Dynamically optimal trajectories for earthmoving excavators

Young Bum Kim, Junhyoung Ha*, Hyuk Kang, Pan Young Kim, Jinsoo Park, F.C. Park

1. Introduction

Interest in unmanned excavators is driven by many considerations, from a desire to reduce the numerous fatal accidents that occur regularly to human operators—particularly in dangerous environments involving highly uneven terrain and unstable ground—to improving task efficiency by automating excavation in an optimal manner. To realize such potential benefits, an unmanned excavator must have the capability to produce optimal motions in a reliable and efficient manner. While there are many possible notions of optimality, our focus in this paper is on excavator motions that are fast, power-efficient, and smooth. These criteria naturally require careful consideration of the excavator dynamics.

In some relevant previous work, [1] reports on a fully automated excavator that senses the dig face, recognizes and localizes the truck, determines a feasible digging path via three planning algorithms: a coarse-to-fine dig point planning, template-based dump planning, and script-based planning. It is important to note that none of these planning algorithms take into account the dynamics of the excavator; only the geometry of the terrain and kinematic constraints are considered. More recently in [2], a high-level hierarchical framework for autonomous construction machinery is proposed, in which the planning, measurement, and control functions are clearly identified and defined. While the general issues involved in planning are addressed, no specific planners are proposed.

In this paper we report on a class of algorithms for generating dynamically optimal excavator trajectories, and more broadly, on an offline simulation and motion planning system for unmanned excavators. The benefits of a computer simulation and planning tool for such purposes are obvious: a means of evaluating and optimizing system performance, from design to motion planning and control, as well as reduced development time and costs with fewer iterations of prototype construction and testing.

Our main technical contribution is a class of motion optimization algorithms for generating the fastest possible minimum torque motions. As has been reported in some of our earlier work, [3], that motion optimization taking into account the dynamics is, although seemingly straightforward in principle, computationally and numerically highly challenging in practice. For one thing, the availability of analytic gradients of the objective function and constraints impacts numerical convergence in a significant way; finite difference approximations of the gradient often result in numerical instabilities and poor convergence behavior. Moreover, the presence of closed loop constraints—the kinematics of a typical excavator involves several closed loops and passive joints that are not actuated—complicates the calculation of the dynamics and the gradients.

Drawing upon some of our previous work [3–5], we develop an optimization algorithm that robustly and efficiently obtains the fastest possible minimum torque motion—that is, without violating any actuator saturation limits, or other constraints on cylinder velocities or accelerations; we call such motions minimum torque/time motions—taking into account the closed-loop structure of excavators, and the soil dynamics during digging and dumping, i.e., in a way that reflects the influence of soil-tool interaction forces during digging, and the effects of soil loads during lifting and dumping. Such motions avoid the undesirable bang-bang characteristics of typical minimum-time trajectories, while being fast, smooth, and energy-efficient. The key to our algorithm lies in the ability to efficiently calculate analytic gradients (and if necessary analytic Hessians) of the objective function, all as a simple by-product of the recursive dynamics calculation—this is achieved by drawing upon the geometric algorithms first derived in [5].

More broadly, our motion optimization algorithm is part of an offline planning and simulation system that integrates disparate components, ranging from a dynamics simulation engine to methods for soil model parameter identification, into a coherent and easy-to-use framework. We describe the basic elements of our overall motion optimization framework. We describe the basic elements of our overall motion optimization framework.
planning and simulation architecture. In particular, using soil models based on the fundamental earth moving equations (FEE) [6] and its variants [7], we describe a soil model parameter identification procedure that relies on measurements of digging forces in actual excavation tasks.

The optimal motions generated by our method are compared with the actual excavation trajectories of experienced human operators. We find that the optimal digging motions, while for the most part similar to those performed by an experienced human operator, do show some important qualitative differences, particularly in the dumping and returning motions. These and other observations are discussed in detail.

The remainder of the paper is organized as follows. Section 2 provides a high-level description of our simulation and motion planning architecture, together with a more detailed description of the main components. Section 3 derives the excavator dynamics for the four main tasks involved in earthmoving, and also describes both the soil model used for the study and a procedure for estimating its parameters. Section 4 presents details of the algorithm for generating minimum torque and minimum time motions. Section 5 presents a comparison of the optimal motions generated by our planner with the actual excavation trajectories performed by experienced human operators. We conclude in Section 6 with a discussion of possible future extensions to our simulator and planner.

2. Simulation and planning architecture

Fig. 1 depicts the overall architecture for our simulation and motion planning system. The three basic functions are dynamics simulation, motion planning and optimization, and soil parameter estimation. The dynamic simulation engine formulates the equations of motion for the four basic excavator tasks (digging, lifting, dumping, and returning). Motions obtained from dynamic simulations can be compared with measurement data taken from field experiments; a direct comparison between, e.g., the force profiles is one means of evaluating the accuracy and reliability of the dynamics simulation.

Digging simulation requires a computational model for the soil dynamics, which in turn requires a method for estimating the soil model parameters. The soil parameter estimation module takes as input measurement data (obtained from field experiments), and estimates the soil model parameters via an optimization procedure. Among the external forces exerted on the excavator, digging forces are typically the largest, so that careful estimation of the soil model parameters is essential for realistic dynamics simulation.
The planner invokes the dynamics simulation engine when generating motions, and takes as input the boundary conditions, i.e., the initial and final bucket pose and velocity, and produces optimal cylinder trajectories for the minimum torque/time criterion. Embedded within the planner are the requisite optimization algorithms, which require analytic gradients; these gradients can be obtained as a by-product of the dynamics simulation as noted earlier.

3. Excavator dynamic modeling

For the purposes of this study we model only the rigid-body dynamics of the excavator, without taking into account the hydraulic actuator dynamics. The methodology presented in this paper can be straightforwardly extended to this case, although the governing equations of motion become more involved.

The dynamic modeling of the excavator is complicated by the presence of four closed loops, and several passive joints. There are two main approaches to the dynamic modeling of such constrained multibody systems: (i) modeling the system by a set of differential equations subject to algebraic constraints (the differential-algebraic approach), and (ii) modeling the system exclusively by a set of differential equations formulated in terms of an independent set of generalized coordinates. For purely dynamic simulation purposes there is little difference between the two approaches. However, for motion optimization purposes, where analytic gradients of the objective function are extremely useful, the latter approach is computationally advantageous and more straightforward.

Assuming that the excavator mechanism consists of a total of \( m \) one degree-of-freedom joints, the first step in the dynamic modeling is to distinguish the \( n \) actuated joints, denoted \( q_a \in \mathbb{R}^n \), from the \( m-n \) passive joints, denoted \( q_p \in \mathbb{R}^{m-n} \) (for typical excavators, \( n = 4 \) and \( m = 16 \)). Define \( q = (q_a, q_p) \in \mathbb{R}^m \). The kinematic loop closure constraints can then be expressed in the following differential form (see, e.g., [8]):

\[
\dot{q} = \Phi \ddot{q}_a
\]

\[
\Phi = \begin{bmatrix} I & -J_{dp} \end{bmatrix}
\]

\[
\ddot{q} = \Phi \ddot{q}_a + \Phi \dddot{q}_a
\]

The matrices \( J_{dp} \in \mathbb{R}^{(m-n) \times n} \) and \( J_{dp} \in \mathbb{R}^{(m-n) \times (m-n)} \) above are obtained by time-differentiating the \( m-n \) kinematic loop closure equations \( g(q, \dot{q}) = 0 \). \( \ddot{q} \) can be further differentiated to obtain \( \dddot{q} \); [8] describes an efficient screw-theoretic method for this differentiation. Below we shall have occasion to use the constraint Jacobian \( J_c \) defined as

\[
J_c = [J_e - J_g] \in \mathbb{R}^{(m-n) \times m}
\]

Having modeled the excavator kinematics in this form, we now convert the excavator into a topological tree structure (which we call a reduced system) by appropriately cutting all closed loops in the original system as shown in Fig. 2. For each cut point, we imagine a generalized force \( \lambda \) exerted at the passive joint that causes the reduced system to undergo exactly the same movement as the original system (see Fig. 3). Physically this generalized force can be identified with the internal force and bending moment at the corresponding point of the original closed-loop system.

Denoting by the vector \( \lambda \) the collection of all the generalized forces \( \lambda \), the dynamics of the excavator can be expressed in the form

\[
M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + V(q) = \tau + J^T \lambda
\]

where \( M(q) \in \mathbb{R}^{m \times m} \) is the mass matrix, \( C(q, \dot{q}) \in \mathbb{R}^{m \times m} \) and \( V(q) \in \mathbb{R}^m \) are the terms associated with the Coriolis and gravitational forces of the reduced system, \( \tau \in \mathbb{R}^m \) is the vector of the torques, and \( \lambda \in \mathbb{R}^{m-n} \) is the vector of constraint forces. Explicit formulations of these matrices, as well as efficient recursive algorithms for the evaluation of the dynamic equations, are provided in [5]. Following [5], \( \ddot{q} \) can then be calculated for the reduced system provided \( (q, \dot{q}, \ddot{q}) \) are assumed given:

\[
\ddot{q} \Delta M(q) \dddot{q} + C(q, \dot{q}) \dddot{q} + V(q)
\]

Please cite this article as: Y.B. Kim, et al., Dynamically optimal trajectories for earthmoving excavators, Automation in Construction (2013), http://dx.doi.org/10.1016/j.autcon.2013.01.007
Note that the left-hand side of Eq. (8) is zero because the torques at passive joint torques: 

\[ \tau = 0 = J^T_p \lambda. \]  

(6)

\[ \tau \] here is the sum of the original torques \( \tau \) with the constraints \( \lambda \). Eq. (6) can be divided into two parts corresponding to the active and passive joint torques:

\[ \tau_a = \tau_a - J^T_a \lambda \]  

(7)

\[ 0 = \tau_p - J^T_p \lambda. \]  

(8)

Note that the left-hand side of Eq. (8) is zero because the torques at passive joints are zero. From Eqs. (7) and (8), \( \lambda \) and \( \tau_a \) can be obtained as follows:

\[ \lambda = J^T_p \tau_p \]  

(9)

\[ \tau_a = \tau_a - J^T_a J^T_p \tau_p. \]  

(10)

\[ \tau_a \] can be obtained from Eqs. (1) and (3), from which \( \tau_a \) and \( \lambda \) are then obtained from (5), (9), and (10). Note that Eq. (4) does not account for the soil forces. Taking these into account, the excavator dynamics equations assume the form

\[ M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + V(q) = \tau + J^T_a \lambda + J^T_f \tau_f, \]  

(11)

where \( J_f \) is the soil–tool Jacobian, \( f_s \) is the soil force exerted on the bucket during the digging, \( f_s \) is the soil–tool Jacobian and \( f_s \) is the force exerted on the bucket arising from soil weight during the lifting (details are given below).

Thus far we have derived the rigid-body dynamics of the excavator in an independent context; we now examine the task-specific dynamics for truck loading, which consists of digging, lifting, dumping, and returning to the original digging position. The next subsections describe, for each subtask, how to formulate the inverse dynamics, which is needed in the motion optimization procedure.

### 3.1. Digging dynamics

During digging, the soil lifting force \( f_s \) in Eq. (11) is not exerted (that is, \( f_s \) should be zero). In this case the equations of motion can be rewritten as follows:

\[ M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + V(q) = \tau + J^T_a \lambda + J^T_f \tau_f, \]  

(12)

The digging force \( f_s \) can be calculated via the soil model, which we describe later in Section 3.5.

### 3.2. Lifting dynamics

During lifting, the bucket is assumed to be filled with the soil obtained during the digging phase. The volume of soil, \( V_s \), is calculated from the closed volume between the ground and bucket-tip path as

\[ V_s = \int \rho \, dt. \]  

(24)

### Table 1

<table>
<thead>
<tr>
<th>Working</th>
<th>Equation number</th>
<th>( P_{\text{max}} )</th>
<th>( Q_{\text{max}} )</th>
<th>( W_{\text{max}} )</th>
<th>( V_{\text{max}} )</th>
<th>N.P.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digging</td>
<td>Eq. (12)</td>
<td>( \bigcirc )</td>
<td>( \bigcirc )</td>
<td>( \bigcirc )</td>
<td>( \bigcirc )</td>
<td>( \bigtimes )</td>
</tr>
<tr>
<td>Lifting</td>
<td>Eq. (15)</td>
<td>( \bigcirc )</td>
<td>( \bigcirc )</td>
<td>( \bigtimes )</td>
<td>( \bigcirc )</td>
<td>( \bigtimes )</td>
</tr>
<tr>
<td>Dumping</td>
<td>Eq. (16)</td>
<td>( \bigcirc )</td>
<td>( \bigcirc )</td>
<td>( \bigtimes )</td>
<td>( \bigtimes )</td>
<td>( \bigtimes )</td>
</tr>
<tr>
<td>Returning</td>
<td>Eq. (4)</td>
<td>( \bigcirc )</td>
<td>( \bigcirc )</td>
<td>( \bigtimes )</td>
<td>( \bigtimes )</td>
<td>( \bigtimes )</td>
</tr>
</tbody>
</table>

### Table 2

<table>
<thead>
<tr>
<th>Minimum torque optimization algorithm.</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_{\text{null}} \rightarrow ) Lifting motion minimum torque optimization (( \Phi_{\text{lift}} ))</td>
</tr>
<tr>
<td>( P_{\text{null}} \rightarrow ) Dumping motion minimum torque optimization (( \Phi_{\text{dump}} ))</td>
</tr>
<tr>
<td>( P_{\text{null}} \rightarrow ) Returning motion minimum torque optimization (( \Phi_{\text{return}} ))</td>
</tr>
</tbody>
</table>

1. Input: Measurement data \( M \), Motion script \( \varphi = [\Phi_{\text{dig}}, \Phi_{\text{lift}}, \Phi_{\text{dump}}, \Phi_{\text{return}}], t_i = \) \( t_f \).
2. Soil parameter \( \varphi \rightarrow \) Soil parameter estimation (\( M \)) described in Section 3.5.
3. \( P_{\text{null}} \rightarrow \) Soil volume \( V_s \) → Digging motion minimum torque optimization (\( \Phi_{\text{dig}} \)).
4. Objective function is given in Eq. (27) with the dynamics specified in Section 3.1.
5. Constraints are given in Table 1.
6. \( P_{\text{null}} \rightarrow \) Lifting motion minimum torque optimization (\( \Phi_{\text{lift}} \)).
7. Objective function is given in Eq. (27) with the dynamics specified in Section 3.2.
8. Constraints are given in Table 1.
9. \( P_{\text{null}} \rightarrow \) Dumping motion minimum torque optimization (\( \Phi_{\text{dump}} \)).
10. Objective function is given in Eq. (27) with the dynamics specified in Section 3.3.
11. Constraints are given in Table 1.
12. \( P_{\text{null}} \rightarrow \) Returning motion minimum torque optimization (\( \Phi_{\text{return}} \)).
13. Objective function is given in Eq. (27) with the dynamics specified in Section 3.4.
14. Constraints are given in Table 1.

Please cite this article as: Y.B. Kim, et al., Dynamically optimal trajectories for earthmoving excavators, Automation in Construction (2013). http://dx.doi.org/10.1016/j.autcon.2013.01.007
shown in Fig. 4. The soil carried in the bucket exerts a force \( f_b \) on the bucket, whereas digging force \( f_d \) should be zero during lifting phase. In general it is difficult to analytically calculate \( f_b \) since it depends on the interior shape of bucket and the motion of the bucket. However, since the soil mass can be readily computed, one can augment the bucket mass with the soil mass for a reasonable approximation of \( f_b \). To derive the equations of motion when the soil mass is augmented to the bucket mass in this form, the corresponding Euler–Lagrange equations can be written as follows:

\[
\frac{d}{dt}\left(\frac{\partial L_k}{\partial \dot{q}}\right) - \frac{\partial L_k}{\partial q} = M_k(q, \dot{q}) + C_k(q, \dot{q})\dot{q} + V_k(q) \tag{13}
\]

where \( L_k \) is the Lagrangian for the augmented soil. Using both Eqs. (11) and (14), the equations of motion for lifting can be written as follows:

\[
(M(q) + M_b(q))\ddot{q} + (C(q, \dot{q}) + C_b(q, \dot{q}))\dot{q} + V(q) + V_b(q) = \tau + J_T^\alpha \lambda. \tag{15}
\]

and the boundary conditions for each of terms in Eq. (16) are given as follows:

\[
M(q, \alpha)\ddot{q} + C(q, \dot{q}, \alpha)\dot{q} + V(q, \alpha) = \tau + J_T^\alpha \lambda. \tag{16}
\]

and the similarity between Eqs. (15) and (11).

3.3. Dumping dynamics

During dumping, the soil volume in the bucket \( V_t \) decreases as a function of bucket angle \( \alpha(q) \). As shown in Fig. 5, we assume that the soil volume \( V_t \) linearly decreases as the bucket angle \( \alpha \) increases, in which case the equations of motion (16) can now be written

\[
\begin{align}
M(q, \alpha)\ddot{q} + C(q, \dot{q}, \alpha)\dot{q} + V(q, \alpha) = & \tau + J_T^\alpha \lambda. \tag{17}
\end{align}
\]

and the boundary conditions for each of terms in Eq. (16) are given as follows:

\[
M(q, \alpha)\ddot{q} + C(q, \dot{q}, \alpha)\dot{q} + V(q, \alpha) = \tau + J_T^\alpha \lambda. \tag{18}
\]

where \( \alpha, \dot{\alpha} \) respectively denote the start and final angles for dumping. Note that each of the terms \( M, C \) and \( V \) in Eq. (16) depends on the bucket angle \( \alpha \) as well as the joint angle vector \( \dot{q} \).

3.4. Returning dynamics

If during the returning task no external forces other than gravity are exerted on the excavator, then the dynamic equations in this case are given by Eq. (4).

3.5. Soil modeling and identification

In general, calculating soil digging forces is complicated by the difficulty in constructing accurate models that capture the static and dynamic properties of the soil. The choice of soil–tool interaction model crucially affects the accuracy and performance of digging simulation. The soil–tool interaction process is summarized in Fig. 6.

Recent contributions to soil–tool interaction modeling include the fundamental earth-moving equation (FEE) [6], learning based models [10], spring-damper models [11], and energy-dissipation models [12]. For our purposes we choose a modified version of the FEE model proposed in [7], illustrated in Fig. 7. This model specifies the soil–tool dynamic interaction in terms of the five soil parameters \( \delta, \phi, \beta, \gamma, \) and \( \zeta \):

- \( \delta \): the angle between bucket blade and soil force \( f_b \) exerted on bucket blade as shown in Fig. 7
- \( \phi \): the angle between the failure plane and resistance force \( R \) as shown in Fig. 7
- \( \beta \): the angle between the terrain surface and the failure plane as shown in Fig. 7
- \( \gamma \): the mass density of the soil
- \( \zeta \): soil cohesion parameter with the same unit as pressure.

The soil force \( f_b \) can be calculated as a function of five parameters,

\[
f_b = d^2 w y g N_w + c w d N_c + V \gamma g N_q \tag{20}
\]

where \( f \) is a set of additional geometric parameters which correspond to the depth of the bucket tip, the width of the bucket, the terrain slope, rake angle, and swept volume. \( f \) can be calculated once bucket pose, bucket shape and terrain shape are known. The derivation of \( f_b \) is given in [9].

We note that this model describes the static force-displacement relation between the soil and tool; while for our purposes this model is sufficient for the most part, for fast, highly dynamic movements of the bucket the accuracy of the dynamic simulation can be expected to degrade.

<table>
<thead>
<tr>
<th>Table 3</th>
<th>Minimum time optimization algorithm.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Minimum time optimization</strong></td>
<td></td>
</tr>
<tr>
<td>1: Input: Measurement data ( M ), Motion script ( \Psi = [\Phi_{\text{dug}}, \Phi_{\text{dump}}, \Phi_{\text{return}}] )</td>
<td></td>
</tr>
<tr>
<td>2: Soil parameter ( \Psi = ) Soil parameter estimation(( M )) described in Section 3.4.</td>
<td></td>
</tr>
<tr>
<td>3: ( \Phi_{\text{dug}}, \Phi_{\text{dump}}, \Phi_{\text{return}} ) — Digging motion minimum time optimization ( \Phi_{\text{dug}} )</td>
<td></td>
</tr>
<tr>
<td>4: ( \Phi_{\text{dug}}, \Phi_{\text{dump}}, \Phi_{\text{return}} ) — Digging motion minimum time optimization ( \Phi_{\text{dug}} )</td>
<td></td>
</tr>
<tr>
<td>5: ( \Phi_{\text{dug}}, \Phi_{\text{dump}}, \Phi_{\text{return}} ) — Digging motion minimum time optimization ( \Phi_{\text{dug}} )</td>
<td></td>
</tr>
<tr>
<td>6: ( \Phi_{\text{dug}}, \Phi_{\text{dump}}, \Phi_{\text{return}} ) — Digging motion minimum time optimization ( \Phi_{\text{dug}} )</td>
<td></td>
</tr>
</tbody>
</table>

Table 4 Specifications for the hydraulic excavator used in experiments.

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating weight</td>
<td>29,300 kg</td>
</tr>
<tr>
<td>Boom length</td>
<td>6.25 m</td>
</tr>
<tr>
<td>Arm length</td>
<td>3.05 m</td>
</tr>
<tr>
<td>Bucket capacity</td>
<td>1.27 m³</td>
</tr>
</tbody>
</table>

Please cite this article as: Y.B. Kim et al., Dynamically optimal trajectories for earthmoving excavators, Automation in Construction (2013), http://dx.doi.org/10.1016/j.autcon.2013.01.007
We now address the estimation of the soil model parameters. In this study we formulate this as an optimization problem, in which the following objective function is minimized:

\[ \text{soil parameters} = \arg \min_{\phi, p, \gamma, \zeta} \int_{t_i}^{t_f} |\Delta F(t)| \, dt, \]  

(24)

where \( t_i \) is the time at which digging starts, \( t_f \) is the time at which digging ends, and \( \Delta F(t) \) is the difference between the measured force and the force calculated from the soil model. The optimization variables are the soil model parameters, and the objective function \( \int_{t_i}^{t_f} |\Delta F(t)| \, dt \) is taken to be the integral of the absolute value of the force difference (which, as shown in Fig. 8, corresponds to the area between the two indicated curves).

4. Minimum torque/time optimization

4.1. Formulation

The generation of minimum torque/time trajectories for excavators can be approached in a number of ways. One possible optimal control formulation involves augmenting the minimum torque functional with a penalty term for the final time \( t_f \), i.e.,

\[ \min_{\tau(t)} \phi(t_f) + \frac{1}{2} \int_{t_i}^{t_f} |\tau(t)|^2 \, dt \]  

(25)

subject to boundary conditions and various physical constraints as summarized in Table 1.

In [13] a local solution to the optimal control problem is found by assuming that the joint trajectories \( q(t) \) are parametrized in terms of B-splines:

\[ q(t, P) = \sum_{i=1}^{m} B_i(t) p_i, \]  

(26)

where \( B_i(t) \) is a B-spline basis function and \( P = [p_1, p_2, \ldots, p_m] \) is the set of control points. The cost functional then reduces to a parameter optimization problem of the form

\[ \min_{P} \phi(t_f) + \frac{1}{2} \int_{t_i}^{t_f} |\tau(P, t)|^2 \, dt \]  

(27)

subject to the same constraints as described in Table 1. The basis functions must be chosen so as to satisfy all the terminal boundary conditions.

For the minimum torque problem in which the final time \( t_f \) is given, it is intuitively clear that for larger values of \( t_f \), the minimum torque profiles become increasingly flatter. If \( t_f \) is allowed to vary, \( t_f \) is increased. A specific algorithm for solving this problem is described in Section 4.2.

4.2. Algorithms

Numerical procedures for solving the minimum torque problem and minimum time problem are described in Tables 2 and 3. Here \( M \) denotes the measurement data containing the cylinder trajectories and corresponding cylinder forces for several digging motion trials, \( \Phi \) is the motion script which contains the initial and final configurations of digging, lifting, dumping and returning motions specified by the user, and \( t_f \) is the duration time for each motion.

For the minimum time problem described in Table 3, the motion optimization procedure is as follows

1. Optimize the minimum torque for a fixed \( t_f \) and check all constraints.
2. Using an appropriate line search algorithm, decrease the final time \( t_f \) if all constraints are satisfied, otherwise increase \( t_f \).

Please cite this article as: Y.B. Kim, et al., Dynamically optimal trajectories for earthmoving excavators, Automation in Construction (2013), http://dx.doi.org/10.1016/j.autcon.2013.01.007
3. Repeat the routine from step 1 to step 2 until some termination condition is satisfied.

The final time $t_f$ can be determined via a standard line search procedure, e.g., golden section search, with an upper bound on admissible values of $t_f$.

To handle the constraints specified in Table 1, it is desirable to enforce hard constraints as much as possible, and to convert some constraints into soft constraints as required.

5. Experimental results

In this section we present experimental results verifying the fidelity of our dynamics simulations, and a comparison of the trajectories produced by our planner with those produced by an experienced human operator.

Physical specifications for the excavator used in the experiments are listed in Table 4.

Displacement sensors and pressure sensors attached each of the cylinders at the boom, arm, and bucket measure the corresponding cylinder displacement and force (Fig. 9).

5.1. Verification of excavator dynamics simulation

We first compare the cylinder force trajectories produced by our dynamic simulation with actual measurements. The cylinder force trajectories are measured for the two cases: (1) without soil–tool interaction, and (2) with soil–tool interaction. Fig. 10 compares the simulated

![Graphs showing comparison of simulated and measured cylinder forces](attachment:image.png)

**Fig. 10.** A comparison of simulated and measured cylinder forces without soil–tool interaction.

Please cite this article as: Y.B. Kim, et al., Dynamically optimal trajectories for earthmoving excavators, Automation in Construction (2013), http://dx.doi.org/10.1016/j.autcon.2013.01.007
and measured forces for repeated trials of a circular reference bucket motion in three dimensions, performed by a human operator. The simulated forces show good agreement with the measurements; the offsets in arm and bucket forces can be traced to small discrepancies in the model parameters (i.e., masses and inertias of the links), as well as unmodelled dynamic effects such as friction, bending, and hysteresis.

For the case with soil–tool interactions, we first estimate the five parameters in the modified FEE soil model using the procedure described earlier. Measurement data has been obtained for eight excavation cycles, in which each cycle consists of digging–lifting–dumping–returning. We use the first digging cycle to estimate the soil parameters, and then compare the predicted and measured cylinder forces over the remaining seven cycles; the results are shown in Fig. 11.

5.2. Optimized motion versus operator motion

We now generate optimal motions using the models and algorithms described earlier, and compare them to the motions generated by an experienced human operator. Minimum torque/time motions are obtained for initial and final times that correspond to those of the human operator motion (Fig. 12). Table 5 compares the task execution times and average torque sums of the human operator motion, the minimum-time motion (using a minimum-time motion planner), and the minimum torque/time motions generated by our planner (Figs. 15 and 16). The results suggest that the operator uses excessive torques compared to our planner’s motions. Note that the minimum-time motion only results in a small improvement in task execution time.

Fig. 11. A comparison of simulated and measured cylinder forces with soil–tool interaction.
One of the interesting findings of our experiments is that there is a notable difference between the generated optimal motion and actual excavation trajectories performed by a human operator, particularly during the arm return motion. We suspect that these differences can be mitigated somewhat by imposing more physically realistic constraints on the hydraulics, and taking into account collision avoidance between the excavator and truck.

At a higher level, this paper has also presented an offline simulation and planning system for unmanned excavators, that integrates an excavator dynamic simulation engine with our optimal motion planner. One of the key elements in both the planning and simulation phases is the use of a soil model. In this paper we have described a nonlinear least-squares soil model parameter identification procedure that relies on measurements of digging forces in actual excavation tasks.

The presented work can be extended in a number of ways. Taking into account the dynamics of the hydraulic actuators is clearly useful, in particular considerations on, e.g., hydraulic actuator limits and heat loss considerations. Although [4] presents one realization of such a model, accurate hydraulic actuator dynamic modeling is a highly complicated task, and it may be profitable to examine hierarchical models that capture features of the dynamics at different resolution scales. Improved soil models, as well as excavation tasks in more extreme unstructured environments, are also being examined in the current context, as well as taking into account ground collision avoidance constraints that arise during typical excavation tasks.

There are also several means of bootstrapping the above procedure into a class of methods for generating real-time suboptimal trajectories; one particular method is described in [14] that involves principal component analysis of a dataset of optimally generated motions.

Table 5

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Measurement</th>
<th>Min time</th>
<th>Min torque</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tilde{f}$</td>
<td>14.9</td>
<td>12.6</td>
<td>14.9</td>
</tr>
<tr>
<td>Average $|\tau(t)|$</td>
<td>1.0145</td>
<td>0.61483</td>
<td>0.61128</td>
</tr>
</tbody>
</table>

Fig. 12. Cylinder forces over a single cycle. Dotted lines separate digging, lifting, dumping, and returning periods from left to right.
Fig. 13. Cylinder trajectories.

Fig. 14. Cylinder forces.

Please cite this article as: Y.B. Kim, et al., Dynamically optimal trajectories for earthmoving excavators, Automation in Construction (2013), http://dx.doi.org/10.1016/j.autcon.2013.01.007
Acknowledgment

This research was supported by a research grant from Hyundai Heavy Industries, with further partial support from SNU-IAMD, ROSAEC-ERC, AIM, and SNU-BK21 Program in Mechanical Engineering.

References


Fig. 15. Minimum time solution. Red curve shows the bucket-tip path of the optimal motion; the blue curve shows the human operator’s motion. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 16. Minimum torque solution. Red curve shows the bucket-tip path of the optimal motion; the blue curve shows the human operator’s motion. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)