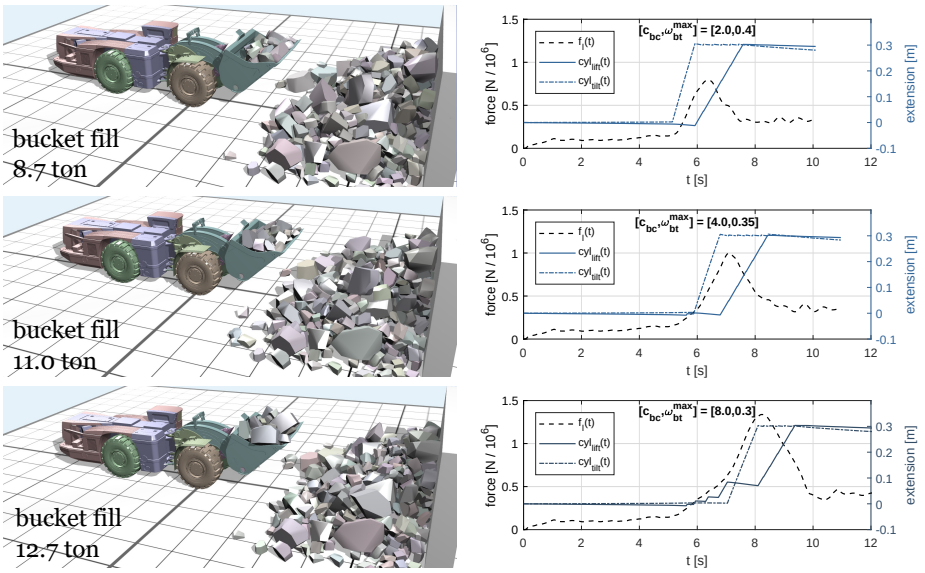


Fig. 7: Different design variable, $[c_{\mathrm{bc}}, \omega_{\mathrm{bt}}^{\max}]$, give different bucket filling and loading time. The right subplots show the cylinder extension and force.

force and cylinder extensions, as function of time, are also shown in Fig. 7. The respective peak force value in the lift cylinder was 750, 1000 and 1300 kN. The most aggressive loading caused wheel slip that was efficiently reduced by the lifting of the boom for increased traction, but the loading time was nevertheless prolonged. The target speed into the pile was 1 m/s. Videos from simulations can be accessed from `http://umit.cs.umu.se/LHD`.

References

1. Acary, V., Brogliato, B.: Numerical Methods for Nonsmooth Dynamical Systems: Applications in Mechanics and Electronics. Springer Verlag (2008)
2. Alart, P., Curnier, A.: A mixed formulation for frictional contact problems prone to newton like solution methods. Comp. Appl. Mech. Eng. 92(3), 353–375 (1991)
3. Blochwitz, T., et al.: The functional mockup interface for tool independent exchange of simulation models. In: Proc. 8th Int. Modelica Conf. pp. 105–114 (2011)
4. Bornemann, F., Schütte, C.: Homogenization of Hamiltonian systems with a strong constraining potential. Phys. D 102(1-2), 57–77 (1997)
5. Boularias, A., Bagnell, A., Stentz, A.: Efficient optimization for autonomous robotic manipulation of natural objects. In: Proc. 28th Conf. Artificial Intelligence (AAAI). pp. 2520–2526 (2014)
6. Dobson, A., Marshall, J., Larsson, J.: Admittance control for robotic loading: Design and experiments with a 1-tonne loader and a 14-tonne load-haul-dump machine. Journal of Field Robotics. 34(1), 123–150 (2017)

7. Duff, D.J., et al.: Physical simulation for monocular 3d model based tracking. In: Int. Conf. Robotics and Automation (ICRA). pp. 5218–5225 (2011)
8. Ettlin, A.: Rigid body dynamics simulation for robot motion planning. Doctoral thesis no 3663 at Ecole polytechnique fédérale de Lausanne (2006)
9. Ferris, M.C., Munson, T.S.: Complementarity problems in gams and the path solver. J. Econ. Dyn. Control. 24(2), 165–188 (2000)
10. Fischer, A.: A newton-type method for positive-semidefinite linear complementarity problem. J. Optimiz. Theory. App. 86, 585–608 (1995)
11. Galtier, V., et al.: Fmi-based distributed multi-simulation with daccosim. In: Proc. Symp. Th. Mod. & Simu DEVS 15. Alexandria, Egypt (2015)
12. Glocker, C., Pfeiffer, F.: An lcp-approach for multibody systems with planar friction. In: CMIS 92 Cont. Mech. Int. Symp. pp. 13–20. Lausanne, Switzerland (1992)
13. Goodin, C.: Sensor modeling for the virtual autonomous navigation environment. In: Sensors 2009. IEEE, Christchurch, New Zealand (2009)
14. Gorissen, D., Couckuyt, I., Demeester, P., Dhaene, T., Crombecq, K.: A surrogate modeling and adaptive sampling toolbox for computer based design. Journal of Machine Learning Research 11, 2051–2055 (07 2010)
15. Haug, E.J.: Computer-Aided Kinematics and Dynamics of Mechanical Systems. Prentice-Hall, Englewood Cliffs (1989)
16. Koenig, N., Howard, A.: Design and use paradigms for gazebo, an open-source multi-robot simulator. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). vol. 3, pp. 2049–2154 (2004)
17. Lacoursiére, C.: Regularized, stabilized, variational methods for multibodies. 48th Scandinavian Conf. Sim. Mod. (SIMS 2007), Göteborg, 30–31 pp. 40–48 (2007)
18. Lacoursiére, C., Härdin, T.: Fmi go! a simulation runtime environment with a client server architecture. In: Proc. 12th Int.Modelica Conf. pp. 653–662 (2017)
19. Layton, R.A., Fabien, B.C.: Systematic modelling using lagrangian daes. Math. Comp. Model. Dyn. 7(3), 273–304 (2001)
20. Lindmark, D., Servin, M.: Computational exploration of robotic rock loading. Submitted manuscript. (2017), `http://umit.cs.umu.se/loading/`
21. Marsden, J.E., West, M.: Discrete mechanics and variational integrators. Acta Numer. 10, 357–514 (2001)
22. Pöschel, T., Schwager, T.: Computational Granular Dynamics, Models and Algorithms. Springer-Verlag, Berlin Heidelberg (2005)
23. Precklik, T., Rude, U.: Leveraging parallel computing in multibody dynamics. Multibody Syst. Dyn. 2, 173—196 (2015)
24. Radjai, F., Richefeu, V.: Contact dynamics as a nonsmooth discrete element method. Mechanics of Materials 41(6), 715–728 (2009)
25. Schenk, O., Gaertner, K.: On fast factorization pivoting methods for sparse symmetric indefinite systems. Elec. Trans. Numer. Anal. 23, 158–179 (2006)
26. Servin, M., Wang, D.: Adaptive model reduction for nonsmooth discrete element simulation. Computational Particle Mechanics. 3(1), 107–121 (2016)
27. Servin, M., Wang, D., Lacoursiére, C., Bodin, K.: Examining the smooth and nonsmooth discrete element approaches to granular matter. Int. J. Numer. Meth. Eng. 97(12), 878–902 (2014)
28. Stewart, D.E.: Rigid-body dynamics with friction and impact. SIAM Rev. 42(1), 3–39 (2000)
29. Visseq, V., et al.: High performance computing of discrete nonsmooth contact dynamics with domain decomposition. Int. J. Num. Mth. Eng. 96, 584–598 (2013)
30. Wang, D., et al.: Warm starting the projected gauss-seidel algorithm for granular matter simulation. Computational Particle Mechanics. 3(1), 43–52 (2016)