M.I.T. Media Lab Perceptual Computing Technical Report No. 433

Originally published as: Towards Augmented Control Systems, in IEEE Intelligent Vehicles 95, pp. 350-355, Detroit MI, Sept. 25-26, 1995

Modeling and Prediction of Human Behavior

Alex Pentland
Massachusetts Institute of Technology
Room E15-387, 20 Ames Street
Cambridge, MA 02139, USA
sandy@media.mit.edu

Andrew Liu
Nissan Cambridge Basic Research
4 Cambridge Center
Cambridge, MA, 02142
andy@pathfinder.cbr.com

Abstract

We describe our research toward building systems that include a complex, multi-state model of human dynamic behavior. This can allow us to predict human behavior over short periods of time, in order to create control systems that intelligently complement the human's action. To accomplish this requires inferring the internal state of the human, and then correctly adapting the remainder of the system to achieve optimal performance. We describe methods for achieving this goal, and report an initial experiment in which we were able to achieve 95% accuracy at predicting automobile driver's actions from their initial preparatory movements.

1 Introduction

Our approach is to modeling human behavior is to consider the human as a Markov device with a (possibly large) number of internal 'mental' states, each with its own particular control behavior, and inter-state transition probabilities (e.g., in a car the states might be passing, following, turning, etc.). A simple example of this type of human model would be a bank of standard quadratic controllers, each using different dynamics and measurements, together with a network of probabilistic transitions between them.

To integrate this human model into an optimal control system encopassing both human and machine it is necessary to know which controller is currently "in charge," and to predict transitions between controllers, so that the remainder of the system (e.g., a car) can configure itself to achieve its best overall performance. However the internal states of the human are not directly observable, so they must be determined through an indirect estimation process. One efficient and robust method of accomplishing this is to use the expectation-maximization methods developed for use of Hidden Markov Models (HMM) in speech processing.

By using these methods to identify a user's current internal (intentional) state, and to predict the most-likely subsequent internal state, we expect to be able to design systems that are able to dynamically reconfigure themselves to better fit the situation. This can potentially allow for higher performance than is possible with a fixed model of the human (assuming similar controller complexity).

2 Simple Dynamic Models

Among the simplest non-trivial models that have been considered for modeling human behavior are single dynamic processes

$$\mathbf{X}_{k+1} = \mathbf{f}(\mathbf{X}_k, \Delta t) + \xi(t) \tag{1}$$

where the function $\mathbf{f}$ models the dynamic evolution of state vector $\mathbf{X}_k$ at time k, and let us define an observation process

$$\mathbf{Y}_k = \mathbf{h}(\mathbf{X}_k, \Delta t) + \eta(t) \tag{2}$$

where the sensor observations $\mathbf{Y}$ are a function $\mathbf{h}$ of the state vector and time. Both ξ and η are white noise processes having known spectral density matrices.

Using Kalman's result, we can then obtain the optimal linear estimate $\hat{\mathbf{X}}_k$ of the state vector $\mathbf{X}_k$ by use of the following *Kalman filter*:

$$\hat{\mathbf{X}}_k = \mathbf{X}^*_k + \mathbf{K}_k(\mathbf{Y}_k - \mathbf{h}(\mathbf{X}^*_k, t)) \tag{3}$$

provided that the Kalman gain matrix $\mathbf{K}_k$ is chosen correctly [12]. At each time step k, the filter algorithm uses a state prediction $\mathbf{X}^*_k$, an error covariance matrix prediction $\mathbf{P}^*_k$, and a sensor measurement $\mathbf{Y}_k$ to determine an optimal linear state estimate $\hat{\mathbf{X}}_k$, error covariance matrix estimate $\hat{\mathbf{P}}_k$, and predictions $\mathbf{X}^*_{k+1}$, $\mathbf{P}^*_{k+1}$ for the next time step.

The prediction of the state vector $\mathbf{X}^*_{k+1}$ at the next time step is obtained by combining the optimal state estimate $\hat{\mathbf{X}}_k$ and Equation 1:

$$\mathbf{X}^*_{k+1} = \hat{\mathbf{X}}_k + \mathbf{f}(\hat{\mathbf{X}}_k, \Delta t)\Delta t \tag{4}$$

In our application this prediction equation is also used with larger times steps, to predict the human's future state. For instance, in a car such a prediction capability can allow us to maintain synchrony with the driver by giving us the lead time needed to alter suspension components, etc.

Finally, given the state vector $\mathbf{X}_k$ at time k we can predict the measurements at time $k + \Delta t$ by

$$\mathbf{Y}_{k+\Delta t} = \mathbf{h}(\mathbf{X}_k, \Delta t) \tag{5}$$

and the predicted state vector at time $k+\Delta t$ is given by

$$\hat{\mathbf{X}}_{k+\Delta t} = \mathbf{X}^*_k + \mathbf{f}(\hat{\mathbf{X}}_k, \Delta t)\Delta t \tag{6}$$

3 Multiple Dynamic Models

Human behavior, in all but the simplest tasks, is not as simple as a single dynamic model. The next most complex model of human behavior is to have *several* alternative models of the person's dynamics, one for each class of response. Then at each instant we can make observations of the person's state, decide which model applies, and then make our response based on that model. This is known as the *multiple model* or *generalized likelihood* approach, and produces a generalized maximum likelihood estimate of the current and future values of the state variables [13]. Moreover, the cost of the Kalman filter calculations is sufficiently small to make the approach quite practical.

Intuitively, this solution breaks the person's overall behavior down into several "prototypical" behaviors. For instance, in the driving situation we might have dynamic models corresponding to a relaxed driver, a very "tight" driver, and so forth. We then classify the driver's behavior by determining which model best fits the driver's observed behavior.

Mathematically, this is accomplished by setting up a set of states S, each associated with a Kalman filter and a particular dynamic model:

$$\hat{\mathbf{X}}^{(i)}_k = \mathbf{X}^{*(i)}_k + \mathbf{K}^{(i)}_k(\mathbf{Y}_k - \mathbf{h}^{(i)}(\mathbf{X}^{*(i)}_k, t)) \tag{7}$$

where the superscript (i) denotes the i^{th} Kalman filter. The *measurement innovations process* for the i^{th} model (and associated Kalman filter) is then

$$\Gamma^{(i)}_k = \mathbf{Y}_k - \mathbf{h}^{(i)}(\mathbf{X}^{*(i)}_k, t) \tag{8}$$

The measurement innovations process is zero-mean with covariance $\mathcal{R}$.

The i^{th} measurement innovations process is, intuitively, the part of the observation data that is unexplained by the i^{th} model. The model that explains the largest portion of the observations is, of course, the model most likely to be correct. Thus, at each time step, we calculate the probability $Pr^{(i)}$ of the m-dimensional observations $\mathbf{Y}_k$ given the i^{th} model's dynamics,

$$Pr^{(i)}(\mathbf{Y}_k) = \frac{\exp\left(-\frac{1}{2}\Gamma^{(i)T}_k \mathcal{R}^{-1}\Gamma^{(i)}_k\right)}{(2\pi)^{m/2}\mathrm{Det}(\mathcal{R})^{1/2}} \tag{9}$$

and choose the model with the largest probability. This model is then used to estimate the current value of the state variables, to predict their future values, and to choose among alternative responses.

Note that when optimizing predictions of measurements Δt in the future, Equation 8 must be modified slightly to test the predictive accuracy of state estimates from Δt in the past.

$$\Gamma^{(i)}_k = \mathbf{Y}_k - \mathbf{h}^{(i)}(\mathbf{X}^{*(i)}_{k-\Delta t} + \mathbf{f}^{(i)}(\hat{\mathbf{X}}^{(i)}_{k-\Delta t}, \Delta t)\Delta t, t)) \tag{10}$$

by substituting Equation 6.

3.1 Results Using Multiple Dynamic Models

We have used this method to accurately remove lag in a high-speed telemanipulation task by continuously re-estimating the user's arm dynamics (e.g., tense and stiff, versus relaxed and inertia-dominated) [3].

In this case, the state vector $\mathbf{X}_k$ consists of the true position, velocity, and acceleration of the hand in each of the x, y, and z coordinates, and the observation vector $\mathbf{Y}_k$ consists of the position readings for the x, y, and z coordinates. We found that using this multiple-model approach we were able to obtain significantly better predictions of the user's hand position that was possible using a single dynamic or static model.

4 Hidden Markov Dynamic Models

In the above multiple dynamic model, all the processes have a fixed likelihood at each time step. However, this is uncharacteristic of most situations, where there is a fixed sequence of internal states each with its own dynamics. Consider driving through a curve; the driver may be modeled as having transitioned through a series of states $\lambda = (s_1, s_2, ...s_k)$, $s_i \epsilon S$, for instance, entering a curve, in the curve, and exiting a curve. Transitions between these states happened only in the order indicated, with a final transition from other to entering the curve.

Thus in considering state transitions among a set of dynamic models we should make use of our current estimate of the driver's internal state. We can accomplish this fairly generally by considering the Markov probability structure of the transitions between the different states. The input to decide the person's current internal state (e.g., which dynamic model currently applies) will be the measurement innovations process as above, but instead of using this directly in Equation 9 we will instead also consider the Markov inter-state transition probabilities.

While a substantial body of literature exists on HMM technology [5, 6, 8, 11], we will first briefly outline a traditional discussion of the algorithms. After outlining the fundamental theory in training and testing of a discrete HMM, we will generalize these results to the continuous density case applicable to switching between dynamic models. For broader discussion of the topic, [6, 9] are recommended.

A time domain process demonstrates a Markov property if the conditional probability density of the current event, given all present and past events, depends only on the j^{th} most recent events. If the current event depends solely on the most recent past event, then the process is a first order Markov process.

The initial topology for an HMM can be determined by estimating how many different states are involved in the observed phenomenon. Fine tuning this topology can be performed empirically. Figure 1, for instance, shows a four state HMM with skip transitions that we have used to classify complex hand motions.

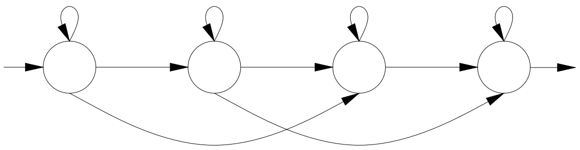


Figure 1: The four state HMM used for recognition, from [4].

There are three key problems in HMM use. These are the evaluation, estimation, and the decoding problems. The evaluation problem is that given an observation sequence and a model, what is the probability that the observed sequence was generated by the model ($Pr(\mathbf{Y}|\lambda)$) (notational style adapted from [6])? If this can be evaluated for all competing models for an observation sequence, then the model with the highest probability can be chosen for recognition.

The Viterbi algorithm provides a quick means of evaluating a set of HMM's in practice as well as providing a solution for the decoding problem. In decoding, the goal is to recover the state sequence given an observation sequence. The Viterbi algorithm can be viewed as a special form of the forward-backward algorithm where only the maximum path at each time step is taken instead of all paths. This optimization reduces computational load and additionally allows the recovery of the most likely state sequence. The steps to the Viterbi are

- Initialization. For all states i, $\delta_1(i) = \pi_i b_i(Y_1)$; $\psi_i(i) = 0$
- Recursion. From $t = 2$ to k and for all states j, $\delta_t(j) = \max_i[\delta_{t-1}(i)a_{ij}]b_j(Y_t)$; $\psi_t(j) = \mathrm{argmax}_i[\delta_{t-1}(i)a_{ij}]$
- Termination. $P = \max_{s \epsilon S}[\delta_k(s)]$; $s_k = \mathrm{argmax}_{s \epsilon S}[\delta_k(s)]$
- Recovering the state sequence. From $t = k-1$ to 1, $s_t = \psi_{t+1}(s_{t+1})$

Note that since Viterbi only guarantees the maximum of $Pr(\mathbf{Y}, S|\lambda)$ over all state sequences S (as a result of the first order Markov assumption) instead of the *sum* over all possible state sequences, the resultant scores are only an approximation. However, [8] shows that this is often sufficient.

4.1 The Continuous Case

So far this discussion of HMMs has assumed some sort of quantization of feature vectors into classes, whereas the innovations processes that will drive our inter-state transitions are continuous. Consequently, instead of using vector quantization, we must employ the actual probability densities for the innovations processes. Fortunately, Baum-Welch parameter estimation, the Viterbi algorithm, and the forward-backward algorithms can be modified to handle a variety of characteristic densities [7]. In this paper, however, the densities will be assumed to be Gaussian. Specifically, from Equation 9,

$$b_j(Y_t) = \frac{\exp\left(-\frac{1}{2}\Gamma_k^{(i)T}\mathcal{R}^{-1}\Gamma_k^{(i)}\right)}{(2\pi)^{m/2}\mathrm{Det}(\mathcal{R})^{1/2}} \quad (11)$$

Initial estimations of μ and σ may be calculated by dividing the evidence evenly among the states of the model and calculating the mean and variance in the normal way. Whereas flat densities were used for the initialization step before, the evidence is used here. Now all that is needed is a way to provide new estimates for the output probability. This can be accomplished by the Kalman filter update equations.

5 An Experiment Using Hidden Markov Dynamic Models

We are now using this approach to identify automobile drivers' current internal (intentional) state, and their most-likey subsequent internal state. In the case of driving the macroscopic actions are events like turning left, stopping, or changing lanes. The internal states are the individual steps that make up the action, and the observed behaviors will be changes in heading and acceleration of the car.

The intuition is that even apparently simple driving actions can be broken down into a long chain of simpler subactions. A lane change, for instance, may consist of the following steps (1) a preparatory centering the car in the current lane, (2) looking around to make sure the adjacent lane is clear, (3) steering to initiate the lane change, (4) the change itself, (5) steering to terminate the lane change, and (6) a final recentering of the car in the new lane. In our current study we are statistically characterizing the sequence of steps within each action, and then using the first few preparatory steps to identify which action is being initiated.

To recognize which action is occuring one compares the observed pattern of driver behavior to hidden Markov dynamic models of each action, in order to determine which action is *most likely* given the observed pattern of steering and acceleration/braking. This matching can be done in real-time on current microprocessors, thus potentially allowing us to recognize a drivers' intended action from their preparatory movements.

If the pattern of steering and acceleration is monitored internally by the automobile, then the ability to recognize which action the driver is beginning to initiate can allow intelligent cooperation by the vehicle. If heading and acceleration is monitored externally via video cameras, as in Figures 2 and 3 (the

Figure 2: Video data of Central Square, Cambridge MA.

Figure 3: Vehicle tracks extracted from spatially-rectified video data.

'blob' processing algorithm that extracts the vehicle parameters from video is described in Boer, Fernandez, Pentland, and Liu 1996 [2]), then we can more intelligently control the traffic flow.

5.1 Experimental Design

The goal is to test the ability of our framework to characterize driver's steering and acceleration/braking patterns in order to classify the driver's intended action. The experiment was conducted within the Nissan Cambridge Basic Research driving simulator, shown in Figure 4(a). The simulator consists of the front half of a Nissan 240SX convertible and a 60 deg (horizontal) by 40 deg (vertical) image projected onto the wall facing the driver. The 240SX is instrumented to record driver control input such as steering wheel angle, brake position, and accelerator position.

Eight adult male subjects were instructed to use this simulator to drive through an extensive computer graphics world, illustrated in Figure 4(b). This world contains a large number of buildings, many roads with standard markings, and other moving cars. Each subject drove through the simulated world for approximately 20 minutes, during which time the driver's control of steering angle and steering velocity, car velocity and car acceleration were recorded at $1/10^{th}$ second intervals.

From time to time during this drive text commands were presented on-screen, whereupon the subjects had to assess the surrounding situation, formulate a plan to carry out the command, and then act to execute the command. The variables of command location, road type, surrounding buildings, and traffic conditions were varied randomly thoughout the experiment. The dynamic models used were specific to a Nissan 240SX, as described in the appendix.

The commands included: (1) stop at next intersection, (2) turn left at next intersection, (3) turn right at next intersection, (4) change lanes, (5) pass the car in front of you, and (6) drive normally with no turns or lane changes. A total of 72 stop, 262 turn, 47 lane change, 24 passing, and 208 drive-normal episodes were recorded. The time needed to complete each command varied from approximately 5 to 10 seconds, depending upon the complexity of both the action and the surrounding situation.

Using the steering and acceleration data recorded while subjects carried out these commands, we built three-state models of each type of driver action (stopping, turn left, turn right, lane change, car passing, and drive-normal) using the estimation tools provided by the Entropic's HTK computer software [11].

To assess the classification accuracy of these models we combined them with the Viterbi recognition algorithm, and examined the stream of drivers' steering and acceleration innovations in order to detect and classify the driver's actions. We then examined the computer's classifications immediately after each command, and recorded whether or not the computer had correctly labeled the action.

Recognition results were tabulated at one second after the presentation of a command to the subject. Note that the minimum response time to a command is approximately 0.5 seconds, so that the one-second point is at most one-half second after the beginning of the driver's action. The one-second point, therefore, is roughly $1/10^{th}$ of the way through the action.

To obtain unbiased estimates of recognition performance, we employed the "leaving one out" method. In this method models are trained on seven subjects and then tested on the eighth subject. This is then repeated eight times, each time leaving out a different one of the eight subjects, and the eight sets of recognition statistics are averaged.

5.2 Results

At one second after the command presentation ($\approx$ 0.5 seconds after the beginning of action, and roughly 10% of the way through the action) mean recognition accuracy was $95.24\% \pm 3.1\%$. These results demonstrate that many types of driving behavior are sufficiently stereotyped that they are reliably recognizable from observation of the driver's preparatory movements.

To test whether our sample is sufficiently large to adequately encompass the range of between-driver variation, we compared these results to the case in which we train on all subjects and then test on the training data. In the test-on-training case the recognition accuracy was 98.8%, indicating that we have a sufficiently large sample of driving behavior in this experiment.

While these results are very promising, caution must be taken in transferring them to real-world driving. It is possible, for instance, that there are driving styles not seen in any of our subjects. Similarly, the driving conditions found in our simulator do not span the entire range of real driving situations. We belive, however, that our simulator is sufficiently realistic that comparable accuracies can be obtained in real driving. Moreover, there is no strong need for models that suit all drivers; most cars are driven by a relatively small number of drivers, and this fact can be used to increase classification accuracy.

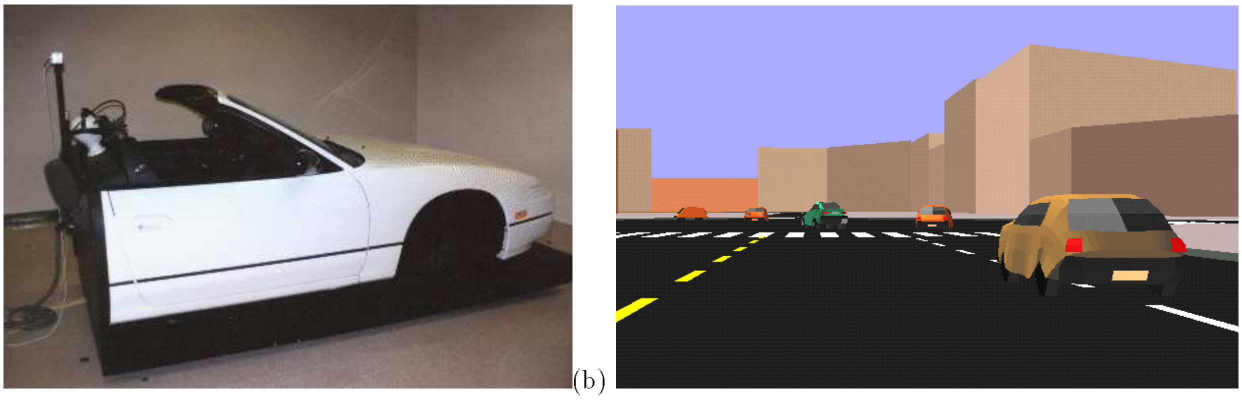
6 Conclusion

We have demonstrated that we can accurately categorize drivers' actions very soon after the beginning of the action using our behavior modeling methodology. Because of the generic nature of the driving task, there is reason to believe that this approach to modeling human behavior will generalize to a wide variety of human-machine systems. This would allow us to automatically recognize the people's *intended action*, and thus to build control systems that dynamically adapt to better suit the human's purpose.

Acknowledgements. We would particularly like to thank Thad Starner for his work in developing HMM technology for gesture understanding.

References

[1] M. Land. Predictable eye-head coordination during driving, *Nature*, Vol. 359, pp. 318-320, 1992.



(a) (b)

Figure 4: (a) Nissan Cambridge Basic Research simulator, (b) part of the simulated world seen by the subjects.

[2] Boer, E., Fernandez, M., Pentland, A., and Liu, A., (1996) "Method for Evaluating Human and Simulated Drivers in Real Traffic Situations," *IEEE Vehicular Tech. Conf.*, Atlanta, GA.

[3] M. Friedmann, T. Starner, and A. Pentland. Device Synchronization using an Optimal Linear Filter, *Proc. ACM 1992 Symposium on Interactive 3D Graphics*, Boston, MA, May 1992.

[4] T. Starner, and A. Pentland. Visual Recognition of American Sign Language Using Hidden Markov Models, *Proc. Int'l Workshop on Automatic Face- and Gesture-Recognition*, Zurich, Switzerland, June 26-28, 1995.

[5] L. Baum. An inequality and associated maximization technique in statistical estimation of probabilistic functions of markov processes. *Inequalities*, 3:1–8, 1972.

[6] X. Huang, Y. Ariki, and M. Jack. *Hidden Markov Models for Speech Recognition.* Edinburgh Univ. Press, Edinburgh, 1990.

[7] B. Juang. Maximum likelihood estimation for mixture multivariate observations of markov chains. *AT&T Technical Journal*, 64:1235–1249, 1985.

[8] L. Rabiner and B. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, p. 4–16, Jan. 1996.

[9] T. Starner. Visual Recognition of American Sign Language Using Hidden Markov Models. Master's thesis, MIT Media Laboratory, Feb. 1995.

[10] T. Starner, J. Makhoul, R. Schwartz, and G. Chou. On-line cursive handwriting recognition using speech recognition methods. In *ICASSP*, 1994.

[11] S. Young. *HTK: Hidden Markov Model Toolkit V1.5.* Cambridge Univ. Eng. Dept. Speech Group and Entropic Research Lab. Inc., Washington DC, Dec. 1993.

[12] R.E. Kalman and R.S. Bucy. New results in linear filtering and prediction theory. In *Transaction ASME (Journal of basic engineering)*, 83D, 95-108, 1961.

[13] A.S. Willsky. Detection of Abrupt Changes in Dynamic Systems. In M. Basseville and A. Benveniste, (Ed.). *Detection of Abrupt Changes in Signals and Dynamical Systems*, Lecture Notes in Control and Information Sciences, No. 77, pp. 27-49, Springer-Verlag, 1986.

6.1 Dynamic car model

The kinematic model is a three-wheeled model with two front wheels and one rear wheel. Given access to a system clock, we can figure out the amount of time elapsed since the last frame, Δt. To calculate the distance traveled since the last frame, we calculate

$$d_{lin} = v\Delta t$$

where the function d_{lin} is the linear distance traveled since the last frame and v is the speed of the car. To calculate the new position of the car, we must consider whether the car is turning or moving straight ahead. If moving ahead, then

$$\vec{P}_{new} = \vec{P} + (d_{lin}\vec{H})$$

If the car is making a turn, then

$$\vec{P}_{new} = \vec{P} + (r_{circle}\sin(\theta_d)\vec{H})$$

where the angular travel of the car θ_d is given by

$$\theta_d = d_{lin}/r_{circle}$$

and the turning circle of the car, r_{circle} is given by

$$r_{circle} = \frac{d_{WB}}{\tan\theta_{FW}}$$

d_{WB} is the wheelbase of the car.

The new heading of the car is figured from the following equation

$$\vec{H}_{new} = 0.5(\vec{V}_{LFW} + \vec{V}_{RFW}) - \vec{P}_{new}$$

where $\vec{V}_{LFW}$ is the vector position of the front left wheel and $\vec{V}_{RFW}$ is the vector position of the front right wheel. Finally, we normalize the new heading vector

$$\vec{H}_{new} = \frac{\vec{H}_{new}}{||\vec{H}_{new}||}$$

The dynamical equations are equally simple for the vehicle. The new velocity of the car is computed based on the energy produced by the engine balanced by the drag from air and rolling resistance and braking.

To calculate the new velocity, we first compute the kinetic energy based on the present velocity using

$$KE_{old} = \frac{1}{2}Mv^2$$

where M is the mass of the car. Then we calculate the work done by the engine to propel the car forward which is simply given by

$$W_{engine} = aP_{maxeng}\Delta t$$

where P_{maxeng} is the maximum power output of the engine and a is the position of the accelerator ($0 \rightarrow 1$). The limitation of this model is that the work is independent of the rpm. The work done by the brakes to slow the car is given by

$$W_{brake} = F_{maxbrake}d_{lin}(b + 0.6\, b_{park})$$

where $F_{maxbrake}$ is the maximum braking force of the brakes, b is the postion of the brake pedal ($0 \rightarrow 1$) and b_{park} is the parking brake which is either 0 or 1.

The contribution of potential energy, if the world is not flat, can be expressed as follows:

$$\Delta PE = (P^y_{new} - P^y)Mg$$

where P^y is the current y-coordinate and P^y_{new} is the newly computed y-coordinate. The acceleration of gravity, g, is given in m/s^2.

To calcuate the effect of drag, we first find the total force resulting from the air drag and road friction.

$$F_{drag} = \mu_{air}v^2 + \mu_{road}Mg$$

where μ_{road} is the coefficient of friction of the road and μ_{air} is the drag coefficient of the car. The total deceleration energy is just the sum of the drag energy and the brake energy.

$$E_{drag} = d_{lin}F_{drag} + W_{brake}$$

So our final energy balance equation looks like

$$KE_{new} = KE_{old} + W_{engine} - \Delta PE - E_{drag}$$

$$v_{new} = \sqrt{\frac{2KE_{new}}{M}}$$

The skid state of the car depends on the speed and steering wheel angle and road surface coefficient of friction. In our experiments we used constants that apply to a Nissan 240SX and a specified road/tire configuration. In this model, if the speed is < 11.2 mph then the car always regains traction. Basically, if the following condition is satisfied

$$\frac{\theta_{SW}}{e^{-0.0699v}} < 24.51$$

then the tires are within their adhesion limits and the car will not skid.