Learning to Predict Resistive Forces During Robotic Excavation

Sanjiv Singh

Field Robotics Center
Carnegie Mellon University
Pittsburgh, PA 15213-3890

Abstract— Few robot tasks require as forceful an interaction with the world as excavation. In order to effectively plan its actions, our robot excavator requires a method that allows it to predict the resistive forces experienced as it scoops soil from the terrain. In this paper we present methods for a robot to predict the resistive forces and to improve its predictions based on experience. We start with a simple analytical model of a flat blade moving through soil and show how this analysis can be extended to account for the phenomena specific to excavation. In addition, we examine how representation of the learning problem and methodology affect prediction performance using several criteria.

I. Introduction

The excavation task is stated as follows. *Given an autonomous excavating robot, a sensor that can determine the shape of the terrain, and, a goal configuration of the terrain, produce and execute a sequence of robot actions to reach the goal configuration without violating specified constraints, in a reasonable number of steps.*

More informally, we would like a robot excavator that is able to excavate a volume of soil according to a specification. Excavation tasks range from loading from a pile of soil, to cutting a geometrically described volume of earth as is required in the digging of a trench or a foundation footing. Two such excavators are shown in Fig. 1.

Previously we have proposed a method to automatically plan digging motions for excavators given a kinematic description of the mechanism, a state description of the terrain and the goal state to be achieved [SINGH92, SINGH94a]. The method can be described as follows: before every dig, a planner forms the set of all the feasible digs that the excavator might perform. From this set, one dig that optimizes a specified cost function (e.g. amount of soil excavated) is chosen. A complete plan is formed by concatenating such one dig plans until the goal is achieved. Statement of the set of all "feasible" digs requires a method which will distinguish between plans that will succeed and those that will fail. While the geometric constraints are easy to specify (e.g. a test that will determine if a particular plan will fail because it requires the excavator to reach beyond its work space, or, to violate the shape of the goal terrain), force constraints are harder to specify. It is important to specify force constraints because in general there are many plans that are geometrically feasible but cannot be accomplished because the excavator cannot generate the necessary forces. In this paper we develop methods to estimate forces required to perform excavation for a given excavator and a given terrain. Such a method can then be used to properly state the force constraint for automatically planning digging motions.

In the following sections we first examine related work done by other researchers. Next we develop a simplified analytical model of a flat blade moving through soil. We show how this method can be extended to account for phenomena specific to excavation, using learning methods.

II. Relation To Other Work

There has been some research on the operation of earthmoving machinery [ZELENIN86, ALEKSEEVA85] that explicitly addresses the issue of estimating forces necessary to overcome the shear strength of soil. Unfortunately, this work is mostly stated in empirical terms for specific types of machines and it is not clear how to extrapolate the methodology for arbitrary mechanisms. A considerable amount of research has explored finite element analysis of the soil plasticity, for example [CUNDALL88a]. While this work performs a painstaking analysis of soil displacement, it is not suited for our purposes since it requires a very large number of iterations of numerical integration for each problem. Faster, but more approximate methods have been developed for the purposes of simulation [LI93] but the emphasis is on "realistic" looking motion of soil, not on accuracy of the force model.

There has been some work in the field of agricultural engi-

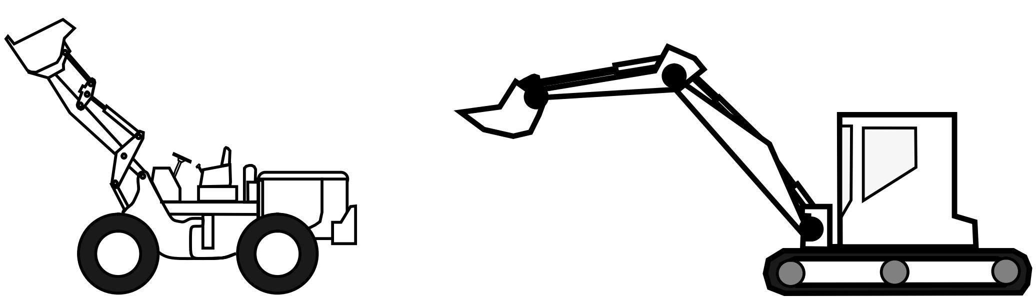

Fig. 1 Earthmoving Machines: loader (left), front shovel (right).

neering that has been directed at producing estimates of cutting resistance for tilling implements [HETTIATRATCHI88, GILL68], using well understood physical principles. This work is important in understanding the mechanics of simple motions of a blade moving through soil. We have found that this analysis accounts for first order effects and allows for order of magnitude predictions. Our work extends this analysis to account for phenomena specific to excavation.

A more complete description of the work reported in this paper can be found in [SINGH94b].

III. The Mechanics Of Excavation

We start with some basic principles in soil mechanics that are necessary to develop the mechanics of soil-tool interaction. Next we develop a static force analysis for the case of a flat blade moving through soil and discuss the relevance of this model to the case of an excavator bucket moving through soil.

A. Basic Soil Mechanics

To understand failure in a soil mass, it is necessary to review some basic theory in the area of soil mechanics. Fundamentally, we want to develop the notion of *shear strength* and *shear stress*. The basis of soil strength is ascribed to Coulomb who noted that there appeared to be two mechanical processes that determine the shearing strength of a material. He noted that one process (friction) was proportional to the pressure acting perpendicularly on the shearing surface. The other process (cohesion) seemed to be independent of normal pressure. Coulomb modeled the shear strength of a soil, τ, as a sum of these two components:

$$\tau = c + \sigma \tan \phi \tag{1}$$

where c is the cohesion, σ is the normal pressure acting on the internal shear surface and $\tan \phi$ is the coefficient of sliding friction. ϕ is also called the angle of internal friction and is directly visible as the angle of repose of a pile of dry, uncompacted granular material like sand or sugar. When soil shears, it does so along a surface called a *failure* surface. The *shear strength* of a soil can be understood to be the resistance per unit area to deformation along a surface of failure. *Shear stress*, on the other hand, is the pressure that pushes soil to move along a failure surface.

While many phenomena such as particle size, chemical composition, compaction, and moisture content may affect soil properties, gross soil behavior can be studied satisfactorily by the simple parameterization (c, ϕ). In case the soil is compacted or is saturated, these parameters (if they could be measured accurately) are “effective” values. Several laboratory instruments exist to characterize soil in such a manner, although obtaining soil properties *in situ* (in the field) has the advantage that the soil is in its natural state. Since our case involves interaction of the soil with a tool, we will also need parameters for soil-tool friction(δ) and soil-tool adhesion (c_a).

B. The case of a flat blade moving through soil

Lets consider a simple case of soil-tool interaction to start. A flat blade (infinitely wide and high) is pushed along a horizontal trajectory as shown in Fig. 2(a). Qualitatively, the total force applied by the blade is a function of the shape of the terrain, soil properties, soil-tool interaction, and tool motion. An examination of the mechanics reveals that there are distinct phenomena that contribute to the total reaction force experienced by the blade. The total resistive force is comprised of forces to shear soil along the failure surface, overcome friction between the blade and soil, overcome adhesion between the blade and soil, and to accelerate the soil in front of the blade. Of these, the force necessary to shear the soil is most significant while the adhesion between the blade and soil is negligible. At low speeds, the forces required to accelerate soil in front of the blade can be ignored; hence our analysis will be *static*.

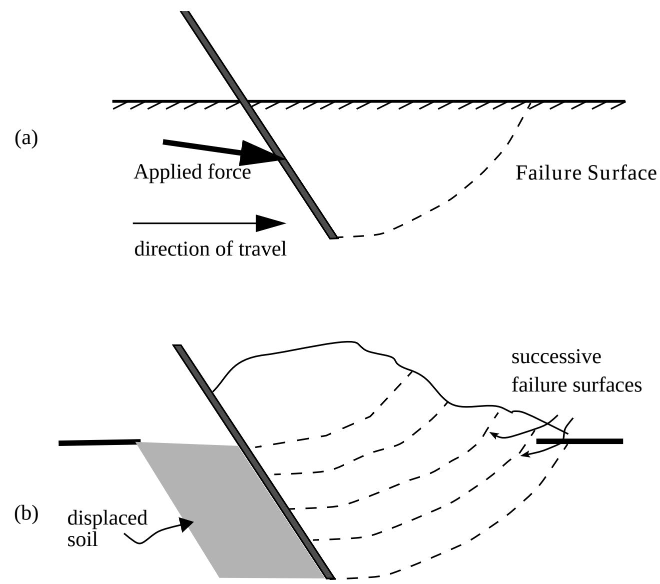

Fig. 2 (a) A flat blade moving through soil. Soil shears along an internal surface known as the rupture surface and a wedge of soil slides along this surface. (b) State of soil after some movement of blade. The soil has failed successively along failure surfaces and the displaced soil now rests on the moving wedge. The displaced soil is called *surcharge* or *overburden*.

C. Predicting resistive forces using a static analysis

One reason to start with the case of a wide, flat blade is that this scenario has been examined extensively in civil engineering and agricultural engineering. Civil engineering has developed an analysis of the passive pressure applied by foundations and retaining walls on adjacent soil masses [TERZAGHI47]. Agricultural engineering has developed analyses of resistive forces experienced by tools necessary to perform tillage.

In this section we develop an approximate equilibrium model for the mechanics of a flat blade, called the *wedge* model[1]. We will model the case just before the soil fails along a new rupture surface using the approximation shown in Fig. 3. As the blade moves from left to right, soil rides up the metal and up the rupture surface. The blade resists the soil with a force equal to the sum of the perpendicular force that the blade provides, the frictional force (soil-tool) and the adhesion (soil-tool). Similarly, the soil resists shearing by a force equal to the

[1] after the shape of mass of soil that is presumed to move along the rupture surface.

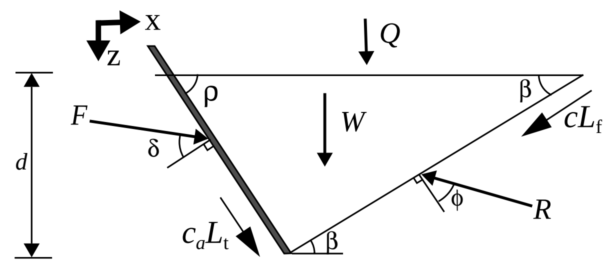

Fig. 3 Static equilibrium analysis using an approximation of the failure surface. W is the weight of the moving soil wedge, L_t is the length of the tool and L_f is the length of the failure surface, Q is the surcharge pressure, R is the force resisting movement of the wedge and F is the total resistive force. From [McKyes85].

frictional force (soil-soil) and the cohesion along the entire rupture surface. Force equilibrium equations for a blade of unit width can be written as

$$\sum f_x = \begin{array}{l} F\sin(\rho+\delta) + c_a L_t \cos\rho \\ -R\sin(\beta+\phi) - cL_f\cos\beta \end{array} = 0 \tag{2}$$

$$\sum f_z = \begin{array}{l} -F\cos(\rho+\delta) + c_a L_t \sin\rho \\ -R\cos(\beta+\phi) + cL_f\sin\beta + W + Q = 0 \end{array} \tag{3}$$

After some manipulation, the total resistive force F, (ignoring adhesion) for a tool of width w, can be written as:

$$\begin{array}{c} F = \left(\gamma g d^2 N_\gamma + c d N_c + q d N_q\right) w \\ \angle F = \pi/2 - \rho - \delta \end{array} \tag{4}$$

where

$$N_\gamma = \frac{\cot\rho + \cot\beta}{2[\cos(\rho+\delta) + \sin(\rho+\delta)\cot(\beta+\phi)]} \tag{5}$$

$$N_c = \frac{1 + \cot\beta\cot(\beta+\phi)}{[\cos(\rho+\delta) + \sin(\rho+\delta)\cot(\beta+\phi)]} \tag{6}$$

$$N_q = \frac{\cot\rho + \cot\beta}{[\cos(\rho+\delta) + \sin(\rho+\delta)\cot(\beta+\phi)]} \tag{7}$$

and $\angle F$ is the angle of the resistive force from the horizontal. (4) is written in this way because it is a familiar form also known as the *Fundamental Equation of Earthmoving* (FEE) [REECE64]. Coulomb proposed that the shape of the failure surface is such that the force required to produce failure is minimum. This shape has been accurately modeled using logarithmic spirals, but here we will approximate it by a plane (a straight line in two-dimensions). Since the soil frictional properties alone determine the failure zones in soil cutting, the most likely slip angle β is that which causes N_γ to be minimum. In effect, this identifies path of least resistance for the soil to fail. When different values of β (between 0 and 90 degrees) are tried, $N\gamma$ varies and the most likely value of β is found at the minimum value of $N\gamma$, and is then used to calculate the other terms.

Prediction of the reactive force starts with calculation of β and the N factors. These are only dependant on the soil parameters and the geometry of tool. If the soil properties are constant and the tool moves without changing its rake angle, the N factors need not be recomputed. (4) is solved for the tool at regular increments along the line of travel. At each increment, the amount of soil displaced (by the movement of the tool) is calculated and is used a measure of the surcharge (Fig. 2(b)). Although it is possible to obtain order of magnitude predictions using this procedure for a tool moving along an arbitrary path, there are several caveats to using this analysis:

- it ignores all dynamic effects such as the forces required to accelerate the soil mass.
- it approximates shape of failure surface;
- it presumes that the surcharge is uniform over the failure wedge. In fact, the surcharge due to the displace soil is concentrated close to the blade.
- it assumes that the soil being pushed in not confined in any way. This allows us to ignore dilation of the soil, which is an otherwise important effect.
- it models forces just before failure. If the blade moves purely horizontally, the soil mass in front of the blade is constantly being brought to failure. However, if the blade moves in arbitrary directions, not only will the wedge geometry vary, but also failure might not be always be imminent.

D. Extension to the excavation scenario

Above we have developed an approximate model for an infinitely wide blade moving through soil along a horizontal trajectory at a constant depth. Let's compare the case we have modeled to that of an excavator bucket going through typical digging motions.

First, excavation tools have finite width and usually are longer than they are wide. The tillage literature has developed special models for long tools because they cause failure in the soil lateral to direction of movement as well as along the direction of movement. This means that in the general case, we will not be able to use the two dimensional analysis developed above. McKyes [MCKYES85] notes that in the case the an excavation tool has side-walls (as with excavator buckets), then the walls help push the soil into the bucket and constrain the failure to a volume directly ahead of the bucket. That is, a two dimensional analysis will suffice. Second, the cutting surfaces of many excavation tools (dozer blades, excavator bucket) are curved and in some cases shaped such that after a certain amount of soil is "scooped", the soil becomes captive and additional travel results in compression in addition to shearing. This stands in contrast to the flat blade assumption of our model above. Third, excavation often occurs in uneven terrain resulting in varying depth as opposed to the relatively constant depth assumed in the case of tillage. Lastly, excavation tools will be required to rotate at various rates as opposed to tillage tools which are typically required to translate at a fixed orientation.

So far we have considered only very simple motions of the excavating tool. We could use the FEE to predict the forces during realistic excavation. We find that to start with, the wedge model provides reasonable prediction, but as the tool proceeds, the quality of the prediction decreases. We have conducted a series of experiments in our experimental testbed (described in the next section) and have recorded forces during movement of an excavator bucket. Fig. 4 compares the recorded resistive forces to those predicted by the wedge model. Below we discuss how prediction can be improved using a variety of methods.

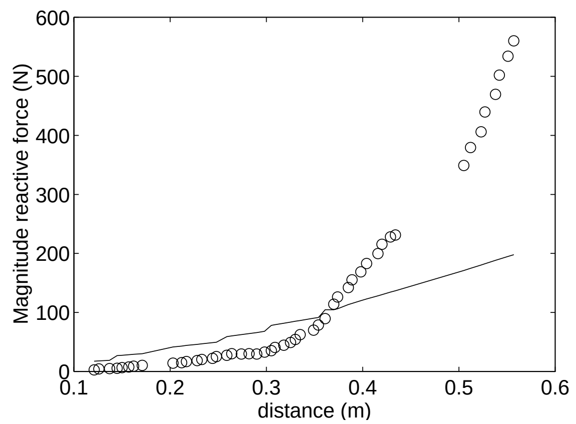

Fig. 4 Use of the wedge model to compute resistive force during excavation.

IV. EXPERIMENTAL TESTBED

We have developed a testbed to conduct experiments in subsurface sensing and excavation. The testbed consists of a sandbox (2.5m x 2.5m x 1m), a Cincinnati Milacron T3 hydraulic robot outfitted with an excavator bucket and force sensor, and a laser range finder. The setup of our testbed is shown in Fig. 5.

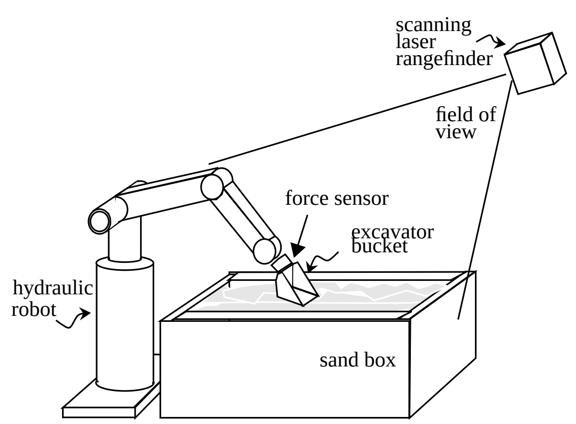

Fig. 5 Testbed

The robot we use is a large industrial manipulator with an end effector payload of approximately 125 lbs. We have attached a small excavator bucket with a volume of $0.01m^3$ as an end effector. The laser range scanner produces an image of the terrain such that the value of each pixel in the image is based on the distance from the scanner to the world along a ray that sweeps in a raster fashion. The scanner has a 60 degree horizontal field of view and a 45 degree vertical field of view. We have adapted a perception and mapping system developed at CMU [HOFFMAN92] to produce terrain elevation maps for our task. An example terrain map of the testbed is shown in Fig. 6 Each cell in the terrain map is 5 cm square.

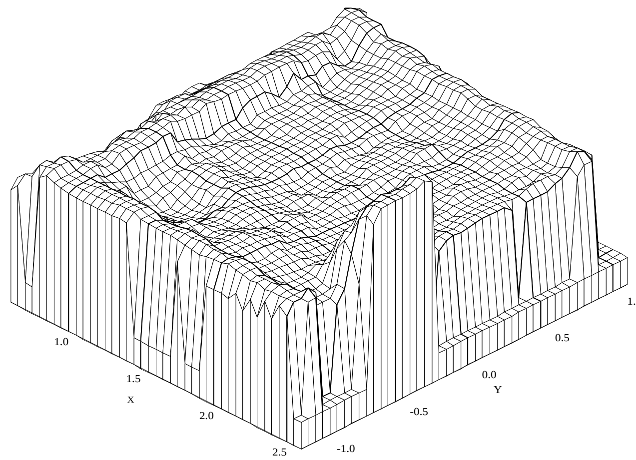

Fig. 6 Elevation map of testbed.

To illustrate the difference in prediction produced by the combinations of representations and learning methods, we consider two examples of excavation. Fig. 7 shows two digs that were performed. We will compare the force data collected from these experiments to force data predicted based on prior experiments (approximately 4000 measurements from 30 digs are used).

V. LEARNING TO PREDICT RESISTIVE FORCES

We use "learning" to loosely mean the ability to improve performance with experience. For our purposes, we are specifically interested in a function approximation scheme that is able to better predict resistive forces based on experimental data. In this section we examine two fundamental factors in function approximation— *representation* and *methodology.* We contend that good representations significantly impact learning methods. In particular we claim that representations motivated by physical models provide a substantial benefit. We examine three learning methods (*global regression*, *memory based learning* and *neural nets*) and show how these differ in terms of performance using several criteria (accuracy, training time, prediction time and memory requirements).

We start with a physical model— the FEE. Given a set of force data along with shape of the terrain and the trajectory followed by a excavating tool, we seek the unknown parameters that best fit the observations. For various reasons this model is awkward to work with and is deficient in capturing the complexity of soil-tool interaction during excavation. In response we have developed a formulation motivated by the wedge model but that is easier to work with. Ideally, we are looking for is a set of basis functions, *G*, that are linearly related to the

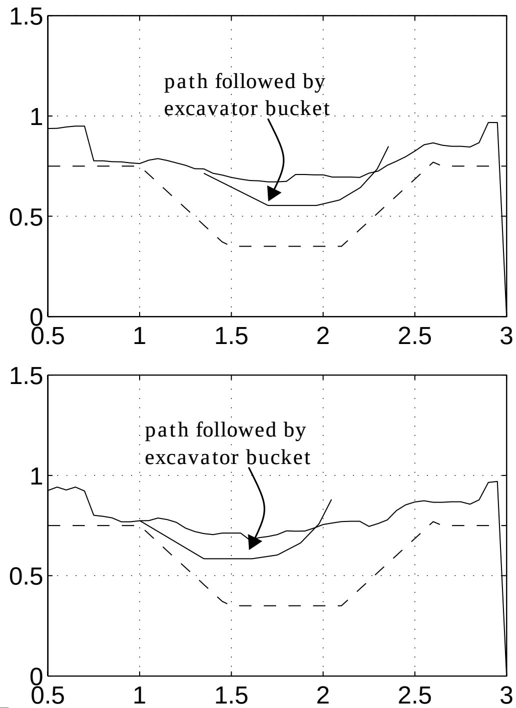

Fig. 7 Two digs. (a) and (b) show elevation profiles of the soil in testbed along the length of the trench that is to be created (marked by dashed lines). Forces at the bucket tip were measured at approximately 8 hz. α_a = 25 degs, α_b = 30 degs, $d_{1a} = d_{1b}$ = 0.38 m, $d_{2a} = d_{2b}$ = 0.26 m.

unknowns. That is:

$$G_1 k_1 + \ldots + G_n k_n = F \tag{8}$$

where the k_i are the unknown parameters. In this section we show how G is selected and how the semantics of G affect performance. For every force reading taken during excavation, we can write a separate equation. This is a familiar case of an over-determined system and can be readily solved.

A. Fitting wedge model parameters to observed data

Since the FEE is highly non-linear, complete inversion to determine the unknowns (δ, ϕ, c and γ) is difficult and we will need a numerical method. If we pose the determination of unknowns as a minimization of the error between observed outputs and predicted outputs, a large number of methods can be used (for example, the *downhill simplex* method and *conjugate gradient descent* [PRESS88]). In general, non-linear minimization is a difficult process requiring iterative refinement. Since we present the wedge model only for comparison, we have used a brute force approach in which we simply search for the values of the unknown variables (c, ϕ, δ, γ) that best fit the observed force data. This will involve an exhaustive search that is exponential in the number of unknowns.

Brute force search to find the parameters (ϕ, c, δ, γ) from the wedge model that best fit the observed force data results in a number of solutions that are very close in their error metric. Ranking the solutions by their ability to minimize the error, we notice that there are a large number of solutions within 1% of each other, suggesting that there are numerous local minima in the function. The advantage of this method is that the representation is very compact and once the unknowns are found, prediction is fast. On the downside, the form of the wedge model and the accompanying FEE is hard to work with— the non-linear nature and multiple local extrema make data fitting a difficult task. Our brute force method that requires a very large number of evaluations of the FEE. Discretizing the four unknown parameters into just 20 intervals, the search required several days of computation.

Although we are able to find a set of parameters that produce a mean absolute error of approximately 100 newtons[2] in the magnitude of the resistive force, qualitatively the predictions produce average estimates, and do not fit the measured data well at any part of an excavation. Accuracy is improved over the prediction done using nominal parameters (we roughly know the values of (ϕ, c, δ, γ) for our testbed) in that the errors are attenuated but the fundamental deficiency in accounting for second order effects remains. The wedge model does provide a bound on the errors that will be acceptable by other methods. Below we will see how this method compares to others.

B. Selecting a linear basis

Let us turn our attention to a representation that is easier to work with. To review the problem at hand, we are looking for a method to predict future resistive forces based on past experimental data. The resistive force, F, is a function of the robot's action (A) and the world state (W):

$$F = f(A_i \times W_i) \tag{9}$$

If we choose G to be an appropriate encoding of A,W we can write

$$A_1 k_1 + \ldots + A_n k_n + W_{n+1} k_{n+1} + \ldots + W_{n+m} k_{n+m} = F \tag{10}$$

where n and m are the number of variables needed to encode A and W respectively. Since we are interested in both the planar components of F, we will need to write two separate equations one for each of F_x and F_z. Now it is possible to use a least-squares method to determine the unknown k_i.

It remains to be shown how to choose the basis functions G. We present three ways to build a linear basis. To start we consider a "naive" formulation. This formulation is based simply on a compact geometric representation of actions and world states. Next, we use the analysis of the mechanics to motivate the basis functions. Finally, we suggest a reduced formulation, based on empirical results.

Consider the trenching action in Fig. 8. Lets say that we perform a number of digging actions. Before the digging starts, we record the shape of the terrain as a set of l_i and c_i from an elevation map. While the excavator moves along a path, we record the position of the bucket and the forces at the bucket tip measured by a force sensor at a fixed interval of time. For each

[2] This is the best possible case since this number is simply the residual error from fitting the entire data set.

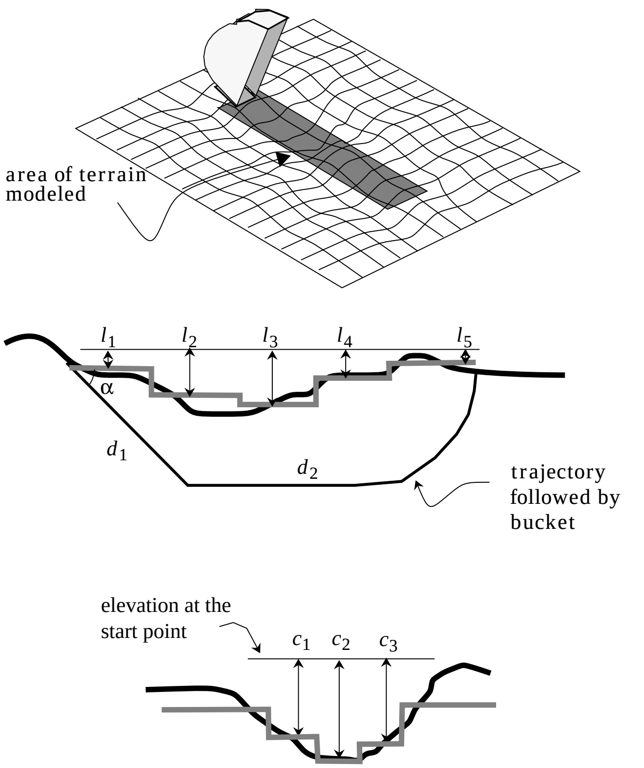

Fig. 8 Encoding a trenching action and the state of the terrain. Here we encode an atomic dig with (α, d_1, d_2). The world state is represented by a sub-sampled elevation profile along the trench to be excavated (l_i) and across the trench (c_i).

force reading we create the basis shown in (11). For a particu-

$$\alpha, d_1, d_2 \qquad l_1, l_2, l_3, l_4, l_5 \qquad c_1, c_2, c_3 \tag{11}$$

lar dig, α and l_i are fixed. d_1, d_2, c_1, c_2 and c_3 vary with the position of the bucket (Fig. 9).

This formulation is purely geometrical. The next formulation will use insights gained from the above analysis. For instance, we know that the resistive force is proportional to the square of the depth of the tip of the cutting surface, and that the resistive force depends on the profile of soil in front of the cutting surface. We also know that the resistive force depends on the volume of soil displaced (surcharge) and on the angle of the cutting surface. A *mechanics-based* formulation is shown in (12) where v is the volume of soil displaced. Fig. 10 shows how the semantics of these parameters. From experimentation we have found that some of the variables in the basis do not have much of an effect. For example, the further one gets from the cutting surface, the less likely the terrain is to affect the resistive force. For comparison we have also looked at a reduced

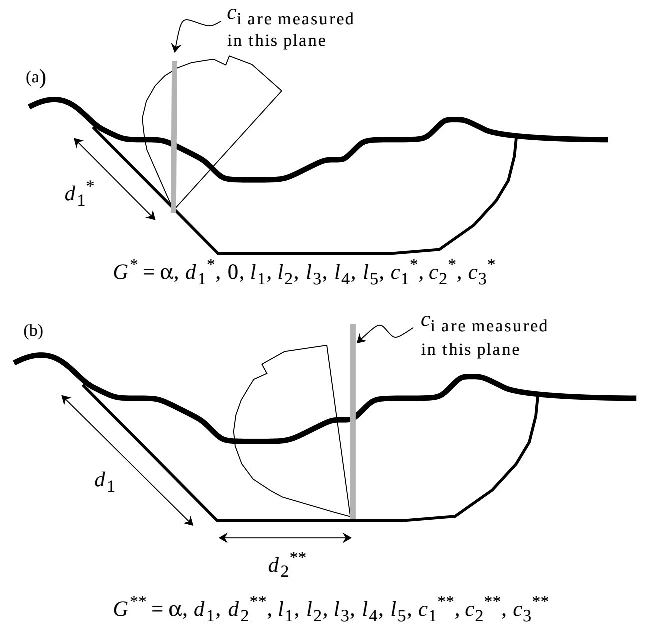

Fig. 9 Two examples of computing G. The bucket moves along a trajectory described by (α, d_1, d_2). (a) the bucket has moved a distance d_1^* along d_1 (b) the bucket has moved d_2^{**} along d_2.

$$\rho, d_1, d_2 \qquad l_1^2, l_2^2, l_3^2, l_4^2, l_5^2 \qquad c_1^2, c_2^2, c_3^2 \qquad v \tag{12}$$

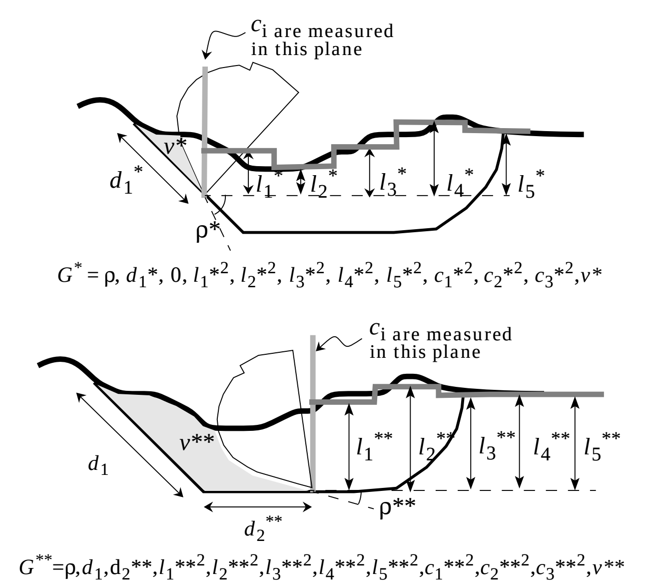

Fig. 10 Computing G based on an analysis of the mechanics. Terrain ahead of the cutting surface is encoded as a sequences of depths (squared values used). Additionally, we encode the rake angle, and volume of soil displaced.

G(13). Below we examine how differences in formulation

$$\rho, d_1, d_2 \qquad l_1^2, l_2^2, l_3^2 \qquad c_2^2 \qquad v \tag{13}$$

affect prediction performance.

VI. Learning Methods

A. Global Regression

Given a set of basis functions G, the unknowns, K, are found by solving the normal equations:

$$K = \left(G^{\mathrm{T}}G\right)^{-1}G^{\mathrm{T}}y \tag{14}$$

(14) is a *batch* method— all data are processed simultaneously. This method can be modified into an *iterative* scheme where data are incorporated sequentially, as they arrive. It is noteworthy that if the basis functions are motivated by the mechanics, global regression is able to produce good predictions in the presence of noise and in the absence of data neighboring a query point. However, complex systems are not easily represented by methods that model all the data simultaneously, with the net result that even if the residual fit error is low, no part of the function is represented well.

The net result of global regression is two sets of 12 parameters (one each for f_x and f_z). It is instructive to examine the magnitude of these weights. To a first approximation they tell us that distance along d_1 and d_2, depth of the bucket tip, l_1, and the volume displaced, v, are significant factors. The weights also show rapidly decreasing effect from elevation of the soil as one gets further away from the bucket tip (l_1.. l_5). This agrees with the intuition developed in the analysis

That global regression produces better results than the wedge model is testimony to the fact that the wedge model is not able to capture the complexity of soil-tool interaction during excavation (Fig. 11).

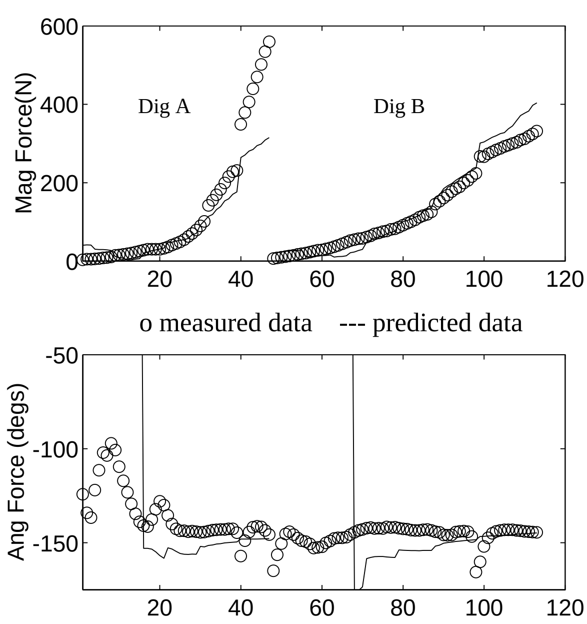

Fig. 11 Prediction using global regression with mechanics based basis functions.

In the batch case, the complexity of this method is dependant on the method used to solve a linear over-determined system. Assuming LU decomposition is used to invert the matrix $G^{\mathrm{T}}G$, the number of computations required are $Dn^2 + nD^2 + D^2/3$ where D is the number of unknowns and n is the number of data points. Once the parameters are known, prediction time is constant.

B. Memory Based Learning

Instead of modelling all the data simultaneously, predictions could be based on local models that are fitted to the data in the neighborhood of a query point. Such methods require that all experiences $\{(X_1, Y_1),..., (X_n, Y_n)\}$ are explicitly stored in memory and hence the term "memory based". A simple memory based method is *nearest-neighbor.* In this method, when a query is made, the memory is searched for the n nearest points in the input space. The dependant values of these points are averaged to provide a prediction. Since there is no model whatsoever, this method is prone to noise. The only recourse is to average over a larger neighborhood. Instead of simply averaging the n nearest points, we might form a linear model over this subset— the surface of the function is approximated by hyperplane patches. A further extension is to weight the points used in the linear model by their proximity to the query point. Intuitively, the further a point is in the input space from the query point, the less it should be weighted. This method is commonly called Local Weighted Regression (LWR) and has been described in [CLEVELAND88, ATKESON91, MOORE93].

Moving from global to local models further improves prediction. Fig. 12 shows force data that are predicted using a local weighted regression scheme. In addition to the method described above we have used a scaled Euclidean distance metric that weighs input parameters by a measure of how strongly they predict the output. Relative importance of input parameters is obtained by analyzing the entire training set using a statistical package called MARS [FRIEDMAN88].

As usual, there is a trade-off in ability to represent the underlying function and noise rejection. Using a local model can do a lot to improve fidelity of fit, even if the basis functions are approximate. Conversely, the weaker the model, the more prone local methods will be to noise and missing data. Use of LWR is predicated on the search for the n nearest neighbors to a query point and for practical purposes a prediction using LWR will require $O(n)$ computations. One way to deal with large data sets is to train on a randomly chosen subset of the training data.

C. Neural Nets

Another way to deal with complex functions is to use a nonlinear mapping between the unknowns and the basis functions. One such method is a back-propagation neural net. Back-propagation works by repeatedly showing a network sample inputs along with the true (perhaps noisy) outputs, while the network learns by adjusting its weights. It is common to connect inputs and outputs with one or more layers of computational nodes called *hidden units*. The simplest computational nodes sum weighted inputs and pass the result through a non-linearity. A node is characterized by an internal threshold θ and by the type of nonlinearity. A commonly used nonlinearity is called the

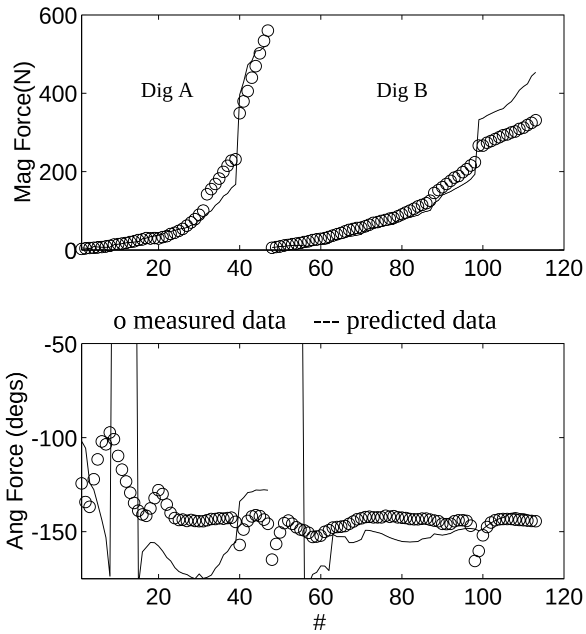

Fig. 12 Prediction using default local weighted regression with mechanics based basis functions.

sigmoid and is defined:

$$\sigma(\alpha) = \frac{1}{1 + e^{-(\alpha - \theta)}} \tag{15}$$

Fig. 13 shows prediction for the two exemplar cases we have been following.

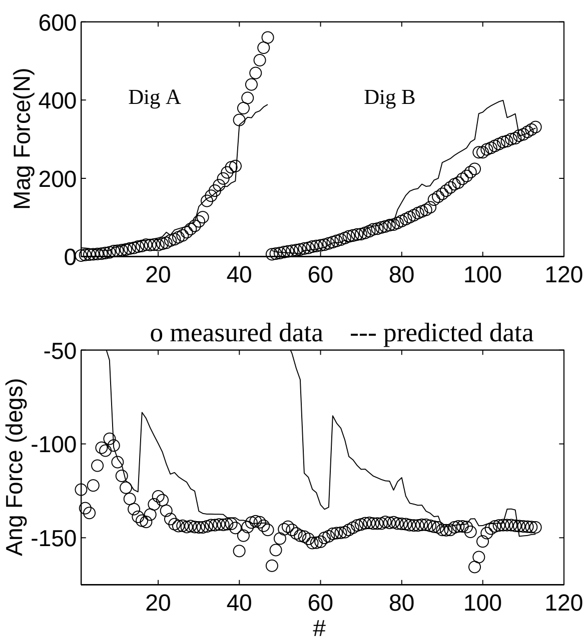

Fig. 13 Prediction using a back propagation neural network (30 hidden units) with mechanics based basis functions.

Training neural networks to approximate a function is somewhat of a black art. There are many parameters to be adjusted and in general, the designer has to try a large number of combinations of parameters to gain satisfactory results. For our application we have found that a single layer back-propagation network with 30 hidden units gives good results when it has been shown the inputs and outputs approximately 10,000 times. There is no clear way of describing the amount of computation required to train a neural net; in our case training a network requires 10-30 hours of computation time. However, once trained prediction is very fast and the representation is very compact.

D. Criteria

So far we have qualitatively compared the various methods with the mechanics based formulation of the basis functions. Below, we present quantitative results to aid in comparison of the method and the various representations.

The first criteria to compare methods and representation is *accuracy*. We have conducted a systematic comparison of the average accuracy that results from the various representations and methods discussed above. Each data set was analyzed using 10-fold cross-validation. The data set was divided into ten blocks of near-equal size. For every block in turn, each method was trained on the remaining blocks and tested on the hold-out block. Results, averaged over all test blocks, thus reflect the predictive performance on unseen cases. The mean absolute error is used as the measure of performance.

TABLE I Mean Absolute Error (N) between predicted and observed magnitude of resistive force. The results are based on a 10 fold cross-validation with the exception of the results for back propagation which was trained on a block of 3000 points and tested with the remaining 1000 points.

	Naive Basis. (11)	Mechanics based basis (12)	Reduced Mechanics based basis (13)
Global Regression	81.64	69.52	70.22
3 Nearest Neighbors	79.27	72.37	72.44
LWR (default)	65.83	65.37	64.15
LWR (MARS scaling)	63.65	62.92	68.82
Back Prop (30 HU)	83.18	64.67	66.91

This comparison confirms a couple of points. First, the mechanics based basis functions provide significantly better accuracy for 3 out of the 5 cases. Improvement is smallest with LWR and in this case it appears that the local nature of the method is able to represent the underlying function well. Second, removing those variables which were detected to have a small influence on the output values does not degrade performance significantly and in one case even improves the error metric. Across methods, apart from discounting global regression and nearest neighbor, it is hard to make a case for particular method based on accuracy alone.

Next we compare methods based on training time, prediction time, ease of incorporating new data and memory requirements during prediction (TABLE II).

TABLE II Other criteria to compare learning methods. *D* is the number of basis functions, *n* is the size of the training set, and *h* is the number of hidden units. Execution times are approximate times for a Sun Sparc 10.

	training time	prediction time	ease of incorporating new data	memory requirements
Wedge Model	days (using brute force search)	~ 10 ms	complete retraining/ search required	4 constants
Global Regression	~10 secs	~1ms	iterative scheme possible	2 sets of D constants
Memory Based Learning	~1 hr	~100ms	easy	all data must be stored: nD
Neural networks	~10 hrs	~1ms	some retraining required	2 sets of h constants

Atkeson has pointed out that if there is a continuously growing training set intermixed with queries then memory-based learning is appropriate [ATKESON91]. Local methods are able to reduce interference between data both spatially (in the input space) and temporally, and generalization depends on the density of the samples. For the excavation domain this means that if very little is known about the soil-tool interaction in a given environment, and we would like to rapidly improve prediction, then memory based methods will be useful. However, if a fixed training set is available before queries are made, then an approach like neural nets will be preferable.

VII. Summary

In this paper we have presented methods to improve prediction of resistive forces experienced during excavation. We started with a simple model of a blade moving through soil and showed how this model is deficient for our purposes. We have used the analysis to develop a formulation which can improve predictions with experience. Upon experimental evaluation, we find that representation based on physical models is superior to a naive representation based purely on geometry.

Acknowledgement

The author would like to thank Andrew Moore for helpful discussions regarding Memory Based Learning. Thanks to Justin Boyan for discussions and help with neural networks

Refrences

[SINGH92] Singh, S. and Simmons, R., Task Planning for Robotic Excavation, Proc. IEEE/RSJ International Conference on Intelligent Robots.

{SINGH94a] Singh, S., Developing Plans for Robotic Excavators, Proceedings ASCE Conference on Robotics for Challenging Environments, 1994.

[ALEKSEEVA85], Alekseeva, T. V. and Artem'ev, K. A. and Bromberg, A. A. and Voitsekhovskii, R. I. and Ul'yanov, N.A., Machines for Earthmoving Work, Theory and Calculations, A. A. Balkema, Rotterdam, 1992.

[ZELENIN86] Zelenin, A. N. and Balovnev, V. I. and Kerov, L. P., Machines for moving the earth, 1992.

[CUNDALL88] Cundall, P., Board, M., “A Microcomputer Program for Modelling Large Strain Plasticity Problems”, Numerical Methods in Geomechanics, Balkema, Rotterdam1988.

[MCKYES85] McKyes, E., *Soil Cutting and Tillage*, Elsevier, Amsterdam, 1985.

[GILL68] Gill, W. R., VandenBerg, G. E., “Soil Dynamics in Tillage and Traction”, Agriculture Handbook No. 316, Agricultural Research Service, US Department of Agriculture, 1968.

[HETTIARATCHI88] Hettiaratchi, D.R.P., “Theoretical Soil Mechanics and Implement Design,” Soil and Tillage Research, Vol. 11, 1988, pp.325- 347, Elsevier Science Publishers.

[ATKESON91] Christopher G. Atkeson, Using Locally Weighted Regression for Robot Learning, Proceedings International Conference on Robotics and Automation, 1991.

[LI93] Xin Li and J. Michael Moshell, Modeling Soil: Realtime Dynamic Models for Soil Slippage an Manipulation, Proceedings of SIGGRAPH93

[SINGH94b], Singh, S., Resistive Forces during Robotic Excavation: Analysis, Experiments, and Methods to improve prediction based on empirical data, Technical Report, Robotics Institute, Carnegie Mellon University, To Appear.

[TERZAGHI47] Terzaghi, K., Theoretical soil mechanics, Wiley, 1947.

[REECE64] Reece, A. R, The fundamental equation of earth moving mechanics, Proceedings of Institution of Mechanical Engineers},179-3F, 1964

[HOFFMAN92] Hoffman, R., and Krotkov, E., “Terrain Mapping for Long Duration Autonomous Walking”, In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Raleigh, 1992.

[PRESS88] Press, W. H and Flannery, B. P. and Teukolsky, S. A. and Vetterling, W. T., Numerical Recipes, Cambridge Press, 1988.

[CLEVELAND88]. S. Cleveland and S. J. Devlin, Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting, Journal of American Statistical Association, 83(403), 1988.

[MOORE93] A. W. Moore, General Memory Based Learning, A Brief Explanation of the concept, the software and the C interface, 1993.

[FRIEDMAN88], Friedman, J. H., Fitting functions to noisy data in high dimensions, Proc., Twentieth Symposium on the Interface, Wegman, Gantz and Miller, 1988.