Automated Train Operation System for a Mine

Girish Keshav Palshikar

Reference Number: 1120

Contact Author:
Girish Keshav Palshikar
Principal Scientist
Tata Research Development and Design Centre,
Tata Consultancy Services Limited, 54B Hadapsar Industrial Estate, Pune 411014, India.
Phone: +91 (20) 6608 6400
Fax: +91 (20) 6608 6399
Email:gk.palshikar@tcs.com
Automated Train Operation System for a Mine

Girish Keshav Palshikar

1. Principal Scientist, Tata Research Development and Design Centre, Tata Consultancy Services Limited, 54B Hadapsar Industrial Estate, Pune 411014, India. Email:gk.palshikar@tcs.com
ABSTRACT

Mining operations offer many opportunities for the use of IT and communication technologies to improve automation and thereby deliver higher levels of productivity, cost, quality and safety. In this paper, we discuss a particular application, namely, building a fully automated train control and operation system within a mining area. We first outline the basic architecture and functionality of such a system and then present a real-life case-study where the railway system within a remote iron ore mine in Canada was converted to a fully automated train operation system. This hard real-time safety-critical system required the use of advanced technologies such as embedded systems (e.g., on-board train controller), radio communications, formal methods, optimal scheduling and extensive simulations. We focus on the design aspects for maintaining the system in safe states.

Keywords: Railway systems; Real-time safety-critical systems; Mining automation; Formal methods.

INTRODUCTION

Mining operations offer many opportunities for the use of IT and communication technologies to improve automation and thereby deliver higher levels of productivity, cost, quality and safety. For example, one can deploy a radio-frequency based ad-hoc network of sensors of various kinds (temperature, water, methane, CO or CO₂ etc.) which can help monitor dangerous conditions. A centralized system can then collect and analyze the data received from these sensors (every minute, say) and decide whether there is likely to be an unsafe situation (e.g., fire, methane explosion or water flooding) at any particular location within the mining tunnels. The system can then automatically take appropriate preventive actions such as raising alarms for evacuation, stopping relevant mining equipment, starting pumps to remove excess water, shutting tunnel doors for isolating the unsafe spot etc. Other applications include manpower planning and scheduling, control of mining equipment to achieve productivity goals, reliability analysis of mining equipment and so forth. In this paper, we discuss one particular application of IT and communication technologies, namely, building a fully automated train control and operation system within a mining area.

Human errors, such as over-speeding or signal overrun, are a cause of many train accidents. Moreover, a human train driver is subject to fatigue, boredom and other distractions, which can lead to failure to detect or effectively handle unsafe situations (such as obstructed tracks) in real time. Human train operators' performance is often sub-optimum in operations such as speed control, which leads to energy wastage. Due to these reasons there is a trend towards increasing automation in railway operations. Metro (subway) train operations, other rapid transit systems such as intra-airport trains, goods transport systems within a port or a factory, and trains within mines in remote areas are prime candidates for automated train operations.

Naturally, the level of automation of train operations within a system may vary depending on the needs and costs involved. In semi-automated train operations, the human driver remains in charge of operating a train, except for speed and brake control, which are handled by an automated controller. In the next level of automation, a train is run by a fully automated controller, but a human driver is available inside the train, and can override automation and take charge in case of failures or emergencies. In fully automated train operations, there is no human driver on board and hence the train is entirely operated by an automated controller. Metro train networks in Copenhagen, Singapore, Dubai, Tokyo and Kuala Lumpur now include parts with fully automated train operations. Parts of metro networks in many other cities, such as London, include semi-automated train operations. Such fully automated train operations are also particularly suitable for passenger-less trains, such as those that transport extracted ore within a mine.

The benefits of automated train operations can be summarized as follows (Fischer (2011)): (i) improved safety due to elimination of human errors and automated failure detection and handling; (ii) more predictable and timely train operations, with better adherence to schedules; (iii) optimum train movement leading to energy savings; (iv) reduced operational costs, including less human staffing requirements; (v) possibilities of adding new trains or improving schedules to the increase track and train utilization, leading to more profits.

The basic idea in automated train operations is to instrument the track and train infrastructure to monitor the current state of all resources at every instant and have a sophisticated centralized automated movement planning and control system that analyzes the field data, constructs a “true” picture (system state) of outside reality, sends control instructions every instant to all trains for their next tasks, such as start, stop or speed changes or instructions to track-side devices such as switches that allows a train to go from one track to another. Note that most fully automated train operations do not depend on outside visual signals along the
track, since there is no human on board any train. Real-life automatic train operation systems have to deal with issues such as compliance to regulatory standards and interfaces with existing equipment.

THE ATO-2000 SYSTEM

Overview

ATO-2000 (Palshikar (2001)) is an example of a fully automated driverless train control system within a Canadian iron ore mine. The system controls about 8.5 miles of tracks divided into two main tracks and several sidings. The system can have up to 9 electrically powered goods trains, each containing up to 20 cars, running simultaneously. Each train picks up ore rocks from one of the three pockets (which are storage points where the ore rocks dug from the mine tunnels are stored), transports and then unloads the ore into a crusher complex. Then the empty train returns to the pocket and continues as before.

Fig. 1 shows the various sub-systems in the ATO-2000 system. The trains are driverless, with an on-board computer (OBC) that actually runs the train as per the commands from a centralized controller. The track-side equipment includes tunnel doors (some part of the track is underground), more than 100 track circuits to detect track occupancy, 36 power sensors for overhead power, and 24 switches that allow trains to change tracks. The tracks are controlled by 10 wayside interface units (WIU), each of which consists of a computer, with an operator console to mainly allow a human operator to operate the trackside devices. A human operator can use a WIU to observe the state of the part of the track controlled by that WIU and to issue commands to track-side devices if necessary. There are various speed limits specified for various parts of the track, which may change dynamically; e.g., due to maintenance work. The system may also contain human-operated trains (e.g., for inspection) and thus their presence must be detected and taken into account when operating the automated trains. Human operators may block some part of the track for a specified time period, e.g., for maintenance, which must be taken into account, when controlling the train movements. Fig. 2 shows a small part of the track, along with some of the track-side equipment. The main motivation for this automation was the harsh weather conditions (temperature could drop as low as –70°C in winter), which made it very difficult for human train drivers to work.

The Railway Control System (RCS) is the central system that observes, plans, controls and manages the trains and the track-side equipment so as (i) to meet operational goals like throughput (tones of ore delivered to the crusher complex per day), productivity (total number of trains and cars, used along with the total time taken), ore mix constraints etc.; and (ii) to always keep the system in a safe state that meets all the specified safety requirements.

Each train and each WIU sends its current status every second to the RCS via a TCP/IP based radio communication network. The train status includes its current position, speed, and direction as computed on-board. WIU status includes switch and door positions, power status, occupancy indicated by track-circuits etc. Upon receiving this field data, the RCS computes the next course of action for each train and new values for each track-side equipment and dispatches the necessary commands to the OBC in each train and to each WIU. The commands are then executed by the computers in OBC and WIU and their status is also included as part of the field data sent to the RCS.

It is important that the RCS maintain an instantaneous state of the system that is always up-to-date and corresponds to the outside reality. Hence the field data received and the commands generated by RCS are validated for correctness and safety (the commands should not lead the system to any unsafe state). If the received field data has any errors (e.g., the reported speed of a train seems wrong), then RCS assumes the old state for at most two consecutive cycles. If there is any error in any generated command (e.g., the new speed specified for a train seems wrong) then that command is not sent out. If a train or WIU does not receive a command from RCS for two consecutive cycles, then they have a default fail-safe action (e.g., the OBC applies brakes and ignores all further commands until the train stops).

The ATO-2000 system has a stringent 1-second processing cycle, making it a hard real-time system. This 1-second cycle is sub-divided into several sequential sub-frames: receive status messages from trains and WIU, validate this field data, update system status database, generate new commands, validate and send these commands to trains and WIU. The stringent safety requirements make this a safety-critical system; Leveson (1995).
Movement Planning

The movement planner in RCS decides a contiguous route – called enforceable authority (EA) – for each train, along with a movement plan that includes start time, end time, various stops or track changes along the route, speeds at different time periods etc. The EA is essentially the “remaining” route for the train, and hence keeps changing as the train moves along. The movement plan for a train is continuously monitored and updated, particularly for speeds and stops, depending on the state of the system (track-side equipment and state of this as well as other trains). The system sends various commands to the track-side equipment to ensure that the train can move as per its movement plan; e.g., tunnels doors are opened or switches are appropriately aligned before the train reaches that point. Note also that the movement plan of a train may depend on the movement plans of other trains; e.g., train A approaching switch S on track 1 must halt before reaching S to allow train B to change from track 1 to track 2. The movement plan must respect the various speed limits that may be imposed in various parts of the tracks. Also, the movement plan must respect all the relevant mechanical constraints such as the time and distance required by a train to come to a stop (the braking table) or to attain a new speed from its current speed. The movement planner consists of a sophisticated dynamic planning algorithm that simultaneously plans the movements of all trains as well as state of all track-side equipment, taking into account all safety and other constraints (speed limits, braking tables, blocked tracks, non-ATO trains, human operators etc.), so as to meet the operational goals; Miyatake and Ko (2010); Khmelnitsky (2000); Howlett (2000); Golovitcher (2001).

Safety Monitoring

The various mechanical, electrical, communication, hardware or software components in this system may fail in various ways; e.g., hard breakdown of a device, missing or delayed messages, corrupted data etc. For this reason, many components have individual fault-tolerance facilities, either based on simple replication or corrective measures. For example, each track-circuit occupancy sensor is duplicated and is required to report a fail-safe value. Despite this component-level fault-tolerance, the system may still reach an unsafe situation, due to multiple faults that can cause distortion in the system’s view of the reality.

Many types of unsafe scenarios may occur in this system. For example, train collisions (e.g., due to EA overlap or failure of a train to stop when commanded), derailment (e.g., due to speed limit violation by a train, switch mis-alignment, train overloading, track wear and tear etc.), mechanical failures (e.g., broken rail, axle dragging, overheated bearings etc.), device failures (e.g., a door failing to open), speed limit violations, brake failure, train entering into closed tunnels or blocked tracks, unexpected change of track for a train, un-commanded train movement (e.g., rollback on a slope), track obstruction (e.g., falling rocks), fire in train or WIU etc. Additional unsafe scenarios may occur in loading or unloading of the ore.

An important goal of the system is to always maintain the system in a safe state and to ensure that the various failures are handled appropriately and in time so that accidents and unsafe scenarios do not occur. Accidents are costly in terms of damage to property and human life and also affect productivity and throughput of the system. To achieve this goal, the system made use of several strategies. First, it used formal methods during software development, which advocate the use of mathematical techniques in various phases of the software development life-cycle. Requirements were formally stated using the Z notation Spivey (1987), which is based on mathematical logic, which exposed several issues in English statements of the requirements, such as ambiguity, incompleteness and incorrectness. These formal specifications were then analyzed before they were implemented. The use of formal methods in safety-critical systems, in particular in railways, is well-understood (Boraiv 1998; Cimatti et al 1998; Guiho and Hennebert 1990; Hirao and Fukuda 1998); see also Raghavan (2004).

A separate safety monitoring sub-system performs two important tasks. First, it validates the received field data from OBCs and WIUs to detect any errors or inconsistencies. This is important to avoid incorrect picture of the outside reality (e.g., actual positions of trains and track occupancies) which may lead to accidents such as collisions. Second, it validates the outgoing commands generated by RCS to ensure that they would not lead to any unsafe system state. Examples of data validation checks are as follows.

1. For each train, reported north location should really be more northwards than its reported south location.
2. For each train, all contiguous track segments from the south to north location should indeed be reported as occupied by the corresponding track-circuit occupancy sensors.
3. For each train, the computed distance between its south and north location should be close to the train length.
4. All contiguous track segments reported as occupied should be consistent with the values of the switches within these segments.
5. For each train, the computed direction of movement (using changes to reported north and south train locations) should be consistent with commanded direction of movement for the train.
6. For each train, the computed speed (using changes to the reported north and south train locations) should be “close” to the commanded speed for the train.
7. For each train, the computed speed is within the speed limit applicable to each track segment currently occupied by it.
8. For each train, the computed train occupancy (using north and south locations and corresponding track occupancies) should be within the commanded EA for the train.
9. Detect any “unexpected” occupancy reported by any track-circuit occupancy sensor (e.g., an occupied track even though there is no train currently at that position).

Examples of validation checks for the output commands generated by RCS are as follows.
1. For each train, the commanded EA is contiguous and contains the train’s current occupancy.
2. For each train, the commanded EA does not include any track segment that is blocked, occupied, without power or contains a broken rail.
3. For each train, the commanded EA should not overlap with commanded EA of any other train (except possibly for end-points).
4. For each train, the switches within the commanded EA are correctly aligned.
5. For each train, all the doors within the commanded EA are open.
6. For each train, the commanded EA and current EA are consistent with the reported direction of train movement.
7. For each train, the commanded speed must be within all the applicable speed limits within the EA.
8. For each train, the commanded EA must be long enough to enable the train to stop within the EA.
9. For each pair of adjacent trains, their EAs must be separated by at least 40 meters.
10. If any new command is issued to any device that will change its state, the device must not be in the current or commanded EA of any train.
11. If any new command is issued to any track segment (e.g., block or unblock), then that track segment must not be included in the current or commanded EA of any train.

Testing using Real-time Simulations

The system needs to be tested against realistic scenarios that may happen in the environment, including various normal traffic movements as well as abnormal conditions and failures such as over-speeding, failure of signals, switches or doors etc. Typically, the status of the trains, tracks and other devices is reported to the application program to be tested through TCP/IP messages. Note that while field testing is crucial, it is not possible to create many of the unsafe situations on real tracks for testing purposes, due to obvious reasons. The use of a realistic and real-time simulator is then crucial for safety testing.

We had built a simple simulator for railway tracks, track-side devices and train traffic. The simulator was developed in VC++ for MS-WINDOWS platform. The simulator was usable for any track topology and included built-in features for creating errors and faults (e.g., randomly or through user-interaction). It was built on top of a powerful discrete event simulation engine and had a simple and flexible scripting mechanism for creating a wide range of powerful and comprehensive traffic scenarios and conditions, including various normal movements and conditions, load conditions, timing dependent conditions, errors and faults in devices etc. The system provided facilities to observe and control the simulation though a simple graphical user interface and had support for storing, importing/exporting, gathering, displaying and analyzing simulation data. The simulator works in real-time in the sense that it sends all the status messages in real-time to RCS over TCP/IP, exactly as a real OBC or WIU would send. It also receives and processes the commands from RCS in real-time, exactly as in the final system. The RCS is not aware that it is actually communicating with a simulated environment (OBCs and WIUs).

CONCLUSIONS

Mining operations offer many opportunities for the use of IT and communication technologies to improve automation and thereby deliver higher levels of productivity, cost, quality and safety. In this paper, we discussed a particular real-life system called ATO-2000 which is a fully automated driverless train operation system for movement of iron ore from mine tunnels to crusher complex in a remote Canadian mine. The main motivation for this automation was the harsh weather conditions (temperature could drop to −70°C in...
winter), which made it very difficult for human train drivers to work. This hard real-time safety-critical system controlled and operated 9 goods trains over 8.5 miles of track, part of which was underground. Important features of this system include the use of formal methods and real-time simulations to ensure that the system is always in a safe-state, as defined by a variety of safety criteria. The system is fully operational and so far has been operating without any major accidents; Palshikar (2001). The state of the art has advanced considerably since this system was built, and now many components in such systems are readily available.

ACKNOWLEDGEMENTS

I would like to thank Prof. Mathai Joseph, Prof. K. V. Nori and the entire ATO-2000 project team for giving me the opportunity to work on this challenging system. I appreciate the enthusiasm and support that Charlie Gipson gave for Formal Methods throughout the project. Thanks to Dr. Pradip for suggesting that I write about the ATO-2000 project for IMPC 2012.

REFERENCES


FIGURE CAPTIONS
Fig 1 – Sub-systems in the ATO-2000 system.
Fig 2 – A simple part of the railway track and associated equipment.
Fig 1 – Sub-systems in the ATO-2000 system.

- Safety Monitor System
- Railway Control System (RCS)
- Ethernet LAN (TCP/IP)
- Radio-frequency WAN (TCP/IP)

WIU: wayside interface unit to monitor and control track-side equipment (switches, doors, occupancy and power sensors etc.)
OBC: on-board computer housed in a locomotive to monitor and control a train

Fig 2 – A simple part of the railway track and associated equipment.

- Switches used to change tracks
- Occupancy sensors to detect whether a portion of track is occupied or not
- Tunnel doors

Numbers indicate the length of the track segment (in meters)