SANDIA REPORT

SAND2000-8206
Unlimited Release
Printed November 1999

Optimization of Automatic Train Control For Energy Management and Service Reliability

S. P. Gordon, R. M. Wheeler, T. J. Sa, D. A. Sheaffer

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Prices available from (703) 605-6000
Web site: http://www.ntis.gov/ordering.htm

Available to the public from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A06
Microfiche copy: A01

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

SAND2000-8206
Unlimited Release
Printed November 1999

Optimization of Automatic Train Control For Energy Management and Service Reliability

Susanna P. Gordon and Richard M. Wheeler
Systems Research Department

Timothy J. Sa
Systems Studies Department

Donald A. Sheaffer
Exploratory Systems Technology Department

Sandia National Laboratories
P.O. Box 969
Livermore, CA 94551-0969

ABSTRACT

A new generation of automatic train control systems is currently under development in the commuter-rail transit industry. These systems will utilize radio communication between wayside control computers and trains in order to provide high precision train control beyond the capability of today's automatic systems. The Bay Area Rapid Transit (BART) system is developing such a modern control system in collaboration with Harmon Industries. This system, called the Advanced Automatic Train Control (AATC) system, will allow for precision train locating and control, and will facilitate coordination of the trajectories of multiple trains. This system will be capable of running trains more closely together and decreasing the time a train requires to traverse the system, while simultaneously operating with a more modest traction power infrastructure, and providing a smoother, more comfortable ride to commuters. We have collaborated with BART to develop a simulator of the AATC system and the traction power system, and we have utilized this simulator as a testbed for the development of advanced train control techniques. Several train control algorithms, including one employing a neural network for train voltage prediction, have been developed and tested in the simulator. Smoother train trajectories, reduced power infrastructure requirements, and reduced energy consumption have been demonstrated. Improved service reliability is also expected to result.

ACKNOWLEDGMENTS

The authors are indebted to the staff of the BART Research and Development Department for their collaboration and support in the successful pursuit of a more efficient, reliable, and comfortable transportation system. In particular, we would like to thank Eugene Nishinaga and John Evans for many useful discussions, Marcus Thint for irreplaceable help in tracking down obscure BART information, and David Lehrer for the development of the Train Control Simulator, without which none of this work would have been possible.

We would also like to thank Professor K. S. Narendra of Yale University for providing invaluable insight into the role of artificial intelligence in practical applications, and specifically as applied to BART train control. In addition, we greatly appreciate the contribution of his student Chen Xiang, who conducted critical early experiments in neural network prediction of train voltages, putting us on the road to success.

This work was conducted under the BART/Sandia Cooperative Research and Development Agreement SC94/01316.

CONTENTS

Figures

Tables

OPTIMIZATION OF AUTOMATIC TRAIN CONTROL FOR ENERGY MANAGEMENT AND SERVICE RELIABILITY

Executive Summary

The San Francisco Bay Area Rapid Transit (BART) system is in the process of developing the Advanced Automatic Train Control (AATC) system, which will not only increase the capacity of the BART system and reduce trip time, but will also allow for the design of sophisticated control techniques to improve system efficiency, reliability, and passenger comfort. BART is collaborating with Harmon Industries to develop and implement the AATC system, employing radio ranging technology developed by Hughes Aircraft Company (now Raytheon) in order to both accurately determine train locations and to reliably communicate train control commands. Wayside computers will determine desired train trajectories and will command train speeds and accelerations through the radio network. Trains will incorporate only simplistic on-board logic to command motor propulsion in response to the wayside commands. This control system will allow trains to run closer together than the present track-circuit-based automatic system, thus increasing system capacity, and will also increase average train speeds, thereby reducing the time required to traverse a given route. Moreover, the system's wayside intelligence architecture will facilitate control techniques that coordinate the trajectories of multiple trains in an area for both control- and power-related objectives.

We have collaborated with the BART Research and Development Department to demonstrate targeted enhancements to the AATC system, in order to reduce traction power infrastructure requirements, reduce energy consumption, improve service reliability, and create smooth train trajectories for a more comfortable ride. We have integrated BART's simulator of the AATC system with our traction power model. We have also added the capability to incorporate enhanced "non-vital" control algorithms, and to measure their relative merits against a small set of important metrics, such as minimum train voltage and energy usage. The resulting AATC Train and Power Simulator (ATAPS) has provided a critical testbed for the development and assessment of enhanced control techniques.

Demonstrated control enhancements include an algorithm to avoid low train voltages, utilizing neural network technology to predict low voltages before they occur. This technique exploits AATC capabilities to reduce traction power infrastructure requirements, which are largely driven by the need to prevent low voltages, potentially deferring tens of millions of dollars in capital costs. Several algorithms were also demonstrated which smooth train trajectories during crowded, or "interfered," conditions. These include algorithms to avoid unnecessary brake/acceleration cycles immediately before station stops, during acceleration and speed maintaining close behind another train, and following substantial delays that can cause backups. These algorithms generate smooth train trajectories, making for a more comfortable ride, and are expected to improve train motor reliability by avoiding unnecessary mode-changes between propulsion and regenerative braking. Algorithms briefly investigated but not implemented in the ATAPS simulator include coasting, coordinated starts and stops for improved energy efficiency, and reduced peak substation power.

Acronyms, Abbreviations, and Other Terms

ATAPS	AATC Train and Power Simulator – a simulator of the AATC control system and the traction power system
ATC	Automatic Train Control – The current track-circuit-based train control system currently in use on the BART system.
AATC	Advanced Automatic Train Control – The new radio-communication-based train control system under development by BART and Harmon Industries
BART	Bay Area Rapid Transit – The commuter rail service in the San Francisco Bay Area.
CBTC	Communication-Based Train Control – The next generation of automatic train control systems using radios for train-to-wayside communication.
EPLRS	Enhanced Position Location Reporting System – a radio ranging technology originally developed by Hughes Aircraft Company (now Raytheon)
jerk	the rate of change (derivative) of acceleration
MJ	megajoules – a measure of train or substation energy (=MW * seconds)
MODRAILS	Model of DC RAIL Systems – the traction power model that is currently incorporated into the ATAPS system simulator
mph	miles per hour – a measure of train speed
mph/s	miles per hour per second – a measure of train acceleration
MW	megawatts – a measure of train or substation power
PL	Performance Level – a simple control parameter that limits train speeds and acceleration rates to pre-tabulated values
TCS	Train Control Simulator – the simulator of the ATC and AATC systems that is the core of the ATAPS simulator
transbay tunnel	The BART tunnel, or "tube," under the San Francisco Bay between Oakland and downtown San Francisco
WCSD	Worst Case Stopping Distance – the distance a train will take to stop assuming the most conservative stopping trajectory
Wye	The area on the BART system in Oakland where multiple lines merge

Introduction

The Bay Area Rapid Transit (BART) system services commuters in the San Francisco Bay Area, utilizing heavy rail commuter trains primarily to move people between the East Bay communities and the San Francisco peninsula. As BART expands its service area with extensions to the lines in the East Bay, increased capacity is required to funnel all of this additional traffic through the tunnel under the San Francisco Bay and through the downtown. Short of a massive engineering project to add additional track infrastructure in parallel with the current line, trains must be scheduled closer together to meet the expected capacity requirements with the existing infrastructure. The current Automatic Train Control (ATC) system is reaching the limits of its capabilities; in order to increase the capacity of the system much further, a new control system is required. BART, in collaboration with Harmon Industries, is in the process of developing an Advanced Automatic Train Control (AATC) system[1] to replace BART's current ATC system. As was the case for ATC, the trains will be controlled by station computers at the wayside; however, spread-spectrum radios[2] rather than hard-wired circuitry in the tracks will be employed to determine train locations and reliably transfer control information. This new communication-based train control (CBTC) system will provide finer speed and acceleration control, as well as more precise train locating capabilities than the current system.

In addition to fulfilling the immediate need to increase capacity on the BART system, AATC will be able to take advantage of its fine control of train trajectories and its wayside control architecture in order to improve service, reliability, and passenger comfort, while reducing energy infrastructure and usage costs. AATC will decrease the time to travel the length of a line by increasing average train speeds. This will reduce the total number of train cars required on the system at any given time by 20 cars in 2002, saving $2 million in capital cost per car as well as operational and maintenance costs. AATC will also be capable of coordinating the motion of multiple trains, including trains traveling in opposite directions but sharing power resources. For example, when there is insufficient power available for more than one or two trains to accelerate in a track area, control of trains in that area can be coordinated to limit the number of trains simultaneously accelerating. This is possible because the control logic is resident in wayside computers, which calculate speed and acceleration commands for all trains within their control zones. A system with train control intelligence on-board the train would not allow for such coordinated control.

In order to develop coordinated control strategies, we have collaborated with BART to develop a simulator of the train control and power consumption of the AATC system. We have employed the simulator to develop enhanced train control algorithms to supplement the safety-critical vital control. These algorithms do not attempt to globally optimize the control system with respect to a cost function, but rather they modify the baseline vital control strategies to smooth out train trajectories, and to reduce energy consumption and power infrastructure requirements, through coordination of multiple trains. Several control algorithms were developed, including (1) delay recovery, which smoothly and efficiently controls trains approaching or stopped behind a delayed train, (2) interference management, which controls closely-following trains to avoid oscillatory brake/acceleration cycles, and (3) low voltage avoidance, which limits the power

consumption of multiple trains in an area to prevent low voltage problems. The latter two are particularly important in a CBTC control system, because trains can travel so closely together that interference and low voltages can become severe problems if ignored. Other control algorithms briefly investigated include (4) coasting, (5) coordinating train movement such as starts and stops at stations to enhance reuse of regenerated braking energy, (6) limiting needle peaks in power demand at substations to some specified level, and (7) power-limited acceleration. In two recent papers, we have briefly described these non-vital algorithms as enhancements to the AATC system.[3,4]

This report will first discuss issues related to the development of modern train control systems and important train control concepts, and will then discuss the specifics of the Advanced Automatic Train Control system under development at BART. The following section will describe the simulator that we have employed for algorithm development. Finally, the last section will discuss the enhancements to the AATC and will demonstrate the capabilities of the developed algorithms through simulated train control data.

Train Control Issues

The Future of Automatic Train Control

Many automatic train control systems currently in operation, such as in San Francisco and Washington, utilize circuitry connected directly to the running rails to locate and communicate with the trains. These circuits divide the length of track into "fixed blocks," which are typically hundreds to thousands of feet long. These systems are limited to determining whether or not a train is present in a given control block, but not where within the block the train is located. This leads to a large uncertainty in the train location determination, and requires trains to be separated by correspondingly large distances for safety. In addition, only a limited number of train speeds may be commanded due to a limited communication bandwidth. This causes trains to often travel more slowly than the safety-limited speed limit, since they must be commanded with the highest selectable speed command below the safe speed. The stopping profile for station stops is particularly subject to this limitation, with trains slowing in a "stair-step" braking profile described by periodic discrete drops in speed, with steps typically of 10 to 20 miles per hour.

Several transit contractors, including Harmon Industries and Alcatel Canada Inc., are currently attempting to develop the next generation of automatic train control systems – communication-based train control (CBTC) systems utilizing radios in place of track circuits. CBTC systems will determine the location of trains more accurately, and will define train control "blocks" based on the instantaneous locations of the trains, rather than based on fixed locations on the tracks. Thus, these systems are referred to as "moving-block" control systems. All of the competing automatic systems must attempt to satisfy the needs of the transit industry. For example, transit properties often express the need for establishing common standards, open architectures, and interoperability, so that transit systems in the future will not be tied to a particular supplier or contractor. Also near the top of the list of transit requirements is the need for a reliable system that can be implemented with no down time or reduction in service quality. Transit properties such as BART that are upgrading from fixed-block automatic control systems to new CBTC systems must be able to run the old and new systems together in what is called "mixed-mode" automatic operation during the transition period. Sensitivity to risk management at transit properties may prove to be an over-riding issue in the future.

The AATC system is unique among the developing systems in that the radios will be utilized to locate the trains as well as to communicate with them. Competing systems will instead locate the trains with separate technology, such as using a train's measured speed to project its location from markers at fixed locations along the tracks. Since the AATC system uses the same radio system for train locating and communication, the quantity of new hardware required for system installation is minimized. Due to its precision train control and wayside intelligence, AATC will also be capable of more optimal performance, providing the smoothest possible ride for its passengers while minimizing trip time and taking full advantage of the available power infrastructure.

BART System Layout

For reference, a diagram of the BART system is shown in Figure 1. The end-of-line cities are indicated, as well as the location of downtown San Francisco, the Oakland "Wye" where the lines share tracks, and the transbay tunnel that crosses under the bay from Oakland to San Francisco.

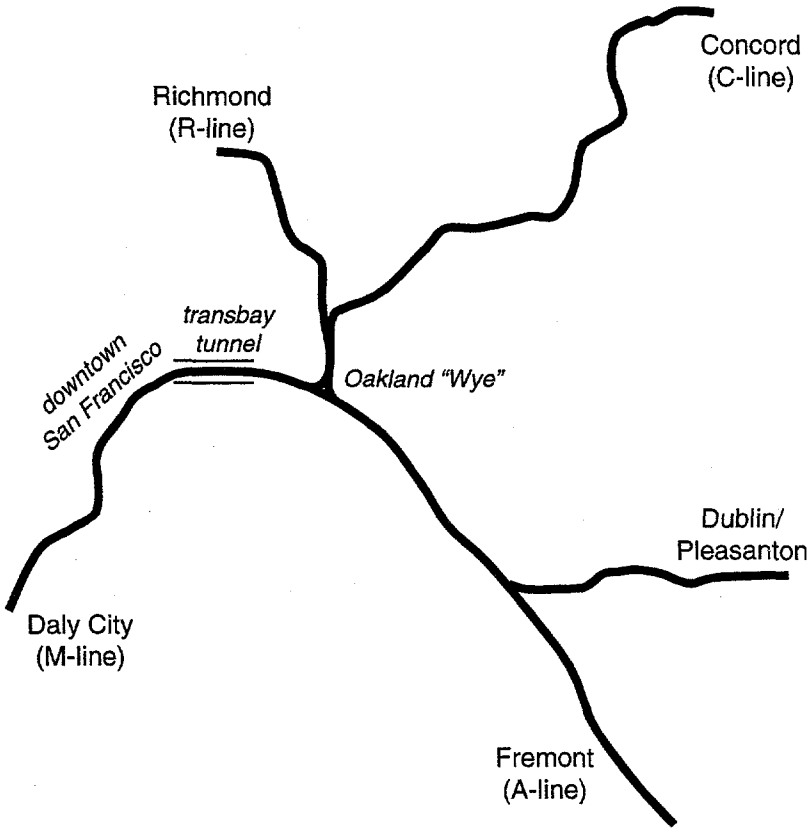

Figure 1. Sketch of the BART system lines and primary features

Trip Time and Its Consequences

Trip time is, as it sounds, a measure of the time it takes a train to traverse the system. AATC will reduce trip time as compared to the current ATC system in two ways. First, as mentioned above, the present "stair-step" braking profile will be replaced by a more efficient reduced-brake-rate profile. This will allow trains to continuously travel at the maximum safe speed while slowing for station stops, rather than at speeds well below this limit, and thus will save time during braking.[4] In addition, since AATC will allow speed commands every 1 mile per hour, trains will be able to travel at the civil speed limit governed by the track geometry, rather than at the next speed command below this limit. On the present system, for example, trains can be commanded to travel at 36mph or 50mph, but not in between. In a zone with a civil safe speed limit of 48mph, the present system would give a 36mph speed command in order to avoid

exceeding the safe speed. The new AATC system will be able to command the maximum 48mph speed.

One of the benefits of the AATC control system will be a reduction in the number of trains on the system at one time. This will be accomplished by reducing trip time from one end of the line to the other. When a train reaches the end of the line, it must be turned around and prepared to begin a run back up the line. If the train arrives at the end of the line with enough time to turn around before the next dispatch time, then it becomes the next scheduled train. If, on the other hand, there is not sufficient time, then another train must be available for the next run. Thus, a small reduction in trip time can potentially reduce the number of trains that BART must have operational at once. This is extremely important, because BART trains, which typically consist of 10 cars each, cost $2 million per car.

Delays can not be tolerated in the dispatch time at the end of the line, because the train lines on the BART system merge and share tracks in an area called the "Oakland Wye" in the middle of the system. A significant slip in schedule can result in a train missing its timeslot through this merge, thereby causing a significant schedule disruption. For the same reason, significant delays along the lines approaching the Wye can not be tolerated. Thus, the schedule through the Wye constrains the applicability of enhanced control algorithms which significantly increase trip time.

On today's system, a delay is recorded when a train reaches the end of the line more than 5 minutes late. In 12/93, there were 6000 car-hours of delay on the BART system. About 1/3 (100 events) were related to vehicle problems, 1/3 were due to control system problems, and 1/3 were from miscellaneous causes such as gunfire.

Since the AATC system is still under development, there is not yet any field experience to verify the trip time that has been calculated in simulations, nor has the schedule been created for the new system. Thus, it is not possible a priori to determine how much trip time can increase before another train is required on the system. Thus, we have investigated enhanced control techniques in the context of the tradeoff between trip time and the various energy-related metrics. When the AATC design is more mature, it may then be possible to quantify trip time as well as energy usage in terms of costs, and to minimize overall system costs.

Schedule Adherence

Adherence to the schedule is presently enforced primarily with adjustments to the dwell time at stations. It is also possible to speed up trains somewhat in areas with high speed limits by utilizing the top "Performance Level." However, Performance Level 1 (PL1), which increases the top train speed from 70 to 80mph, is not used during normal operations and is rarely used to catch up a train which is behind schedule, because it causes reliability problems with the train motors. Since trains are scheduled to utilize their top available speed at all times, the only flexibility in the schedule occurs during the time stopped in stations. Scheduled station dwell times are typically 20 seconds long, but can be as short as 10 seconds. In order to minimized dwell time, a delayed train is given a dispatch command as soon as its doors are open at the station. This allows the train's operator to close the doors as soon as they are clear. Unfortunately, this often does not reduce the effective dwell time, because delays often occur

during rush hour when many people enter and exit each train and a significant amount of time is required to clear the doors. Once a train gets behind schedule during rush hour, it is very difficult to catch up again. It is possible to catch up by skipping stops, but this solution is not acceptable to commuters.

Approximately once each day, a train is delayed enough to move back one time slot in the schedule (i.e. delay = headway). When this occurs, personnel at Central Control must manually change the train IDs so that each train on the line becomes the next train in the schedule, and the system once more appears on time.

Headway and Interference

Headway is defined as the time between consecutive trains passing a point on the system. As long as trains are sufficiently far apart, they do not affect each other's behavior. Each train moves at its top available speed at all times, and accordingly each train's trip time is at a minimum. However, when trains get close enough together, they begin to "interfere," braking occasionally to maintain a safe following distance. Below this headway, referred to as the Uninterfered Crush, the trip time increases as shown in Figure 2. The minimum sustainable headway is known as the Steady State Crush. Reducing the headway further results in significant backups, and each train takes longer to traverse the section of track than the last. In other words, there is no steady state trip time for headways below the Steady State Crush.

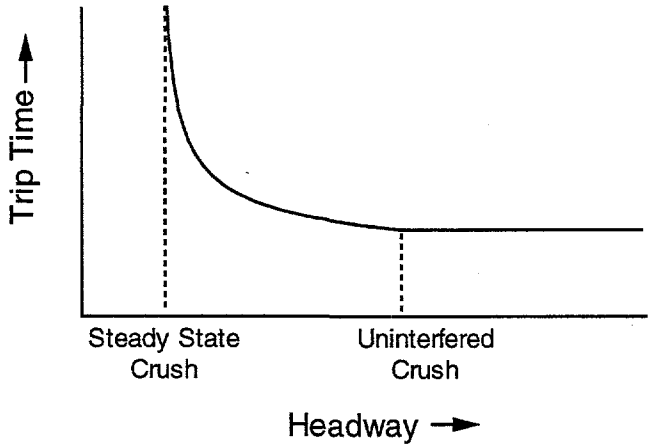

Figure 2. Sketch of the steady-state relationship between trip time and headway

On the present BART ATC-controlled system, the system Steady State Crush is 110 seconds. This is the minimum headway that can be achieved in steady state over an entire route. Locally, the Steady State Crush in some areas is somewhat smaller than this. The system Uninterfered Crush, above which trains operate without interacting at all, is 150 sec. When considering these figures, keep in mind that these headways are specific to the BART system layout, and are not intrinsic to the control system. For example, trains approach the San Francisco downtown area from Oakland at 70mph, which corresponds to a long stopping distance and requires a commensurately long following distance. If the speed limit where reduced, then the crush

headways would be smaller. This alternative has not been acceptable at BART, since reducing train speed would increase trip time, which is considered to be of primary importance. Similarly, headway must include enough time for station stops, so shorter dwell times at stations would allow for shorter headways. However, reducing dwell times at BART stations would also pose difficulties, and would probably not be possible during rush hour.

The new AATC control system will allow for significantly shorter headway operation, with the same system layout as the current system. Using the simulator discussed later in this report, BART has estimated that the system Steady State Crush will be about 95 sec, and the scheduled headway will be about 120 sec. A crush of 80 sec will be possible locally to recover from backups, but will not be sustainable or possible system-wide. As for the ATC system, it may be possible to further reduce headway by reducing speed limits. However, it is unclear how to evaluate the tradeoff between short trip time, which is important to passengers, and reduced headway for increased system capacity, which is also important but entails significant capital expenditures for additional trains.

Traction Power System Basics

Power is generated and fed into the high voltage third rail by both power substations and regeneratively braking trains. A regenerating train is effectively running its motors in reverse, slowing down by producing power rather than accelerating by using power. Power flowing from substations and trains must flow through the powered third rail in the form of a DC current to reach a power-consuming train. In the process, power is lost to heat in the rail; the more distance the power must travel, the more is lost. In addition, when power must be conducted over long distances, the voltage on the third rail drops between the trains, causing the voltage at the power-generating train to rise and the voltage at the power-consuming train to fall. If this voltage difference becomes too large, the power-generating train will begin to intentionally burn energy in on-board resistors in order to prevent its third rail voltage from rising too high, and a substation closer to the power-consuming train may provide the required power instead. Also, the voltage at the power-receiving train may drop to the point of effecting the performance of the train's motors. Thus, to efficiently use regenerated energy, a braking train must be relatively close to an accelerating train ($\leq$~1mile via the third rail). These two trains may be on the same line, or traveling in opposite directions on opposing lines.

Substation third rail voltages are currently produced by rectifiers, which do not provide for active voltage control. Voltage "regulation" of these rectifiers typically allows the substation voltage to drop by 6% at 100% of the rated power load. For example, a 5MW substation producing the nominal 1050V no-load voltage will drop to about 990V when providing the rated 5MW. These substations are capable of operating for short periods at several times their rated loads, so the substation voltage can potentially drop briefly by hundreds of volts. Substations are rated to provide 150% of their rated load for 2 hours; 300% for 1 minute, 5 times equally spaced over 2 hours; and 450% for 15 seconds, once within 2 hours. BART may replace some substations with cryogenically cooled equipment to improve voltage regulation. Alternatively, some substations may be equipped with thyristors, which will allow for active voltage compensation to prevent voltage sags. Such controllable substations will not be considered in this study; only train

control will be discussed. However, the ultimate intelligent control system could simultaneously optimize train and substation control to minimize energy usage.

Low train voltages are typically caused by multiple trains accelerating simultaneously, since acceleration uses significantly more power than maintaining speed. For example, train cars consume up to 800kW while accelerating, but only a few hundred kW to maintain speed, depending on the speed and grade. Conversely, a regeneratively braking train produces roughly 600kW per car, and injects this power back onto the third rail if another train is nearby to use it.

Train power and voltage are not currently measured by BART, but the new line of "C"-cars will have the capability of being outfitted to measure and transmit this data in the AATC system. Future system architectures could incorporate this capability, and train control algorithms could use measured power-related data for better system optimization.

AATC System Architecture

"Baseline" AATC System

The Advanced Automatic Train Control system under development at BART promises to achieve shorter headways than the current fixed-block automatic control system by utilizing a radio communication system to both command and accurately locate trains, and by defining moving control blocks based upon the train locations. Following distances will be minimized, constrained only by safety requirements, while the large position uncertainty of a fixed block system will be removed.

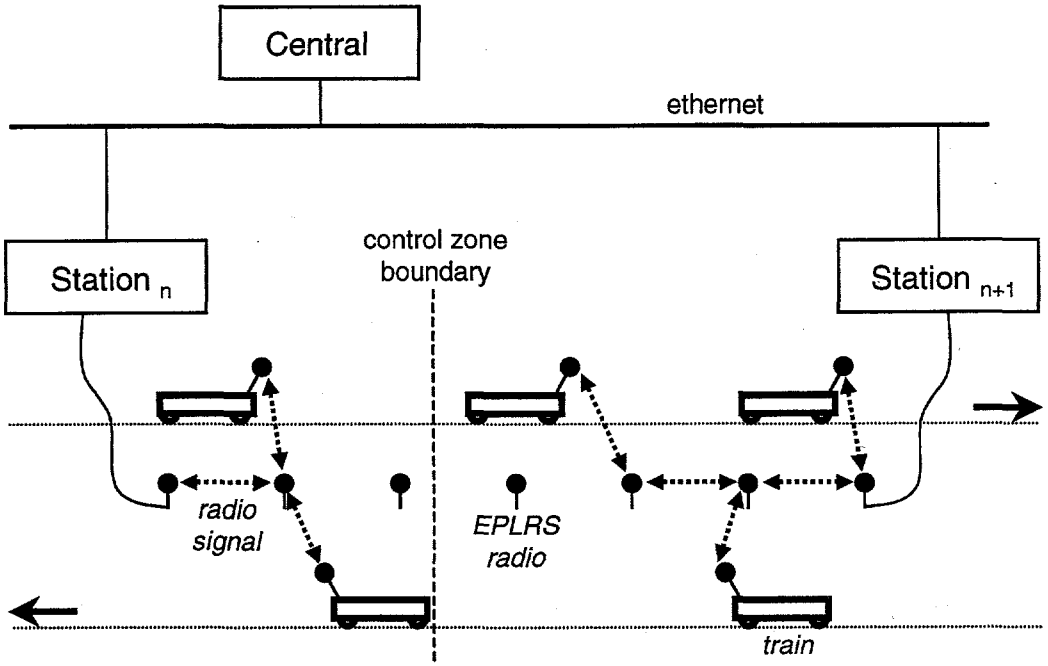

Figure 3. Sketch of the AATC communication system architecture

The AATC system is expected to provide this capability through more precise train locating and control.[1] The system will employ the Enhanced Position Location Reporting System (EPLRS), a spread-spectrum radio ranging technology originally developed by Hughes Aircraft Company (presently Raytheon) and capable of simultaneous train tracking and communication.[2] EPLRS provides high resolution position determination and reliable transfer of command information both inside and outside of tunnels. The propagation time of the radio messages to and from the trains will be used to determine the train locations to within ±15 feet. Trains will communicate from on-board radios through a network of wayside radios to station computers, which will control the trains in local areas through speed and acceleration commands updated every half-second. The radios will carry messages, including speed commands resolved to every one mile-

per-hour, as well as variable acceleration commands. At the moment, additional station-stop equipment is needed to meet the BART requirement of station stops to within ±3 feet. The architecture of the AATC system is shown in Figure 3.

Central control will communicate with station computers through an Ethernet connection. Central primarily controls the Performance Level (PL) of trains, their dispatch times from stations, and their destinations (i.e. their paths through the Oakland Wye). PL's allow for predefined reductions in speed limits and acceleration rates. Adjustment of train trajectories through PL commands is extremely limited. Speed and acceleration commands for trains are calculated locally at the wayside by the station controllers. Each station computer controls several miles of track and can control up to 20 trains on either side of the station. There are 17 station controllers between Fremont and Daly City, a complete train route. Radio communication between stations and trains enables a moving block control system in place of the current fixed block system. In other words, it will be possible to adjust speed commands to trains at arbitrary times rather than only when they pass the interfaces between track circuits, which nominally occur every 700 feet (range from 200 to 1300 feet) on the present ATC system.

As shown in Figure 4, there are two important control loops in the AATC system. The loop for station-to-train communication takes between 0.25 and 0.5 seconds. In this command loop, the train sends its speed and acceleration, as measured on board, through the EPLRS radio network to the station controller. The station computer uses the time of arrival (TOA) of the train's message at the wayside radio to determine the train's location, and computes speed and acceleration commands for this train within 33.3 milliseconds. These commands are then sent back to the train via the radios.

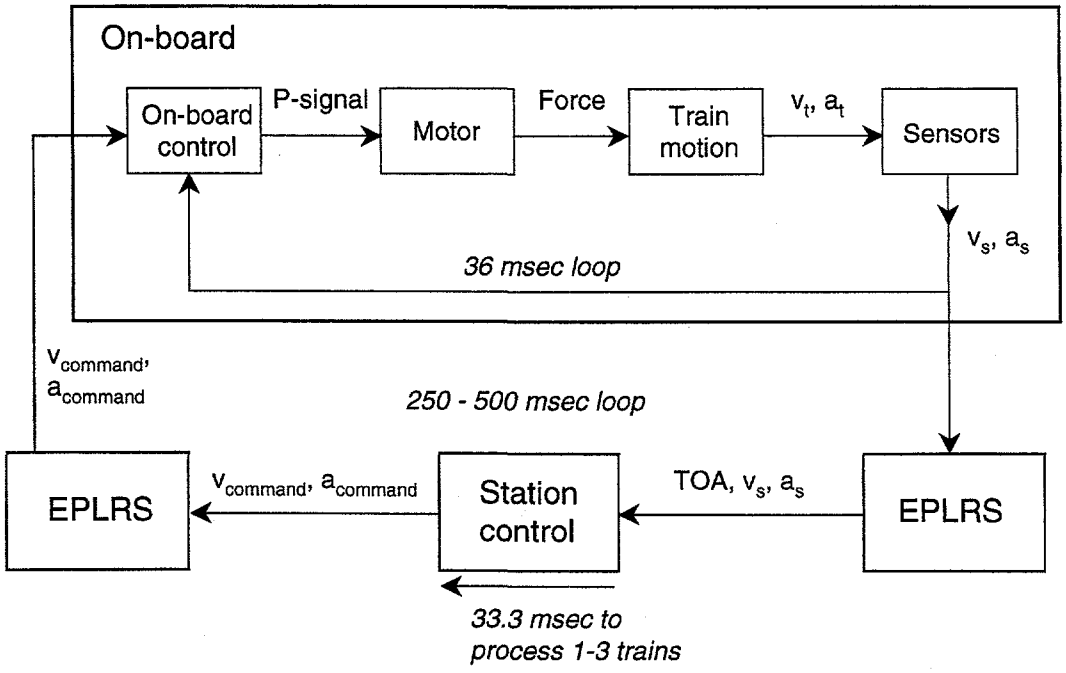

Figure 4. Diagram of the station-to-train and on-board control loops

The second, faster control loop is the on-board feedback control loop. The on-board controller sends a propulsion signal (P-signal) to the motors. The motors produce some force on the train, which affects its velocity and acceleration. These are then measured and compared to the velocity and acceleration commands, and a new P-signal is calculated. Comparing the train's measured speed with the speed command, the controller determines whether the train is in accelerating, speed maintaining, or braking mode. If the train is in an accelerating or braking mode, then feedback is used to match the acceleration command. If it is speed maintaining, then feedback is used to attempt to match the speed command. This loop is repeated every 36 msec.

"Enhanced" AATC System

The commands which will be sent to the trains from the wayside will be generated by a combination of two computer processors – a vital and a non-vital processor. The vital processor will be responsible for implementing "baseline" control algorithms and generating safety-critical train speed and acceleration commands, and for communicating commands to the trains. The non-vital processor, working in parallel, will add enhancements to the baseline vital control in order to meet non-safety-critical objectives such as reliable service and reduced energy usage. This processor will receive information about the states of trains in its control zone from the vital processor, as well as information about trains in neighboring zones through an Ethernet connection. When a change to the baseline train behavior is desired, the non-vital processor will pass suggested train control commands to the vital processor. If the suggestions are more conservative than the vital commands, then they will be sent to the trains in place of the vital commands. That is, the smaller speed command and the more negative acceleration command of the vital and non-vital commands will be sent to the trains.

A diagram of the enhanced AATC control architecture is shown in Figure 5. Information critical for enhanced algorithm implementation is shown in bold. The information in italics will not be available in early implementations of the enhanced control system. The "RF Data Bus" refers to the EPLRS radio system which communicates with the trains.

The vital and the non-vital processors will communicate once every control cycle (0.5 seconds). In the current architecture, it will not be possible for the non-vital processor to modify the vital commands after they are calculated but before they are sent out to the trains. Instead, it will need to project ahead to make suggestions for implementation in a future command cycle. Each command cycle, the vital computer will compare its calculated commands to those suggested by the non-vital processor in the last cycle. In addition, the non-vital processor will compute its suggested commands based on train state information it received from the vital processor in the last cycle. Thus, the non-vital computer must always suggest commands assuming they will not be implemented for three command cycles, or approximately 1.5 seconds.

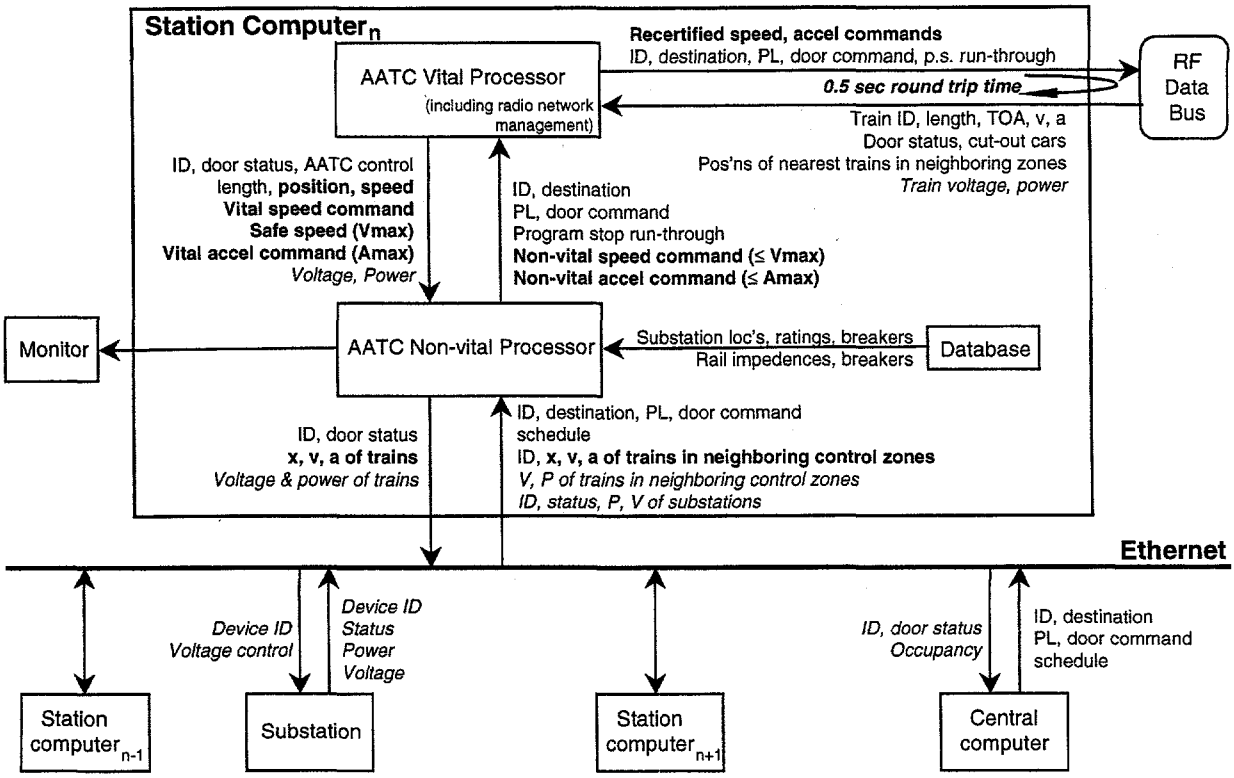

Figure 5. Detailed sketch of the station computer communication network

AATC Commands and Stopping Distance

There are two commands used to control train motion: a speed command and an acceleration command. The speed command serves as the primary control command, and the acceleration command plays a secondary role. The trains have three distinct modes of operation: acceleration, braking, and speed maintaining. If the speed command is 10mph greater than the current train speed, as measured on-board, then the train accelerates at the command acceleration. If the speed command is 0 to 10mph greater than the current speed, then the train uses on-board feedback control to attempt to maintain speed at 2mph below the speed command. The maximum available acceleration while in this speed-maintaining mode is governed by the acceleration command. Finally, if the speed command is less than the current speed, then the train brakes at the brake rate given by the acceleration command.

Trains nominally accelerate at 3 miles-per-hour-per-second (3mph/s), or at half of this rate at some reduced Performance Levels. Regenerative braking ("closed loop" braking or "Brake 3 high") under the new control system will likely be at less than 1mph/s. Friction braking ("open loop" braking or "Brake 3 low") is at 3mph/s. Open loop braking is used to initiate braking; to complete stops, because regenerative braking becomes ineffective at low speeds; and in emergencies, such as a loss of communication with the wayside computer. A minimum brake rate of 0.4mph/s is also enforced by the on-board control.

The on-board train control is unaware of the track grade, and measures train acceleration with an on-board accelerometer, which is affected by grades. For example, the accelerometer on a train that is stationary on an uphill grade will measure an apparent acceleration forward. Each 1% grade corresponds to an apparent acceleration of 0.2mph/s. Trains accelerate more slowly than the acceleration command on uphills and more rapidly on downhills, due to this systematic accelerometer error.

A train begins braking when an obstruction, such as another train or an open gate, appears within its "worst case stopping distance" (WCSD), which is shown in Figure 6. The braking profile shown assumes a worst case stopping scenario, which is initiated when communication between the wayside station controller and the train is lost. The initial speed is assumed to be the command speed. After 2 seconds without communication from the wayside, the train attempts to stop as quickly as possible. The worst case braking trajectory then begins with the train in full acceleration. The train must jerk limit to no acceleration, meaning that the acceleration is reduced to zero at a rate limited by the maximum allowable jerk. The train next experiences a mode change, while the motors are reconfigured from propulsion mode to braking mode, and then jerk limits into braking. The initial few seconds of braking are assumed to have only 60% of the cars applying brakes, followed by 80% of the cars. The available adhesion for a stop is assumed to be sufficient for a maximum brake rate of 1.5 or 2 miles-per-hour-per-second, depending on whether the track is covered or uncovered. These low brake rates correspond to an assumption that the track is wet, with somewhat better adhesion assumed on covered track.

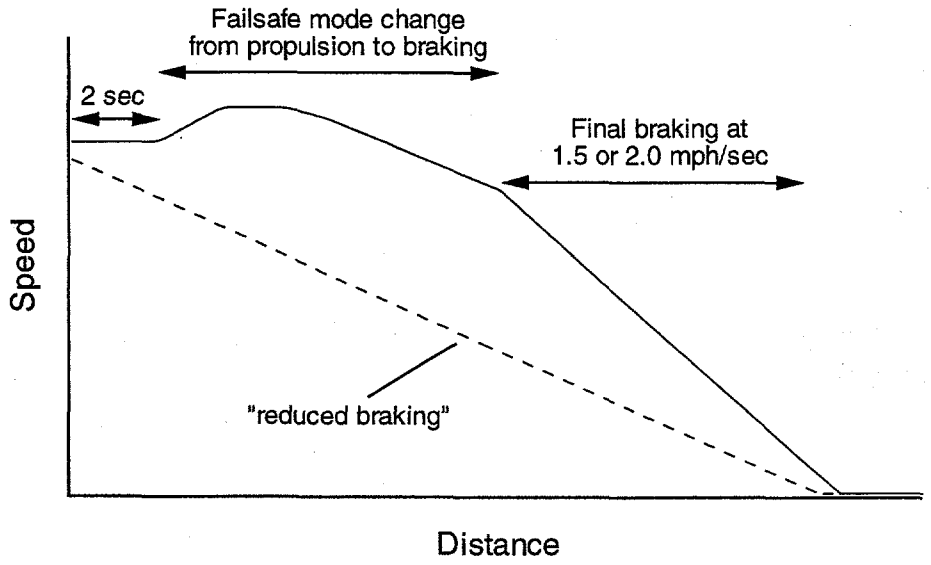

Figure 6. Sketch of the train braking trajectory in the worst case stopping model

When an obstruction appears within the calculated WCSD, a "reduced brake rate" is calculated which will stop the train behind the obstruction. The reduced brake rate command is shown in the figure as a dashed line. Each command cycle, this brake rate command is recalculated based on the new train state and the current location of the obstruction. If that obstruction is a train which is moving away, then the brake rate command will be recalculated to stop behind the new location of that train. It is always assumed that a train must stop behind the current location of

the train in front of it, ignoring the fact that that train may be moving away. This is known as the "brick wall" assumption, because all obstructions, moving or not, are treated like stationary brick walls.

As discussed above, after the mode change to braking, the worst case brake rate is 2.0mph/s on covered track, and 1.5mph/s on exposed track. The commanded reduced brake rate is not allowed to be more than 80% of these values. If the required reduced brake rate ever exceeds 1.6mph/s on covered track or 1.2mph/s on exposed track, then a potentially unsafe condition exists and Brake 3 drops, stopping the train at full deceleration.

While a train is braking, it is continually checking whether an obstruction still exists in its Worst Case Stopping Distance to determine whether it is necessary to continue braking. For example, a gate in its path may switch so that it is no longer an obstruction. The WCSD is first checked for a speed command 10mph above the current train speed, if this is below the track speed limit. If no obstructions exist in this distance, then this speed is commanded, and the train will switch back to propulsion and begin accelerating. Otherwise, the process is repeated for 6mph and then 2mph above the current speed. If none of these are safe, then the train continues on its reduced braking profile.

On-board vs. Wayside Intelligence for Enhanced Control

As the new moving block control systems are developing, one recurring debate involves the location of the intelligence for commanding train trajectories. In the AATC system design, the train control speed and acceleration commands are calculated by wayside station computers. An alternative strategy under consideration by some systems places the intelligence on-board the train. These system designs limit the control from the wayside to determining limits of authority. In other words, a wayside computer keeps track of the locations of trains and states of interlocks, and communicates how far a train may go safely. The train then determines its own trajectory, constrained by the authorized distance it is allowed to travel.

Although on-board intelligence will determine a train's trajectory, it may be possible for the wayside control to implement non-vital optimization algorithms through simple Performance Level (PL) commands. PL's on the BART ATC system comprise a few distinct modes of operation, characterized by reduced speeds and acceleration rates. Similar PL's could be developed to limit energy or power usage, for example, by requesting a pre-specified power-limited acceleration rate. It is unclear whether commands more complex than PL's could be transmitted from the wayside and interpreted by the train.

We believe algorithms that depend only on one or two train states could potentially be implemented on-board, but not algorithms requiring coordination of multiple trains. For example, algorithms to avoid acceleration/brake cycles due to interference could potentially be handled by on-board intelligence, as long as each train is aware of the location and velocity of the train in front of it and the track grades (i.e. hills) between them. Unfortunately, this information would not be sufficient for a train to avoid attempting to draw more power than is

available. Power management through on-board speed command control would require the wayside to inform each train of the locations and power demands of all trains in its vicinity. Transmission of this quantity of information may be bandwidth-limited by the communication system. In addition, the on-board non-vital control algorithms would have to contain a set of rules by which trains could unambiguously determine train prioritization, analogous to the rules used by automobile drivers when they meet at four-way stops. Alternatively, the motors themselves could be modified to reduce power demand automatically as voltage drops so as to prevent severe low voltages. The disadvantage of this approach is that train power demand can not be allocated according to train priority, for example providing full power to a train leading a backup and slowing the following trains. Also, care would have to be taken to make the motor feedback on voltage stable, so that oscillations in acceleration would not result.

The station controller, in contrast, is already aware of the trajectories of all trains in its zone, so wayside intelligence at the station can coordinate train motion without adding further burdens to the radio communication system. Neighboring zone controllers will have to communicate with each other, since power utilization is not constrained by control zone boundaries, but this non-vital information can be transmitted via an Ethernet connection between stations. Again, a set of rules will have to be developed to allow unambiguous prioritization of trains between zone controllers. In the on-board train intelligence architecture, if a wayside intelligence decides how to optimally coordinate the trains in its zone, it is unclear what type of commands it can transmit to the trains, since PL's would probably not provide fine enough control.

On the other hand, some of the desired non-vital algorithms potentially could be implemented as simple PL-like commands from the wayside controller. For example, reduced acceleration or brake rates, reduced speed limits, power-limited acceleration, and coasting could be requested this way. The wayside computer would require sufficient information – including the locations, speeds, and accelerations of all trains in its zone and neighboring zones – processor capability, and intelligence to recognize when and where to apply these PL's. Some flexibility may be lost as compared to a system where the wayside commands train speed and acceleration, since PL's typically represent discrete operating modes. For example, discrete power-limited acceleration modes might be predefined with corresponding PL codes. To add versatility, the PL command could be expanded to contain a mode and an additional parameter. For example, a PL command could contain a "mode" corresponding to power-limited acceleration, and a "mode parameter" containing the power limit to be enforced in kW/car. Even this expanded PL command may not achieve the flexibility of speed and acceleration command generation at the wayside, which would allow for any arbitrary speed profile.

We would argue that many non-vital control algorithms would be difficult if not impossible to implement with an architecture utilizing on-board intelligence to generate speed and acceleration commands based only on system speed limits, obstructions, and PL commands from the wayside. The algorithms that *can* be implemented may be less flexible and more problematic with divided intelligence. Algorithms implemented on-board the train would require more information from the wayside than is available in the baseline system. In addition, algorithms implemented at the wayside would require the wayside computer to send commands to the trains that attempt to guide the on-board controllers to do the right thing. Complications are more likely to arise from using this indirect approach than from using the direct method of allowing the wayside intelligence to control the trains directly, as in the AATC system architecture.

Intentionally Left Blank

AATC Train And Power Simulator (ATAPS)

In order to test and refine the AATC system before implementation, the Train Control Simulator (TCS) was developed at BART to accurately simulate the motion of trains. The TCS then became the central part of a larger AATC Train and Power Simulator (ATAPS), which supplemented the safety-critical control system in the TCS with a non-vital controller, allowing for the addition of enhanced control algorithms for coordination of multiple trains. ATAPS also contains a system traction power model, which allows analysis of power-related issues, such as power consumption and low voltages. In its current version, the trains are modeled in both directions of travel over the entire length of a single line, whereas the power model extends over a subsection of that line.

As the simulator runs, it generates output files which record detailed train trajectory and power-related data as functions of time. These data allow for a detailed analysis of the impacts of various control algorithms. In addition, the Benefits Assessment Module evaluates a small set of overall power- and control-related metrics for each substation and each train, such as the number of low voltage events and the number of motor mode changes from propulsion to braking. The user can use this information to easily assess the overall impact of an algorithm without looking at the detailed output data. We believe the ATAPS simulator provides a powerful test-bed for development and testing of novel train control algorithms in the context of both train motion and power consumption.

Simulator Functions and Interfaces

Train Control Simulator (TCS) [3,4,5]

Development of the Train Control Simulator (TCS) began at BART in the early nineties to explore interest in a moving block train control system. The first versions of the simulator were rudimentary – designed specifically to evaluate the advantages of a moving block control system compared to a fixed block system in terms of headway. Eventually the data provided by these simulations led BART to go forward with the AATC project.

To date, BART has validated the TCS against system-wide revenue service data collected by BART's Central Control System computer, and against data collected during the first phase of AATC system testing. Both have indicated that the simulator accurately models the control and movement of BART trains. With the aid of this simulation, BART has further optimized its existing vehicle control system, which had gone unmodified for nearly a decade. The simulator has developed into a key technique in the validation of such modifications, allowing for more exhaustive testing than might otherwise be possible.

The BART TCS is capable of simulating conventional track circuit-based control systems, as well as moving block systems, or a mixture of both. Multiple trains may be simulated

simultaneously, each with its own configuration; length, weight, performance profile, and speed selection algorithm are all user definable. Output reports may be generated which log events, such as station departure times and trip times.

Many potential problems have been revealed using the simulator, some far sooner than they might otherwise have been discovered. For example, the simulator revealed that a cyclical behavior could result if two trains that had stopped close together attempted to accelerate up to speed. During acceleration, the following train would be forced to repeatedly stop accelerating, and sometimes begin braking, in order to maintain a safe distance from the train ahead. This phenomenon can lead to an extremely rough ride, as well as spikes in power demand that can lead to unacceptably low train voltages. To avoid this behavior, the following train must be commanded either to wait before beginning to accelerate, or to accelerate at a lower rate. We have developed a method for calculating an optimal reduced acceleration rate for the following train, which will be discussed later in "Interference During Acceleration."

Traction Power Simulator (MODRAILS)

The traction power model, MODRAILS (Model of DC Rail Systems), was developed at Sandia to simulate the BART electrical system on a single line of track including both directions of travel. Although the model could be run on any line, it currently models a zone on the M-line bounded by the Oakland West (KOW) and 24th Street San Francisco (MTF) power substations. This zone includes the track section that travels through the tunnel under the San Francisco Bay, and through several closely-spaced passenger stations in downtown San Francisco. Thus, the model allows for analysis of train behaviors in areas with long distances between stations, as well as in areas with frequent stations stops.

Table 1 contains the absolute locations (in thousands of feet) of important power-related objects defined in the MODRAILS function. This table is useful as a reference when creating an input file for a desired simulation run, and when trying to interpret train locations in the output data files. The locations were derived from the locations in the BART Track Plans, and then mapped to corresponding locations in the DF.SDF and FD.SDF simulator track database files. In the transbay tunnel and through San Francisco, where the model is currently configured the FD database corresponds to BART's M1 line, and DF to M2. Since these locations are hardwired into the model at present, no other track database files are supported at this time. Substations provide DC electrical current to the third (high voltage) rail that powers the trains. The third rails that power the two directions of travel are connected only at the gap breaker (MCG) in the middle of the transbay tunnel and at all substations except KTE and MTW in the tunnel. These connections allow power to flow between the third rails. The endpoints of the traction power zone that is currently modeled are defined by the KOW and MTF substations. Passenger stations are also listed for reference; these locations correspond to the locations where the heads of trains stop at stations. The locations listed on the DF and FD lines for the same station are displaced from each other by the 700 foot length of a train, since the heads of trains traveling in opposite directions stop at opposite ends of the station.

Table 1. Summary of important track features

Feature Type	Name	FD (M1) location /1000 feet	DF (M2) location /1000 feet
Station	*Oakland West [M10]*	*132.469*	*79.727*
Substation	**KOW**	**132.627**	**78.870**
Substation	**KTE**	**141.901**	**69.599**
Gap Breaker	**MCG**	**151.501**	**60.015**
Substation	**MTW**	**161.233**	**50.333**
Station	*Embarcadero [M16]*	*163.367*	*48.861*
Station	*Montgomery St [M20]*	*165.243*	*46.985*
Station	*Powell St [M30]*	*167.594*	*44.634*
Substation	**MPS**	**167.623**	**43.903**
Station	*Civic Center [M40]*	*170.257*	*41.971*
Substation	**MSS**	**175.426**	**36.133**
Station	*16th St Mission [M50]*	*176.157*	*36.105*
Substation	**MTF**	**180.123**	**31.437**
Station	*24th St Mission [M60]*	*180.827*	*31.435*

MODRAILS uses the location and power consumption or regeneration of each train in the above zone at a given moment to calculate the voltage of each train and the power being produced by each substation. The system solution is found in the steady state, so the model does not support studies of transients or instabilities in the power network. The primary utility of the model is in evaluating the severity of low voltage problems and the efficiency of regenerated traction power usage.

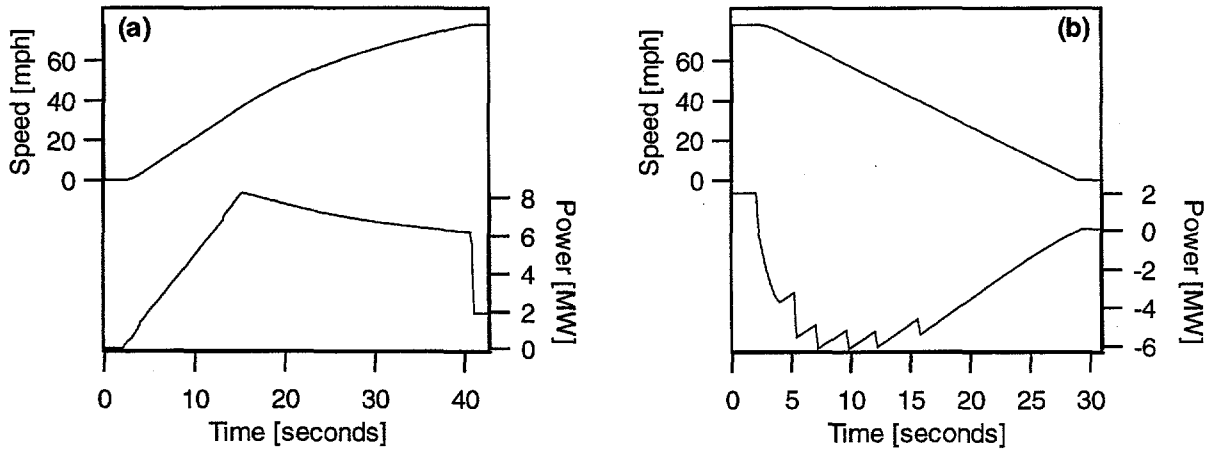

Figure 7. Calculated speed and power consumption of (a) an accelerating and (b) a decelerating train

The power demand of each train is first calculated by the Power and Maximum Acceleration Model (PMAM) within the TCS. This module calculates the power consumed or regenerated by a train, and the maximum possible acceleration which the motors can provide given the current

train state. This function not only impacts power calculations, but also limits train trajectories to physically realizable accelerations. Examples of calculated power consumption as functions of time for an accelerating and a braking train are shown in Figure 7. Negative power consumption corresponds to power regeneration by the braking train.

Given the calculated power consumption and location of each train on the line, MODRAILS translates all infrastructure and train information for a linear section of track into a DC electrical circuit. Incorporated into this model are substation, crossbond, and gap breaker locations. Crossbonds are connections between the running rails in the two directions of travel. This is relevant because the running rails serve as the low voltage power return path, and are grounded at the substations. Each train is treated as two separate power sinks (or sources) consuming (or regenerating) half of the total train power, one located at the head of the train, and the other at the tail. This avoids overestimating low voltages by realistically distributing the power load over several hundred feet of track. Train voltages are limited to a maximum of 1150V during regenerative braking. Power is allowed to flow out of substations onto the third rail, and from running rails into grounds, but not in the reverse directions. Rail resistance may vary in discrete sections, which allows accurate modeling of the presence of low resistance contact rail in some sections of track.

The set of equations solved by MODRAILS are predominantly linear equations describing current flow through resistors. However, the equations corresponding to trains are nonlinear, since trains are modeled as power sinks or sources. Substation voltage equations are also nonlinear, as they are functions of substation power. MODRAILS linearizes these equations, starting with initial voltage guesses, and then iterates the solution using the Newton method until the guesses converge. Convergence is simply defined by a solution with all train and substation voltages within a small fixed voltage (0.1V) of the guesses for that iteration. During the simulation, the solution for these voltages on a given time step is used as the initial guess for the subsequent time step. This method significantly reduces the number of iterations required for convergence as compared to starting with a nominal guess each time.

This model was originally developed in order to analyze the relative merits of an energy storage unit to prevent voltage sags in the transbay tunnel. It was then modified for the purposes of this project and integrated with the BART Train Control Simulator. For a more detailed description of MODRAILS, please refer to Chapter 3 and Appendix B of the Sandia report "Energy Storage and Alternatives to Improve Train Voltage on a Mass Transit System."[6] This report describes the original version of MODRAILS, but the current version is substantially the same.

Sample train- and substation-related output data produced by MODRAILS during a typical simulation are shown in the figures below. In Figure 8, train location, speed, command speed (dashed), acceleration, command acceleration (dashed), power consumption, contact rail and running rail voltages (Vc and Vr), and the differential voltage (Vc – Vr) are shown as functions of time. This train begins at a station, accelerates up to speed, and then decelerates for the next station stop. It repeats this trajectory through three subsequent stations, and then travels through the tunnel under the San Francisco Bay. In general, while the train is accelerating, power is consumed and the train's voltage drops. When the train is regeneratively braking, the voltage floats up to a maximum of 1150V. Additional trains in the area add complexity to the voltage solutions. The speed and acceleration commands shown do not match the trajectory during the

station-stop, because the final braking and stopping for stations is controlled on-board rather than by AATC commands from the wayside station computer. Ultimately, it is expected that the entire train trajectory including station stops will be commanded from the station computer.

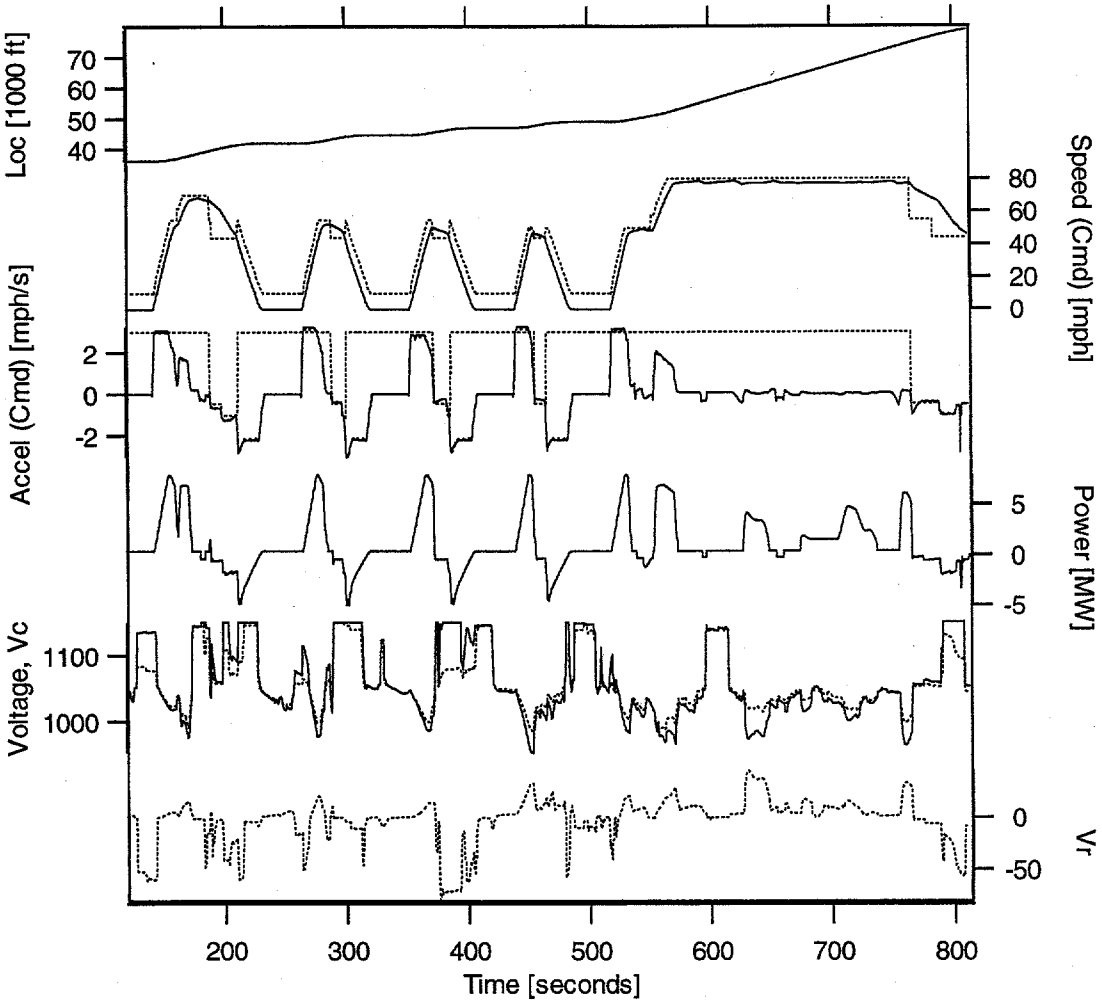

Figure 8. Plot of output data for one train

Figure 9 contains data from the same simulation run for one of the power substations. The power produced by the substation as a function of time, as well as the substation voltage, is shown. Power may only flow out of the substation, so power is always positive. When trains are regenerating nearby, the power drops to zero, and the voltage can float up well above the nominal 1050V.

Substation outages may be simulated by setting the substation nominal voltage to a low voltage, such as 500V. Since system voltages are always well above this value, the substation output power will always be zero, and the voltage will always be floating.

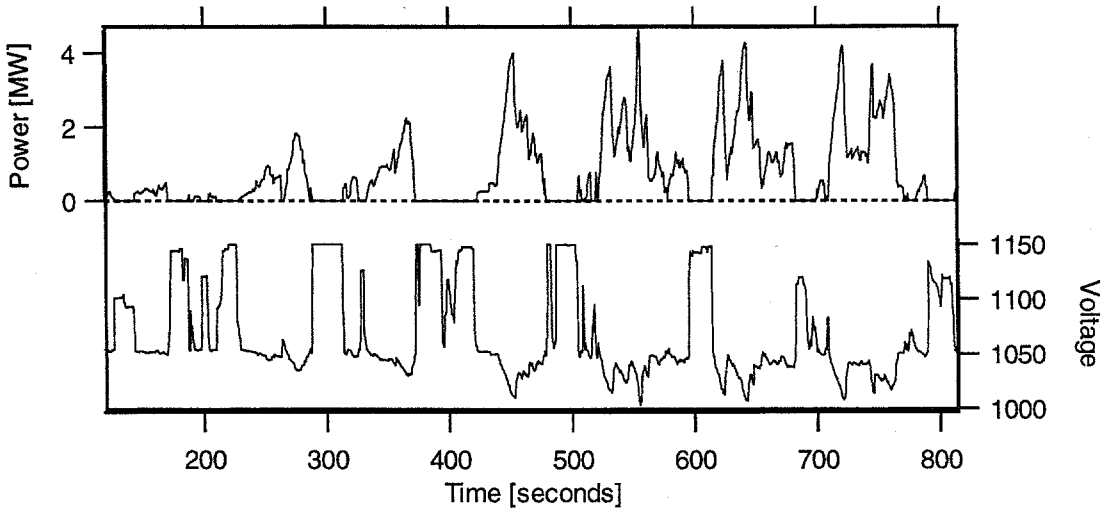

Figure 9. Plot of output data for one traction power substation

Non-vital Algorithm Simulator (NAS)

The non-vital algorithm simulator module allows the user to simulate enhancements to the baseline AATC train control system. In the implemented system, the enhanced algorithms will reside in a non-vital computer, which will communicate suggested train control commands to the vital computer. The algorithms will use information about the state of the system received from the vital computer and through an Ethernet connection from neighboring station controllers to calculate the suggested commands. Communication back and forth between the vital and non-vital computers will occur only once per half-second command cycle, so the enhanced algorithms must be able to overcome significant time delays between when suggestions are made and when they are implemented.

In the ATAPS model, once each half-second control cycle, information about the state of the system passes from the TCS to NAS, including what the trains are doing and what they are being commanded to do. NAS then has the opportunity to evaluate whether or not to suggest changes to the train commands for future control cycles. Communication between the vital and non-vital control and between the vital control and the trains results in a 1.5 second delay between the time when the non-vital controller makes a suggestion and the time when the information returns indicating that it has been implemented. This time delay must be accounted for in any enhanced algorithm design. We are not certain at this point whether this delay of 3 command cycles will be exactly reproduced in the final implemented AATC system. The important point is that the simulator represents a time delay of several command cycles, and that all non-vital control algorithms use logic which can handle such delays. If the delay in fact turns out to be two or four cycles rather than three, then the algorithm logic will need to be adjusted accordingly, but the fundamental logic should still apply.

Non-vital enhanced algorithms are incorporated into NAS as separate functions. Any number of such functions may operate simultaneously. Any coordination of multiple algorithms would also be added to the NAS module.

Benefits Assessment Module (BAM)

The Benefits Assessment Module evaluates control- and power-related metrics as the simulator runs, and outputs a file containing the results when the run terminates. For example, BAM counts substation power spikes, and tracks the minimum voltage experienced by each train. This module acts only as a bean-counter and does not contribute any information to the enhanced algorithms. All available system information is fed into BAM and processed, but no information is returned. Thus, BAM provides the user with summarized results from a simulation run, but does not impact the run in any way.

Function Interfaces

The MODRAILS and NAS modules are interfaced to the TCS in the BART function "run_sim." This function calls the function "ModTrainCmds," which serves as an interface to all of the remaining Sandia functions. Suggested speed and acceleration commands calculated by NAS are returned by this function call for implementation by the TCS. This is shown schematically in Figure 10.

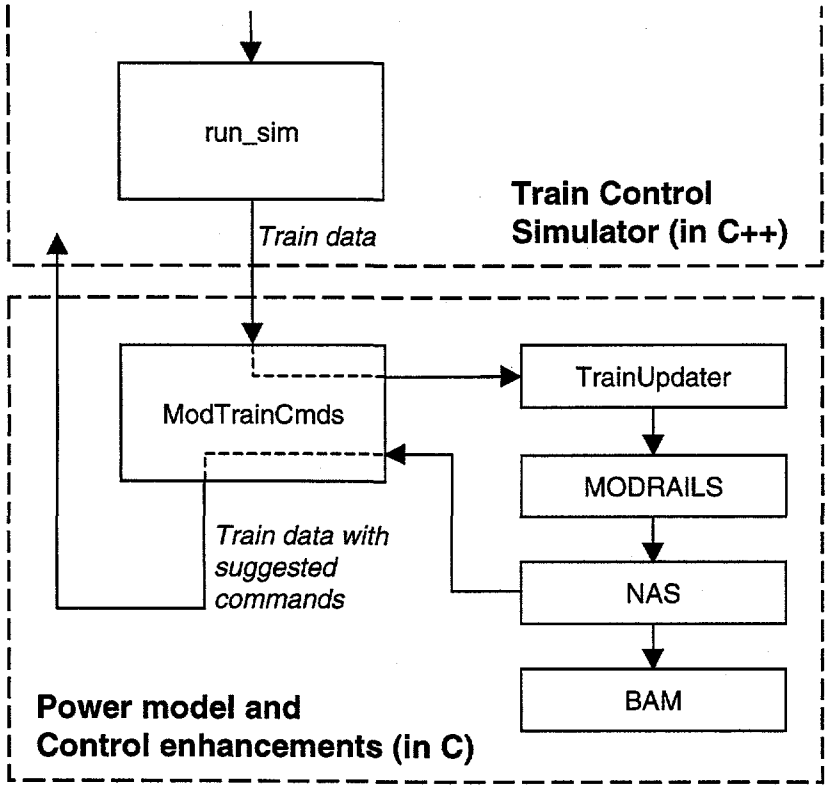

Figure 10. ATAPS function interface structure

The design of the integrated simulator is heavily influenced by the fact that the BART code is written in C++, while the Sandia-generated code is written in C. In order to minimize mixing code of different languages, the interface between the BART and Sandia code consists of a single function call. Using this function as a container for other functions that do the "real work" allowed BART and Sandia, in our collaborative relationship, to easily update our separate sections of the code independently. In addition, a power model capable of modeling more complex track geometry than a single line could be substituted for MODRAILS at some future date. The only exception to this single-interface philosophy is the train motor model, PMAM (described with MODRAILS above), which is interfaced separately into the TCS as a C++ function. This was necessary because PMAM calculations directly impact train acceleration.

The ModTrainCmds module calls the function TrainUpdater, which sorts out which trains on the line are within the zone of the electrical model and initializes the train voltage guesses with either nominal values or the values from the previous time step. These initialized trains are then passed on to MODRAILS, where the electrical circuit is solved for train voltages and substation powers. This information is stored in an output file, and is also passed on to the NAS module, where the states of the trains and the power system are evaluated and new suggested train control commands are generated. These non-vital commands are returned for implementation in the TCS back through the original call of ModTrainCmds. BAM updates its stored metrics at the end of each time step, and only records this data in an output file when the simulation terminates. The diagram above shows the flow of information through the simulator schematically, but does not represent the actual coded calling sequence. In fact, the function ModTrainCmds calls all of the other Sandia functions.

Summary of Sandia-developed Functions

ModTrainCmds – Modify Train Commands
This function is the main interface to the BART Train Control Simulator. It receives a pointer to a structure containing all the trains in the system. It calls the other functions, described below, that process the data contained in the structure and ultimately write modified speed and acceleration commands back into it. The Sandia output files are also generated from within this function.

TrainUpdater – Update Train Data
This function "qualifies" the trains in the system for use by other functions. Among other things, it ensures that trains are in the proper order and are within the zone of the traction power simulator, MODRAILS. It also assigns initial voltage guesses to the trains, using either nominal voltages or, wherever possible, voltages calculated in the last time step.

MODRAILS – Model of DC Rail Systems
This function models a linear section of the BART traction power system – currently the M-line between Oakland West KOW and 24th Street Mission MTF substations. It calculates the voltage of the trains and the power provided by each substation. It returns the number of iterations it needed to converge on a solution.

NAS – Non-vital Algorithm Simulator
This function uses the track and schedule information, and the states of all active trains and substations to calculate suggested AATC velocity and acceleration commands. It contains a number of train control algorithms suited to a variety of conditions such as clearing a backup or preventing interference.

BAM – Benefits Assessment Module
This function evaluates train- and substation-related metrics, such as the average power from each substation and the lowest voltage of each train. These metrics aid the user in evaluating the impacts of various enhanced train control algorithms. It writes this information into a data structure whose pointer is returned to the calling function.

traction_power – train Power and Maximum Acceleration Model (PMAM)
This is the only Sandia-created function which has been converted to C++ by BART and incorporated directly into the Train Control Simulator. It calculates the power consumed or regenerated by a train and the maximum possible acceleration that the motors can provide given the current train state. This function not only impacts power calculations, but also limits the train trajectory to physically realizable accelerations.

Input and Output

SIM Input File

In its current form, the simulator assumes that the first track database specified in the input SIM file is FD.SDF and the second is DF.SDF. These are the only track databases currently supported, as described in the MODRAILS section above. The correct ordering is required for the lines to be matched to the appropriate electrical database information contained within MODRAILS.

The ordering of train starting locations is also critical to the operation of the simulator. The trains on the same line must be specified in descending order, so that the first train has the highest absolute location, the second train has the next highest location, and so on. If the trains are specified in a scrambled or inverse order and enter the zone of the electrical model this way, the Sandia code will respond with an error. However, trains on the same line may be given the same initial location outside of the electrical zone, as in the example below on the FD track. The train simulator will dispatch these trains one at a time, and they will enter the electrical zone in the correct order.

The sample SIM input file below includes italicized comments pertaining to operation of the Sandia code. More complete comments on SIM input file parameters may be found in the BART Train Control Simulator documentation.[5] A description of this example follows the file text.

EXAMPLE.SIM file:

```
name=2+2 TRAINS, DF backup run, Optimize on
descr=AATC, backup at Montgomery[M20] on DF, nominal trains on FD
; When backup moves, FD trains near stations M30&M40
NumberOfTrains = 4
AATCControl=ON
Optimize=ON     ; This parameter toggles the use of the Sandia suggested speed and acceleration commands
TrainBufferDist=35
CivilLimitsOnly=TRUE
StationTargetVelocity=44.0
LoadTrackDatabase = FD:fd.sdf          ; This is the first track database specified; it will be treated as the M1
                                       line by the Sandia code.
LoadTrackDatabase = DF:df.sdf          ; This is the second track database specified and it will be treated as
                                       the M2 line by the Sandia code
OutputFile = EXAMPLE.DAT               ; This is used as the root name for all TCS output files. ATAPS output
                                       files described in "Output Files" below use the SIM filename, not this one. It is
                                       recommended that this filename match the SIM filename of this input file.
Length=10
waysidecontrol  = 0.5
outputrate = 1.00
speedcommand  = 70
OutputDetail = SUMMARY
RestrictILK=FALSE
InterfaceZones = OFF
SORS = OFF
700FootRule = OFF
ForceDwells=30.0
Train=1                                ; This is the first train specified in the input file; it will have a Train ID of 0
TrackDatabase = FD
StartLoc = M10                         ; This starting location is a station  just outside the zone of the Sandia electrical
                                       model
DispatchTime = 0                       ; A dispatch time of zero is the default; this line is for clarification only
Train=2                                ; This is the second  train specified  in the input file; it will have a Train ID of 1
TrackDatabase = DF
StartLoc = 46985                       ; This starting location on the M2 line is inside the zone of the Sandia electrical
                                       model
DispatchTime = 716
Train=3                                ; Train ID's continue to be assigned sequentially
TrackDatabase = FD
StartLoc = M10                         ; This starting location is the same as for the first M1 train.  This train will
                                       dispatch second.
DispatchTime = 140                     ; Specifying an absolute dispatch time provides a high degree of control of train
                                       phasing
Train=4
TrackDatabase = DF
StartLoc = 46185                       ; This is the second M2 train and must have a smaller location than the first
PaceTrain=1                            ; This is useful for generating delay statistics
```

In the example above, a backup is modeled on the DF line. Initially, the trains on the FD line are located at Oakland West station (M10), just short of the electrical model zone which begins at Oakland West substation (KOW). The first train is dispatched immediately, and the second follows at a headway of 140 seconds. These two trains behave nominally. On the DF line, a backup is simulated by starting one train in Montgomery station (M20), and starting a second train 800 feet behind it. The first train sits in the station until its dispatch time of 716 seconds, which allows the nominal FD trains enough time to move into the area. When the first DF train clears the station, the following train will automatically move forward behind it. The train stopped in Montgomery station is given an initial location in feet rather than the station identifier M20, because the latter option starts the train at the entrance to the station rather than at the desired location where it would stop for passengers.

Output to the Screen

The output to the screen by the TCS has been supplemented by Sandia to give some information about the electrical simulation during the course of a simulation run. Each time step, the number of trains in the zone of the electrical solver is printed to the screen in the message "___ trains in TU," where TU refers to the TrainUpdater module. This allows the user to monitor the number of trains under enhanced control and in the traction power model. This information is useful, for example, if the user wishes to end the simulation run by hitting the ESC key after all of the trains have left the zone of the solver.

In addition, during the course of each time step, a "." character is printed to the screen each time the iterative solver in MODRAILS has tried 25 guesses without converging. Since the solver typically converges in about 10 iterations or less, the appearance of "....." on the screen indicates a convergence problem in MODRAILS which should probably be corrected. Error messages are also printed to the screen.

Output Files

The Sandia code can create three different types of output files, which are given the same name as the SIM input file read by the simulator, and have file types TAP, BAM, and YAL. These files originate from the ModTrainCmds function. The sections below describe the contents and intended uses for each file type. Not all output files are appropriate for all simulations.

Train and Power (TAP) Data Output File

This output file contains detailed information about the states of the trains and substations for each time step. There are two possible formats for this data which may by selected. The first format, affectionately called the "oldTAP" format, is preferable for readable viewing of the data in a text editor or word processor, but requires substantial post-processing to load into a data analysis program for plotting purposes. This format lists information about each train and substation in separate tab-delimited rows, and repeats the listing for each timestep. An example of an oldTAP format TAP file is shown below, followed by descriptions of the data it contains. Some explanatory comments have been added in italics. This output was produced by a run of the EXAMPLE.SIM file above.

EXAMPLE.TAP file with oldTAP format:

```
Time      Iterations          (the file begins with these headers, which are explained below)
SSIndex   SSPower   SSVolts
ID  Line  Position  PowerPerTrain  TrainSpeed  SpeedCmd  AccelCmd  VHC  VHR
748.50    8                      (first timestep with electrical models running)
0         0.034     1054.744       (first substation)
1         0.047     1054.451       (second substation)
2         0.047     1054.449       (repeated for each substation...)
3         0.158     1053.180
4         0.346     1051.078
5         0.673     1051.178
6         0.000     1148.074
7         0.000     1149.133
2   1     168341.297  3.514  51.285  55.000  3.000  1051.312  -1.697  (first train)
0   1     175663.875  -5.061  39.277  38.801  -0.400  1098.562  -51.438  (second train)
1   2     46986.289  0.822  2.098  9.000  3.000  1048.062  3.308  (repeated for each train)
3   2     46185.000  0.400  0.000  0.000  -0.450  1048.062  3.332
749.00    6                      (second timestep)
0         0.034     1054.757       (repeated for each substation...)
1         0.047     1054.463
2         0.047     1054.459
3         0.128     1053.528
4         0.378     1050.728
5         0.507     1052.119
6         0.000     1147.474
7         0.000     1148.880
2   1     168379.344  2.895  51.617  55.000  3.000  1052.795  -2.234  (repeated for each train)
0   1     175692.281  -4.403  37.801  37.254  -0.400  1099.088  -50.912
1   2     46988.516  1.185  3.703  11.000  3.000  1047.399  4.402
3   2     46185.000  0.400  0.000  0.000  -0.450  1047.637  4.331
749.50    5                      (repeated for each timestep)
...
```

Descriptions of oldTAP data headers:

Time – simulation time [seconds]
Iterations – number of iterations used by MODRAILS to converge on a solution for this time step
SSIndex – the subscript of the substation in the substation array defined in MODRAILS
SSPower – energy produced by substation for this time step [MW]
SSVolts – voltage at substation for this time step (contact rail minus running rail voltage) [Volts]
ID – train ID as determined by appearance in input file (see "SIM Input File" above)
Line – 1 for M1 (FD) line, 2 for M2 (DF) line
Position – location of train for this time step [feet]
PowerPerTrain – power consumed by train including all cars for this time step [MW]
TrainSpeed – velocity of train [MPH]
SpeedCmd – AATC speed command [MPH]
AccelCmd – AATC acceleration command [MPH/second]
VHC – voltage on the contact rail at the head of train [Volts]
VHR – voltage on the running rail at the head of train [Volts]

The second TAP format, or "newTAP" format, is designed for efficient importing into spreadsheet or data plotting packages. This format lists of all the train and substation information for a given timestep in a single tab-delimited row. An example of a newTAP format TAP file is shown below, followed by descriptions of the data it contains. Like the oldTAP example above, this output was also produced by a run of EXAMPLE.SIM file. The data for each timestep is contained in a single row, which is wrapped due to its width.

EXAMPLE.TAP file with newTAP format:

Time	Iterations	KOW_v	KOW_p	KTE1_v	KTE1_p	KTE2_v	KTE2_p	MTW1_v
	MTW1_p	MTW2_v	MTW2_p	MPS_v	MPS_p	MSS_v	MSS_p	MTF_v
	MTF_p	Loc_10	Speed_10	VCmd_10	ACmd_10	Power_10	Vc_10	Vr_10
	Loc_12	Speed_12	VCmd_12	ACmd_12	Power_12	Vc_12	Vr_12	Loc_21
	Speed_21	VCmd_21	ACmd_21	Power_21	Vc_21	Vr_21	Loc_23	Speed_23
	VCmd_23	ACmd_23	Power_23	Vc_23	Vr_23			
748.500	8.000	1054.744	0.034	1054.451	0.047	1054.449	0.047	1053.180
	0.158	1051.078	0.346	1051.178	0.673	1148.074	0.000	1149.133
	0.000	175663.875	39.277	38.801	-0.400	-5.061	1098.361	-50.447
	168341.297	51.285	55.000	3.000	3.514	1051.010	0.086	46986.289
	2.098	9.000	3.000	0.822	1047.992	3.408	46185.000	0.000
	0.000	-0.450	0.400	1048.062	3.332			
749.000	6.000	1054.757	0.034	1054.463	0.047	1054.459	0.047	1053.528
	0.128	1050.728	0.378	1052.119	0.507	1147.474	0.000	1148.880
	0.000	175692.281	37.801	37.254	-0.400	-4.403	1098.134	-49.395
	168379.344	51.617	55.000	3.000	2.895	1051.839	0.146	46988.516
	3.703	11.000	3.000	1.185	1047.399	4.402	46185.000	0.000
	0.000	-0.450	0.400	1047.637	4.331			

...(repeated for each timestep)

Descriptions of newTAP data headers:

Time – simulation time [seconds]

Iterations – number of iterations used by MODRAILS to converge on a solution for this time step

KOW_v – voltage at the KOW substation for this time step (contact rail minus running rail voltage) [Volts]

KOW_p – power produced by the KOW substation for this time step [MW]

KTE1_v– voltage at the KTE1 substation for this time step

KTE1_p – power produced by the KTE1 substation for this time step

...(repeated for each substation by name)

Loc_10 – location of train for this timestep [feet]. *The number suffix indicates the line (1 = M1, 2 = M2) and train ID. This train is on M1 with an ID of 0. The same suffix is used for all data associated with the same train.*

Speed_10 – velocity of train [MPH]

VCmd_10 – AATC speed command [MPH]

ACmd_10 – AATC acceleration command [MPH/second]

Power_10 – power consumed by train including all cars for this time step [MW]

Vc_10 – the voltage of the train on the contact rail.

Vr_10 – the voltage of the train on the running rail. *The voltages Vc and Vr correspond to the calculated voltages at the head or the tail location of the train, whichever location has the minimum voltage differential Vc - Vr.*

...(repeated for each train with appropriate identifying suffixes)

BAM (Benefits Assessment Module) Output File

This file type contains a summary of the performance of the trains and substations relative to train- and substation-related metrics. An example of a BAM output file is shown below. This output was produced by a run of the EXAMPLE.SIM file above. The measured metrics are defined below the file listing.

EXAMPLE.BAM file:

Total time of electrical simulation = 84.0
Average MODRAILS iterations = 5.3

SubName	AvgPower[MW]	Num300Events	SecP>300	Num450Events
KOW	0.049	0	0.0	0
KTE1	0.070	0	0.0	0
KTE2	0.070	0	0.0	0
MTW1	0.262	0	0.0	0
MTW2	0.500	0	0.0	0
MPS	1.062	0	0.0	0
MSS	0.083	0	0.0	0
MTF	0.038	0	0.0	0

Avg sys power = 2.135

Line	ID	Time@MinVoltage	Loc@MinVoltage	MinVoltage	Num750Events	SecV<800	SecV<880	NumBrkEvents	SecInZone
1	0	716.5	172726.547	1020.274	0	0.0	0.0	1	84.0
2	1	760.5	47366.148	972.164	0	0.0	0.0	1	84.0
1	2	745.5	168123.688	1021.465	0	0.0	0.0	1	84.0
2	3	760.5	46207.746	981.664	0	0.0	0.0	1	84.0

Descriptions of BAM metrics:

Total time of electrical simulation – total time in seconds that the electrical model was active
Average MODRAILS iterations – average of the iterations taken by the electrical solver for each time step
SubName – substation name (e.g. KOW, KTE1, ...)
AvgPower – average power produced by the substation over the time of the electrical simulation [MW]
Num300Events – number of times the substation exceeded 300% of its rating
SecP>300 – total time in seconds spent above 300% of substation rating
Num450Events – number of times the substation exceeded 450% of its rating
Avg sys power – average power produced by all substations over the time of the electrical simulation
Line – 1 for M1 line, 2 for M2 line
ID – train ID as determined by appearance in input file (see "SIM Input File" above)
Time@MinVoltage – simulation time at which the train experienced the lowest voltage [seconds]
Loc@MinVoltage – location at which the train experienced the lowest voltage [feet]
MinVoltage –lowest voltage experienced by the train [V], which occurred at the noted time and location
Num750Events – number of times the train experienced a voltage less than 750 V
SecV<800 – total time over which the train experienced a voltage less than 800 V [seconds]
SecV<880 – total time over which the train experienced a voltage less than 880 V [seconds]
NumBrkEvents – number of times the train switched from propulsion into braking mode

SecInZone – time in seconds spent by train in the zone of the electrical model during the course of the simulation. *This parameter may be used to measure trip time only for simulations run long enough for each train to traverse from one end of the electrical zone to the other. In the example above, the simulation was run for 84seconds, and all four trains spent the entire time in the zone.*

YAL Output File

This format was used by our colleagues at Yale University to train a neural network to estimate train voltages and substation powers. These neural nets were developed to provide enhanced algorithms with rapid estimates of these parameters, avoiding use of the iterative solver in the MODRAILS function. Like the TAP output file, the YAL output file records train and substation information for each timestep. However, this output file contains significantly less information than a TAP file, limiting the outputs to those required for neural net training. Part of the YAL output file produced by a run of EXAMPLE.SIM is shown below, followed by descriptions of the data it contains.

EXAMPLE.YAL file:

T12Loc, T12P, T12V, T10Loc, T10P, T10V, T21Loc, T21P, T21V, T23Loc, T23P, T23V, KOW, KTE1, KTE2, MTW1, MTW2, MPS, MSS, MTF

168341, 3.514, 1050.92, 175664, -5.061, 1148.81, 46986, 0.822, 1044.58, 46185, 0.400, 1044.73, 0.034, 0.047, 0.047, 0.158, 0.346, 0.673, 0.000, 0.000

168379, 2.895, 1051.69, 175692, -4.403, 1147.53, 46989, 1.185, 1043.00, 46185, 0.400, 1043.31, 0.034, 0.047, 0.047, 0.128, 0.378, 0.507, 0.000, 0.000

...(repeated for each timestep)

Descriptions of YAL headers:

T12Loc – the location of the train [feet]. The number suffix indicates the line (1 = M1, 2 = M2) and train ID. This train is on M1 with an ID of 2. The same suffix is used for all data associated with the same train.

T12P – power consumed by train including all cars for this time step [MW]

T12V – the voltage of the train (contact rail minus running rail voltage) [Volts]. *This voltage corresponds to the minimum calculated voltage at the head or the tail location of the train.*

...(repeated for each train with appropriate identifying suffixes)

KOW – power produced by the KOW substation for this time step

...(repeated for each substation by name)

Simulator Validation

In order to test the validity of simulation predictions, it was necessary to compare them to measured empirical data. As has already been discussed, the Train Control Simulator, developed at BART, has been tested against detailed train trajectory data taken at the BART test track, as well as course data gathered during revenue operation. This section will discuss validation of the traction power predictions against train trajectory and voltage data gathered by Pacific Gas and Electric (PG&E).[7] For four months in 1993, PG&E measured the voltage at the MCG gap breaker near the middle of the transbay tunnel, and found that the voltage dropped below 850V on 25 occasions. As for the original version of the traction power simulator, we have simulated the deepest of these measured voltage sags, using BART records of train motion[8] to identify the

locations and behaviors of the trains during these events. We modified MODRAILS to create an output file containing the voltage at the MCG gap breaker for the purpose of this comparison.

For reference, an elevation plot of the M-line of the BART system between Oakland West station and 24th Street Mission station in San Francisco is shown in Figure 11. This is the length of track modeled by the traction power model and the non-vital algorithm simulator in ATAPS. A train on the M1 line would travel from right to left, and M2 from left to right. The figure shows the stations as labeled squares, and the power supply substations in italics. In addition, the gap breaker MCG is shown in the middle of the transbay tunnel. The tunnel presents a unique problem to the power system, because it is not possible the install a substation there. The substations at the ends of the tunnel, KTE and MTW, must provide power to trains nearly two miles away at MCG. Normally, while trains are traveling at constant speed through the tunnel, their power demands stay relatively low. However, trains occasionally experience low voltages in the tunnel when power demand becomes too large. PG&E took measurements of voltages in the tunnel in order to quantify the frequency and severity of this problem. Our previous study used the MODRAILS model to analyze the causes of these voltage sags, and to examine the benefits of potential solutions to the problem.[6]

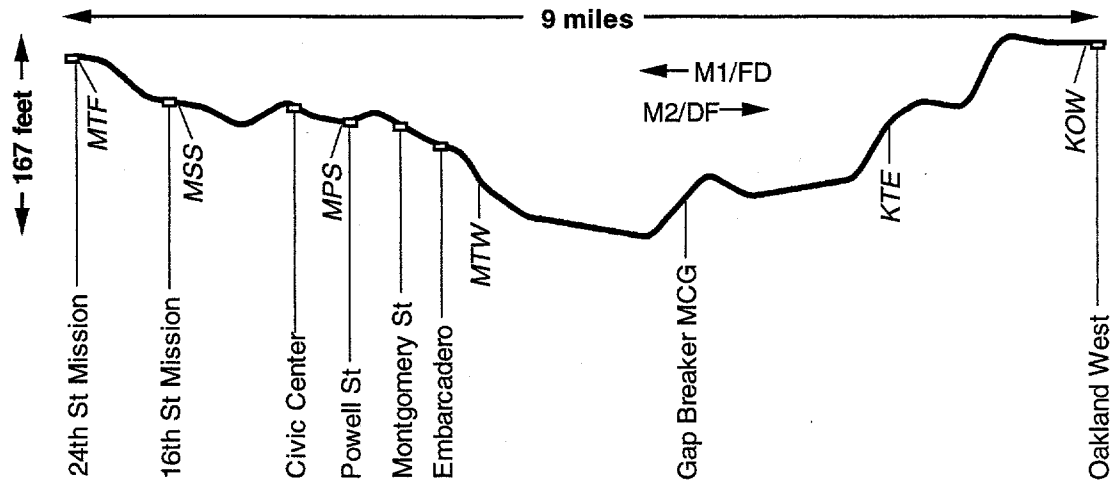

Figure 11. Elevation diagram (not to scale) of the BART M-line versus location in the zone modeled by ATAPS, with passenger stations and traction power features labeled. See Table 1 for track locations.

As may be seen in the elevation figure, there is a substantial hill in the middle of the tunnel under the bay. This hill exists because the BART tunnel was laid on the bottom of the bay, and passes over a spur of Yerba Buena Island. This hill increases the power demand of trains as they attempt to maintain their speed while climbing the hill. The train voltage calculated by ATAPS for a single simulated train passing over this hill while traveling in either direction is shown in Figure 12. The train voltage dips as the train climbs the hill due to the increase in power demand. As the train travels downhill on the far side of the hill, the voltage floats above the nominal substation voltage of 1050V, because the train regenerates power onto the third rail and no other trains are present to use the power. Finally, the train begins climbing out of the tunnel and the voltage dips a second time. The hill has a short 1% uphill grade in the M1 direction, and a longer 2% uphill grade in the M2 direction, which leads to the asymmetry in the voltage plots.

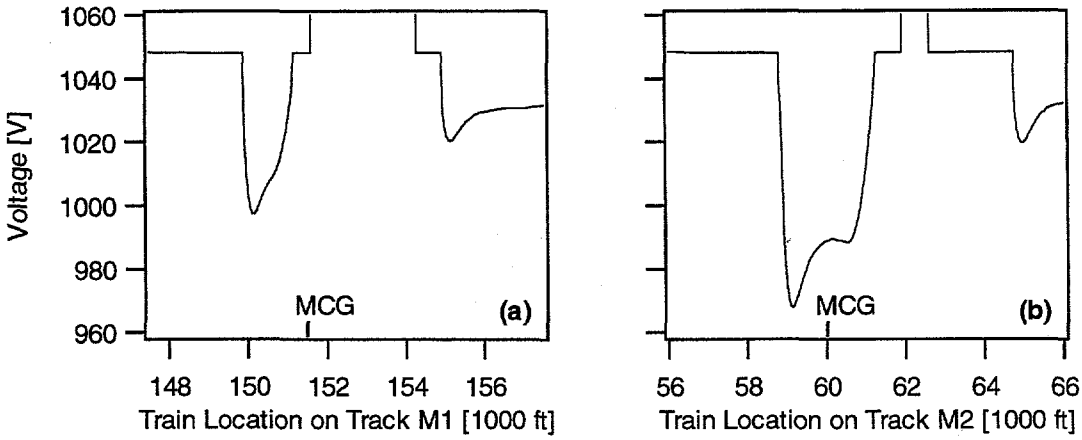

Figure 12. Calculated train voltage corresponding to a single train traveling over the hill in the middle of the transbay tunnel on (a) line M1 and (b) line M2.

One voltage sag recorded by PG&E occurred while two trains went over this hill simultaneously. The data[7] are shown as points and the simulation is represented by a line in Figure 13. This sag was reproduced fairly well by the simulator, with the exception of the transition between propulsion and braking. The current version of the Train Control Simulator predicts a period of coasting during the mode change between propulsion and braking, which is reflected in the flat-line voltages in the previous single-train voltage figures. The data, on the other hand, seems to indicate a slow increase in power demand, as reflected in the slow onset of the voltage sag, which is not reproduced by the simulator. An older version of the TCS, tested for our previous analysis,[6] did not predict these discontinuities in power consumption, but rather assumed a smooth transition, and the voltage data was reproduced more accurately. This argues that mode changes were more accurately modeled in the old version of the simulator. However, the general shape and duration of the voltage sag is still reproduced quite well, so that the accuracy of the model is sufficient for the purposes of the analysis presented in this report.

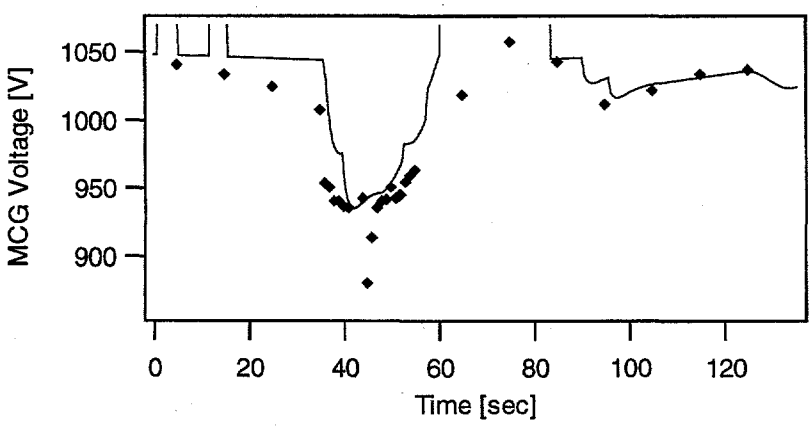

Figure 13. Measured versus calculated breaker voltage for two trains simultaneously traveling over the hill

A similar event occurred early one morning after a train was intentionally started up from a stop near the middle of the tunnel. A voltage sag was recorded as a result of this train accelerating up to speed. A few minutes later, a second sag occurred, which looks very similar to that shown above for two trains speed-maintaining over the hill in the tunnel. From BART records of train motion, it was confirmed that indeed two trains had passed over the hill simultaneously at that time. The data is shown compared to a simulation run of this event in Figure 14, again with the PG&E data[7] represented as dots and the simulation as a line. The agreement between the data and the model is again quite good.

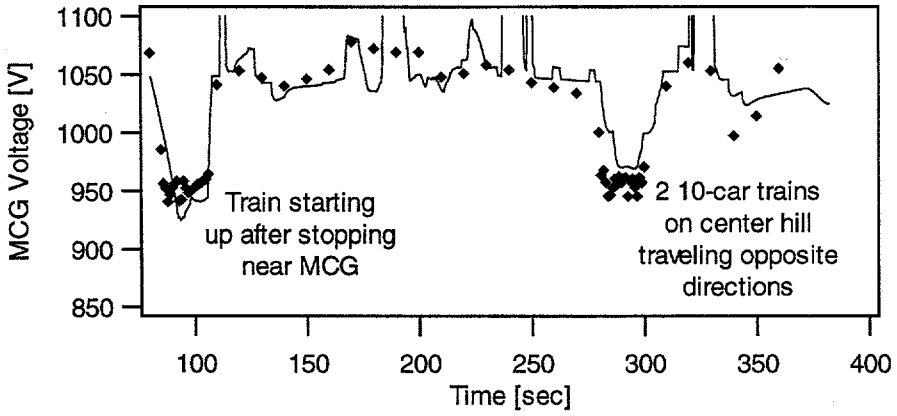

Figure 14. Measured versus calculated breaker voltage for a train starting from a stop near the middle of the tunnel, followed by two trains passing over hill

A long series of repeated voltage sags was recorded in another event due to interference between two trains moving through the tunnel. The lead train moved at approximately 50mph in a 70mph speed zone, while the train behind it attempted to accelerate to 70mph. As a result, the following train repeatedly accelerated, caught up to the slow lead train, and switched to braking. The control system is currently a fixed block system, so acceleration and braking of the following train was initiated each time one of the trains entered a new control block. Each acceleration cycle caused a voltage sag. Coincidentally, as the pair of trains passed near the middle of the tunnel, the lead train finally accelerated to 70mph, causing both trains to accelerate simultaneously up to speed. The worst voltage sag occurred at this point. The exact sequence of voltage sags could not be reproduced with the simulator due to the complexity of the event. However, the voltage calculated for a substantially similar event with trains behaving in a similar manner is shown alongside the data in Figure 15.

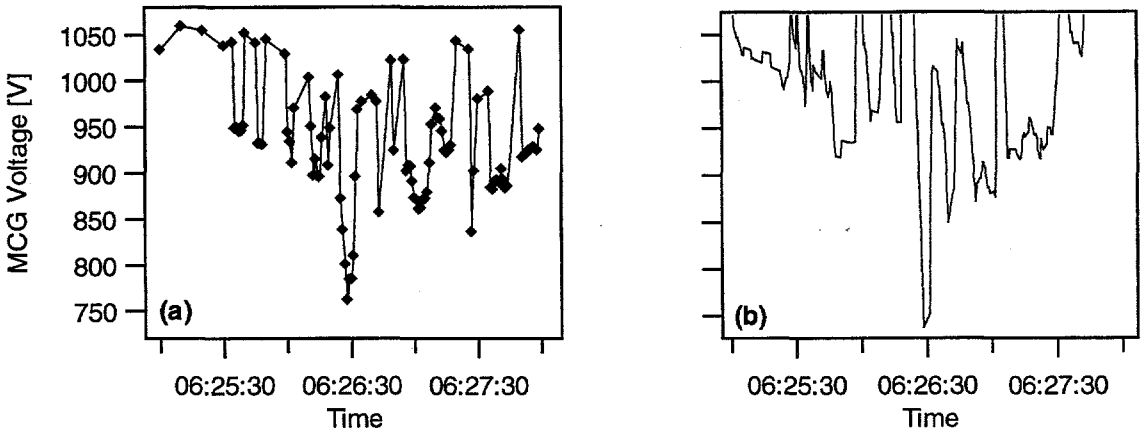

Figure 15. (a) Measured[7] and (b) calculated breaker voltage for a slow lead train with an interfering train behind it causing voltage sags

The lowest voltage sag was measured when a SORS reset occurred on the system while two trains coincidentally were traveling through the tunnel. A SORS reset is a control action used to return train speed limits in an area from a restricted speed condition to normal speeds. Restricted speeds are typically used during maintenance for safety reasons. The measured voltage sag[7] had a somewhat longer duration than that calculated by the simulator, as shown in Figure 16. However, the two momentary voltage recoveries exhibited by the data but not by the simulation indicate that the trains probably had motors shut down due to the low voltage, which would necessarily increase the duration of the event. Trains are nominally set to shut down their motors if the supply voltage drops to 750V in order to avoid damage to the motors. The simulator does not predict motor shut downs, or the reduced motor performance that occurs at voltages below 880V. Thus, the actual train acceleration would be less that that predicted by the model, and the acceleration would have longer duration. The simulator does replicate the approximate severity of the voltage sag during this event, which is sufficient for the purpose of developing control techniques to prevent such events from occurring.

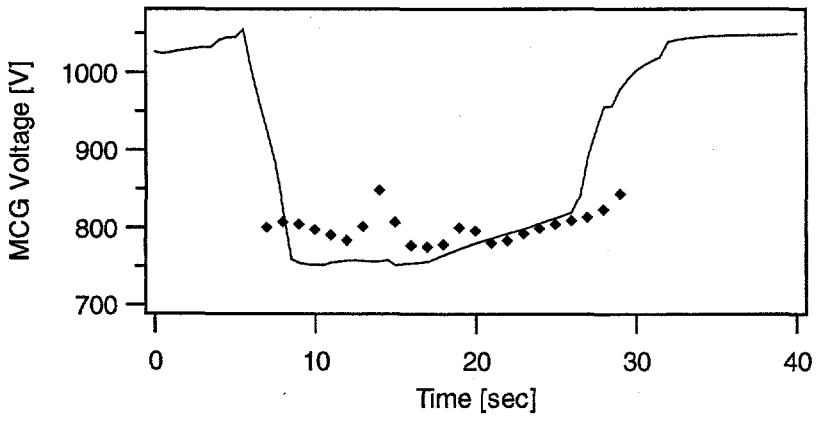

Figure 16. Measured versus calculated breaker voltage for a "SORS Reset" event that caused a severe voltage sag

In summary, the ATAPS simulator replicates the measured voltage data in the tunnel with sufficient accuracy for the purposes of enhanced control development. The simulator may also be useful for studies of regenerated energy usage, as long as care is taken to examine areas of the system that are not in close proximity to the ends of the modeled region. That is, the current version of the model may be used for energy consumption studies in downtown San Francisco, in the middle of the modeled region. However, near Oakland West, energy flow to and from the Oakland Wye would not presently be accounted for, leading to inaccurate results.

For studies of the traction power system that may influence power infrastructure decisions, we would recommend that the simulator be validated against a more complete set of data, including train trajectory information, train voltages, and substation voltages and power flow.

Enhanced Control Algorithms

The ultimate goal of enhanced train control is a system optimized with respect to a well developed and complete cost function. This function would represent the relative value of such things as trip time, energy and power usage, schedule, and rider satisfaction. Although global optimization is a worthy goal, we believe that it is important to begin tackling the problem of enhanced train control by first solving more localized and well-defined problems. Therefore, we have attempted to design a few control algorithms that solve specific problems, and to assess tradeoffs in their impacts on various important metrics. Once some intuition has been developed in this way, it may be possible in the future to design a more globally applicable algorithm.

Appendix A summarizes the scenarios, control and state variables, constraints, uncertainties, and objectives that are relevant to train control optimization. From the long list of possible control objectives, we have concentrated on the five shown in Table 2. For each power- or service-related objective, we have listed the relevant metric with which to measure it. In our pursuit of a more efficient and reliable control system, our primary objectives for enhanced control include reduced energy infrastructure costs, reduced overall energy usage, and improved service reliability. Fortuitously, in most cases the algorithms developed to achieve these goals provide the additional benefit of improved passenger comfort. Given a trip time constraint, a smoother ride typically corresponds to a reduction in unnecessary acceleration and braking cycles, which in turn reduces peak power consumption and overall energy usage. Thus, algorithms which target improved service often have energy-related benefits as well. In general, both energy and reliability goals are linked to smoother train trajectories, which are also more comfortable to experience.

Table 2. Enhanced control algorithm objectives and metrics

Objectives & *Metrics*	
Power-related	
Avoid energy infrastructure costs	Low train voltages
Reduce overall energy usage	Energy consumption
Service-related	
Improve service reliability	Train-minutes of delay Motor mode changes
Improve passenger comfort	Smoothness of service
Minimize trip time	Trip time

The architecture of the BART system presents some interesting challenges regarding distributed control. Although the intelligence in the system is on the wayside, rather than on the trains, it is distributed among multiple station controllers. Control of any given train is passed from one controller to another as the train progresses along a line. The baseline AATC will control trains

as independent entities, each with its own separate mission. Thus, each station computer must be aware of only the trajectories of its own trains and perhaps a few trains in neighboring zones that are either about to enter its control zone or are potentially in the way of its controlled trains. Enhanced control algorithms that attempt to impact power consumption or coordinate multiple trains will require station computers to be aware of a more global picture of activities on the line. In addition to avoiding sudden discontinuities in train commands as trains cross control zone boundaries, for example, we may wish to coordinate the timing of a train leaving a station with the braking of another train into another station that lies in the next control zone, in order to make more efficient use of regenerated energy. Trains in different control zones share power from common substations and can provide power to each other through regeneration and transmission along the powered third rail. Thus, station controllers can not make intelligent power-related decisions, while ignoring trains in neighboring control zones.

For simplicity, we have assumed one large AATC control zone for initial algorithm development in the simulator. Additional work will be required to test the behavior of the developed algorithms in a system architecture with multiple station controllers.

The enhanced control algorithms that we have investigated are summarized in Table 3, along with the corresponding primary objectives. For each algorithm, we will describe the conditions under which it is meant to operate, the methodology we have used to approach the problem, the results of algorithm testing, and suggestions for upgrades.

Table 3. Summary of enhanced control strategies and objectives

	Algorithm	Primary Objectives
1	Avoid low train voltages	Avoid energy infrastructure costs Improve reliability
2	Smooth interfered headway operation:	
2a	Interference during acceleration	Improve reliability Improve passenger comfort
2b	Interference near station stops	Improve reliability Improve passenger comfort
2c	Interference during delay recovery	Reduce train-minutes of delay Improve passenger comfort Reduce energy costs
3	Coordinate starts and stops	Reduce energy costs
4	Coast	Reduce energy costs Reduce trip time
5	Reduce peak substation power	Reduce future instantaneous demand charge

Avoid Low Train Voltages

An algorithm that prevents the voltage at each train from dropping below some minimum value by limiting power consumption can lead to large benefits in avoided energy infrastructure costs, in addition to reducing energy consumption. The ultimate objective for this algorithm is to find a control strategy that will allow rush hour traffic to continue operating on the system during a substation power outage with the minimum required traction power infrastructure. This can be achieved by slowing trains as required in order to keep the system operating smoothly with the available power. The payoff can be substantial, because such a control strategy may be sufficient to delay the need for traction power system upgrades when the schedule is modified for short headway operation. A BART Extension Service Plan Study conducted by Parsons De Leuw in 1993 recommended $75 million worth of upgrades to the power system to handle traffic levels in the year 2002 based on the requirement that the system be capable of powering rush-hour traffic during a substation outage.[9] In some cases, a localized reduced Performance Level (using half of the normal acceleration rate and no speeds greater than 60mph) was found to be sufficient to avoid the need for some additional power capacity. However, reducing PL commands is not a very effective method for reducing peak power demand. Since outages during rush hour constitute very rare events, it seems reasonable to avert or delay power infrastructure costs with a control strategy that reduces power demand as needed during an outage.

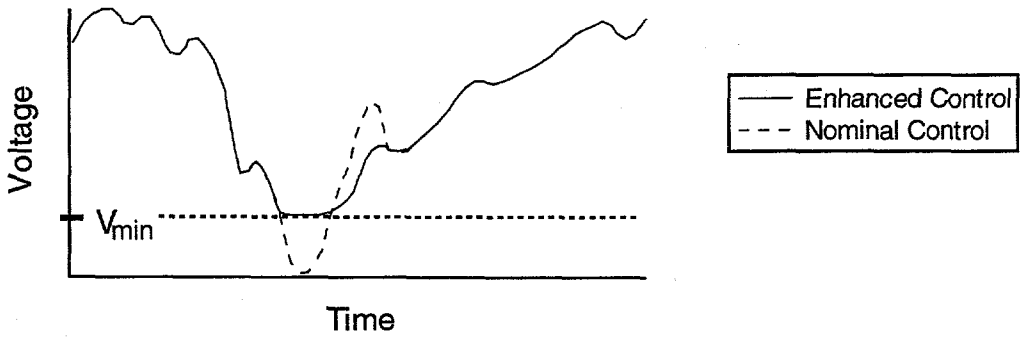

Figure 17. Sketch demonstrating algorithm to avoid low train voltage

Even with a power infrastructure that is sufficient to run trains normally the vast majority of the time, occasional coincidences of multiple trains accelerating can also cause power shortages. When several trains that are close together demand power at the same time, if there is insufficient power available nearby due to either track geometry or an outage, then the voltage at the trains can drop precipitously. If the train voltage drops below 880V, train motor performance begins to degrade. If the voltage drops below 750V, the motors currently on many BART trains will shut down in order to avoid damage from excessive current flow. Even with motors that do not shut down, it is inefficient to allow severe voltage sags, since low voltages typically correspond to large power losses to heat in the rails. The typical response when low voltages occur is to add more traction power infrastructure until the system is sufficiently robust as to be able to handle any possible situation, which may occur under relatively normal operating conditions. Moreover, since the system must be able to operate during an outage at a substation,

additional power capacity is installed so that the voltage will be maintained at some reasonable level even in this circumstance. Enhanced control can avoid low voltages by regulating power usage, thereby avoiding, or at least deferring, tens of millions of dollars of traction power capital equipment costs.

In order to maintain the voltage of all trains above some reasonable minimum, which is somewhere in the neighborhood of 800V, it is necessary to predict train voltages based upon the trajectories of all nearby trains, and then allocate the available power in such a way as to maintain the voltage at all trains while minimizing the impact on the schedule. This algorithm, with such a high payoff, is not surprisingly quite difficult to achieve. Power consumption can rise quickly enough to take the voltage from a comfortable range to well below the desired minimum in a matter of seconds. Thus, it is not sufficient to measure or calculate train voltages and react as the voltage drops too low; but rather, potential problems must be recognized before they materialize. However, train voltage is a non-linear function of power demand, which makes it difficult to predict quickly and reliably.

It is possible that a simpler solution to low voltage problems may be achieved through on-board rather than wayside control, because this would avoid the problems of predicting low voltages and mastering the time lag between wayside command and execution. On-board logic could reduce power demand as the voltage drops, with power approaching zero as the voltage approaches the desired minimum. Some care would be required to implement this solution so that the system is stable and resistant to oscillations in train acceleration.

The wayside control approach allows for more flexibility in the way that limited power resources are allocated. An on-board controller would react to a low voltage condition by reducing the power demand of a train based solely on the voltage measured on-board that train. By contrast, a wayside controller could take into account the schedule, and prioritize the power allocation to the trains. For example, if two trains traveling in opposite directions on a line are both accelerating, and there is only sufficient power available for one of them, then the on-board control solution would be to cut the power demand of each in half. However, if one train is on time, and the other is critically behind schedule, then it may be desirable instead to allocate most of the power to the high priority train and allow the other to coast. A wayside-controlled algorithm would allow such decisions to be made on a case-by-case basis.

Algorithm Approach

We have investigated an algorithm to maintain train voltage in any train control situation, such as during a backup, with a normal traction power system. We believe that the same technique may be applied in an outage condition. The concept was demonstrated in the simulator for the section of track in the middle of the transbay tunnel. Rather than employing a slow but accurate system model such as MODRAILS to predict voltages, we have employed neural network technology to predict approximate train voltages. Data produced by multiple simulator runs were used to train a neural network to predict train voltages based on local power demand patterns. Using the neural network to provide a functional approximation of the system, an algorithm then calculates reduced acceleration commands for trains so that the predicted voltage never drops excessively.

The algorithm logic begins by estimating the power consumption of each train, P_{train}. Since any suggestion made by the algorithm will not go into effect immediately, the algorithm assumes that the worst case for power demand may occur; that is, it assumes that a train in propulsion will accelerate at full power, and a train in braking will start coasting. Given the train locations and worst-case predicted power demands, the neural net (described below) is then run to predict the worst-case voltage, V_{train}, for each train that is in propulsion. If any train's estimated worst-case voltage is below the minimum desired voltage, e.g. 800V, then acceleration commands are reduced to prevent this from occurring.

For each train with a voltage close to the minimum, e.g. below 900V, its acceleration command is reduced by a factor related to the severity of the voltage sag and the power consumption of the train. Specifically, the train's acceleration command is multiplied by the factor

$$e^{-(V_{sag}/300)^2 * (P_{train}/0.7)},$$

where $V_{sag} = 900V - V_{train}$, and P_{train} is the worst-case estimated power consumption of the train. The farther below 900 Volts that a train's voltage drops, the more its acceleration command is reduced by this multiplier. The power factor is included here because a train's voltage can potentially drop due to power consumption by other nearby trains. The acceleration commands to trains with both low voltages and high power consumption are preferentially reduced. A factor related to the schedule could be included here as well, so that trains closer to on-time would be preferentially slowed in deference to those that are delayed.

The acceleration command to each low-voltage train is reduced once, and then the worst-case power demands are recalculated with the reduced acceleration commands. Train voltages are again estimated by the neural network, and the process is repeated until no trains are predicted to drop below the minimum acceptable voltage of 830V.

This voltage-maintaining algorithm is designed to run continuously, and to run last in the non-vital processor. Other algorithms running first may suggest reductions to the acceleration commands sufficient to avoid low voltages, so existing suggestions are used to modify the initial assumption of maximum train acceleration. This algorithm will then reduce the commands further if necessary to insure low voltages will not occur.

As will be discussed below, the algorithm does not have a large impact on train trajectories unless an abnormal control situation such as a backup leads to low voltage conditions. However, it is possible to adjust the impacts of this algorithm by varying the minimum allowable train voltage according to the needs of the moment. Such an adjustment to the algorithm could be accomplished by sending a parameter to the non-vital station computer from Central control. This parameter would need to control all of the related algorithm constants: minimum voltage (830V here), low voltage (900V), and voltage sensitivity (300). If delay prioritization were added to the algorithm, the sensitivity of this factor could also be adjusted.

Before discussing the testing of this algorithm, we will first discuss neural networks in general, and as they apply to this application.

Neural Networks For Non-Linear Mapping

Neural networks have been found to do an excellent job of mapping nonlinear functions. Their advantage is that once they are trained and tested, they are computationally very efficient. A very practical use for them in nonlinear control problems is to compute the input/output relationship of a nonlinear process in real-time. Nonlinear models often require computations in which one iteration or time step takes on the order of minutes to complete, even on a super computer. This may be too slow to be used in practical advanced control algorithms. A large advantage of neural networks is the ability to perform nonlinear mapping of the model off-line in simulation where all possible input/output relationships can be tested. The trained neural network algorithm can then be run in real time with advanced control algorithms using standard computer resources.

It is very important to test the neural network over the entire multidimensional mapping space. This guarantees that the output is always predictable and stable. One of the largest problems with neural networks is that their black box approach leaves skeptics with a lack of confidence in their output stability. Proving the stability of the network in a simulation is critical for success. Once tested, it is a simple job to incorporate the neural net algorithm into an advanced controller for real time implementation.

The neural network is viewed mathematically as a black box with inputs and outputs similar to the system or function it is trying to map. An extensive set of training data must be obtained from the system before training begins. The input-output data set is called the training set. The data is used to train the neural network in an off-line simulation mode. The back-propagation algorithm is used to train the network with the data until the desired minimal error is reached. Afterward, the network is static and is only run in the feed-forward mode, meaning no more training occurs.

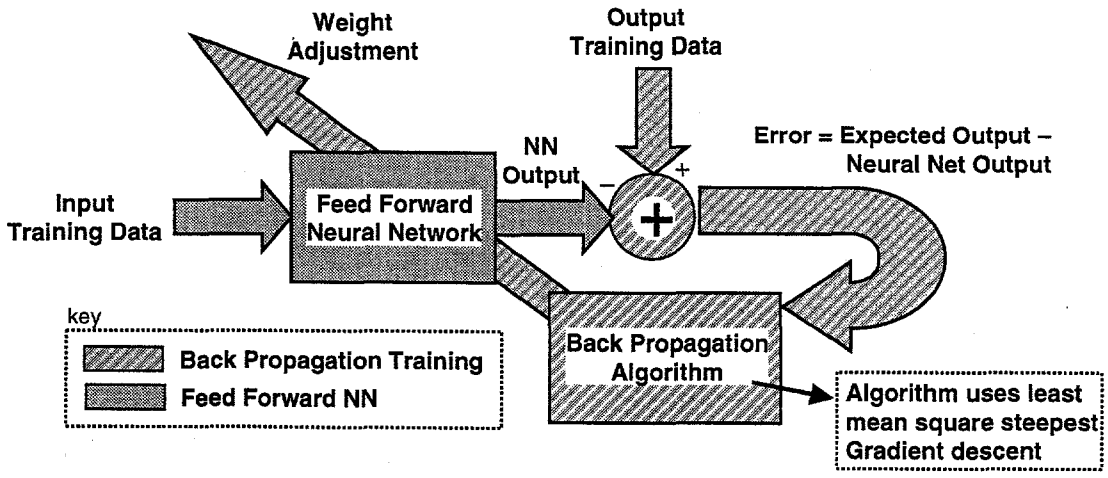

Figure 18. Block diagram of the basic neural network back propagation algorithm

A diagram of the back propagation algorithm is shown in Figure 18. It separates the training section or back propagation algorithm of the neural network, indicated with crosshatching, from

the feed forward algorithm, shown in gray. For a training simulation, both the forward and backward algorithms are run together. The back propagation uses the difference error between the feed forward neural network output and the output training data to adjust the weights in the feed forward section of the network. The process is iterative and can take thousands of iterations to complete a training process. For a more complete description of the algorithm, refer to the references.[10,11] After training is complete only the feed forward section of the algorithm is used.

The study of test cases and the understanding of the problem the neural network is to solve is critical for success. The brute force approach of taking a random sample of input-output data from an unknown system and training a network will be unsuccessful in most cases. The best approach is to develop the most complete set of input-output data possible from the observer's point-of-view, take 80% of that data using a random data selection process, and train the neural network until the desired error is obtained. Specifically, training is complete when the difference between the network output and the training data output reaches the desired minimum. Finally, the network may be tested against the remaining 20% of the data not used for training.

Neural Network Architecture

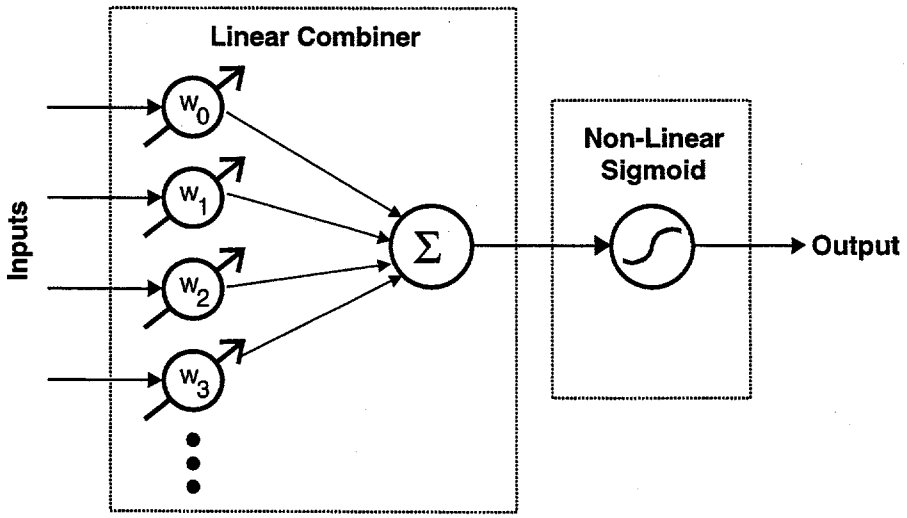

Figure 19. Diagram of a single neuron in a neural network

The way a neural network achieves this non-linear mapping technique is by connecting together an architectural system of its basic building block, the neuron. A single neuron consists of a linear combiner and a non-linear sigmoid function as depicted in Figure 19. The linear combiner is the sum of weighted products of each of the inputs and is linear. The sigmoid function is non-linear and is the inference section of the neuron. One intuitive way to view a single neuron is by examining the linear combiner and the sigmoid sections. A linear combiner is similar to a matched filter. If the input matches the filter weighting, the output of the summer is large resulting in a sigmoid output of 1, meaning the neuron has fired. Conversely, a poor match produces a low level combiner output and a sigmoid output of 0, meaning the neuron has not fired. Any partial match from the input and combiner produces a sigmoid output more linear in nature and is considered to be a partial hit. In this case, the output is considered fuzzy. Applying an input to a single neuron will yield one of 3 results, a strong correlation or match of input, a

weak or no match, or a percentage of a match. Connecting a network of these neurons together and training them on input/output data using the back-propagation algorithm provides a technique for highly non-linear mappings of multi-dimensional systems.

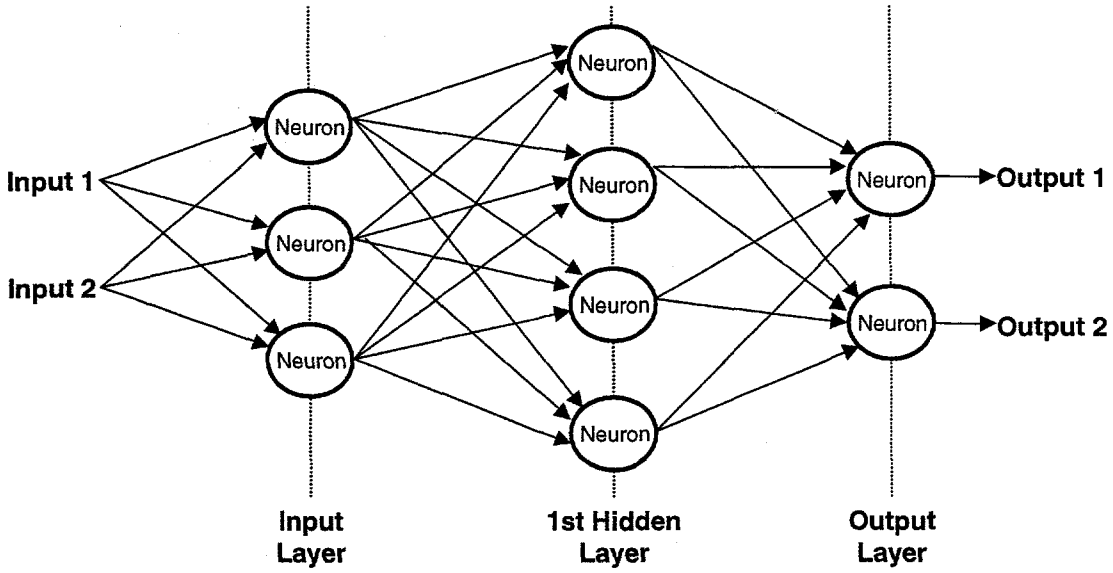

Figure 20. Diagram of a two input / two output neural network with three layers

One of the most common architectures of neural networks is called the 3-layer network, as depicted in Figure 20. There is an input, hidden, and output layer to the network. The actual number of neurons per layer is entirely dependent on the data set and the application. Selection of this is done empirically. The figure shows only a small 2-input, 2-output network with a 4 neuron hidden node network for simplicity. Most applicable networks have more inputs, outputs, and neurons. Also, several layers of hidden nodes are often used. You will notice that every possible output from the previous layer is connected to each neuron input of the next layer. This is a critical part of the network allowing each neuron at the current layer to observe all of the inferences made from the previous layer.

Neural Nets Applied to Train Voltage Prediction

Applying neural networks to a problem as sophisticated as train voltage prediction is complex due to the large number of possible train configurations and the possible variations in power substation configurations. Initially, our collaborators at Yale University, Professor K.S. Narendra and his research group, used a neural network to calculate train voltages based on train locations and power demands. This exercise demonstrated the capability of neural networks to predict train voltages, because the estimated train voltages from a network were shown to be extremely accurate, within ten volts of the voltages predicted by MODRAILS. Unfortunately, this network was not useful in general, but only worked for the case of four trains traveling in each direction. The location and power of each of the eight trains were fed into a sixteen input network, and the eight train voltages were outputs. If more or fewer than four trains were present on each line, the network was not applicable.

In order to generalize the network, it was simplified to two inputs and one output associated with a single train. The location of the train and the total power consumption of all trains in its vicinity were input, and the voltage of the train was output. This neural network was much less accurate than the eight input network, but applied for any number of trains, given a single power system configuration (e.g. all power substations functioning nominally). After these initial experiments, we continued this work at Sandia, continuing on a path to an accurate, generally applicable neural network.

Initially, we attempted to generalize to any power substation configuration, allowing substation locations and ratings to be inputs into a larger neural network. Unfortunately, this approach turned out to be too complex for this stage of the research, and we simplified our goal to demonstrating a network that could accurately predict train voltages in any train control situation for a specific power substation configuration. We focussed our efforts on the critical section of the BART tunnel under the San Francisco Bay. Low voltages can occur here easily, because there is only a power substation at either end of the tunnel with three miles of track in between. As a result, too many trains demanding power near the center of the tunnel can cause severe low voltage conditions. Although this is a smaller scope of the voltage prediction problem, it is a critical area of the BART system. Low voltages occur in the area from time to time, as was demonstrated by the measurements by PG&E,[7] and these affect the reliability of the train motors and are disruptive to passengers. As the trains are scheduled more closely together, these problems will occur much more frequently. Developing a neural net to accurately predict voltages in the tunnel is a significant step.

A neural network requires input/output information for training. As previously discussed, this is a difficult task. The initial set of training data consisted of the location, power consumption, and voltage of trains in the vicinity of the transbay tunnel generated by the simulator. Data was created representing both nominal and off-nominal train behaviors, such as interference and backups. Various numbers of trains were made to interfere in locations throughout the tunnel area, in an effort to simulate the entire space of train control situations. Initially, the effort went into developing small networks with few inputs to speed up the training process. Often, to determine the success of a selected architecture, 1000 training runs were made, requiring computer processing times in the order of days. (A training run, or epoch, consists of a single pass through all of the training data.) Minimizing the size of the training data set minimizes the training time, allowing performance evaluation at a faster rate. Once a simple network was achieved that could roughly approximate voltages, more complexity was added by increasing the number of inputs to include more situational detail, and the network performance was improved.

Our first network that produced successful training had three inputs. The first input was the distance between the train and the KTE substation at the San Francisco side of the tunnel. The other two inputs were given by the total power consumed by all trains within ±2500 feet of the train on either track: (1) power on the track the train was on, and (2) power on the track in the other direction. The two tracks are symmetric with respect to the locations of the substations supplying power, so there is no distinction between trains on one track versus the other. The only important distinction is whether the power demand in the area is on the same track as the train whose voltage is being calculated, or on the opposite track. This is because the high voltage third rails for the two tracks are tied together in very few locations, so voltages on the two tracks can be substantially different.

A data set of 14000 input and output data points were used for the training. Each time the back-propagation algorithm ran through a complete set of the input/output data, called an epoch, a root-mean-square (RMS) error was calculated between actual train volts and neural net predicted train volts. This was the best way to observe how the training was progressing. As long as the general trend of the RMS error was downward, the training was improving. No change in the RMS error meant the network was finished or stuck. Conversely, a general trend upward indicated the weights were becoming unstable and training needed to be terminated.

We developed and trained our neural network computer code using Matlab tools. A typical training session is shown in Figure 21. This was the session for the 3 input neural network. The plot shows the RMS error in train voltage between the neural network predicted voltage and the actual voltage plotted versus training runs or epochs. Notice the slope of the curve decreases at different rates. Initially, the slope drops off very fast, and then it slows down to a more modest rate. At this point, very conservative learning rate parameters are being used for the learning to prevent the steepest descent algorithm from making large jumps from iteration to iteration. Too large of a jump in this early stage may cause the training to go unstable, basically because the forward propagation weights are far from their correct values. After 250 epochs, the learning rate is increased to speed up the learning process. At this point, it is possible to speed up the process because the weights are now more stable and the search for the optimal set of weights can now be more aggressive. Notice again the descent increases initially and then slows down to a more modest descent. Knowing how and when to very the learning rates is purely empirical and based mostly on operator experience. In many cases, the learning rate can be very aggressive at first, and then slow down as the network zeroes in on the optimal set of weights. The best strategy can vary greatly from application to application.

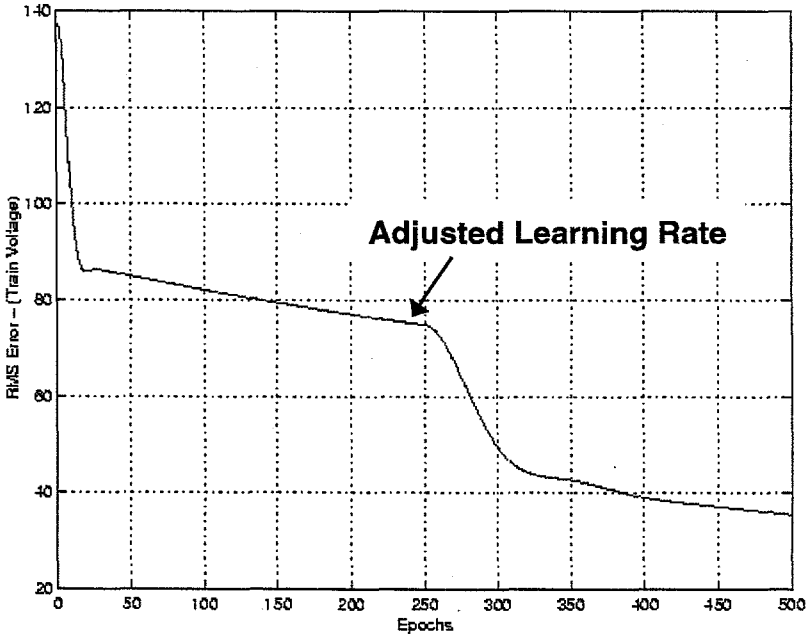

Figure 21. Back propagation training RMS error versus epochs for the 3 input neural network predicting train voltage

Training of the three input neural network was stopped after 4500 epochs with a desirable error of only 42 Volts. This was considered quite good, since nominal train voltages are about 1000V, making the overall network resolution of 4.2%. A data set that was not used for training was used to test the network. The error for the second set of data was 4.6%. At this point, we were not convinced this was the best that could be achieved, but felt that this network had value and that it was worth pursuing the same architecture with more inputs.

A complete neural network simulation was made mapping the entire 4-dimensional space of the system (i.e. 3 inputs plus one output). The result yielded the contour plot shown in Figure 22. The plot is very intuitive for the observer and gives confidence in this architecture. The constant lines of voltage form a partial ellipse with respect to "distance from substation" and "zone power." When the trains consume power near the two power substations, located at 0 and 19300 foot distances, the system is more intolerant to low voltages. Power being consumed in the center of the tunnel is much more sensitive to low voltage conditions. Although, there are 2 power zones, one for each track, the plot was basically indifferent to either zone. In summary, the network computed the obvious. More power consumed further from substations produces low voltage conditions, exactly as one would suspect.

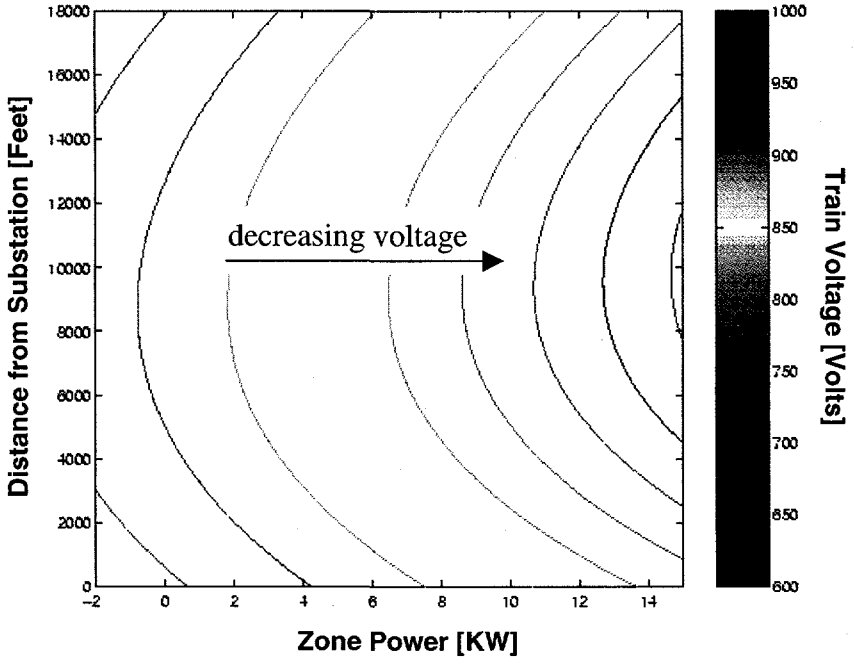

Figure 22. Contour plot of train voltage versus train location and zone power consumption

A more complex neural network architecture of the same type was attempted next with 3 power zones. This was a seven input neural net, with one input for train location and six for power consumption in separate zones near the train. One power zone was the same as before – 5000 feet long and centered on the train location. The remaining two power zones consisted in 5000 foot long track sections on either side of the center zone. We will refer to these as the front

and the aft zones, referring to their location with respect to the KTE substation, not to the direction of travel of the train. Two inputs correspond to each zone, representing the total power on each track consumed by all trains in the zone. Training was terminated after 5000 epochs and the error was down to 26.5 volts. Again, we did not feel that the error was sufficiently low, but it was improving. A new architecture was developed with 5 zones, as depicted in Figure 23. There were 2 forward power zones in front of the train, each 5000 feet in length. The zone nearest the front of the train is denoted the forward zone and the one further away denoted the forward-forward zone. Also, there were 2 aft zones, denoted the aft zone and the aft-aft zone. This network had 11 inputs, 10 for zone powers and one for the distance to the KTE substation. This network produced the best results for predicting train voltage. The error was 16.5 volts rms.

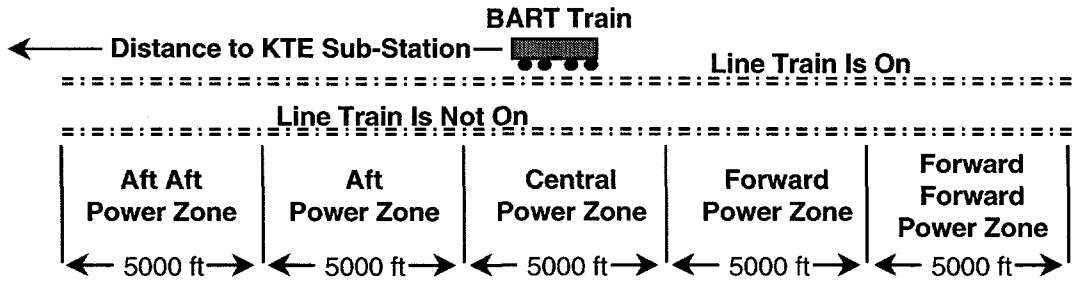

Figure 23. Layout for neural network voltage prediction algorithm

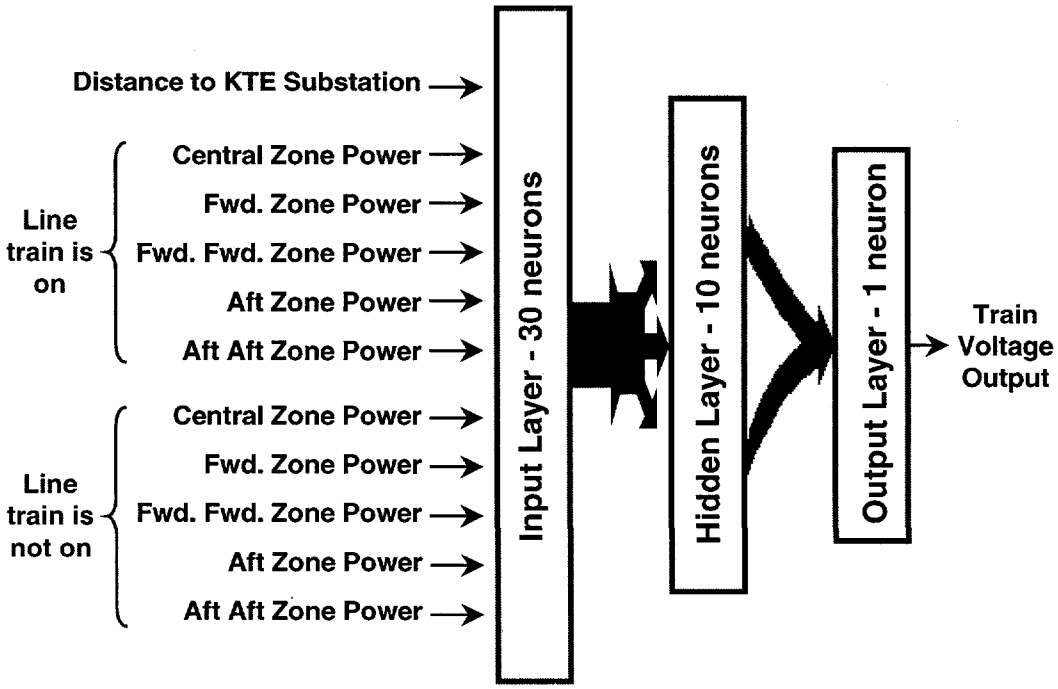

Figure 24. Neural network input/output and layer architecture

A block diagram of the 11 input feed-forward neural network is shown in Figure 24. It shows the 11 inputs going into a layer of 30 neurons, a hidden layer of 10 neurons, and an output of

layer of 1 neuron. As stated above, the network produced an error of 16.5 volts on training data and an error of only 18.5 volts for untrained data runs. The input layer requires an array of 12 by 30 weights. There are 11 inputs to each neuron, one for each data set input, and the extra input is for a bias weight input. Although not mentioned earlier, bias weight inputs are standard practice in neural networks and are used to compensate for any offsets in the data. The second layer or hidden layer uses a 31 by 10 weight array. Again, these include 30 inputs for each neuron from the previous layer's output and one bias weight for each neuron. The output layer had an 11 by 1 weight vector at its input, including 10 weights from the previous node and one bias weight. Basically, the neural network architecture and the 3 weight matrices are all that is needed for implementation of the network.

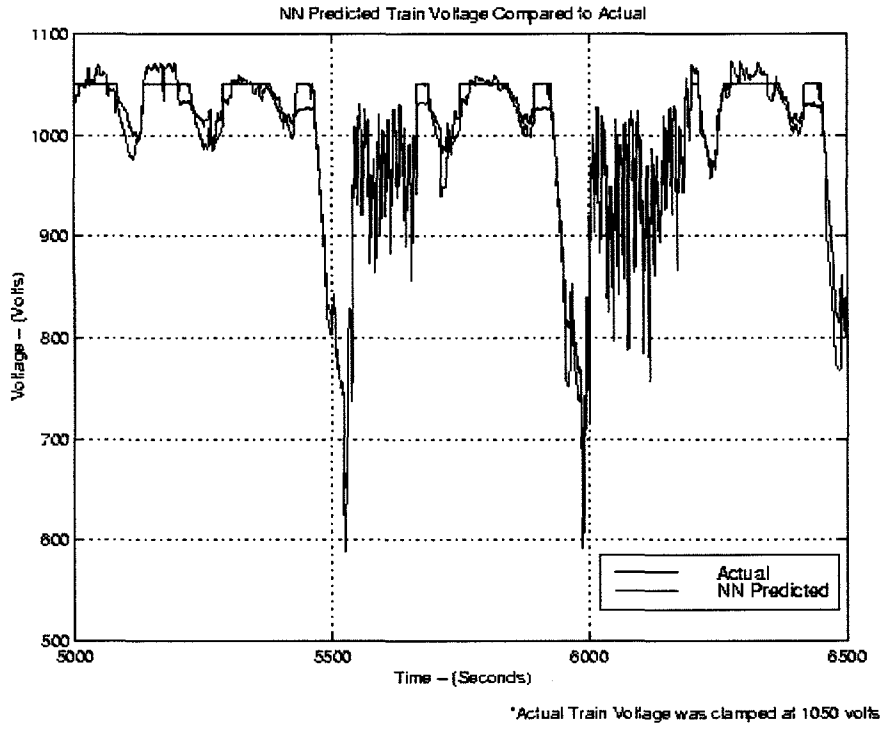

Figure 25. Neural net-predicted train voltage compared to MODRAILS calculation

The train voltage calculated by MODRAILS in the simulator is compared to the voltage predicted by the neural network in Figure 25. All voltages are plotted versus simulation timestep. These voltages correspond to severe low voltages in the tunnel due to the periodic acceleration of interfering trains. Several observations can be made from the plot. Firstly, the dynamics of the train voltage on the line are very erratic and non-linear in nature. Second, the neural net predicted voltage tracks the simulator voltage very well. For the purposes of an algorithm to avoid low voltages, the most important features are that low voltages are accurately predicted when they occur, and that they are not predicted were none exist. Otherwise, misdetections could allow low voltages to occur, which is exactly what we are trying to avoid. In addition, false alarms from the neural net could trigger actions in the control system in response to phantom low voltage conditions. Considering the RMS error is 18 volts for untrained data, making errors of greater than 50 volts is highly unlikely. Here, the 3 sigma rule

of the standard deviation from the mean indicates that these conditions will only occur less than 1% of the time. Observe the voltage conditions below 800 volts in Figure 25. In most cases, the neural net predicted voltage tracks the simulator voltage very closely. A noticeable error occurs where the simulator voltage dips below 600 volts at 5550 seconds. Here, the neural network voltage predicts 650 volts, resulting in about a 3 sigma error. Fortunately, the voltage conditions in this case are so far below 750 volts that the inaccuracy will not effect the control system.

Simulator Testing Results

After implementing the neural network for voltage prediction in the ATAPS simulator, we tested the neural net further as algorithm testing proceeded. We did not calculate RMS error, but rather noted any conditions under which the neural net prediction of low voltages failed. In general, we found the network to be quite robust, and to always detect low voltage events reliably. Low voltage prediction errors of greater than 30 Volts occurred predominantly at times when multiple trains were present in the area during a low voltage event, but not all of the trains experienced low voltages. The network sometimes predicted that an extra train was experiencing the low voltage event, and it occasionally missed one of the affected trains. However, a case was not found were the network missed the primary trains causing the event, and thus the algorithm could depend on its predictions to consistently avoid low voltages.

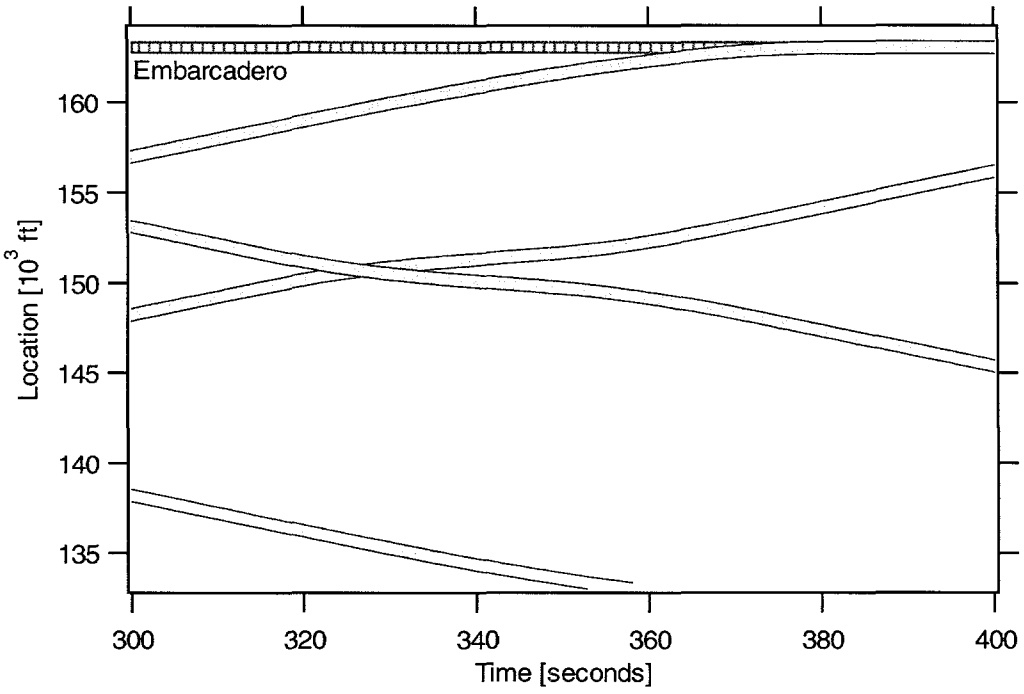

Figure 26. Train trajectories during a SORS reset near MCG

The train control algorithm successfully avoids low train voltages during abnormal train accelerations in the transbay tunnel. Figure 26 shows the train trajectories during a SORS reset event similar to the one discussed regarding Figure 16. Four trains are modeled, each displayed as a gray band denoting the full 700-foot length of the train. Trains traveling towards

Embarcadero station move upward on the graph as time progresses to the right, and those traveling away from the station and toward Oakland West slope downward. For 30 seconds in the middle of the simulation, the two trains near the middle of the tunnel are commanded to reduce speed from 70 to 27 miles-per-hour. The speed command is then returned to 70mph, and both trains attempt to accelerate up to speed. The decrease and subsequent increase in speed of these two trains may be seen as changes in slope in the location plot. The resulting train voltages for the two accelerating trains are shown in Figure 27. Under nominal control, the voltage at both trains drops well below the level where motors on the real cars would shut down.

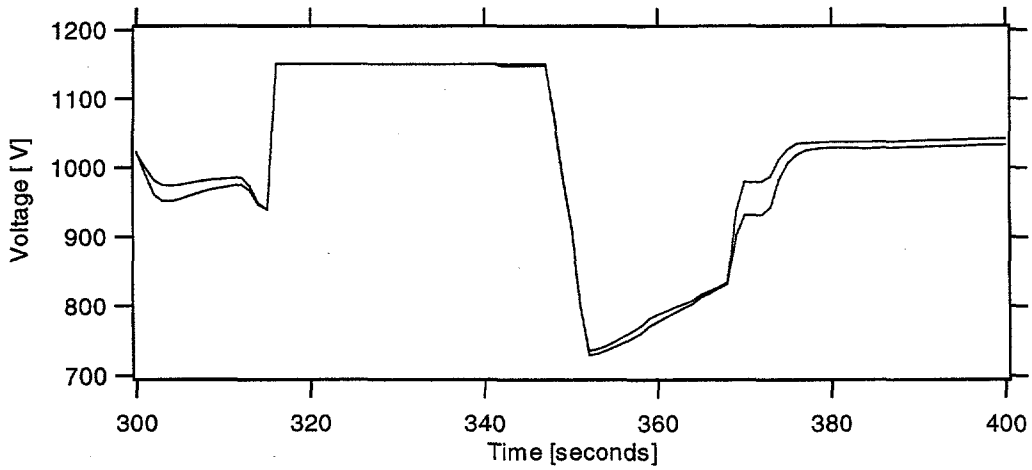

Figure 27. Train voltages during the SORS reset with nominal control

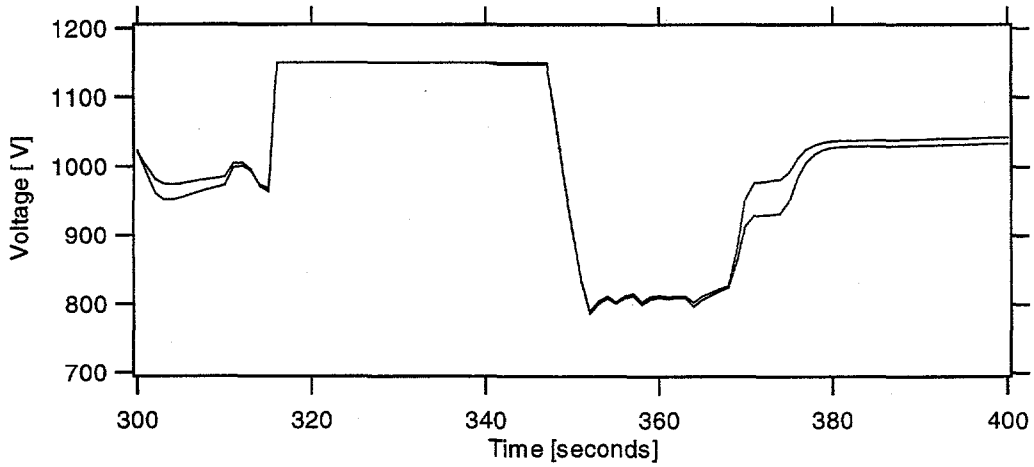

Figure 28. Train voltages during the SORS reset with enhanced control

Enhanced control employing the low-voltage-avoiding algorithm during the same event maintains the train voltages above 800V, as shown in Figure 28. The trip time impact is about one second due to the reduced acceleration after the SORS reset, which is a very small price to pay. The voltage is a bit choppy while the algorithm is active because the logic used is rather simplistic. As described above, during each timestep the algorithm decreases the acceleration command from the nominal value until the predicted worst case voltage is at an acceptable level.

It achieves this by reducing the command in discrete jumps, rather than finding the optimal point. However, this is not a fundamental result of the technique, and could be improved through more sophisticated logic, as will be discussed in the next section.

The train speed (in mph), acceleration (in mph/s), and power demand (in MW) of one of the accelerating trains during the SORS reset event are shown as functions of time in Figure 29. The speed command and acceleration command are also shown as dashed lines on the speed and acceleration axes. The reduction of the train's acceleration command by the enhanced control, and the resulting reduction in power demand to a sustainable level, are apparent in the figure.

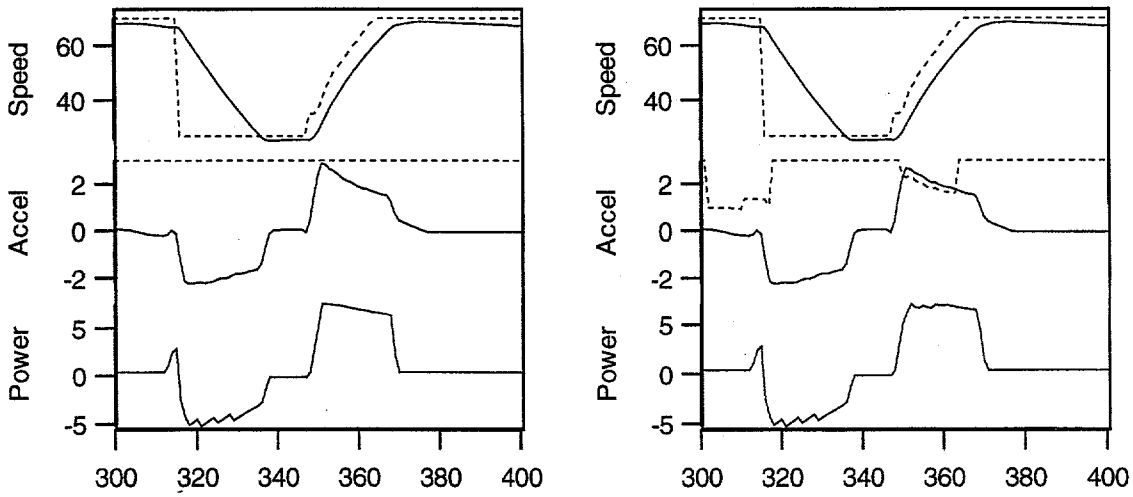

Figure 29. Train speed, acceleration, and power with (left) nominal and (right) enhanced control during the SORS reset event

Figure 30 shows simulated train trajectories during another off-nominal train control situation – a backup in the middle of the transbay tunnel on the DF (M2) line. Only trains on the line experiencing the backup are shown. The other line has six trains dispatched with an 80 second headway, so the trains are minimally spaced. In this simulation, a train stops in the middle of the tunnel until two more trains have stopped behind it and another is approaching. The lead train then accelerates up to speed, and the following trains attempt to follow suit. The resulting trajectories may be seen in the top of the figure. In the middle plot, the acceleration and acceleration command (dashed) are shown in detail. The oscillations in acceleration experienced by the second train are due to interference that will be described in the "Interference During Acceleration" section below. Finally, the bottom of the figure shows the voltages of the trains on both lines. Notice that the low voltages are not confined to the trains on the line with the backup, but also appear on the opposite line, M1. The lowest voltage occurs before the oscillations in acceleration begin, demonstrating that the interference is not the source of the voltage problem. Rather, the problem is again the simultaneous acceleration of multiple trains.

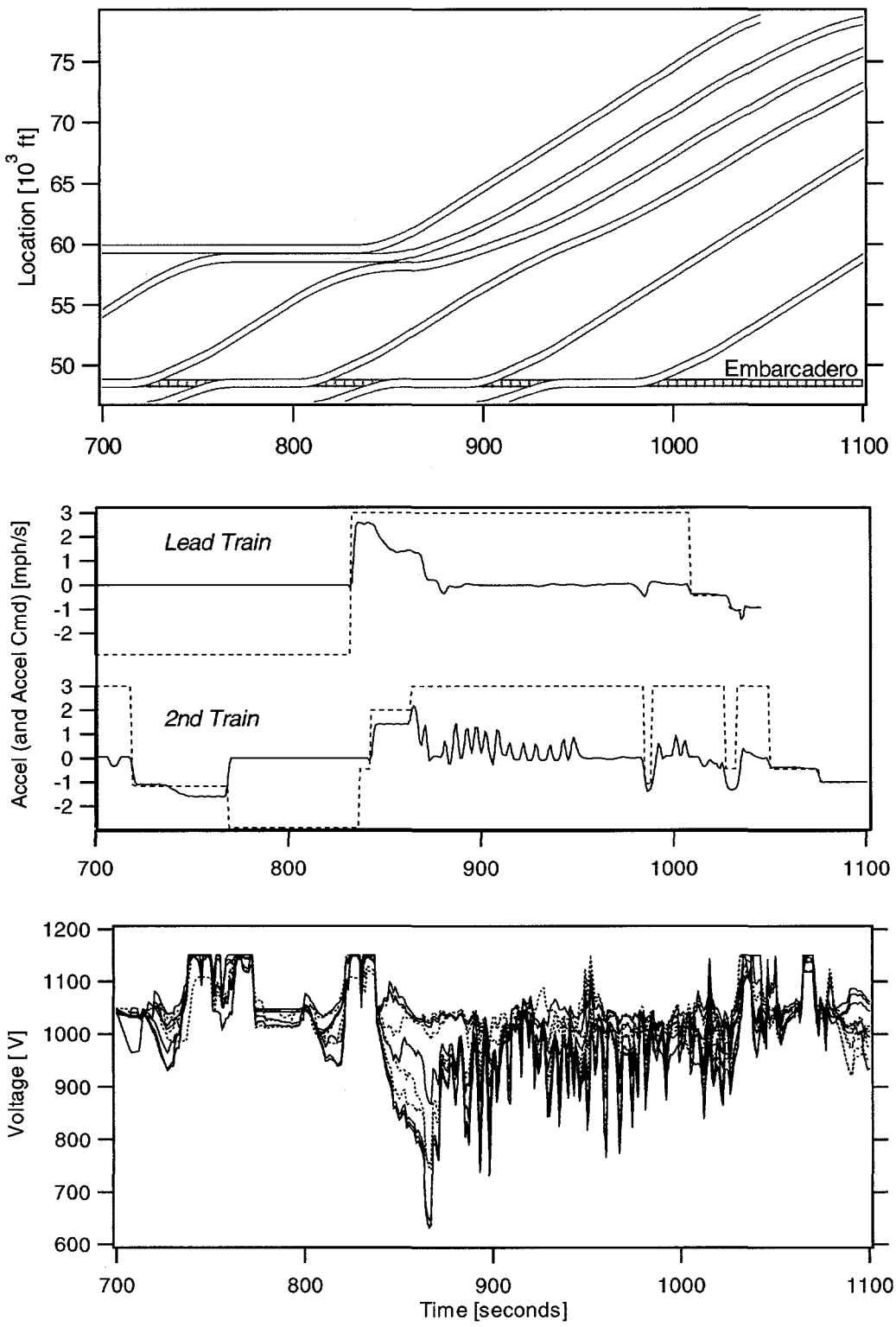

Figure 30. (top) Train locations versus time for nominal control during a backup in the transbay tunnel, (middle) acceleration of the first two trains, and (bottom) train voltages, shown with M1 trains dotted and M2 solid.

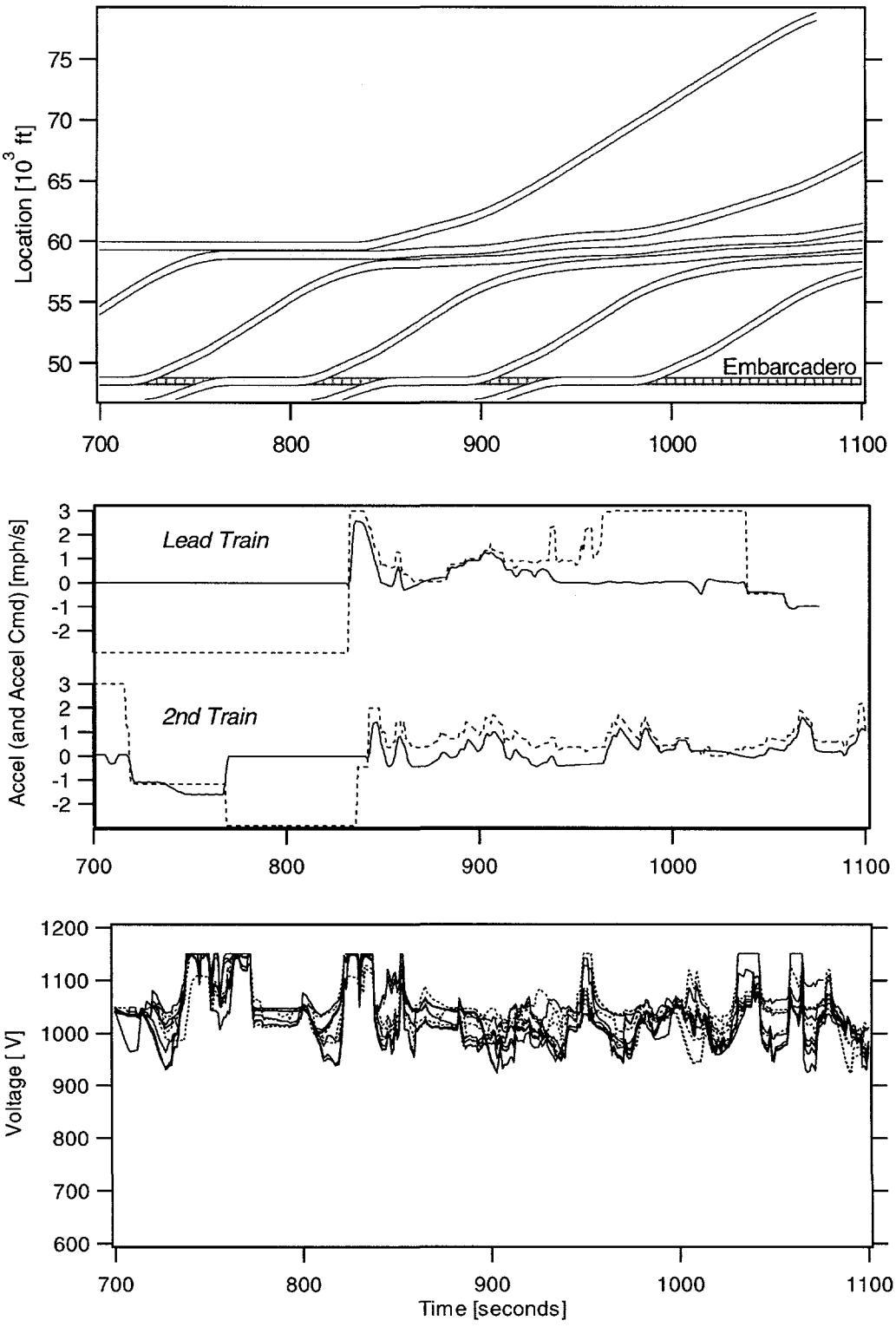

Figure 31. Same as Figure 30, but with enhanced control.

Figure 31 shows the same simulated backup situation, but with the enhanced control algorithm operating. The decreased acceleration commands are clearly sufficient to prevent low voltages from occurring. As a result of the reduced accelerations, the lead train is delayed by 30 seconds in its arrival at the next station, and the headways of the following two trains are increased by 67 seconds and 17 seconds, respectively. The remaining train headways are all close to 80 seconds, as in the nominal case. The trains on the opposite line are not delayed.

It is important to note that the nominal trip times would actually be somewhat longer than that predicted by the model, since train behavior in the model is not impacted by the low voltages. In reality, the extended period of low voltage in the tunnel would cause the trains to operate with reduced performance, and motor shutdowns would occur on the lowest voltages. Feedback from the power model to the train control model would need to be added to the simulator in order to accurately represent these effects and predict the correct triptime. Thus, the increase in trip time caused by the algorithm logic will not be quite as severe as represented by the present simulator.

If the algorithm were updated to incorporate a train prioritization based on each train's delay, as suggested in the algorithm description, then the lead train in this simulation would be given the highest priority and therefore the most power. This improvement to the algorithm could allow the lead delayed train to arrive in the next station much more quickly – sooner than the nominal control system, in fact, since the train could be given full power with sufficient voltage to achieve full acceleration. The following trains would share the remaining available power.

Finally, the algorithm was tested against simulations of nominal train behavior with 97-second headways in order to determine the impact of the algorithm during normal operations. This is an important issue, since the algorithm is designed to be operating at all times. A simulation of the worst-case-voltage phasing of trains on the two lines is shown in Figure 32. The worst case here is defined as the relative train locations that generate the lowest voltage that could be found empirically. The trains on the two lines were phased in order to maximize power demand in the transbay tunnel. As may be seen in the figure, the minimum voltage occurs when two trains are approaching the middle of the tunnel, simultaneously climbing the hill located there (see Figure 11). In addition, the headway between trains was varied a bit around the nominal 97 seconds in order to maximize power demand at the moment of the minimum voltage. The train locations versus time are shown in the top of the figure, and the train voltages are shown on the bottom. The plots look quite similar for both nominal and enhanced control, with a minimum train voltage of 877V under nominal control, and 882V for enhanced control. Triptime is increased a maximum of 2 seconds by the enhanced control, and is not impacted at all for most trains.

In summary, it seems practicable to run this algorithm at all times. During nominal train control situations, the impact on train behavior will be minimal. During off-nominal conditions that cause low voltages, on the other hand, the algorithm is capable of preventing low voltages and their associated motor cut-outs, rough rides, and reliability problems.

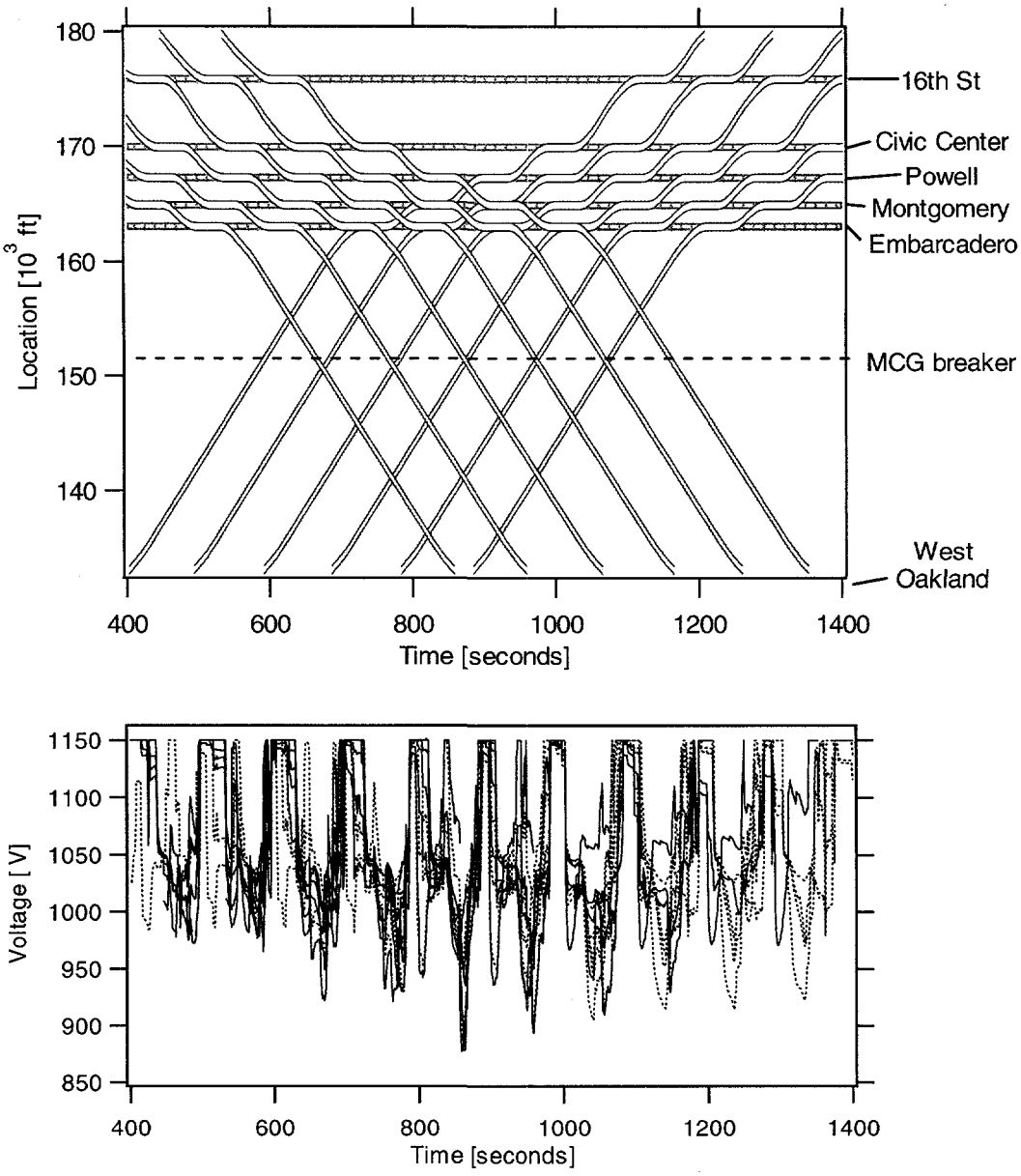

Figure 32. (top) Train locations versus time during the worst case phasing of 97 second headway trains, and (bottom) corresponding train voltages, shown with M1 trains dotted and M2 solid.

Suggested Upgrades

Distribute Train Power Consumption

Some of the jumps in acceleration command exhibited by the current version of the algorithm are caused by a simplifying assumption that we made in the neural net inputs that could easily be corrected. The neural net used here treated each train as consuming its full power at the location of the head of the train. The neural net sums power consumption in track zones, so the power

consumption of a train jumps discontinuously from zone to zone as the train moves. If, on the other hand, the power is distributed over the 700-foot length of the train, then the power would gradually shift from one zone to the next as the train moves. For example, if a 700-foot-long 10-car train were located 350 feet from the zone boundary, straddling two power zones, then half of the train power would be summed into each zone. If it were 100 feet from the zone boundary, then 6/7 of the power would be summed into one zone, and 1/7 into the other. This would help to reduce discontinuities in the predicted voltage during the simulation, which lead to discontinuities in the suggested acceleration commands. In addition, the location of the central zone should be defined by the location of the middle of the train whose voltage is being calculated, rather than the head of that train. These modifications should definitely be implemented in any future voltage-predicting neural nets such as those described here.

Prioritize Trains with Respect to Delay

As was mentioned previously, in addition to voltage sag severity and power demand, the decrease in acceleration command of a train could be linked to the "priority" of each train. In order to accomplish this, the schedule will need to be incorporated into the algorithm, so that it is possible to judge how much each train is delayed. Once this is known, the power may be preferentially consumed by trains that are more behind schedule. In a backup situation caused by a delay, the lead delayed train will be the farthest from on-time, and will be given the most power. This solution would take better advantage of the wayside control architecture provided by the AATC to more optimally control the trains.

Include More Locations and Power System Configurations

Although the algorithm discussed here employs a neural network specific to the tunnel location with a normal power system, we expect the algorithm will handle train behaviors that cause low voltages at any location and during outages. For implementation in other areas or with other traction power configurations, the neural net will either need to be generalized to work in any power system configuration, or a separate network will need to be trained for each location and power condition. A generalized algorithm could be quite challenging to develop. It would need to have additional inputs indicating the locations and power ratings of the two power substations bracketing the train location. We were not able to confirm whether this type of network can be successfully trained to handle any power configuration, but it is definitely worth pursuing.

If such a network could be successfully implemented, then that single network could be used in any location on the system and during outages by simply modifying the substation data inputs. In the worst case, if this can not be accomplished, then the same network design described here could be retrained for each location and power outage condition of concern, and the correct network could be selected as appropriate to the situation. For example, the non-vital computer could have a set of networks applying to the various track regions under its control, and another set of networks for each outage condition. Creation of these networks will simply require generating a set of training data for each location and outage condition, and training networks using the methodology described above.

Unless information regarding the state of the power system is directly available to the wayside station computers, a network specific to an outage condition would need to be switched on by a

controller at Central in the event of a power outage. This could be controlled simply by sending a single number corresponding to the desired neural net configuration to the affected station computer.

Continue Network Training with Real-Time Data

To improve the accuracy of the neural nets as the system is exercised in the future, train voltages could be measured on the system real-time, and this data could be used to continue to train and refine the neural net to improve its accuracy, and thus to increase the effectiveness of the algorithm over time. The neural net in training could be periodically validated and then substituted for the previous version in the control algorithm.

Improve Algorithm Efficiency and Smoothness

Assuming the simulator correctly reproduces the response of a train to a reduced acceleration command, the power does not decrease until the command drops below the maximum achievable acceleration at a given speed. At high train speeds, the maximum current limit of the motors limits the acceleration rate. For example, at 60mph and full power, the maximum acceleration rate is below 2mph/s. The train acceleration rate is unaffected unless the acceleration command drops below this rate. Thus, in order to decrease the power consumption and thereby improve the voltage of a train going this fast, the acceleration command must be stepped down more than this. The current simplistic logic is unaware of this fact, and simply steps down a bit at a time until the voltage improves. Instead, it would be better to immediately step down enough to impact the power consumption on the first step. This may be possible with a better understanding of the behavior of the on-board proportional controller as a function of acceleration command and speed.

In addition, the algorithm logic steps down acceleration commands discretely until the minimum expected voltage is acceptable. Instead, it may be worthwhile to explore the possibility of more optimal control. Once a neural network exists to predict train voltage, the gradient of this function could be used to determine the impact of reduced power consumption on voltage. The optimal power distribution between the participating trains could then be found. This control strategy is non-trivial to develop, but would lead to smoother and more optimal control.

The algorithm also assumes a top speed of 80mph is possible and could go into effect at any time. This means even a train in the tunnel going 70mph (in Performance Level 2) is assumed to be capable of drawing full power and is given a reduced acceleration command to avoid the voltage sag which could result. If it is known that the maximum speed of 70mph will never be violated, then trains going 70mph would only require power for speed maintaining at most, and fewer potential voltage sags would be predicted.

Smooth Interfered Headway Operation

Interference occurs when a train runs so closely to the train in front of it that it is eventually forced to brake in order to maintain a sufficient following distance. The new generation of CBTC control systems will allow trains to run close to the minimum safe following distance, so any slight changes to a train's trajectory or station dwell time may lead to interference. Like cars on a densely packed highway, trains can accelerate and break repeatedly in response to each other's movements, wasting energy, abusing the motors, and causing an uncomfortable ride for the passengers.

In 1996, BART investigated the frequency of short delays on its system, and found that the system experiences about 55 delays of 3.5 minutes or longer every day, 35% of which are greater than 5 minutes.[12] When the system is scheduled in the future to run with trains separated by less than 2-minute headways, these delays could lead to backups behind the delayed trains, leading to severe interference between the affected trains. In addition, there are over 100 delays between 2 and 3.5 minutes daily, which will also lead to significant interference. Thus, the "off-nominal" condition of interference will become more the rule than the exception as the scheduled headways become shorter to meet additional demand.

The current generation of fixed-block control systems can already exhibit interference behavior; however, the severity of such events is moderated by the infrequent changes in train speed commands. In fixed block systems such as BART's ATC system, trains are only given new speed commands when they or the train in front of them cross from one control block into another, and these blocks are at least several hundred feet apart. The new moving-block control systems, on the other hand, will be capable of changing train speed commands more than once every second, and therefore could exhibit much more severe interference events. An example of such a problem was shown in Figure 30, where a train accelerating immediately behind another train experienced rapid and repeated cycling of its acceleration.

Although the baseline AATC system would certainly have extreme difficulty handling such events, the system also provides the means to correct the problem and to handle interference in a smooth and efficient manner. Enhancements to the AATC system may be designed to avoid oscillatory acceleration/brake cycles, and, in general, to smooth the trajectories of closely-following trains. Due to the frequency and severity of these events, such improvements to the control system will be important in a future of short headway operations. In addition to improving the comfort of passengers, removing unnecessary acceleration cycles will have beneficial impacts on system reliability and the energy bill, just as your driving habits impact the lifetime of your car and your gasoline bill.

The ultimate goal for smoothing train interference is a single enhanced control algorithm that will handle any situation involving interference. There are many causes of interference, including delays in stations, stops between stations, trains accelerating close to one another, trains traveling close together in the midst of frequent station stops, and trains traveling below the commanded track speed limit due to equipment failure. However, all of these events lead to similar problems, which could be handled by one intelligent algorithm. The creation of such an

algorithm requires the solution to a complex control optimization problem, so that trains will always be given commands that lead to smooth trajectories. Rather than tackling this global problem initially, we have pursued specific algorithms to handle focussed problems of this type. Once some intuition has been gained through this pursuit, we believe that it will be possible in the future to solve the more general problem.

The following three sections, "Interference During Acceleration," "Interference Near Station Stops," and "Interference During Delay Recovery," describe the interference-related algorithms that we have developed to solve focussed problems. Implementation of these algorithms should improve passenger comfort without significantly increasing trip time, since they avoid interfered train behaviors which also take time. Thus, these algorithms are designed to run in the AATC non-vital computer at all times to smooth train trajectories.

Interference During Acceleration

When two trains that are close together accelerate up to speed, the stopping distance of the rear train increases due to its increased momentum, and oscillations in acceleration can result. A trajectory exhibiting interference during acceleration is shown in Figure 33 under "nominal control." This phenomenon is rooted in the calculation of Worst Case Stopping Distance (WCSD, see "AATC Commands and Stopping Distance"), which is proportional to the velocity squared. If the following train accelerates at the same rate as the lead train, its expanding stopping distance quickly overtakes the lead train, and it must stop accelerating. Once its following distance increases sufficiently, acceleration resumes, and the process is repeated. In simulation, this cycle was repeated approximately every five seconds throughout the acceleration profile. We have developed an enhanced control algorithm to prevent this situation by continuously adjusting the acceleration rate of the following train to maintain the required following distance, as shown in the figure under "enhanced control." This algorithm prevents unnecessary mode changes from braking to acceleration, thereby reducing wear-and-tear on the motors and improving reliability, and at the same time reduces the likelihood of low voltages, saves energy, and produces a smoother, more comfortable ride.

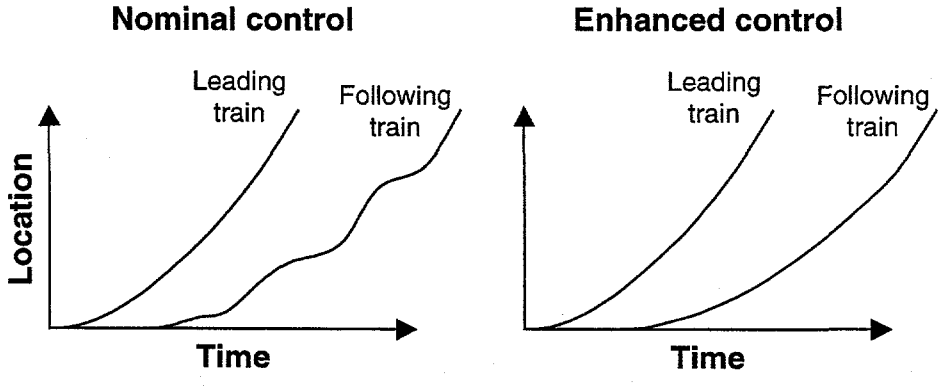

Figure 33. Sketch of algorithm to smooth interfered operations

The algorithm was also designed to prevent braking due to hills. As a train travels, its WCSD will increase if the average grade in front of the train becomes more down-sloped, because it takes more distance to stop on downhills. An example of the hills on the BART M-line may be seen in the elevation and grade plot shown in Figure 34. The lowest elevations here correspond to the track in the transbay tunnel. In this area, there are quite substantial track grades, reaching 4% in one location, or an elevation gain of 4 feet for every 100 feet traveled. The grades shown are averaged over the 700-foot length of a train, which is the effective grade under a train at a given location. If a train is traveling as close as possible to the train in front of it, then it will have to brake as it approaches a downhill track section in order to leave additional following distance for the longer WCSD. It is a simple matter to maintain a sufficient following distance at all times so that braking is never required, and it is possible with care to follow at the minimum allowable distance that avoids such braking.

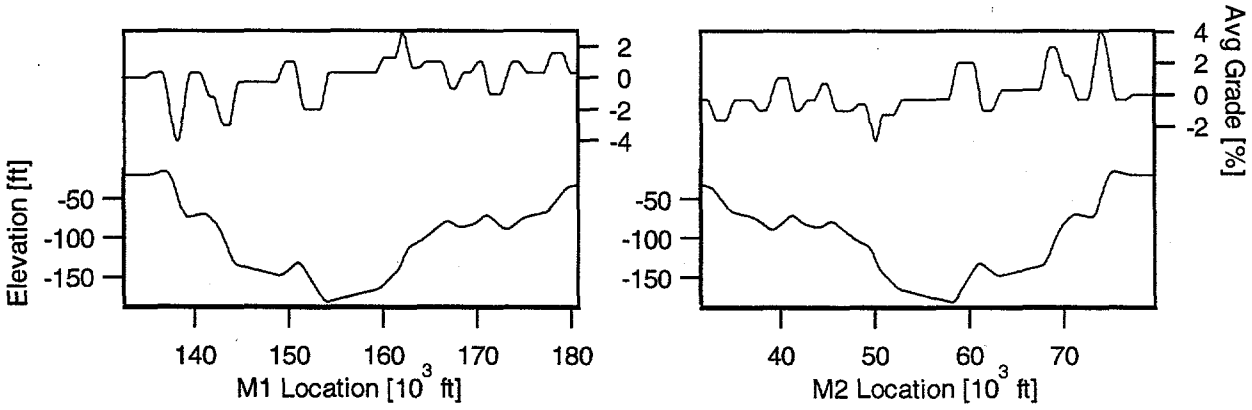

Figure 34. Elevation and grade between Oakland and Embarcadero stations on the M1 (FD) and M2 (DF) lines.

Finally, we made some attempt to handle the situation where a train is traveling below the track speed limit, and the train behind it repeatedly catches up, brakes, falls behind, and then accelerates back to the speed limit. The implemented algorithm will prevent this problem if the leading slow train is being commanded by the AATC system with the lower speed command. If, however, the train is traveling slowly due to an equipment failure or a misinterpretation of the AATC speed command by the on-board logic, then repeated accelerations will still result.

Algorithm Approach

We have developed an enhanced control algorithm that maintains a following distance for each train that is greater than the largest stopping distance expected between a train's location and the next station stop. The WCSD is first pre-calculated as a function of location and speed. The "Maximum WCSD" is then calculated based on this function by taking the maximum of the WCSD at each speed between each location and the next station location. Figure 35 shows the stopping distance and maximum stopping distance between Oakland West and 24th Street stations as functions of location and speed. If the following distance of a train, or the spacing between a train and the train it is following, is maintained at all times above this maximum stopping distance for its location and speed, then braking due to interference will not occur. In

order to maintain this separation, when a train is accelerating close behind another accelerating train, its acceleration rate is calculated such that its following distance increases to match the increase in its maximum stopping distance. If a train is following another train that is not accelerating but is traveling slowly, then the rear train slows to match speeds with the lead train when it is following at its maximum stopping distance.

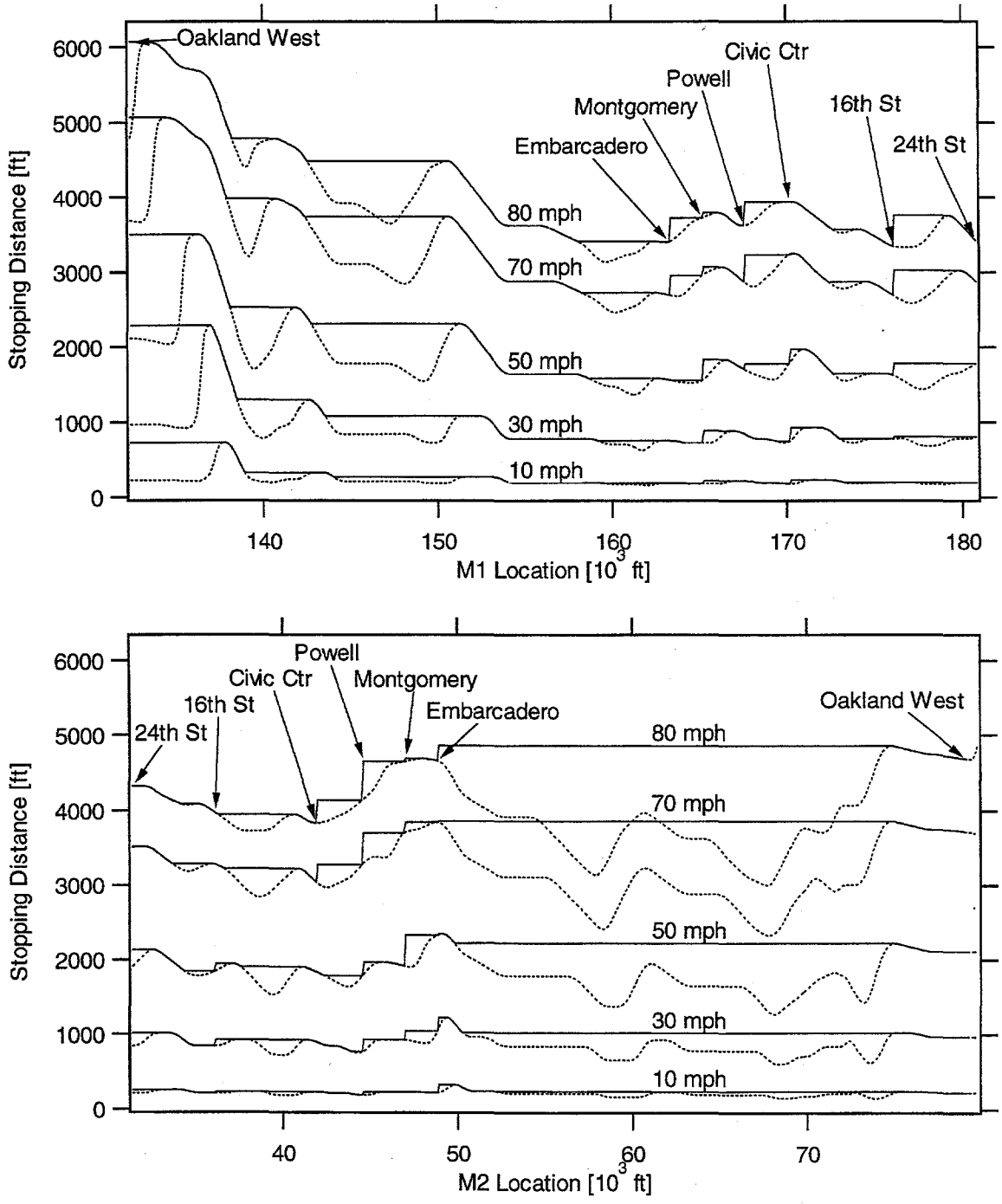

Figure 35. Plot of the WCSD (dashed) and the Maximum WCSD (solid) versus location on the M1 (FD) and M2 (DF) lines

Before detailing the algorithm logic, we will first briefly discuss the critical importance of correct accounting for time delays in the calculation of reduced acceleration commands. If these acceleration commands are generated using the current train state information in the non-vital computer, the rear train in a pair of accelerating trains will still accelerate too fast and cyclical interference will still occur. It is important to project ahead to predict the locations and speeds of the trains at the time that the acceleration command will take effect in order to calculate the correct reduced acceleration rate. The non-vital computer has information that is a bit old, and its suggested commands take a while to go into effect. In the simulator, commands suggested by the non-vital controller do not go into effect for three command cycles, or 1.5 seconds, after the train state information is collected from the train. The enhanced control algorithm projects the train trajectories forward by 1.5 seconds so that, when the train finally acts on the suggested acceleration command, it is in approximately the state that was predicted and the command successfully avoids interference.

For each train, the algorithm logic begins by using the location, speed, and acceleration of the train to project the expected location and speed after the 1.5 second delay time. It then looks for the train in front of, or leading, this train, and likewise projects its location and speed forward by the delay time. The projected separation between the trains is compared to the Maximum WCSD (MaxWCSD) of the rear train for its projected location and a speed command of 10mph above its projected speed. This is the nominal speed command for an accelerating train. If the train separation is larger than this MaxWCSD, then the algorithm does nothing. If the separation is insufficient, however, then the speed and acceleration commands for the rear train are reduced. The speed command is first reduced to 6mph above the projected speed. If this is greater than the speed command of the lead train, then the speed command is reduced further to match unless this would induce braking. A new MaxWCSD, and the derivatives of this function with respect to location and speed, are calculated next based on the reduced speed command. The desired reduced acceleration command is then calculated based on these values so that the distance separating the two trains will increase as fast as the MaxWCSD increases. Finally, a small linear feedback term is added to the calculated reduced acceleration rate in order to compensate for any difference between the actual and the desired optimal following distance. If the train separation is more than desired, then the calculated acceleration command is increased linearly with this error. If the separation is too small, the command is likewise decreased. This term compensates for any accumulated small errors in the acceleration command calculation.

A constraint placed on the action of this algorithm is that it will never command a train to brake. If the calculated reduced acceleration rate ever becomes negative, indicating a need to decelerate, then the algorithm suggests a zero acceleration command, which corresponds to a coasting command. Likewise, a speed command less than the train speed is never suggested. If necessary, the vital controller will eventually command braking to slow the following train. This does occur if a train catches up to a slow train and must brake to match speeds. The combination of vital braking followed by non-vital speed command reduction works well, as shown below.

Simulator Testing Results

We focussed on the M2 line in the transbay tunnel between Embarcadero and Oakland West for algorithm testing, since this area allows plenty of space for acceleration to full speed, and for

multiple train interference simulations. This line also demonstrates an extreme case of avoidable braking due to grade changes. We first tested the algorithm logic by simulating two train accelerating up to speed after stopping in various locations in the tunnel. Figure 36 shows simulation runs with and without the enhanced algorithm operating after two trains stop just outside of Embarcadero station. The trajectory and power consumption of the second train is shown as a function of location as the two trains accelerate up to full speed.

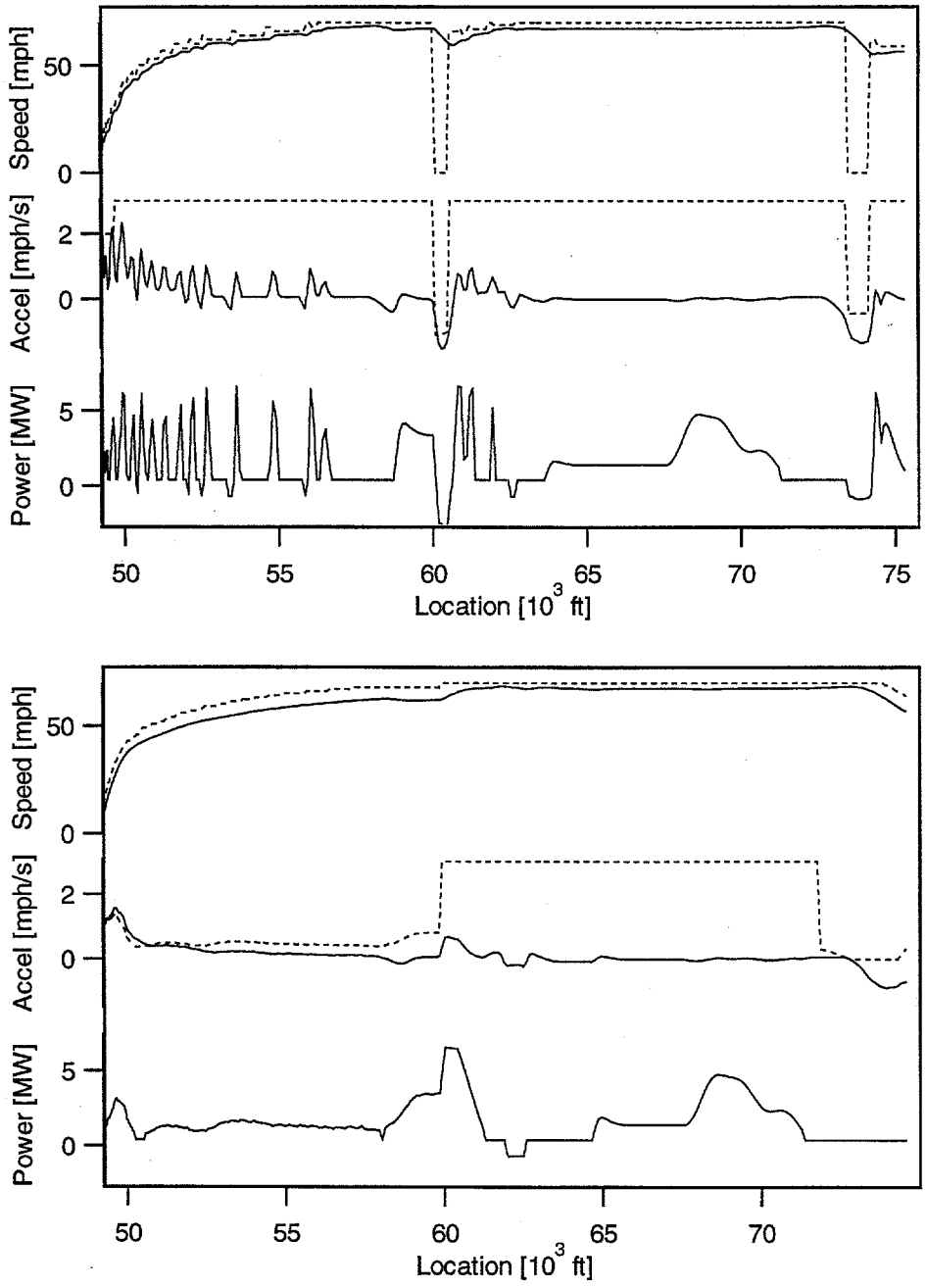

Figure 36. Two trains accelerate from a stop near Embarcadero, with (top) nominal and (bottom) enhanced control. Speed, speed command (dashed), acceleration, acceleration command (dashed), and power of the second train are shown.

In addition to smoothing the trajectory of the second train during the accelerating phase, the algorithm also prevents the two subsequent occasions when the nominal control system commands the train to brake. These events occur in locations where downhill sections of track cause the WCSD to increase, necessitating a longer following distance. Using the MaxWCSD function in place of WCSD in the algorithm prevents this behavior by always enforcing the largest following distance that will be required until the next station stop.

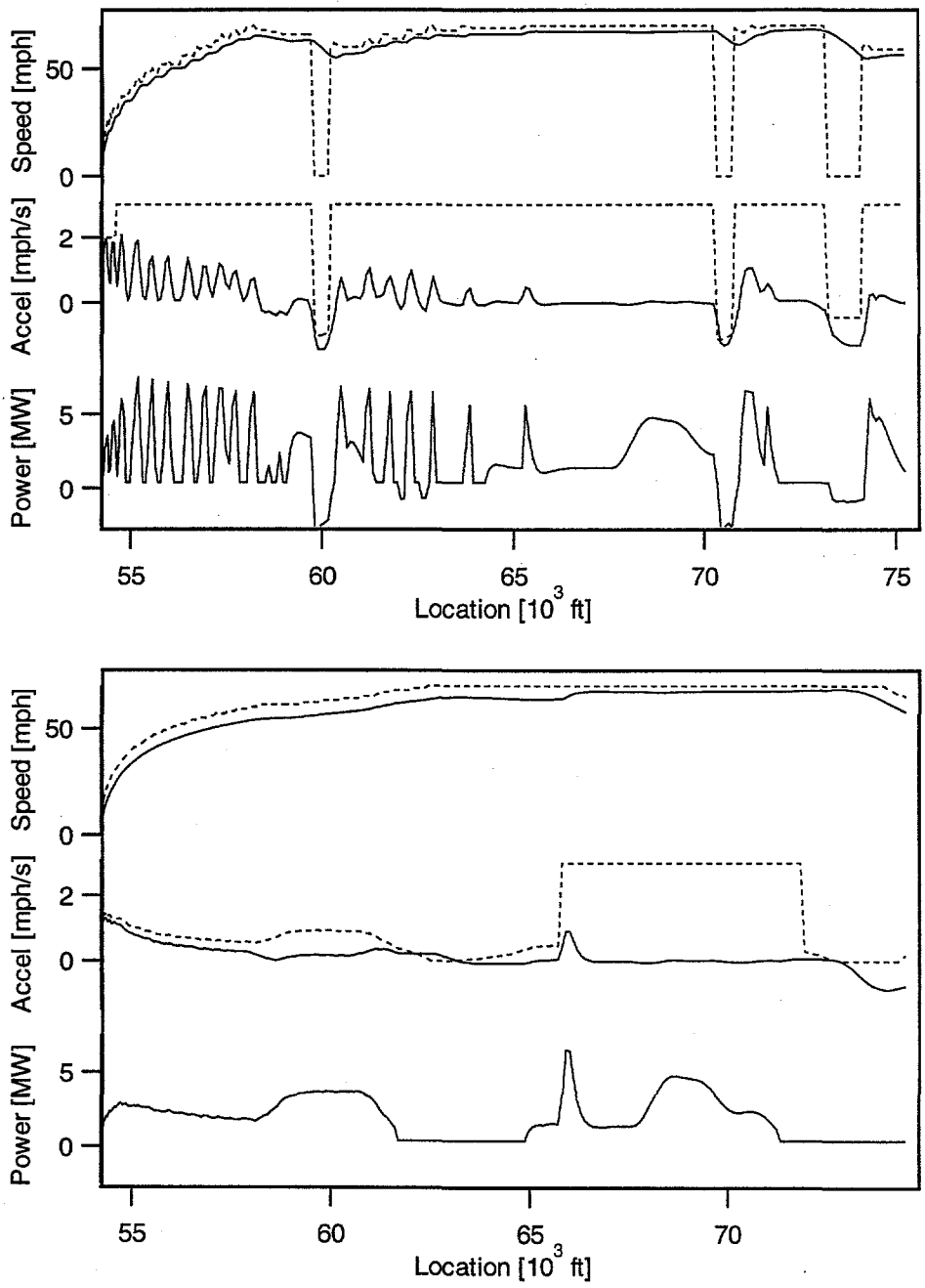

Figure 37. Same as the last figure, but after a stop 5000 feet short of the middle of the transbay tunnel

Figure 37 shows a similar pair of simulation runs, again with and without the algorithm operating. In this case the trains were stopped halfway between Embarcadero and the middle of the tunnel. Under nominal control, the first braking event due to a downhill section occurs before the second train has completed accelerating up to speed, so the train trajectory is even more complex. Under enhanced control, the situation is again much improved.

The clear difference between acceleration command and acceleration in the enhanced control plot is caused by the presence of hills. On a steep uphill, as near 60000 feet on the M2 line, the train's accelerometer is fooled by an extra apparent acceleration forward due to the hill, as was discussed in "AATC Commands and Stopping Distance." The on-board controller therefore believes that the train is accelerating at the command acceleration, even though it is actually accelerating more slowly. Likewise, on downhills, the train accelerates faster than it is commanded. The version of the enhanced control algorithm tested in the simulator does not compensate for this error, but rather depends on the following-distance feedback term to correct for it. As a result, the train is never following quite at the optimal distance, and on sufficiently steep hills the algorithm can fail. The final implemented version of this control algorithm should incorporate a term to correct the acceleration command directly for grade, as will be discussed in "Suggested Upgrades" below.

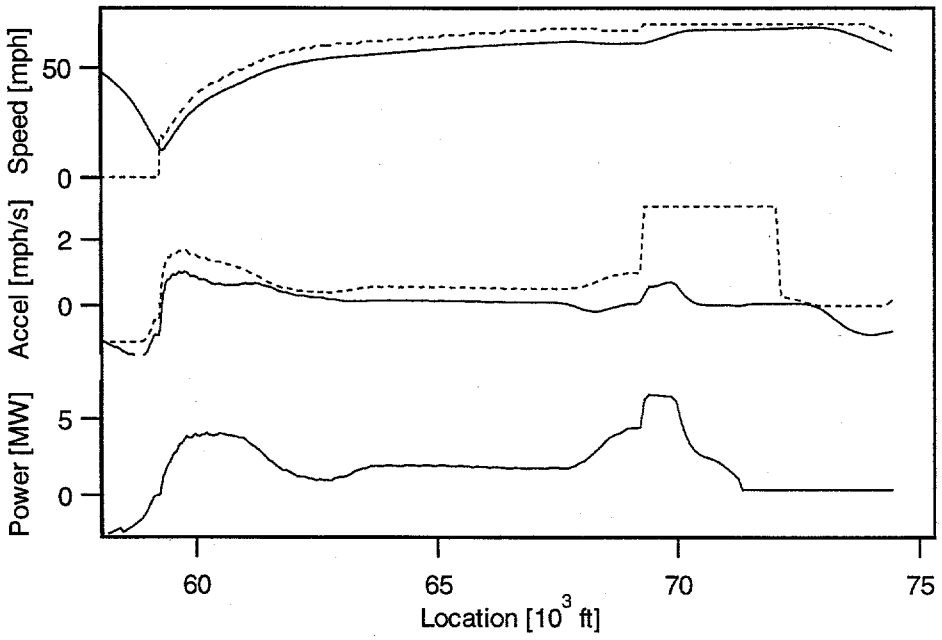

Figure 38. A train brakes but does not stop behind a stopped train in the tunnel, and then both trains accelerate. The speed, acceleration, and power of the second train under enhanced control are shown.

As demonstrated in Figure 38, this algorithm functions during acceleration whether or not the interfering train comes to a complete stop. In this simulation, a train was stopped near the middle of the transbay tunnel, and the train behind slowed but did not stop before the lead train began accelerating up to speed. The following train, shown in the figure, smoothly switches from braking commanded by the vital station computer, into acceleration at the reduced

acceleration rate commanded by the non-vital computer. This smooth transition occurs because the non-vital computer sends the suggested reduced acceleration rate even while the train is braking behind the stopped train, working under the assumption that the lead train could begin to accelerate at any time. Thus, when the vital computer does begin to move the lead train, it is already armed with a suggested reduced acceleration rate for the rear train, which it received from the non-vital computer in the last timestep.

After thorough testing with two trains, the simulator was tested with simulations of multiple trains. Five trains were stopped in the middle of the tunnel in order to explore multiple train interactions and low voltage impacts. Figure 39 shows a simulation run with nominal control of a line of five trains separated by 78 second headways, the minimum headway out of Embarcadero. The lead train stops for 100 seconds in the middle of the tunnel, forcing the second train to stop behind it, and the third train to slow down significantly. The figure shows the speeds, speed commands (dashed), accelerations, and acceleration commands (dashed) of the leading four trains. All of the interfering trains display oscillations in their acceleration profiles. The second and third trains also experience braking events that occur after the acceleration profile is complete. This is again due to interference from grade-induced increases in stopping distance. The fourth train eventually catches up to the interfering group of trains, slows down behind the third train, accelerates with interference, and then exhibits the same hill-induced braking just at the end of the simulation run. The same situation mitigated by enhanced control is shown in Figure 40. The algorithm smoothes out the acceleration profiles, and also removes the post-acceleration interference.

The voltage of the second train in these simulations is also shown in the figures, demonstrating the low voltage that occurs while multiple trains accelerate in the middle of the tunnel. The enhanced control algorithm improved the voltage situation, preventing the severe low voltage that occurred under nominal control. In other similar simulation runs, the voltage dropped well below 750V when several trains coincidentally accelerated at the same time under nominal control. Again, the voltage did not drop to unacceptable levels with the algorithm operating. The modified acceleration profiles are apparently sufficient to reduce the power demands and therefore to avoid low voltages, even for five trains stopped in the middle of the transbay tunnel. In addition, the control algorithm reduced energy consumption by as much as 15% by reducing the rail losses associated with the low voltages, and by preventing energy from being wasted by trains accelerating with no place to go.

Figure 41 shows a similar simulation, but in this case the lead train stops 5000 feet short of the middle of the tunnel for 150 seconds, stopping two trains behind it, and subsequently accelerates to a reduced speed command of 30mph. The figure shows the speeds and speed commands of the trains with and without enhanced control. Under nominal control, the following trains exhibit the usual oscillations during acceleration, but also attempt repeatedly to accelerate to the full speed of 70mph, resulting in continued severe interference all the way to the next station. Under enhanced control, the trajectories are much improved.

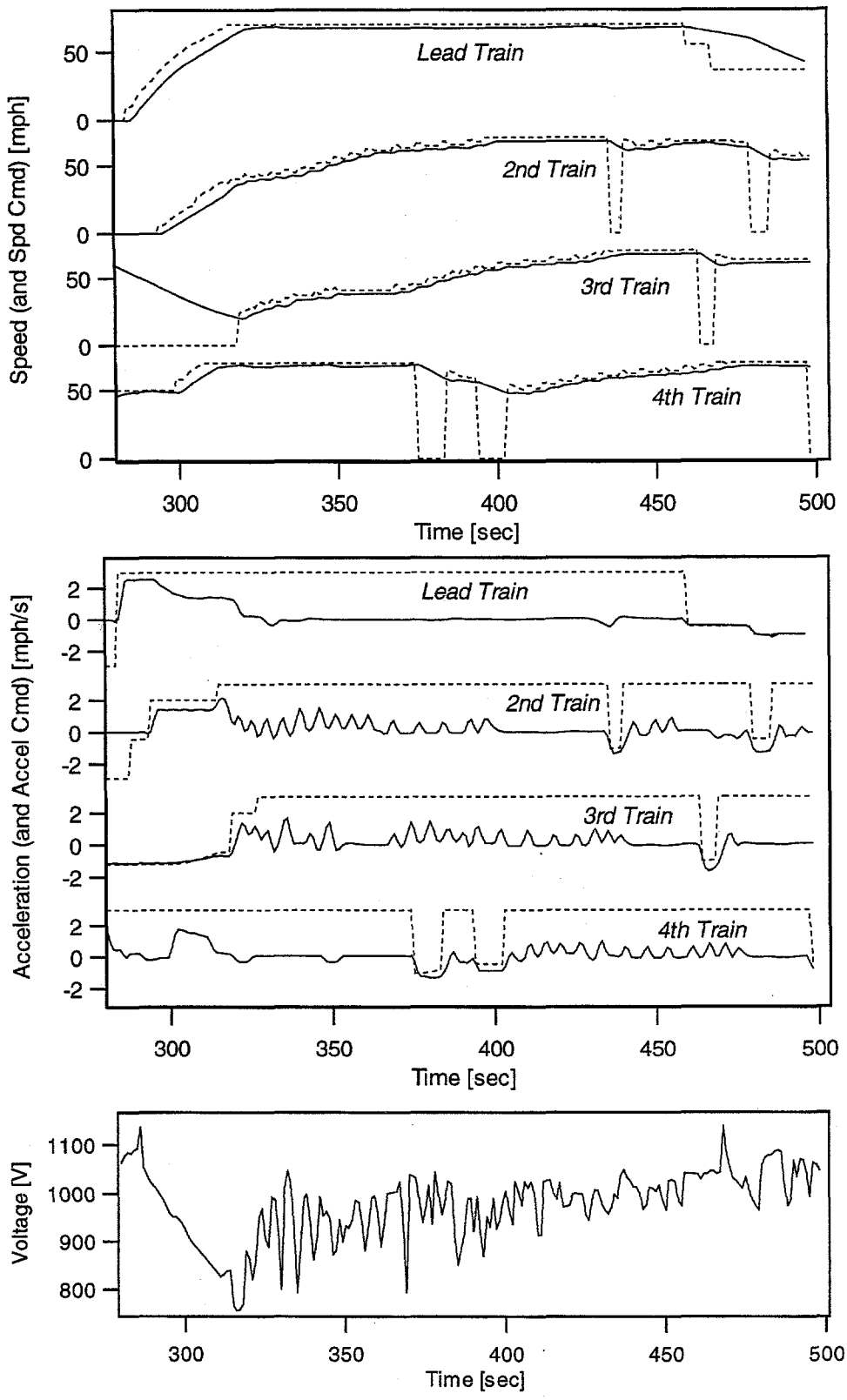

Figure 39. Five trains accelerate in the transbay tunnel under nominal control. Train speeds and accelerations and the voltage of the second train are shown.

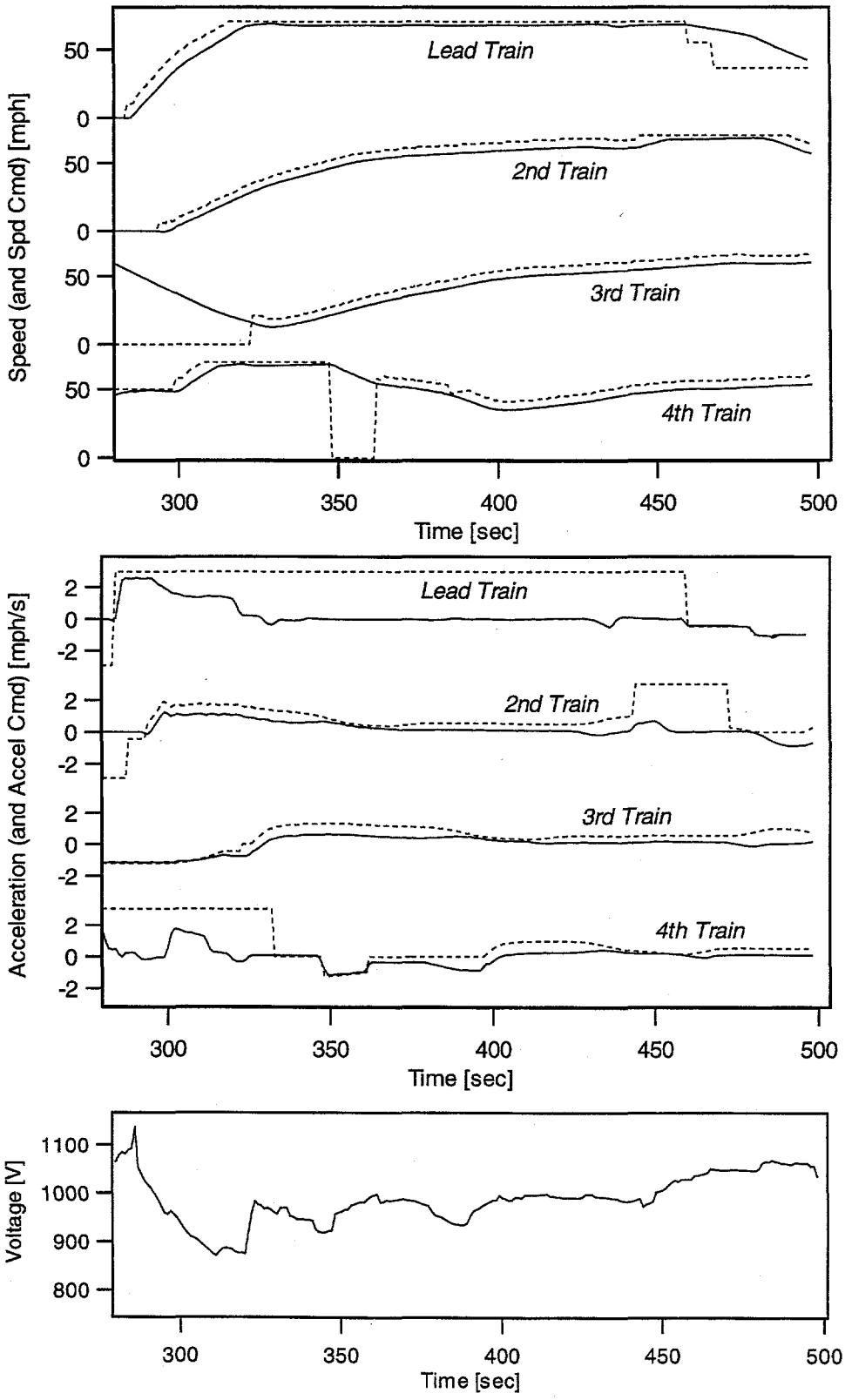

Figure 40. Same as last figure, but for enhanced control.

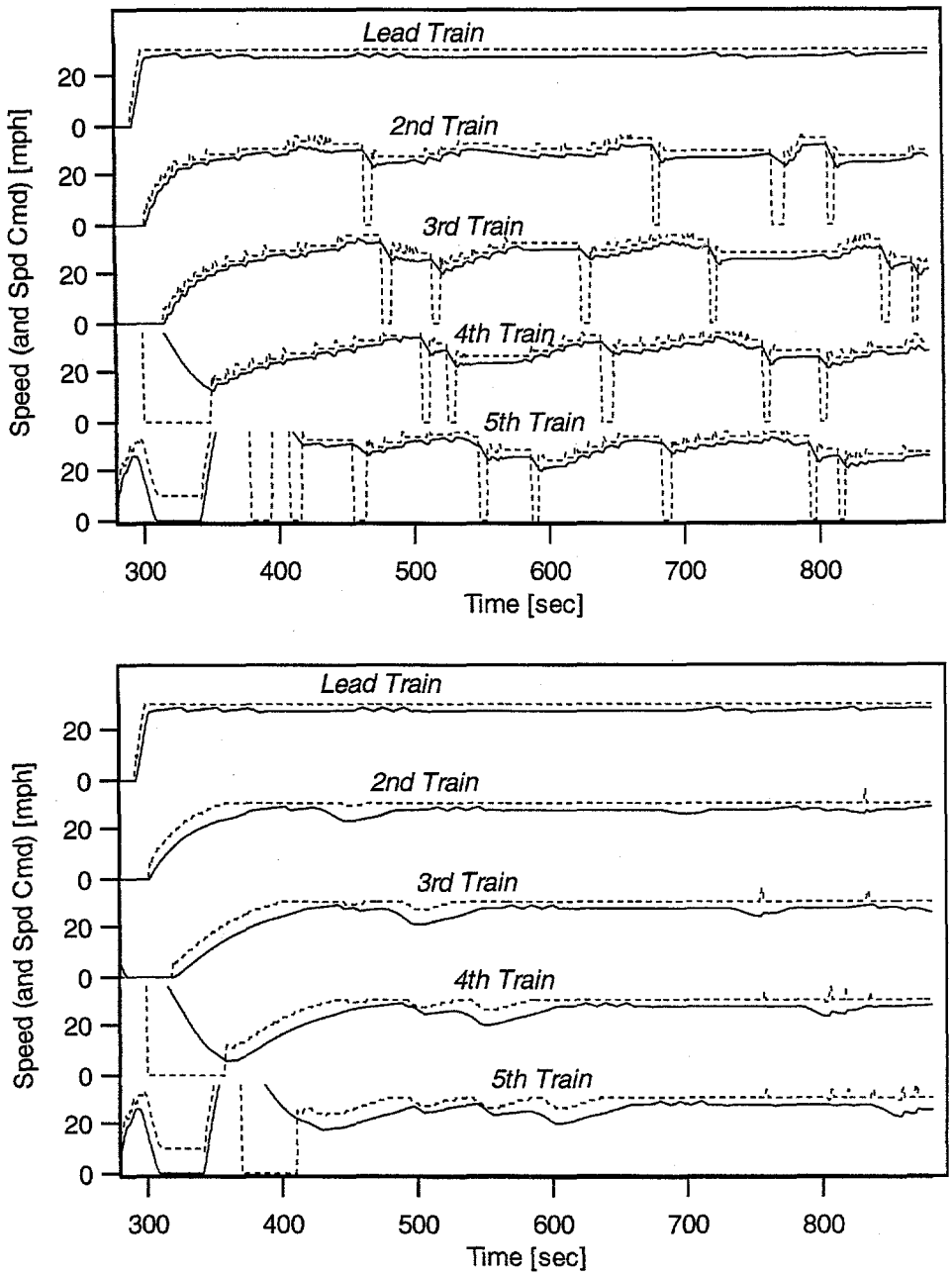

Figure 41. Five trains accelerate behind a slow train in the transbay tunnel under (top) nominal and (bottom) enhanced control. Train speeds and speed commands (dashed) are shown.

The slow oscillations in train speed, which become more evident with each successive train, are caused by the acceleration error due to grade. When the second train reaches the hill in the tunnel, the acceleration command (not shown) is too small on the uphill section, and the speed drops by a few miles-per-hour. The third and following trains must also slow down to avoid catching up. The same problem occurs for each train as it reaches the hill, and so each train shows one more dip in speed than the last. Enhanced control will not exhibit this behavior if the

grade error is properly accounted for in the calculation of acceleration command in future versions of this algorithm, as discussed below.

Suggested Upgrades

Explicitly Compensate for Systematic Accelerometer Error

In its current implementation, this enhanced control algorithm suggests an acceleration command for each train which corresponds exactly to the desired train acceleration rate. However, the on-board controller uses an accelerometer to determine the train acceleration, and is therefore always in error due to the grade under the train. On uphills, trains accelerate more slowly than the algorithm expects and fall behind their desired trajectory, and likewise they accelerate too quickly on downhills. The small position-error feedback term in the calculated acceleration command compensates somewhat for the grade error, but results in trains which are always displaced from their desired following distance, by as much as a few hundred feet on steep hills. In one instance on a steep uphill, a train fell so far behind that the algorithm shut down due to a lack of interference, causing the train to revert to full acceleration momentarily. Repeatedly during algorithm testing, it became evident that the desired acceleration command calculated by this algorithm needed to be increased or decreased to compensate for this systematic accelerometer error in order for the algorithm to function dependably.

Unfortunately, the simulator does not currently provide track grade data to the Non-vital Algorithm Simulator code, so that the algorithm could not predict acceleration due to grade. In the real system (and in future versions of the simulator) the algorithms should be given access to track grade data. To compensate the acceleration command for grade, after the desired acceleration rate has been calculated from the projected train states and the stopping distance, and after the linear position feedback term has been added, the acceleration due to grade should be added on. Acceleration due to grade is (32 ft/sec^2 * grade). The average grade under the train at the time when the command goes into effect should be used, not the average grade over the stopping distance. For example, for a train on a 2% uphill, $32ft/sec^2 * 0.02 = 0.44mph/s$ should be added to the desired acceleration rate. Without this direct compensation, in the coded version of the algorithm, a train must fall behind its desired following distance by 175 ft in order for the linear feedback term to increase the acceleration command by this much. (Stronger linear feedback itself causes undesirable speed oscillations.) If grade were compensated directly, then the linear feedback term would only need to compensate for small errors in train separation, as we originally intended.

Include Interference Due to Slow Trains

A second, but less simple improvement to the algorithm would be to avoid interference due to slow trains. This algorithm will help to avoid interference when a train with a high speed command follows a train with a smaller speed command, by using the speed command of the lead train to command the following train. Such a case is shown was Figure 41. However, if a train travels more slowly than its commanded speed, interference will again result. This may occur, for example, if there is a problem with the on-board speed measurement, or a failure of the on-board control logic.

This problem is non-trivial to solve. We tried quickly to handle this case by setting the speed command of the following train equal to the speed of the lead train plus 2mph, rather than using its speed command. However, this approach led to equally unpleasant results. With a line of five trains, any small fluctuations in the speed of the lead train due to hills led to increasing oscillations in the speed of each subsequent train. It may be possible to implement a more sophisticated version of this, for example by limiting the number of available speed commands in order to stabilize the system, or perhaps by time averaging train speeds to prevent rapid fluctuations.

Interference Near Station Stops

In addition to the interference due to changes in stopping distance described above, avoidable acceleration cycles can also result when trains are close together in a region with closely-spaced station stops. If a train is stopped in a station, and another train approaches, the second train will begin to brake early to stop short of the station. If the train in the station then pulls away, the rear train may accelerate briefly before stopping for the station. Thus, the rear train goes through a brake-accelerate-brake sequence. The baseline control system will command a train to mode-change from braking to propulsion, even if it will begin its station stop braking profile a second later. This sequence may then be repeated at each station along the line if the stations are close together, as is the case in downtown San Francisco.

As for the interference-during-acceleration algorithm, the benefits of this control strategy are smoother service, improved passenger comfort, reduced wear and tear on the motors, and reduced energy costs. In other words, the passengers aren't being thrown back and forth as the train changes modes, the motors aren't changing modes as frequently, and energy isn't being wasted by propulsion/braking cycles. There is little cost to overall trip-time in this case, because interference will slow down the rear train with or without the algorithm running.

Algorithm Approach

A relatively simple algorithm is capable of removing this type of interference. If a train is braking before a station because another train is stopped there, and it is then freed to accelerate by the stopped train pulling out of the station, then it should only accelerate if it is necessary in order to reach the station or if it will add excessively to trip time to remain in braking. Otherwise, it should continue braking at the minimum rate (0.4mph/s) until the station stop.

The implemented algorithm incorporates one adjustable parameter that corresponds to the additional estimated trip time that the algorithm deems acceptable at each station stop. If a train remains in braking rather than using the nominal acceleration command, the trip time to the station stop will increase. The algorithm estimates this increase in trip time, compares it to the parameter, and commands continued braking only if the trip time increase is smaller. It is possible to adjust how often this algorithm goes into effect by adjusting the size of this maximum additional time parameter. For example, if the parameter is set to a small number, like 5 seconds, then the algorithm will only prevent egregious momentary accelerations. If it is made larger, like 15 seconds, then the algorithm will prevent accelerations that last longer. In an area

with frequent station stops, as in the case demonstrated below, there is no reason not to use the larger parameter setting. After interference through several successive station stops, the trip time for the line will be approximately the same with or without the algorithm operating.

Whenever continued braking is deemed appropriate, the delay time between non-vital command generation and implementation requires that the non-vital algorithm begin suggesting a minimum brake rate for the train as soon as it begins to brake due to interference. While the vital computer is commanding a faster brake rate, the suggested minimum brake rate will be ignored. If and when the vital computer then decides that the station is clear and the train can go back into propulsion, it will replace its acceleration command with the more conservative non-vital suggestion from the previous time-step, and acceleration will be prevented.

Simulator Testing Results

The following set of figures show a single simulation run in which four trains were dispatched on each line. The dispatch headways were 70, 120, and 120 seconds, so that the first two trains would interfere through the stations, and the following trains could potentially catch up. The algorithm run shown used a maximum additional time parameter setting of 15 seconds. Only the first three trains exhibited any interference.

Figure 42 shows the train locations versus time for the four trains on each line, and shows the station locations as bars. Again, each train is shown as a gray area representing the full length of the train. The nominal and enhanced control trajectories are overlaid, with nominal control in black and enhanced control underneath in gray. A small change in train location is apparent for the second and third trains, but the trajectory change caused by the algorithm is clearly quite small. The change in trip time due to the action of the algorithm is less than 3 seconds at the end of the run.

The following two figures show the speeds and speed commands (dashed) of the first three trains versus time with and without enhanced control. Figure 43 shows the trains on the M1 line, and Figure 44 shows the trains on the M2 line. When the vital controller commands braking due to interference, the speed command goes to zero. This still occurs with the algorithm running; however, the trains do not subsequently return to propulsion in most cases. For a more direct comparison, the bottom plot on each figure shows speed as a function of location for the second train with and without enhanced control.

With all 8 trains running (4 on each line), the energy savings of running this algorithm was calculated to be only half a percent over the course of the simulation. Thus, energy savings do not play a large role in this case. The primary benefits of this algorithm are improved reliability and passenger comfort due a reduced number of brake/acceleration cycles. It is simply smarter, smoother service.

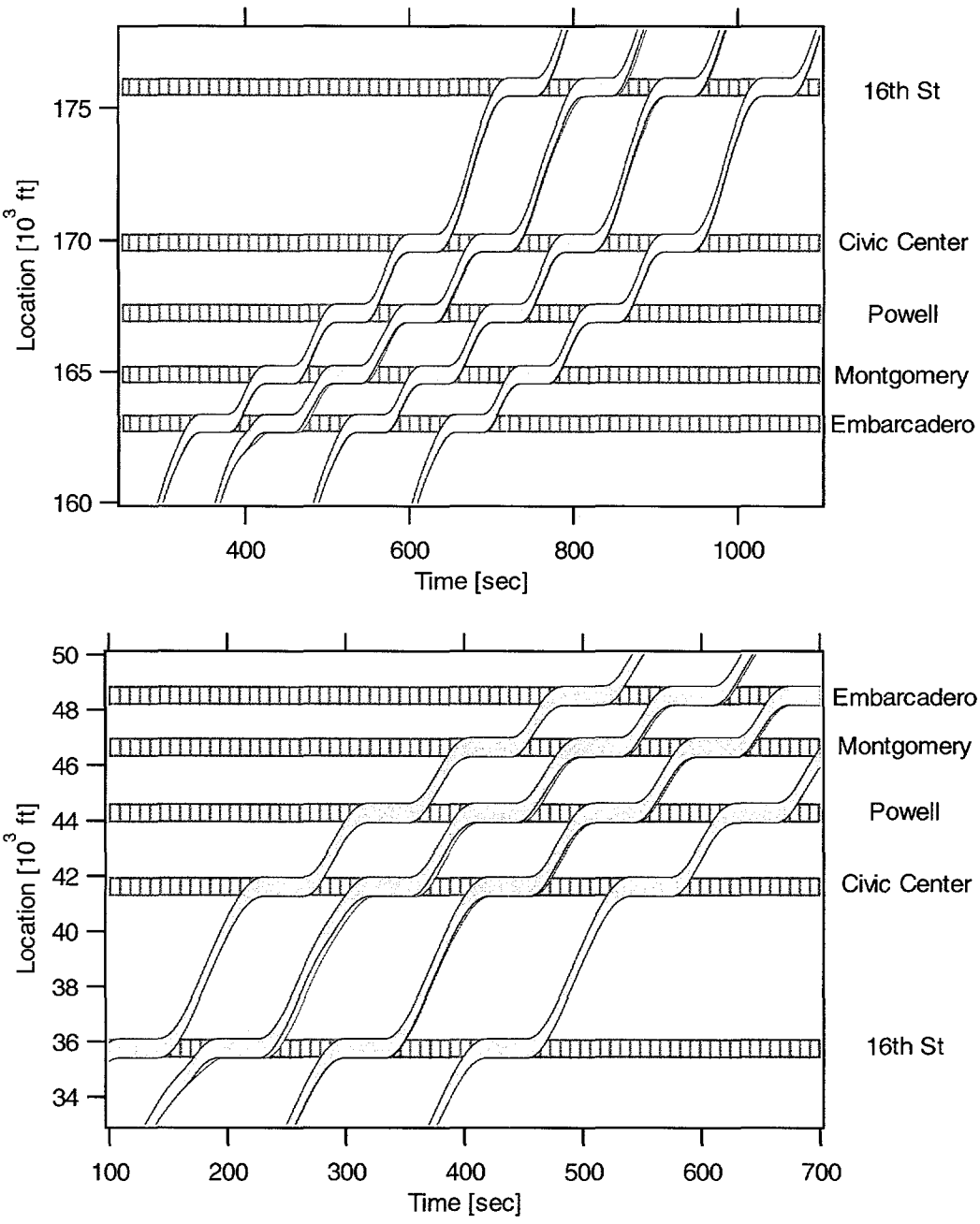

Figure 42. Trajectories of four trains interfering on (top) the M1 line and (bottom) the M2 line through downtown San Francisco.

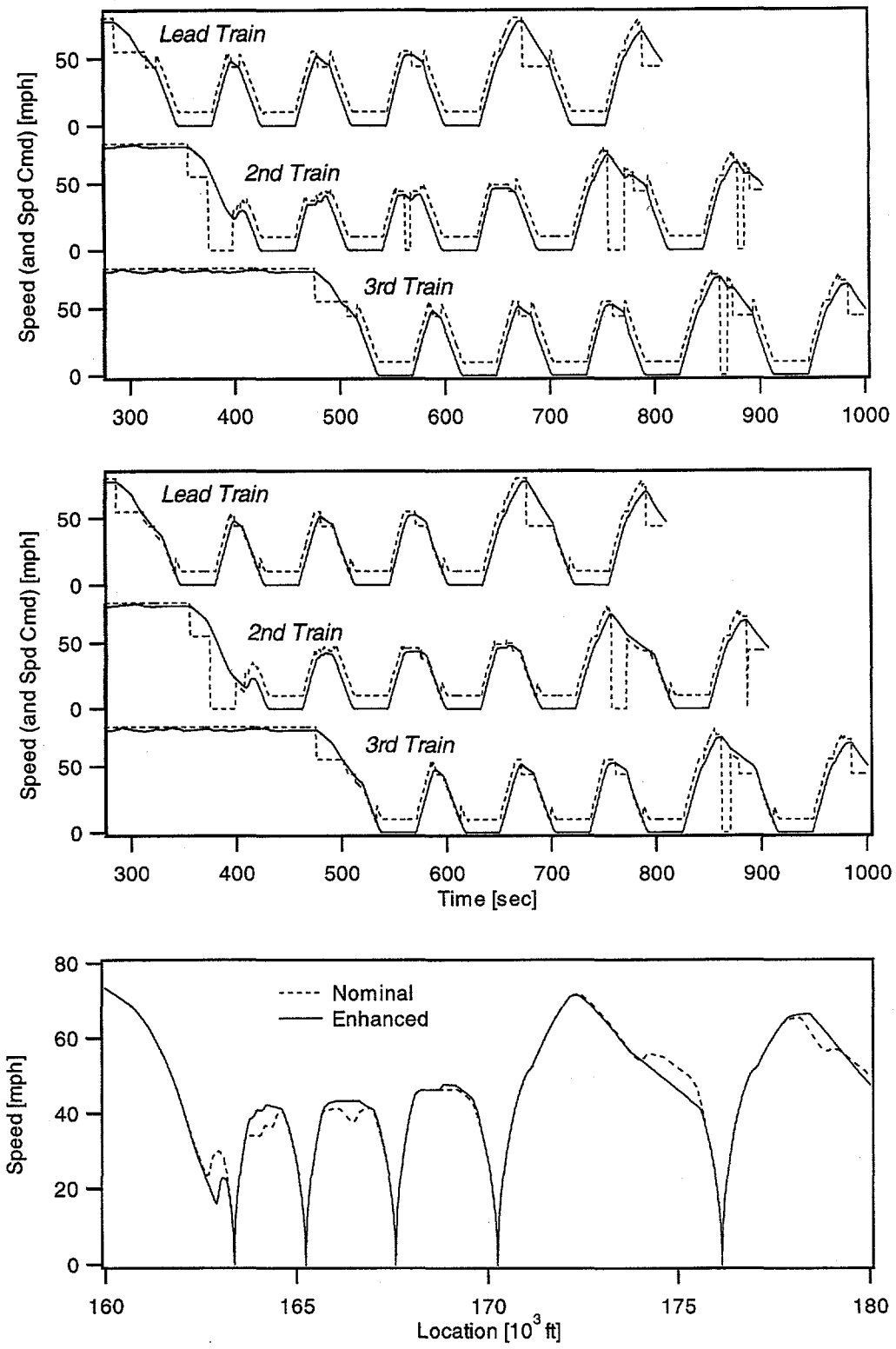

Figure 43. Three M1 trains interfering through a series of station stops. Speed and speed command with (top) nominal and (middle) enhanced control. (bottom) Speed of second train versus location for nominal and enhanced control.

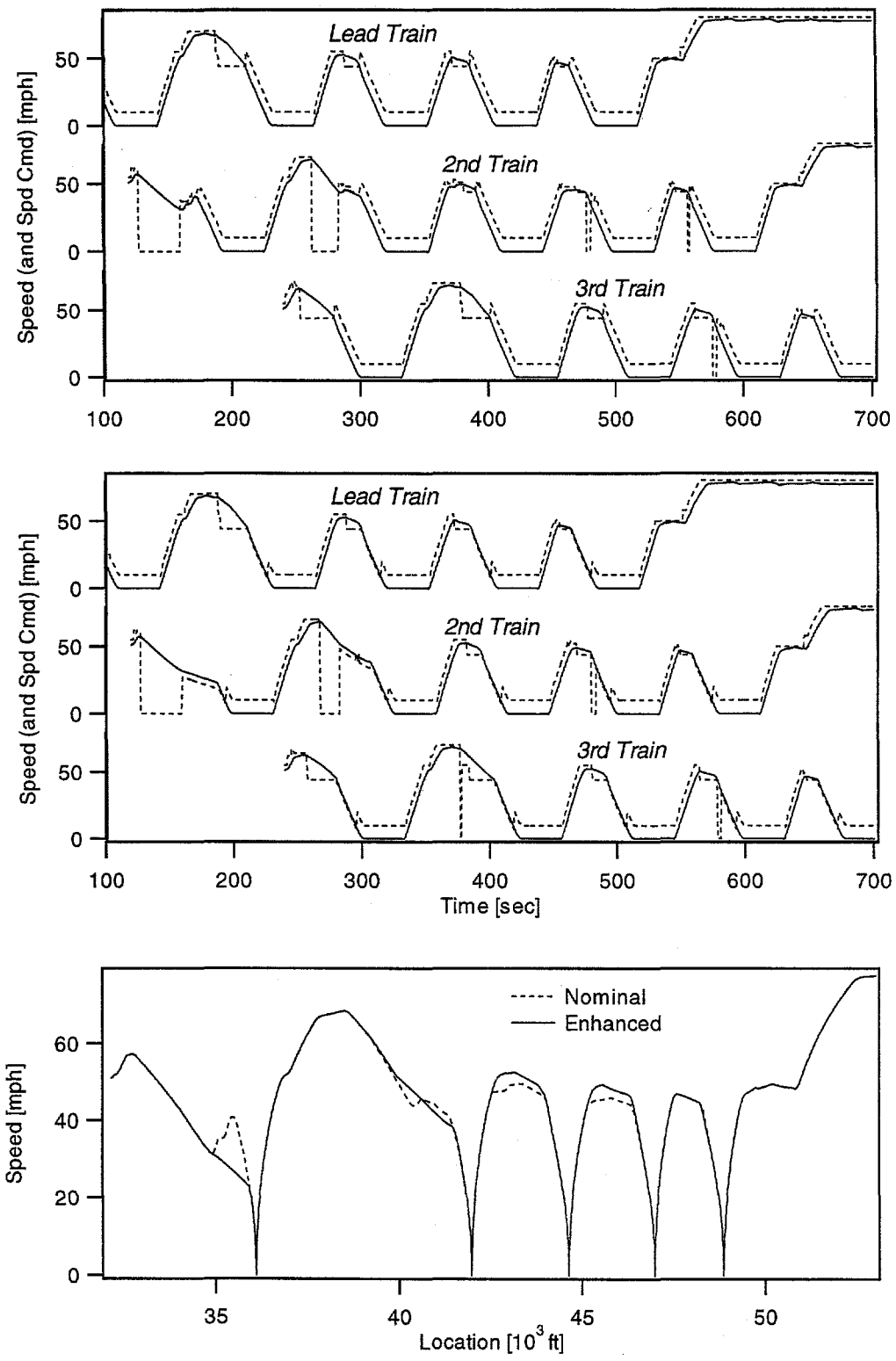

Figure 44. Same as last figure, but for M2 line.

Interference During Delay Recovery

As trains are scheduled closer together in order to increase system capacity, delayed trains that are currently unimportant will become more problematic. Every day, BART experiences roughly 20 delays of longer than 5 minutes, a few of which are over ten minutes long.[12] With scheduled headways of two minutes or less, these delays will cause substantial backups unless enhancements are added to the control system.

Backups can lead to an extreme form of interference, where a line of trains sit one behind the other outside of a station, and the line moves forward one train length at a time as the trains pull into the station one after the other. This behavior leads to spikes in power demand, as the trains repeatedly accelerate and then brake, like a line of cars at a stop sign. In addition to the resulting frustrating ride and the waste of energy, low voltages may result if sufficient power is not locally available for multiple accelerating trains. Although a low-voltage-avoiding algorithm can address this last problem, it is possible to use a more simple algorithm to smoothly and efficiently recover from a backup, while avoiding simultaneous accelerations that can cause low voltages.

With this in mind, we have developed an algorithm that will handle such delays more smoothly than the baseline system. This algorithm prevents unnecessary motor mode changes, and can prevent extreme voltage sags if a delay occurs in an area with limited power availability. In the process, the algorithm also makes delay-recovery a less noticeable event for the passengers.

Algorithm Approach

A train only stops outside of a station under abnormal circumstances, either because it has stopped behind a delayed train in a station or for some other reason. In either case, when a train stops outside of a station, the algorithm recognizes that a delay has occurred and calculates reduced speed commands for any approaching trains in order to prevent them from stopping. If the delay continues for a prolonged period, some trains will eventually be forced to stop in a backup behind the delayed train. When the delay finally does begin to move, the algorithm staggers the starts of any stopped trains so as to avoid simultaneous acceleration which can lead to power spikes and voltages sags. In addition, any trains still approaching the area are controlled so as to avoid stopping. If additional delays occur to subsequent trains in the station, then the algorithm reduces the speeds of all approaching trains accordingly so that they will not be forced to stop. This capability is critical, since dwell times are unlikely to be exactly as scheduled for any number of reasons. As long as additional delays are on the order of the station dwell time or less, this approach is sufficient. However, a substantial delay to a second train can cause a backup to recur at the station. In this case, the algorithm will again keep track of stopped trains and slow down approaching trains as though this were a new event.

Under enhanced control, each train that has stopped accelerates twice – once to come up to a calculated speed, and once to quickly pull into the station when it is clear. The initial version of the algorithm did not accelerate trains into the station, but rather maintained the approaching speed all the way to the station stop. The disadvantage of this approach is that the triptime of the train is increased because of its leisurely arrival at the station, and the headway between trains

departing the station is correspondingly increased. Our objective was to smooth out the train trajectories, and to match or better the headway through the station achieved by the nominal control strategy. We found that the only way to match the trip time and headway capability of the baseline control was to allow each train to accelerate into the station as soon as it was clear.

Simulator Testing Results

Figure 45 shows the results of simulation runs in which a train is delayed in Embarcadero station for 400 seconds on the M1 line. Trains are approaching the backup at 120 second intervals. The graphs show the full length of each train as a shaded region along the location axis. The trajectory is flat when a train is stopped, and sloped when it is in motion. The backup moves through the station with nominal control on the left, and with the delay-recovery algorithm control on the right.

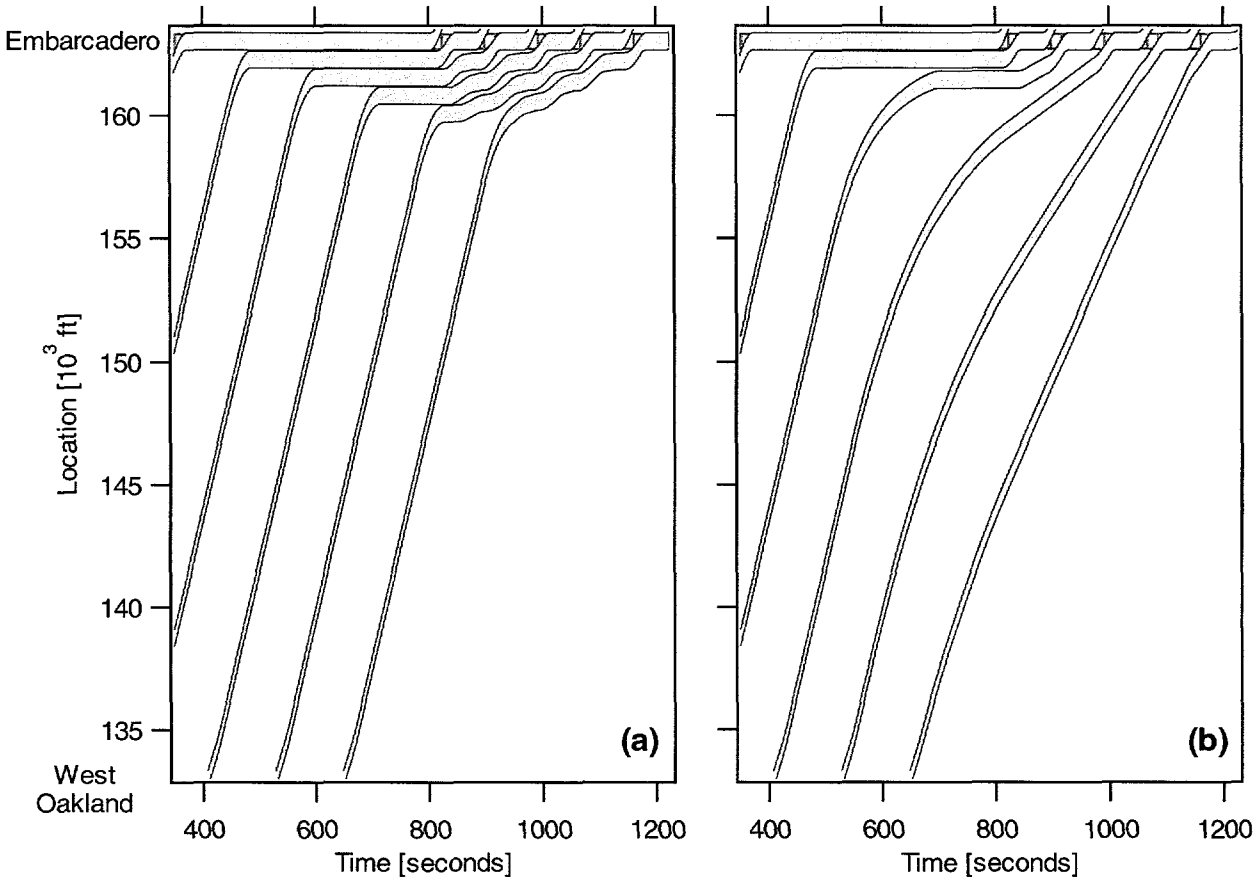

Figure 45. Recovery from a 400-second delay at Embarcadero station with (left) nominal and (right) enhanced control

Under nominal control, several trains stop behind the delayed lead train while it is delayed in the station. Even after the lead train begins to move, additional trains continue to arrive and stop behind the backup. As the backup clears, the line of trains moves forward one train length at a time as the trains pull out of the station. With the delay-recovery algorithm in place, the same

event causes only two trains to stop during the delay, and no further stoppages occur thereafter. The headway at departure from the station is maintained at approximately 80 seconds, which matches the headway achieved by the nominal hurry-up-and-wait approach to within a few seconds. The voltage never drops below 900V under either nominal or enhanced control. Thus, for a backup in a station, where substation power is readily available, low voltages are probably not an issue.

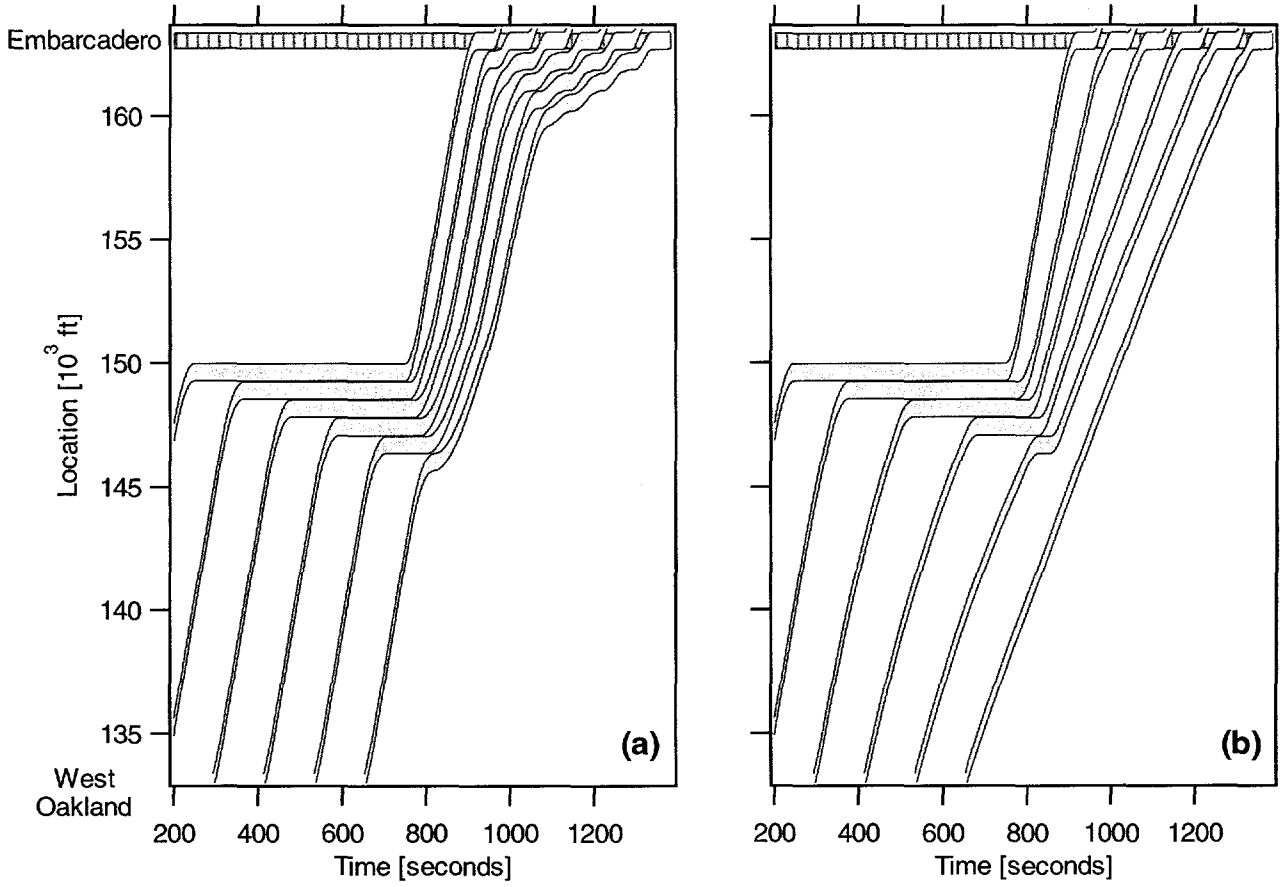

Figure 46. Recovery from a 500-second delay in the middle of the transbay tunnel with (left) nominal and (right) enhanced control

The power-related benefits of this algorithm become more apparent for a delay in an area with limited power availability. Figure 46 shows a 500-second delay in the middle of the BART transbay tunnel, where there is insufficient power available for more than one or two trains to accelerate at once. As shown in Figure 47, nominal control results in a severe voltage sag. In simulation, the train voltage drops repeatedly to the point where train motors would shut down to avoid damage from excessive motor current. This is due to the same acceleration oscillations discussed in "Interference During Acceleration." On the other hand, the voltage remains above 800V with the algorithm in place to stagger the starting times of the trains. The total time spent below 800V aggregated over all trains is reduced from 23 seconds to zero, and the time below 880V, corresponding to reduced performance, is reduced from 158 seconds to 68 seconds. The substations never exceed 300% of their rated power in both cases. This control algorithm not

only prevents motor shutdowns, but also saves energy. For the same backup in the tunnel, enhanced control saves 8% of the energy used compared to nominal control.

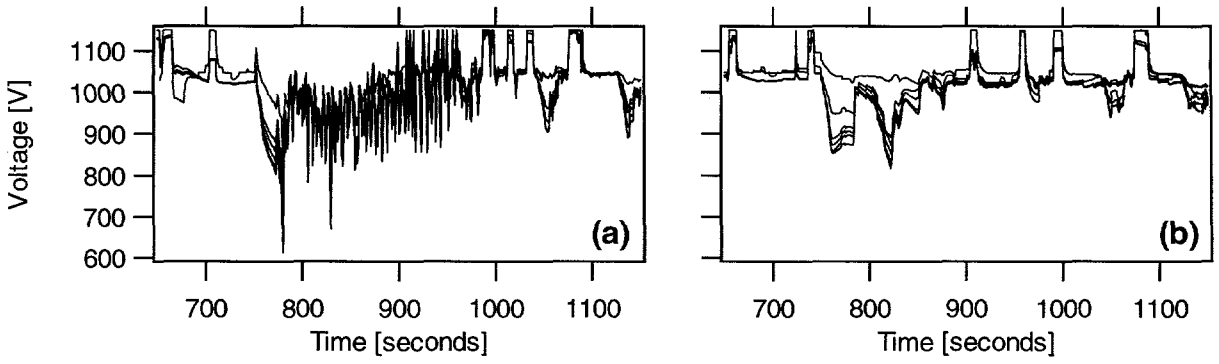

Figure 47. Train voltages corresponding to the delay in the tunnel with (left) nominal and (right) enhanced control

In a similar simulation for a 500 second delay a mile from the middle of the tunnel, which is half way to the nearest substation, and moving toward that substation, the voltage again drops well below 750V under nominal control, and remains above 850V with enhanced control. In this case, the energy consumption drops by 5%.

On the real system, station dwell times will not be precisely equal to the desired dwell, but rather will include some stochastic additional delay time because of passengers blocking doors and the like. If station dwells are delayed for only a short time, on the order of the nominal dwell time of 30 seconds or less, then the algorithm slows down the approaching trains sufficiently to avoid repeated stops. This capability was exercised in the simulations shown above by a statistical random delay of a few seconds added to the station stop of each train.

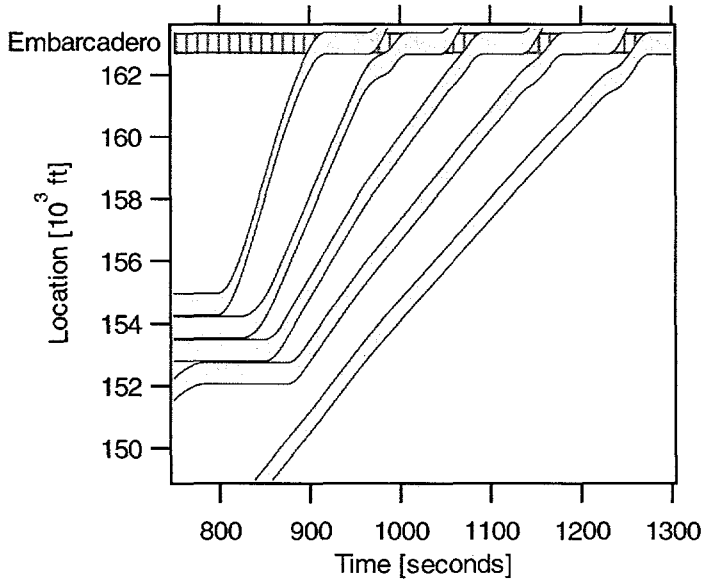

Figure 48. A delay in the tunnel followed by short delays in Embarcadero

Figure 48 shows the algorithm response to a slightly longer statistical delay. In the case shown, a primary delay occurs in the tunnel, causing very tight headway conditions, and the trains are then delayed randomly in Embarcadero station with a maximum statistical delay of 30 seconds – twice as long as in the other examples. The lead train is delayed by 15 seconds in Embarcadero, the second by 2 seconds, the third by 10 seconds, and the fourth by 15 seconds. In each case, the algorithm logic smoothly slows down the approaching trains in response, avoiding any unnecessary stopping before the station. Only the train that is about to arrive at the station slows noticeably in this figure after each station delay. This occurs because, in order to maintain a headway of close to 80 seconds in the station, each train must be very close to the station when the train in front of it departs. If the departure does not occur precisely when expected, then the approaching train must brake. This may be avoided with more buffer time between trains, but a longer headway out of the station will result. This may turn out to be an acceptable trade off, particularly if 80-second headways after delays turn out be impractical due to crowded conditions on the station platform. If the trains rarely depart on time, then the algorithm should anticipate the delays.

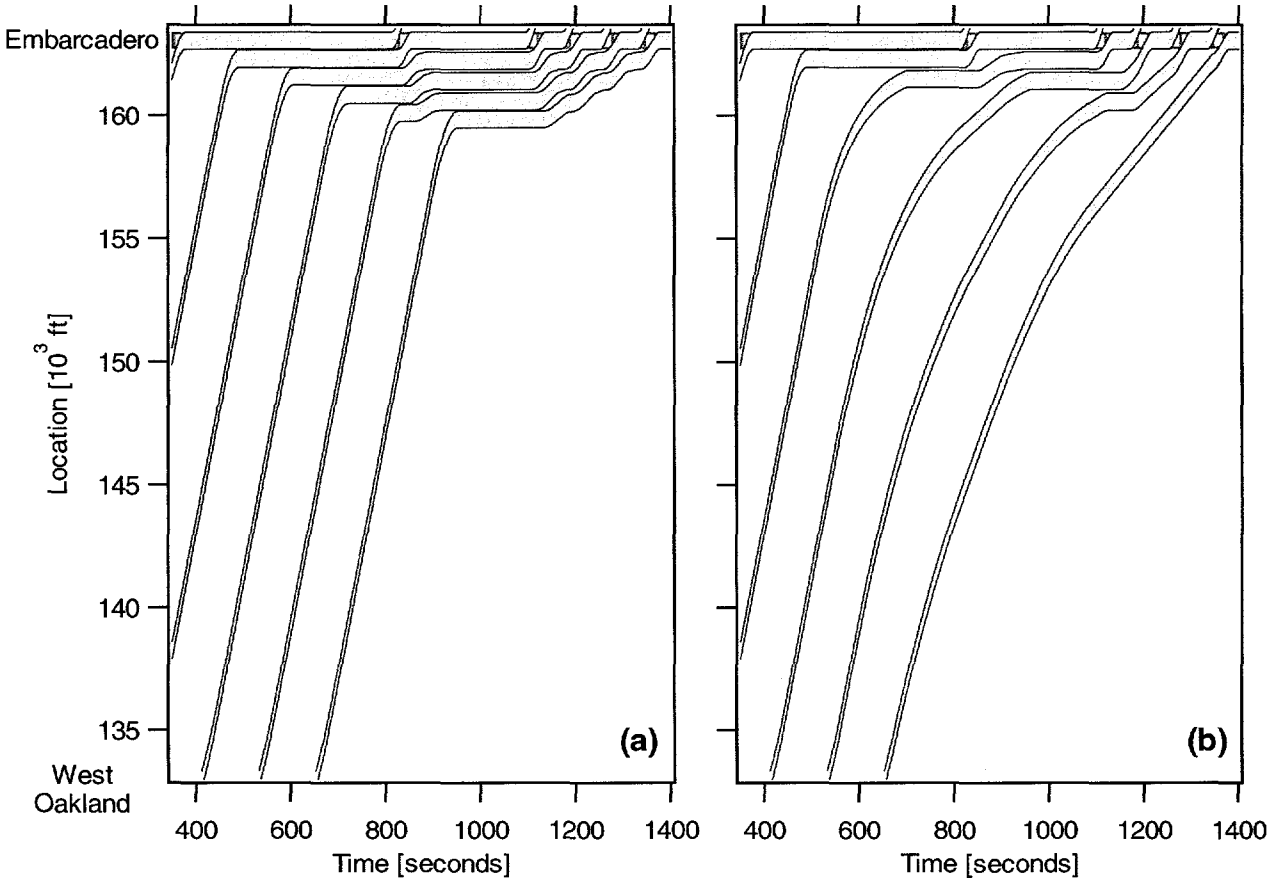

Figure 49. Recovery from a secondary delay at Embarcadero station with (left) nominal and (right) enhanced control

The implemented algorithm will also handle a long secondary backup in the same station. This situation is likely, as a large number of passengers are likely to be waiting after a lengthy delay,

and this may well cause the second train also to be delayed in the station. An example is shown in Figure 49 of a 400-second backup in Embarcadero followed by a 205-second delay on the second train. The second delay here is quite long, so that several trains have to stop outside the station in spite of the control enhancements. However, the logic will handle delays of any duration, and will prevent trains from stopping to the extent that this is possible. For example, for a similar simulation with a second delay of 150 seconds, the last two trains do not stop, whereas all of the trains still stop under nominal control. For these long delays, if additional trains were on the system, no additional trains would stop with the control enhancement, but trains would continue to pile up on the backup under nominal control.

Suggested Upgrades

Control Trains Through Multiple Station Stops

The algorithm logic thus far only commands trains through the first station in front of a backup, as shown here for backups in the tunnel approaching Embarcadero. However, if a delay at the next station, such as Montgomery in San Francisco, causes a backup into this station, the algorithm logic will not be aware of this situation and will continue to assume the track in front of Embarcadero is clear. The algorithm treats each length of track between stations separately.

It should be a natural extension of the logic used here to handle backups through a series of stations. For example, if a train stops between Embarcadero and Montgomery, it should be possible with similar logic to predict the time the train will take to proceed through Montgomery station once it begins moving, and to predict how fast a train approaching from the tunnel should go to arrive at Montgomery just as this train is expected to depart. Such logic could be applied along an entire line, such as from Oakland through all of the San Francisco stations. This algorithm would not be much more complex that the one demonstrated here. However, the programming complexity required to keep track of all of the trains through the series of station stops may be increased significantly.

Improve Algorithm Robustness

If a situation becomes sufficiently complex, such as with bunches of trains stopped in multiple locations, then the algorithm logic used here will not handle the problem optimally. It only looks for the leading stopped train on the line between stations, and trains stopped or approaching immediately behind it. If all of San Francisco was treated as a single zone, and there was a backup-causing delay at Montgomery station and another in the tunnel, then only the one in the station would be accounted for. Logic to handle multiple simultaneous delays would be more complex than that demonstrated here, but would make the logic robust to more complex circumstances.

Coordinate Starts and Stops

It may be possible to reduce energy costs by more efficiently taking advantage of regenerated energy from braking. There are two obvious methods to achieve this: (1) coordinate the timing of trains arriving at and departing from stations, and (2) coordinate trains climbing up and down hills, so that energy is efficiently transferred from the braking (regenerating) train to the accelerating (powered) train.

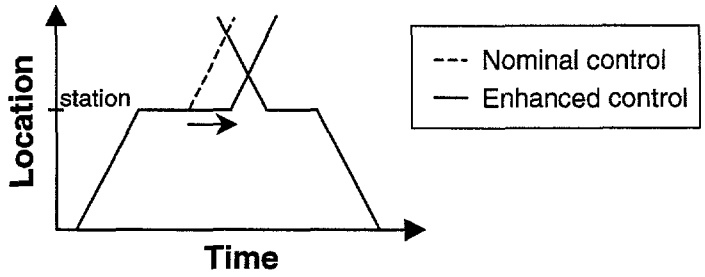

Figure 50. Sketch of an algorithm to coordinate starts and stops

A schematic representation of such an algorithm in action at a station is shown in Figure 50. In this case, the two trains are traveling in opposite directions through a station. The first train to stop in the station is held there by the algorithm to wait for the arrival of the opposing train. Thus, when the train is finally allowed to depart, it will use the energy available from the braking train rather than using power from a substation. Alternately, the approaching train could have been commanded to begin braking early to allow the departing train to leave on-time. Similar methods may be used to cause a train to climb uphill while another train traveling in the opposite direction is coming down the hill, braking to prevent acceleration.

It is unclear how much of BART's regenerated energy will be used by the baseline AATC system. On the M-line, approximately 20% of the energy used is from regeneration under the current ATC system. The regenerated energy will probably be used more efficiently under AATC due to the reduced-brake-rate stopping profile, which will produce a reduced peak power over a longer period of time, and therefore will reduce rail losses. The higher density of trains will also improve regeneration usage, as trains consuming power will be closer to trains regenerating power, again reducing rail and other resistive losses.

The power transfer from a regeneratively braking train to a simultaneously accelerating train in a 70mph speed zone is shown in Figure 51. On the left side of the figure, the speeds of the two trains are shown versus time as they arrive and depart from the same station. The right side of the figure shows the substation power required to power the accelerating train with or without the braking train present. The power transfer can be very efficient if the two trains are close to each other and the start and stop overlap temporally as shown here. These calculations assume a rush-hour crowd on a 10 car train, a 0% grade, a 3mph/s acceleration rate, and a -2mph/s brake rate. With optimal overlap of braking and accelerating trains, 35% of the energy for the

accelerating train is saved. Figure 52 likewise shows a savings of 30% for optimal overlap in a 36mph speed zone.

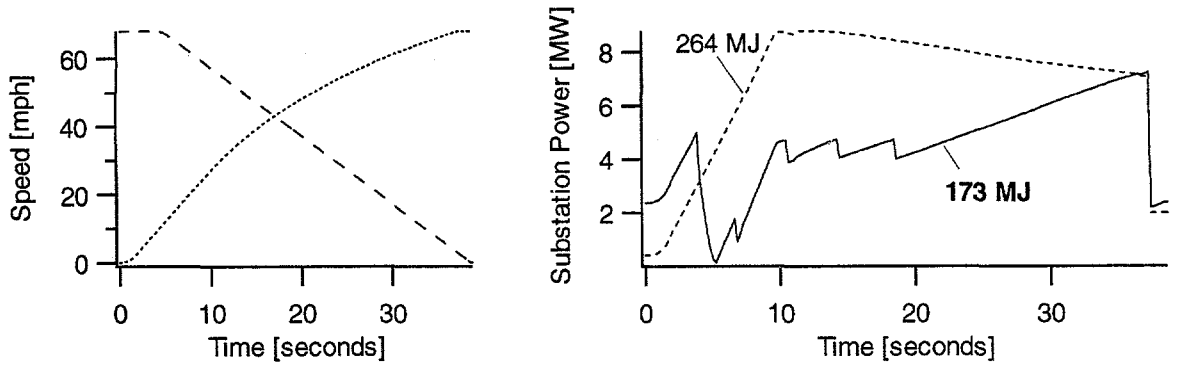

Figure 51. Train speeds and required substation power in a 70mph speed zone, shown with (solid) and without (dashed) the braking train present

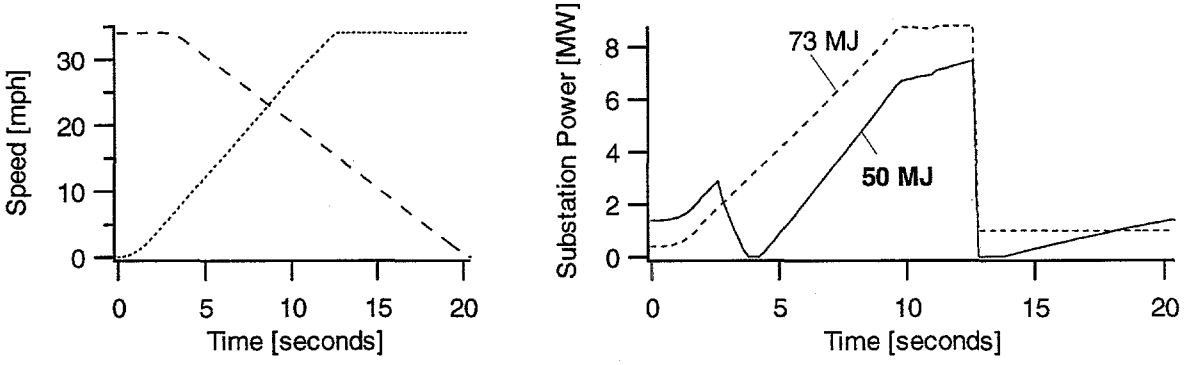

Figure 52. Same as last figure, but in a 36mph zone

It is also possible to transfer energy between trains braking and accelerating in neighboring stations, or even over several stations in dense areas such as downtown San Francisco. Thus, an algorithm can try to coordinate the trains traveling through multiple stations, allowing for more flexibility. On the other hand, in areas with low speed limits like the downtown area, trains take less than 15 seconds to accelerate or brake, as shown in Figure 52. Delays of half a minute or less, which are extremely common, can foil the attempts of an algorithm to coordinate train accelerations.

Delaying trains for the purpose of coordination will only be an option for trains which are on-time, or nearly so. Actions by this algorithm will be constrained by scheduling requirements, especially in approaches to the Oakland Wye, where arrival times are critical. We have not implemented this algorithm in the simulator, because we do not believe the benefit of this control strategy would be sufficient to justify the extreme impacts on trip time. However, it would certainly be sufficient to justify developing a train schedule which encourages efficient energy transfer between trains wherever possible.

Coast

BART trains typically travel at their maximum safe speed at all times, making it nearly impossible to reduce trip time any further during nominal operations. However, there is one exception to this rule: in some sections of track with long distances between stations, the speed limit is 80mph, but trains normally travel at only 70mph. This is done because trains traveling at 80mph and then braking from this speed can experience excessively large motor currents, which can lead to increased failure rates and maintenance costs. Coasting may provide a method to travel at over 70mph in these regions while avoiding the associated maintenance problems. It can not only reduce trip time, but can also conserve energy at the same time.

A simulation of a one-mile long coasting trajectory as compared to a nominal 70mph speed-maintaining trajectory is shown in Figure 53. The coasting trajectory is designed so that friction will slow the train to 70mph when it reaches the location where it begins braking. In this profile, the coasting portion of the trajectory corresponds to zero propulsion force, and therefore the speed drops due to drag and friction. The figure shows both train speed and power consumption versus time for the nominal (dashed) and coasting (solid) trajectories. This data was calculated with an early, simplified version of the simulator for a one mile coast on a flat track with no wind, assuming a rush-hour crowd on a 10 car train, a 3mph/s acceleration rate, and a -2mph/s brake rate. The coasting trajectory saves 3 seconds of trip time and simultaneously saves energy as compared to the nominal trajectory. Energy is saved due to the relative efficiency of the train motors during acceleration versus during speed maintaining. Typically, traction motors are more efficient at high power, so coasting saves energy by avoiding the long period of relatively inefficient speed maintaining.

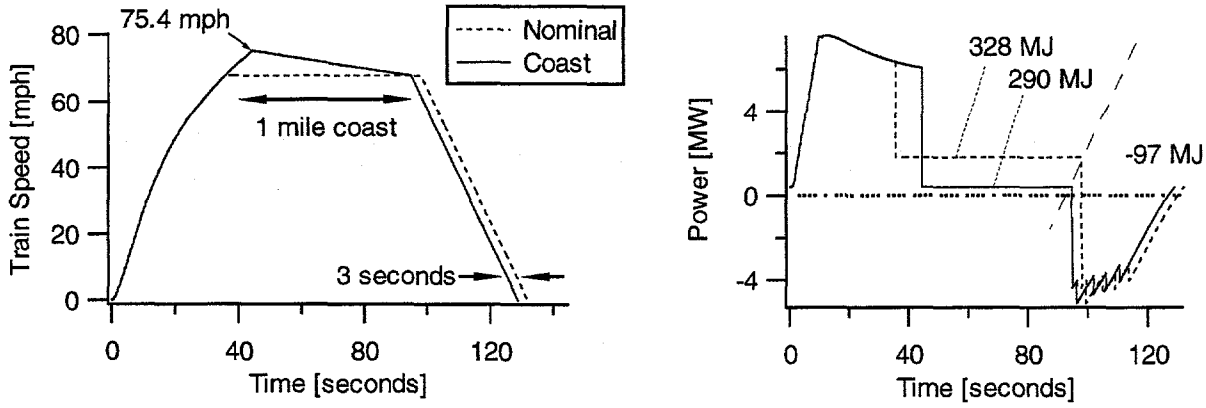

Figure 53. Coasting (solid) and nominal speed-maintaining (dashed) trajectories in an 80mph zone. (left) Train speeds and (right) total power consumption versus time are shown.

On the power graph in the figure, the total energy consumption during the propulsion and braking phases of the trajectories are summed separately. Both trajectories begin braking from the same speed at the same location, so the regenerated braking energy is the same, 97MJ, in both cases. The propulsion phase, on the other hand, is quite different. The coasting trajectory

accelerates for a longer time in order to reach a higher speed, initially using more energy than the nominal speed-maintaining trajectory. During the one mile coast, however, the coasting trajectory uses only "hotel" power, used to run the lights and air conditioning, but uses no energy for propulsion. Overall, the coasting trajectory saves approximately 10% of the energy used during propulsion. If the energy consumption over the entire trajectory, including the regenerative braking phase, is summed, and it is assumed that all of the regenerated energy is reused by other trains, then energy usage drops from 231MJ to 193MJ, for a savings of over 15%.

Before implementing a coasting algorithm, it is important to keep in mind the fact that the coasting trajectory depends upon Davis drag, which includes air and rolling friction, to decelerate the train to 70mph before braking begins. Air friction accounts for almost half of the drag force at 70mph, so a train will decelerate only half as rapidly in a strong tailwind. If coasting were implemented on an outdoor section of track, then trains may be forced to brake from speeds greater than 70mph on windy days. If this is not considered an acceptable outcome, then a more intelligent coasting algorithm could be designed to adjust the top speed before coasting as required. An algorithm could record the average or, to be more conservative, the minimum deceleration of successive coasting trains in a region. The top speed of each train could then be selected so that the train speed would drop to 70mph before braking based on recent empirical data. For example, if the top speed is initially set, as in the figure, to 75.4mph, but trains are only slowing by 3mph rather than the expected 5.4mph before braking begins, then the next train could be accelerated to only 73mph.

At a minimum, there is no reason why a coasting algorithm with a fixed top speed could not be implemented in tunnels, such as in the transbay tunnel. Air speed and track conditions in tunnels are more predictable, allowing for more consistency in train trajectories. A top speed should be chosen so that train speeds will never reach the maximum safe speed of 80mph, taking into account any downhills. This algorithm is meant to avoid braking at high speeds; and braking at 80mph on a downhill, such as the one entering the transbay tunnel, is certainly not desirable. On a long section of track, if the train speed drops to 70mph well before braking is required for the next station, then a second coasting cycle could follow the first.

Some experimentation will be required to implement a coasting algorithm, to determine the optimal trajectory and whether acceleration to higher speeds induces any motor maintenance problems. If acceleration to speeds in excess of 75mph caused problems, for example, then trains could be commanded to repeatedly accelerate to 75mph and then coast between stations on long sections of track. We expect that such a trajectory would save energy as well as trip time. This algorithm would operate on a continuous basis during normal operations, and thus any energy savings would be accrued for every train, accumulating significant savings with no additional cost.

Reduce Peak Substation Power

Control algorithms that reduce peak power usage at substations may reduce substation infrastructure requirements and demand charges on BART's energy bill in the future. The traction power bill at BART is approximately $15 million per year. Of this, approximately two thirds is for energy, and one third is a demand charge. Currently, the demand charge is related to the peak power averaged over half hour intervals and over multiple substations. Thus, "peak power" actually refers to the peak half hour of energy usage, and does not reflect short term spikes in power demand that occur over minutes or seconds at substations. It is unclear whether demand charges will ever be billed for periods shorter than 15 minutes, or at substations rather than over the entire system. With a billing structure involving anything like today's spatial and temporal averages, reduction of substation peak power will have negligible impact on BART's demand charge. As far as the energy infrastructure is concerned, substations are capable of providing peak powers substantially higher than their rated powers for short periods of time, as was described in "Traction Power System Basics." To the extent that substations are rated to handle spikes in power demand, some energy infrastructure costs could be deferred if power spikes could be avoided. Power demand spikes can also cause low voltages, but these will be addressed by the algorithm to avoid low voltages discussed earlier.

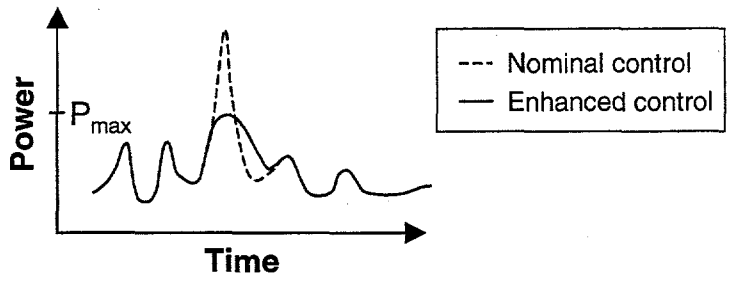

Figure 54. Sketch of algorithm to limit peak substation power

A sketch of substation power as a function of time with and without an algorithm that caps power demand is shown in Figure 54. This algorithm will require logic much like that for low voltages, as described in "Avoid Low Train Voltages," but will be somewhat easier to implement because peak substation power is less strongly non-linear, and therefore easier to predict, than train voltage. A neural network, or perhaps some more simplistic estimate, could be used to predict substation power demand, and train acceleration commands could be reduced in order to limit substation power to some predetermined level. We have deferred development of this algorithm in favor of the low voltage algorithm, since the benefits of this algorithm are less clear. In addition, we found no examples in the simulator of cases where peak power exceeded a substation's rating, whereas we found plentiful examples of low voltage problems.

Power-limited Acceleration

Intuitively, one would think that it would be possible to reduce the peak power consumption of a train by reducing its acceleration rate. However, this is not necessarily the case. The peak power consumption of a train is achieved when the train speed becomes high enough for the motors to reach their current limit. At higher speeds, power consumption remains relatively constant, while the maximum achievable acceleration rate decreases, as shown in Figure 7a. The only way to reduce peak power with a fixed reduction of acceleration rate is to use a rate so low that the motor's current limit is never reached. As one might expect this method leads to a large impact on trip time.

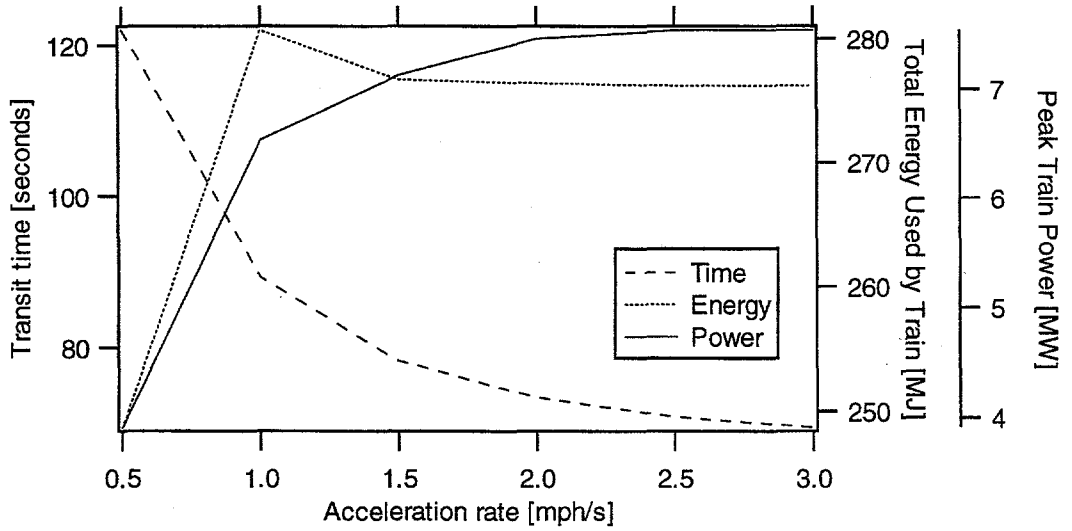

Figure 55. Peak train power, total energy usage, and trip time for a train starting from a stop and travelling one mile versus acceleration rate.

Figure 55 demonstrates this issue with calculations of a train beginning at rest, accelerating with a constant acceleration command up to 70mph, and then speed maintaining, over a total trip distance of 1 mile from its starting location. The graph shows the peak train power consumption and the trip time as functions of the acceleration rate. Total energy usage, which is simply the time integrated power consumption, is also shown. As the train's acceleration rate is decreased, there is very little impact on the peak power until the acceleration rate drops all the way down to 0.5mph/s. At this rate, the train never reaches full speed in the one mile distance, but only achieves 60mph at the end of the trajectory. Trip time, meanwhile, is severely impacted. Reducing the acceleration command from 3 to 1 mph/s increases trip time by 20 seconds, while reducing peak power by only about 15%.

A more effective method to reduce peak power would be to adjust the acceleration command as a function of the train speed in order to limit the power consumption to a pre-defined level. While train speed is low, the power consumption is also quite low, irrespective of the acceleration rate. Thus, the maximum acceleration rate of 3mph/s may be used until the desired peak power is

achieved. As the train speed continues to increase, the acceleration rate may then be decreased as required to prevent further increases in the power demand. We refer to this as a "power-limited" acceleration profile.

The location, speed, and power consumption of three trains accelerating to 34mph are compared in Figure 56. The first train accelerates at a nominal 2.8mph/s. The second accelerates at approximately half of this rate. The third uses full acceleration until it reaches a 5MW power limit, and then reduces its acceleration to maintain a constant 5MW power demand. Both half acceleration and power-limited acceleration reduce the peak power demand, but power-limited acceleration has a negligible impact on trip time.

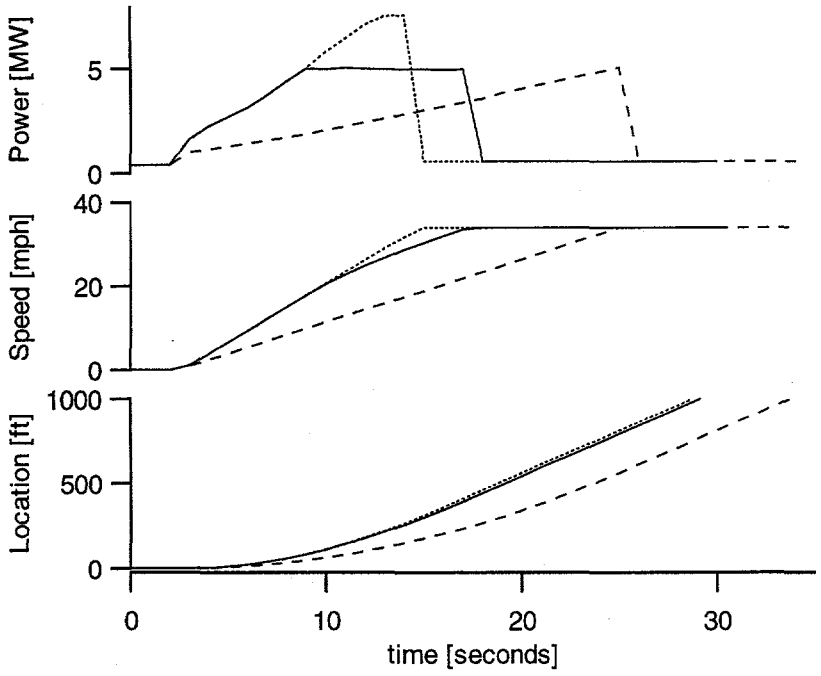

Figure 56. Location, speed and power of a train accelerating to 34mph on a flat grade using power-limited (solid), half (dashed), or full (dotted) acceleration rates

The same comparison of peak power and trip time versus acceleration mode is tabulated in Table 4 for acceleration to three different speeds: 34, 48, and 68mph. In addition, the last column shows the total time over which the power exceeds 5MW. The half acceleration strategy only succeeds in significantly reducing peak power for acceleration to 34mph. When a train accelerates to a higher speed, peak power continues to grow, nearly linearly with speed as shown in Figure 56, eventually reaching powers nearly as high as for full acceleration. For a single train, power-limited acceleration is much more effective at reducing peak power demand while minimizing the increase in trip time as compared to a simple half-acceleration command.

Table 4. Train travels a fixed distance from a stop with various acceleration modes

Speed command	Accel mode*	Trip time	Peak power	time >5MW
68 mph	nominal rate	–	7.6 MW	31 sec
	half rate	+9 sec	7.1 MW	27 sec
	5 MW limit	+5 sec	5.0 MW	–
48 mph	nominal rate	–	7.6 MW	14 sec
	half rate	+7 sec	6.8 MW	10 sec
	5 MW limit	+2 sec	5.0 MW	–
34 mph	nominal rate	–	7.6 MW	6 sec
	half rate	+5 sec	4.9 MW	–
	5 MW limit	+1 sec	5.0 MW	–

*Acceleration mode: nominal = 2.8 mph/s
half rate = 1.5 mph/s
5 MW limit = 2.8 mph/s maximum, power limited to 5MW

The technique of power-limited acceleration may be useful as a technique for reducing overall peak power consumption. However, it is important to note that the power consumption of a single train is reduced but extended over a longer time. If several trains in an area begin accelerating at times five or ten seconds apart, then the total system peak power may not be significantly reduced by this technique. The moment of peak system power will simply have power contributions from multiple trains, rather than from one train at a time, as shown in Figure 57. We have not pursued implementation of this algorithm in the simulator to date, but it would be relatively simple to enforce power limited acceleration for all trains and to determine the resulting impacts on peak power and trip time during nominal and off-nominal control conditions.

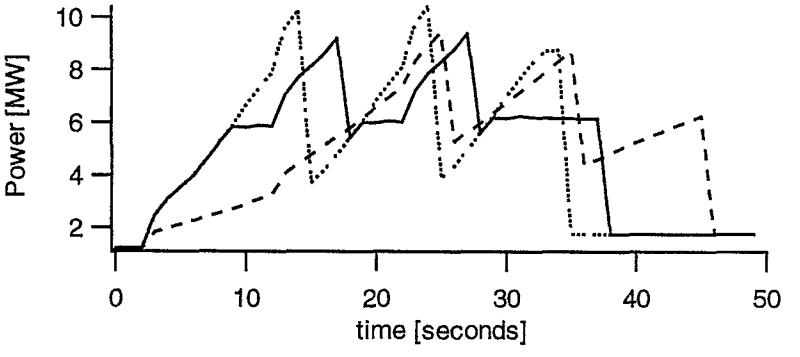

Figure 57. Total power consumption for three accelerating trains, starting from zero speed at 10-second intervals, with power-limited (solid), half (dashed), or full (dotted) acceleration rates

Multiple Algorithms

The algorithms discussed in this report have been developed and tested individually, and have not yet been tested acting together. In fact, this is in some cases not yet possible, because they sometimes were developed in non-overlapping portions of the system. For example, the delay-clearing algorithm was developed for trains on the line approaching Embarcadero from the tunnel, whereas the interference-avoiding algorithm was implemented on the line departing from Embarcadero into the tunnel. These algorithms may be applied more broadly, but this has not yet been implemented. The assumption in the algorithm logic is that the algorithms should be applied in sequence, and that as each algorithm calculates a suggested reduction to the commands, the more conservative value should always been used. However, it is possible that the algorithms will interfere with each other, causing the logic to fail, and further development work may be required to make them work in concert. An "overseer" algorithm may be required to prioritize the algorithms and apply one in preference to another if both are potentially applicable to a situation. It is possible that some artificial intelligence may be required to serve this function, but more simple logic may be sufficient. When a backup occurs in the tunnel, for example, the delay recovery algorithm itself avoids low voltages, and the low voltage-avoiding algorithm may need to be shut down to prevent it from confounding the delay-recovery logic.

Control Optimization

The best of all worlds would be a single algorithm logic that serves all of the desired functions, rather than multiple actors all trying to act as a group. In particular, a single algorithm to avoid interference may be possible, to do the work of the station-stop interference algorithm, the acceleration interference algorithm, and the delay-clearing algorithm. All of these, plus more complex interference behaviors, may define a single class of problem with a single solution. We are currently working with Pamela Williams, Paul Boggs, Patricia Hough, and Juan Meza at Sandia to pursue this goal with a simplistic version of the simulator. If this proves achievable, then we hope to also add the complexity of avoiding low voltages in the future.

Conclusion

Development of a simulator of the train control and traction power systems at BART has been instrumental in assessing the benefits of conversion to a moving block control system from the present fixed block system. In addition, it has proven to represent an invaluable tool for developing and refining the control system before implementation, and for tracking down potential problems in the control system while it is still being designed.

The new Advanced Automatic Train Control system will allow not only more precise control of trains, but also coordination of the commands to multiple trains. Enhanced control algorithms will be incorporated into the system in order to reduce energy capital and operating costs, while simultaneously improving passenger comfort and equipment reliability. To date, we have developed such algorithms for targeted control problems, such as low train voltages and interference. We expect that more global optimization of the control system will be possible in the future.

References

[1] E. Nishinaga, J.A. Evans, and G.L. Mayhew, "Wireless Advanced Automatic Train Control," *IEEE Vehicular Technology Society News* 41 (1994), p. 13.

[2] G.L. Mayhew, J.A. Kivett, J.G. Himes, and J.A. Evans, "Application of Radio Navigation Technology to Advanced Automatic Train Control," *IEEE Position, Location, and Navigation Symposium*, Las Vegas, Nevada (1994), p. 217.

[3] S.P. Gordon and D.G. Lehrer, "Coordinated Train Control and Energy Management Control Strategies," *1998 ASME/IEEE Joint Railroad Conference*, D.Stone and D.Haluza, eds. (IEEE, 1998) pp. 165-176.

[4] S.P. Gordon and D.G. Lehrer, "Service- and Energy-Related Optimization of Advanced Automatic Train Control," *1998 Rapid Transit Conference* (APTA, 1998), CD-ROM.

[5] D.G. Lehrer, "BART Train Simulator Documentation," Bay Area Rapid Transit District (1997).

[6] S.P. Gordon and W.S. Rorke, "Energy Storage and Alternatives to Improve Train Voltage on a Mass Transit System," SAND95-8222 (1995).

[7] M.M. Reading, "BART SMES Application Scoping Study, Phase 0," Pacific Gas and Electric, 007.5-94.4 (1994).

[8] BART, Data Transmission System Log.

[9] Parsons De Leuw, "Extension Service Plan Study: Traction Power System Capability, Operation Years – 1198, 2002, 2010," Bay Area Rapid Transit District (March 1993).

[10] R. Hecht-Neilsen, "Neurocomputing," Addison-Wesley Publishing Company (1989).

[11] H. Demuth and M. Beale, "Matlab – Neural Network Toolbox," Thh Mathworks, Inc. (January 1994).

[12] "Delay Analysis and Its Impact on System Performance," BART Research and Development Department (February 1997).

Appendix A – Summary of Train Control Parameters

For reference, we list below the parameters relevant to enhanced algorithm development. This list is not meant to be comprehensive, but rather to summarize the most important problem parameters.

Scenarios

- Nominal control and power
- Off-nominal control
  - behind schedule
  - interfered headway *(repeated acceleration and braking)*
  - backups due to delayed train in station or stopped train between stations
- Off-nominal power
  - rectifier/transformer outage *(reduced power capacity at substation)*
  - AC bus outage *(no power available from substation)*

Control variables

- train speed command
- train acceleration command
- train departure times from stations
- operating headway *(time between trains entering the line)*

State variables

- train location
- train speed
- train power

Dependent variables

- train acceleration
- train voltage *(= contact rail voltage – running rail voltage)*
- substation power
- substation voltage

Constraints

- safety
  - safe speed
  - worst case stopping distance *(required brake rate)*
- power availability *(minimum required train voltage)*
- speed command selection *(1 mph increments)*
- schedule at the Oakland Wye *(trains must arrive/depart the Wye on time)*
- propulsion/braking physics
- possibly trip time or station stop schedule

Uncertainties

train location *(calculated by station computers from radio time of flight, ±15 feet)*
train speed *(measured on-board with tachometer, ±0.1 mph, no slip detection)*
train acceleration *(measured on-board, systematic 0.22 mph/s error per 1% grade, random error ±0.3 mph/s)*
human factors
    passengers *(e.g. holding doors open at stations)*
    Central control *(e.g. personnel overriding automatic control)*
failures
    trains
    electrification
half second variations *(update time for train information)*

Objectives

minimize headway *(maximize throughput = 1/headway)*
minimize trip time
minimize energy costs *(traction power bill ~$15M/year)*
    energy usage (total kWh) *(increase use of regenerated energy)*
    demand charge *(peak power, averaged system-wide over half hour intervals. In the future, may be for power averaged over 15 minutes, or even instantaneous power.)*
maintain train voltage above a minimum level
minimize train-minutes of delay
minimize infrastructure costs
    train cars *(~$2M/car)*
    traction power equipment *(~$4M/substation)*
improve service reliability
    delay recovery
    schedule adherence
smooth service *(reduce number of unnecessary train accelerations)*

Metrics

Trip time
Energy usage
    total energy from each substation
    total system-wide energy
Power demand spikes
    number of substation power spikes 300% above rated power
    total time substation power exceeds 300% above rated power
    number of substation power spikes 450% above rated power
Train voltages
    minimum train voltage
    number of times train voltages drop below 750V *(where motors shut down)*
    train-minutes below 800V *(too close for comfort)*
    train-minutes below 880V *(reduced train motor performance)*
Train-minutes of delay *(measured at station departures)*
    Number of unnecessary train brake/acceleration cycles *(due to interference)*

Initial Distribution:

Venkat Pindiprolu
Federal Transit Administration
U.S. Department of Transportation
Office of Engineering
TTS-20/Room 6429
400 7th Street, S.W.
Washington, D.C. 20590

Jeff Gordon
U.S. Department of Transportation
Volpe National Transportation System Center
Mailstop DTS-76
55 Broadway
Cambridge, MA 02142

Manuel Galdo
Federal Railroad Administration (RDV-31)
400 7th Street, SW
Room 5106
Washington, D.C. 20590

Steven Ditmeyer
Federal Railroad Administration (RDV-30)
400 7th Street, SW
Room 5106
Washington, D.C. 20590

Daniel Adamson
Deputy Assistant Secretary
Office of Power Technologies
EE-10
U.S. Department of Energy
1000 Independence Avenue SW
Washington, D.C. 20585

Thomas Gross
Deputy Assistant Secretary
Office of Transportation Technologies
EE-30
U.S. Department of Energy
1000 Independence Avenue SW
Washington, D.C. 20585

C. Edward Oliver
Associate Director
Office of Advanced Scientific Computing Research
SC-30
U.S. Department of Energy
19901 Germantown Rd
Germantown, MD 20874-1290

Eugene Nishinaga (8)
Bay Area Rapid Transit
Research and Development
1000 Broadway, Suite 605
Oakland, CA 94607

Tom Sullivan
Transportation Systems Design
6543 Girvin Drive
Oakland, CA 94611

Audrey Strathmeyer
ARINC
2551 Riva Road
Annapolis, MD 21401

Howard Moody
American Association of Railroads
Train Control Technology
50 F Street, Northwest
Washington, D.C. 20001

Larry Milhom
Burlington Northern Santa Fe
Train Dynamics & Research Derailment Prevention
1001 Northeast Atchison
Topeka, KS 66616-1101

Patty De Vlieg
San Francisco Muni Railway
1145 Market Street
Suite 402
San Francisco, CA 94103

John LaForce
South East Pennsylvania Transit Authority
Power Signal Communication
1234 Market Street, 13th Floor
Philadelphia, PA 19107-3708

Martin Lukes
Washington Metropolitan Area Transit Authority
600 Fifth Street, NW
Room 4C05
Washington, D.C. 20001

Mike Francis
Massachusetts Bay Transit Authority
Operation Control Center and Training
45 High Street, 8th Floor
Boston, MA 02110

Pete Furkey
Port Authority Transit Hudson
Signal Division
1 Path Plaza, 8th Floor
Jersey City, NJ 07306

Lou Sanders
American Passenger Transit Association
Transit Research Program
1201 New York, New York
Suite 400
Washington, D.C. 20005

Jim Irwin
London Underground
Signal and Control Systems
30 S. Colonnade
London, E14 5EU
England

Yoshio Michibe
The Nippon Signal Co., LTD.
CBTC Project
15-1, 3-Chome, Taito, Taito-Ku
Tokyo, Japan

Kenji Nakada
The Nippon Signal Co., LTD
CBTC Project, Research and Development
13-8, Kamikizaki, 1-Chome
Urawa-Shi, Saitama, 338 Japan

1	MS	0749	A. B. Baker, 6217
1		9001	M. E. John, 8000 Attn: Center directors other than 8100
1		9014	W. G. Wilson, 2221
1		9014	J. L. Bowie, 2251
1		9014	S. E. Faas, 2251
1		9014	W. S. Rorke, Jr., 2251
1		9056	J. Vitko, 8100 Attn: 8100 managers other than 8112, 8114
1		9102	D. A. Sheaffer, 8416
1		9201	L. D. Brandt,8112 Attn: 8112 staff
10		9201	S. P. Gordon, 8112
1		9201	J. T. Ringland, 8112
1		9201	R. M. Wheeler, 8112
1		9201	P. K. Falcone, 8114 Attn: 8114 staff
1		9201	T. J. Sa, 8114
1		9214	P. T. Boggs, 8950
1		9214	P. D. Hough, 8950
1		9214	J. C. Meza, 8950
1		9214	P. J. Williams, 8950
3	MS	9018	Central Technical Files, 8940-2
1		0899	Technical Library, 4916
1		9021	Technical Communications Department, 8815/ Technical Library, MS 0899, 4916
1		9021	Technical Communications Department, 8815 For DOE/OSTI
1		1380	Technology Transfer, 4212