EXPERIENCE ACQUISITION SIMULATOR FOR OPERATING MICROTUNELLING BORING MACHINES

Alberto Jardón, Juan G. Victores, Santiago Martínez, Carlos Balaguer

Robotics Lab, Engineering Systems and Automation Department, Universidad Carlos III de Madrid, 28911 Leganés, SPAIN

Abstract

This paper describes an innovative modeling, interfacing and training framework and application for microtunneling machines under heterogeneous gravel and sand soils, based on a machine simulator. It is initialized using a selective collection of skilled pilots' know-how during the performance of a pipe jacking microtunneling machine operation, generating a rule-based system based on grouped rules and states that replicate the machine's performance. The adjustment of these states and associated rules allows the creation, setup and analysis of a realistic functional model for tunneling machines. The developed system integrates a friendly Human-Machine Interface (HMI) that closely resembles real machines' pilot cabinets and allows natural interaction with the implemented inference engine through the simulated control panel. Additionally, the framework allows the training of tunneling machine operators by simulation and subsequent gathered data analysis, obviating or reducing mechanical movement times if desired. The virtual pilot's desk allows global training time and cost reductions, and increases safety for future operators and machinery. The HMI is divided into two screens, which replicate the data and the command panels of a real machine's control desk. The presented framework has allowed the first implementation of a jack piping microtunneling machine simulator by means of the developed pilots' steering know-how capture methodology.

Keywords: Tunnelling machine, Robotic automation, Pipe Jacking, Inference Engine, HMI, Simulator, Training.

1. Introduction

The industry of underground construction is expanding; therefore, the associated technology must provide reliable solutions. The use of tunneling machines will also be affected by urban expansion, that implies the construction of new residential zones in places where basic services such as electricity, sewerage, and water and gas supply do not exist. In the last few years, a fast development of tunneling machines has been experienced. The number of utility tunneling (microtunneling, pipe-jacking) and traffic tunneling (large-diameter) projects is growing. Pipe-jacking microtunnel boring machines (which will be referred to as simply MTBM) use a powered cutterhead and steering assembly that is pushed forward by a jacking system from a launching pit, shown in Figure 1. They have mainly been developed in Japan [3],[23],[19] and Germany [6], for sewer work. As the machine advances, sections of permanent pipes are fed in and pushed forward behind the MTBM by the jacking unit. The resulting tunnels are always of circular cross section (typically between 0.5 m and 2.5 m diameter) due to the excavation method, which consists in rotating some form of cutterhead against the soil and rock face. Modern machines have a relatively short driven mechanism. Such drilling assemblies tend to be long compared to their diameters and tend to fit tightly in tunnels.

Most of these machines are steerable and typically use a system of hydraulic jacks to tilt the cutterhead towards the required direction. Changing the direction of tunneling is hence a matter of making the cutter head cut sideways and getting the drive mechanism to follow it. Typically, steering angles can reach 3 degrees at most, allowing curvature radii starting at 250 m. However, many factors can affect these values, such as pipe type, pipe size, joint type, ground condition, and tunnel length.

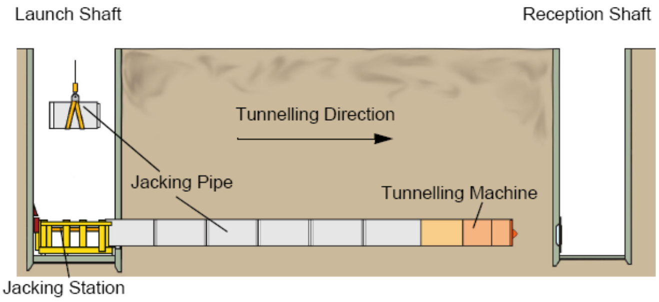


Figure 1: Microtunneling machine operational schema

The first microtunneling project was completed in USA in 1984 [18]. In Spain, the first microtunneling project ever assembled by a Spanish company was not until 1996, by the Eurohinca company [21]. Microtunneling machines are an excellent solution that minimize urban surface disturbances and economic impact during tunnel construction. These machines operate in a variety of relatively soft soils, although some are provided with cutting and crushing heads that are capable of coping with gravel and small boulders and allow them to drill rocky terrains. Maximum tunnel lengths are continuously increasing, from 150 m reported at [14] up to 1200 m and 3000 m [22]. Microtunneling technology is considered highly specialized, and inexperienced execution can severe issues. Reliability can be limited if the geology of a terrain does not fit with ideal machine working conditions, or if the project proposal does not take restrictions and limitations of this technique into consideration.

Typically, these problems are due to undefined and unanticipated conditions that are adverse and beyond the capabilities of the MTBM such as: (i) obstacles that stop the forward progress of the machine, boulders, high concentration of cobbles, (ii) rock layers and hard zones in soft rock or soil, and (iii) plastic clay with mixed ground, unusual slits, and interfaces between materials which present radical differences in density and consistency. These factors can generate inadequate torque, chocking of the MTBM cutterhead, and losing the guidance control and collapsing. Having to rescue the machine by means of digging generates delays, cost overruns, and economic losses, and is not always possible.

To recognize and solve these previously described situations and to be able to avoid similar issues, an experienced crew is required to perform the machine's control properly. Due to tunneling machine – ground interaction, some internal machine variables continuously drift off course, and the operator's control references adjustments must compensate the drift in order to keep the machine under control and safe operation while following the planned path. The growing demand of microtunneling machines is greater than the growth of available qualified professionals. Due to efforts in cost and learning time, it normally takes from 12 to more than 18 months to achieve a certain degree of qualification. The given training is focused on transmitting senior pilots' knowledge to successfully solve common and unexpected machine states in as many different terrains as possible, by means of control of the real machine shared between senior and novel pilots. The senior, however, remains the ultimate responsible of the machine.

2. Training simulators in construction

Performance enhancements are a well-known advantage of simulator training [4]. Several training simulators for construction machinery have been developed in the past by construction companies. Several examples are presented: a) "Versatile augmented reality simulation for training in the safe use of construction machinery", called Var-trainer, is the expected result of a EU 7th Program Frame project coordinated by Ikerlan [30], b) Atlas Copco simulators for Rocket Boomer L2C tunneling machine-drill jumbo jet simulator (updated to a jumbo jet L2E version), and a JCB backhoe 3CX simulator [5], c) CyberMINE ^{TM} immersive simulators, presented in the Bauma fair in 2007 [28], that "transport" the operator in a highly realistic operating environment within a 3D training world, d) Aitemin developed in 2005 a mining machine simulator that also used immersive environments [2].

Most of these simulators closely replicate common control cabinet instrumentation, and possess control functions that are very similar to those that run on real equipment. The operation of sub-systems such as engines, braking systems, hydraulics and drilling heads is mathematically modeled, in order to provide realistic feedback to the training operator. These simulators work with a fixed implicit mathematical model of the machine environment interaction [24], [1], but there is no information about the accuracy of these models.

The proposed framework has been designed to ease the transference of high level know-how, experience, and expertise of senior MTBM's pilots to other organization members. A discrete state virtual machine and a rule based inference engine have been developed to implement the machine control systems and the most influential parameters. To replicate the machine performance and the terrain – machine interaction without using physics engines [10] or implicit mathematical models, a very simple initial rule engine has been implemented, and a replica of machine control and instrumentation panels has been programmed. This way, the model is first tuned by the experts, and then constantly updated considering the experts' suggestions and typical exercise results. After some iterations the rule engine is tuned enough to respond as the real machine does, and it can be used to build a simulator or to be used as a black box model for control algorithms. After a detailed and progressive analysis, more precise and complex rules have been implemented to obtain realistic performance. Used as a training simulator, it has proved useful to focus the training effort on steering. Other machines issues, relevant for maintenance or technical staff but not for pilots, have been avoided. The system has been designed to reduce training time from 12 or 18 months to less than a few, providing actual training quality improvements. The presented development is the first ones focused on the guidance of tunnelling jack piping machines.

3. Framework and simulator architecture

As response to proprietary complex systems' uncertainty that forbids the analysis and modeling of internal machine subsystems and parameters in order to create practical models, the proposed framework assists expert system creation. This methodology for creating virtual machine replicas is based on two pillars: first, the recreation of the real machine interface, which is the cockpit of the pilot machine; and second, the inference engine that obtains and feeds data to the cockpit, designed from the analysis of the system from the expert's point of view, reliable enough for training or to be used as a model in designing machine control systems without using physics engines or implicit mathematical models.

The presented framework facilitates the application of the proposed methodology. It has been implemented by means of two computers linked by TCP/IP, where one of them is an industrial PC with a tactile screen. The application of the methodology requires several stages. The initial approach to analyze the problem of how to steer the MTBM ,was to analyze which are the data sources of pilots and how they manage this data to generate the proper settings inside the cockpit. Efforts must focus on emulating the cockpit's data sources and controls as reliably as possible. After that, it is possible to apply the proposed methodology step by step. The procedure has consisted in four stages: analysis, implementation, debugging, and finally, users' validation. The analysis phase was dedicated to study the process, identify machine states for representation, report main parameters and their interrelationships, as well as designing the simulator's main interaction schema. The second step covered the implementation of the simulator architecture in a software application able to manage all the process and machine variables, defined states and errors. After that, expert validation of preliminary results was performed. The next steps were debugging the HMI and inference engine until they "worked like" the real system. This item was checked by expert pilots and needed several test exercises and iterations. After these processes, the simulator's responses and the real machine responses were close enough to be used as a framework for modeling, training or predictive simulation.

Machine performance is heavily influenced by soil characteristics, geometric and physical properties. The most important ground or rock-mechanical parameters for machine – terrain characterization are: the grading curve, ground water penetration, the consistency limits, the rock/clay mineralogy, the quality of the rock, and the hardness of the rock. These parameters (or a subset among them) are used to adjust machine – terrain interaction. Grain size, over break, as well as lubrication or stoppages affect machine responses modifying predictable variables ranges, which is what the pilot must understand and recognize. As Figure 2 summarizes, the operator proceeds as he would do at the construction site, starting the machine from the instrument panel and loading the proper parameters to start the piping process. The operator must adjust pressures of jack cylinder, cutting head direction and speed, steering cylinders and slurry circuit valves and pumps to start operation. The interaction with the simulator framework, by means of a virtual cockpit panel replica, allows pilots to understand the machine state (guidance, jacking pressures, etc.). The machine responses shown in the data panel are generated by the inference engine.

One of the main characteristics of the proposed framework is that machine's inside parameters are hidden and only the relevant are displayed, as in the real machine. The observable parameters are displayed in the HMI's data panel. The controllable ones are set by means of the control panel replica. Exact copies of the control and data panels are the façade of the application. Both appearances and functionalities are identical to panels used in real machines. The functionality and model setup are performed and tuned in a specific module implemented in C++, denominated Machine–Terrain Interaction Module, presented in figure 2. This part will be explained in the following section.

The initialization data provided, by means of terrain setup files, are used by this module to adjust the machine – terrain responses to the user commands. These responses are based on the actual machine state and previous data, and return adequate values for each element of the data panel. An identical behavior to the real machine's instrumentation panel is accomplished in this way. The simulator architecture lies in graphical representations of the machine progress at the instructor module desk. It also has a data logger module that generates a log file to register time, machine states, pilot commands, terrain responses, trainer-generated events, and pilot corrections. The analysis of exercise's recorded data is useful to estimate time/cost deviations, and to evaluate pilots' performance and skills progress.

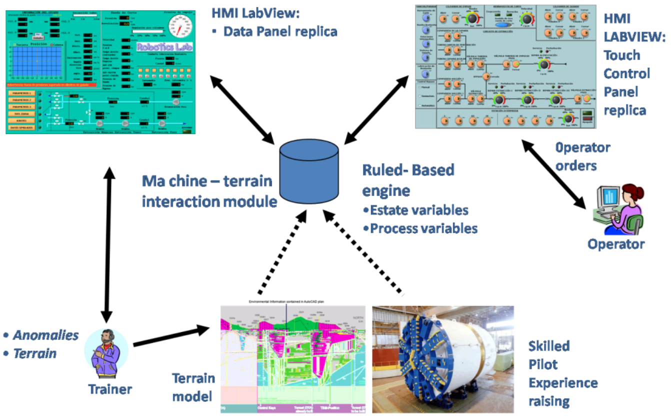


Figure 2: Proposed framework interaction schema

The developed simulator architecture is presented in Figure 3. The basic operation of the interaction module is now described. The basic pillars which shape the friendly human–machine interface can be seen in both sides of Figure 3. These are control panel and data panel, which are implemented using LabVIEW environment [25]. This graphical programming tool has been selected due to its powerful libraries to manage virtual instrumentation [29]. At the center, the inference engine is presented. This module receives references and setup values from control panel. Pilot commands are relative to the steering and adjust cutting and jacking parameters; these references are used to compute the machine responses and are presented at data panel by replicated instrumentation. The trainer can modify some of the data displayed to simulate instrumentation malfunctions, and can intentionally modify terrain parameters or machine state variables to generate anomalies and deviations to verify the training pilot reactions.

This modular architecture allows each part of the simulator to be modified without affecting the rest of the components. The inference engine can be updated without modifying the panel screens, by means of setup files. Details about the inference engine are presented next. Several visits to construction sites and pipe jacking pilot interviews have been made and crucial parameters and their relationships have been gathered. The interfaces and data exchange between modules has been designed to allow the interaction module be updated by means of C++ developed code compiled as a dynamic library.

This architecture easily accommodates formation programs in which the instructor can change the conditions faced by the student during the training to focus on correction of specific learning deficiencies. These formation programs can be gathered in basic training packages so teachers and students can follow a formation plan that is adapted to the current existing demand.

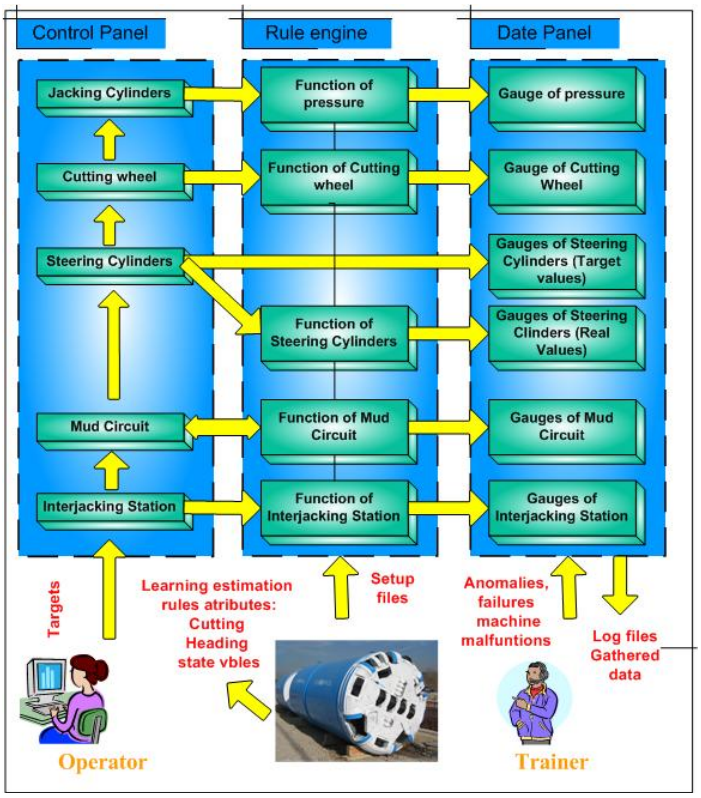


Figure 3: Simulator Architecture

4. Jack Piping Tunneling Machine modeling

As commented before, the first step is to understand the problem of steering a MTBM in order to be able to drive it. Several visits to tunneling workplaces were performed for “in situ” interviews of proven expertise pilots. Some of the maneuvers were also recorded to further analysis. The initial phase main target was to understand the machine overall and subsystems working principles. For example, pilots were asked how to perform corrections when path tracking error appears, which cockpit controls uses and about how they recognizes machine anomalies. From the pilot point of view, all the information about the machine’s state, variables and excavation process are presented at the data panel. These values are visualized in the data panel through the indicators, so pilot (and trainer) decodes data to deduce the machine state and correct the deviations. For example, the expert has report how to steering straight or how to achieve a controlled turn. The slurry circuit is the most influential subsystem for the machine performance. An adequate control of caudal and pressures are critical to avoid machine deviations. Typically, slurry circuit functionalities include three ways of by-pass operation functioning: manual, semi–automatic, and automatic. These slurry circuit working modes involves different behaviors of pumps and valves and a realistic representation on the instrumentation panels (as it is in a real construction site) is critical. Based on the expert know-how a simple but easy expandable structure of rules is generated and composes the rule-based inference engine.

4.1. Machine terrain interaction module

Several considerations are necessary for a complex control system in which the human pilot is an essential part, and the environment facing this human controller is so complicated, that no mathematical model exists. The experience of the pilots is usually expressed as some linguistic “IF-THEN” rules to describe actions to take in some situations. The behavior of the tunneling machine has been specified by the Specification and Description Language (SDL) due to the possibility of representing model’s description and its performance. SDL is a specification developed in the 70’s with the aim of describing communication among processes. The external behavior of the processes is modeled using a message chart. All process interaction takes place using messages, being defined to be ‘signals’ [27]. Not only was the use adapted for simulation, but it is being used in sociological research for describing social groups’ behavior. One of the most important advantages of SDL specification is its modularity, that is, each object can represent a new specification. The use of graphic diagrams to represent these modules facilitates the communication/understanding among beginners in software engineering and programmers, keeping it away from the literal language ambiguity [12]. This provided an excellent tool to capture pilot experience.

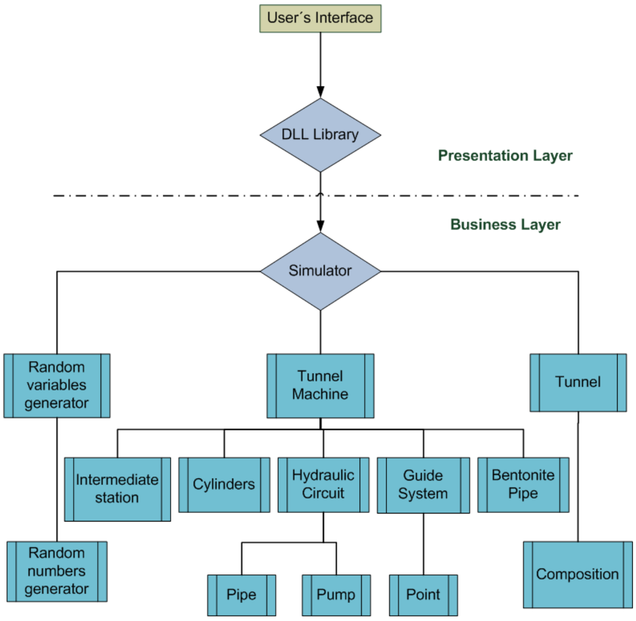


Figure 4: C++ microtunneling framework and application module hierarchy

The decomposition process provides a structure that covers the connection the with presentation layer (HMI) and a business layer composed by the simulator C++ modules, as shown in Figure 4. The simulator modules cover the main machine subsystems (tunnel machine module): the slurry circuit, piping cylinders, guidance cylinders, the main data of target reference and laser tracking system, slurry valves and pumps, bentonite injection, and also intermediate jacking stations. These items govern the internal behavior of the tunneling machine. This module sends messages for start/stop, drilling state, output parameters, error events and failures. The simulator block receives those data and changes the machine states, sending and receiving adequate data to HMI. Additional modules have been implemented for machine terrain modeling.

A special case is the execution of curved bore tunnel projects. Jacking force is a key consideration when specifying a microtunneling project. If the machine requires more force than the pipe being installed behind the MTBM can handle, the resulting stress will likely compromise the integrity of the pipe. A predictive calculation of the maximum effort can help avoid greater problems. The parameters are gathered and stored during simulation. The jacking pipe systems should limit ring stress when accommodating the necessary jacking force at the curve.

As previously mentioned, the terrain characteristics are also crucial for achieving a complete real simulation. The simplest strategy to define terrain properties is to consider it perfectly homogeneous; therefore the selected set of variables will keep its values along the entire tunnel. Anticipated conditions expected to be provided by the model would include: average performance, ranges and the most adverse conditions expected. The success of the predictive simulation is conditioned by the effectiveness of geotechnical data collecting method. Field and laboratory testing of samples from the tunnel envelope or local former records must include: index properties (water contents, unit weight, Atterberg limits...), grain size distribution, density test, viscosity, hardness and strength, and obstructions. A practical and more realistic approach is to introduce randomness in those parameters, in order to simulate heterogeneous terrains. A special set of C++ modules has been defined to model terrains and its variation from setup files that are loaded at initialization or any time during the simulation exercise.

4.2. *Rule–based inference engine implementation*

The rule based engine has been implemented as a state machine so each state represents a real machine situation and a particular set of rules is then applicable. The state diagram, shown on Figure 5, has five states, 1) Start: initial setup routines and tests are performed before drilling; 2) Moving forward: all machine subsystems must work within range; 3) Error: if a subsystem reports an error; 4) Break: simulates some machine subsystem failure and must be reset or break the simulation; 5) Pipe Changing: implies that a pipe has been inserted successfully and a new phase must start to continue the drilling process. These states are enough for fully characterizing the machine functionality.

In Figure 7, an example of implementation of the variables' influences between machine subsystems can be seen. The rules that compose each module of the inference engine are stored as a hierarchy of modules that communicate information to each other. For example, consider two machine subsystems: the steering system and the slurry circuit. The slurry circuit is also divided into two parts: pumps and valves. Every instance of the pump module knows everything about the other instance by means of each interface variable. The slurry circuit uses its pumps and valves to build its behaviour. At the same time, the tunnel machine gets the information from the slurry circuit and the steering system, among others, to modify its behaviour.

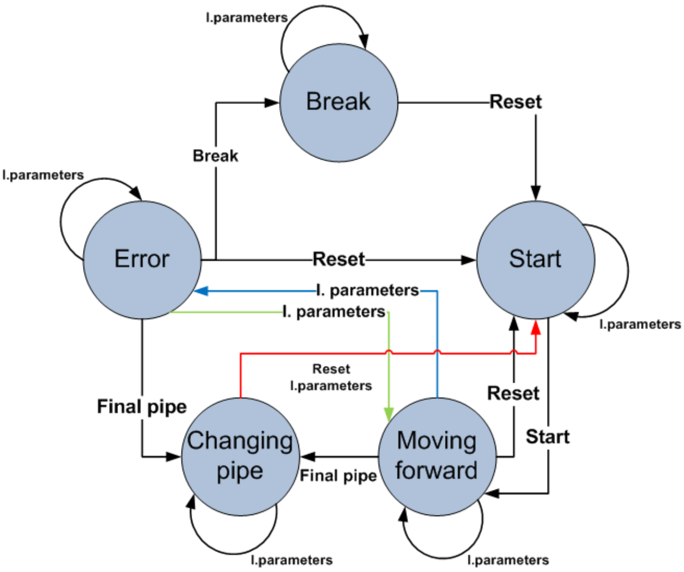


Figure 5: Simulated machine's states

Reciprocally, the general behaviour of the tunnel machine determines the behaviour of its components.

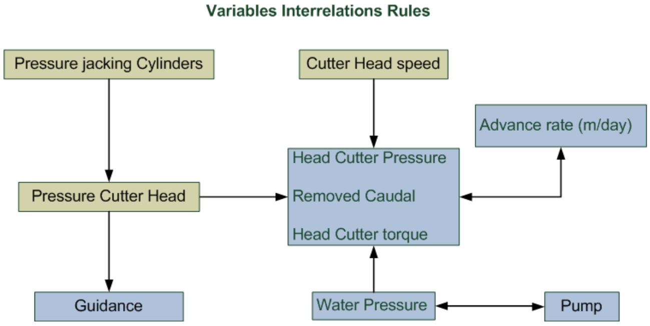


Figure 6: A simplified interrelationship schema of guidance related variables

In 'Moving forward' state, the control of head cutter pressure has been implemented using SDL diagram notation as shown in Figure 6. The left flowchart represents the activation of a rule for increasing or decreasing cutter-head pressure in function of by-pass state. The right flowchart shows how the rule for increasing cutter head acts: depending on the by-pass state and pilot commands, an upward deviation of the machine's head can be obtained. Additionally, an alarm can be shown to the user, and an offset shown in the guide system. This is an example of a specific rule and the actions that are taken. Similar procedures have been adopted for the main subsystem of the machine.

Until now the model is used to simulate the machine – terrain interaction in a homogeneous soil of sands and gravels, but the module for interaction for rocks and clays is under development.

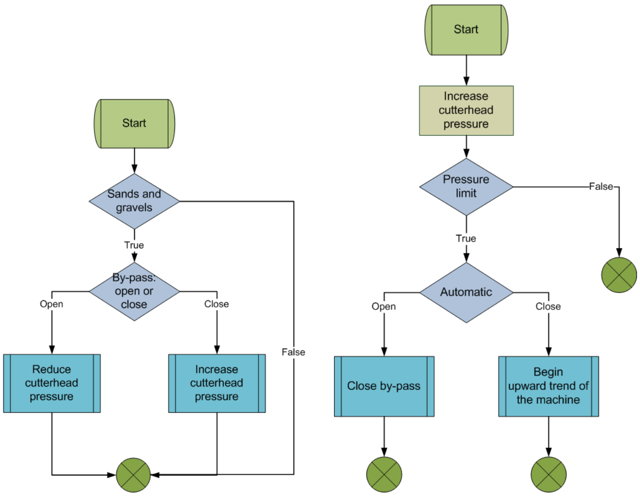


Figure 7: Implemented rule's flowcharts of slurry affectation over guidance

5. Friendly HMI

One crucial issue was to design a model that would replace the real machine to interact with the human operator. Hence, the graphical front-end is an exact replica of cockpit machine controls that communicates with the simulator rule engine. The Human-Machine Interface provides the functionality to simulate the control of the cutting wheel, the jacking cylinders, the steering cylinders, and slurry circuit. These systems are essential to obtain the advance of the machine. The HMI's screens are almost exact copies of the controls used in the real jack pipping microtunneling control cabins. These control cabinets provide all the instrumentation accessible to the pilot in two different panels, both with a schematic representation of the machine main systems. The upper one is the Data Display, and presents analog, digital and steering displays; and the bottom one is the Control Panel, and is provided with control mechanisms: potentiometers, switches and buttons. Comparing the screen images with those of real panels (Figures 9 and 10); it is possible to verify that the reproduction created can help the pilot to easily get familiar with the controls, enhancing the learning experience.

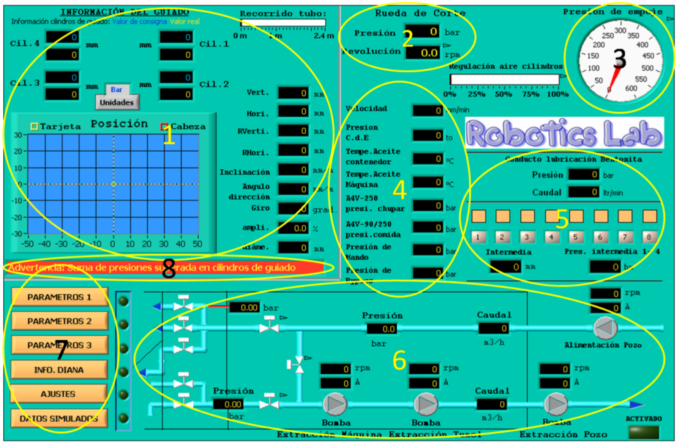


Figure 8: Virtual Data Panel: detail of functional areas

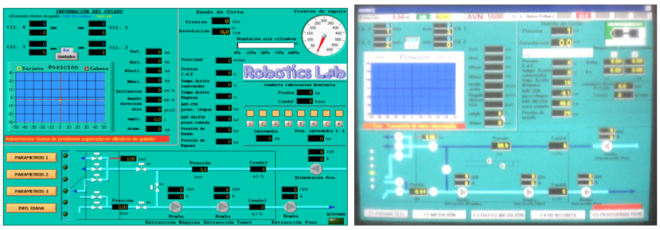


Figure 9: Virtual Data Panel (left): comparison with real one (right)

The details of the Data Panel screen are shown in Figure 8. It is formed by a set of 8 mains areas: 1) Guidance information area: shows the position and orientation of the tunneling machine in the tunnel; 2) Cutting wheel area: current state of interaction between the cutting wheel and soil parameters; 3) Jack cylinders pressure gauge area; 4) Auxiliary parameters area: it allows monitoring the general state of the machine, like the temperature and its mean; 5) Pipeline bentonite injection and intermediate jacking stations area: indicates injection parameters and which station is selected and their pressures; 6) Slurry circuit area: shows pumps activity, caudal, pressures and valves state; 7) Parameter buttons to set the simulator working conditions; 8) Message window area to display help and error messages.

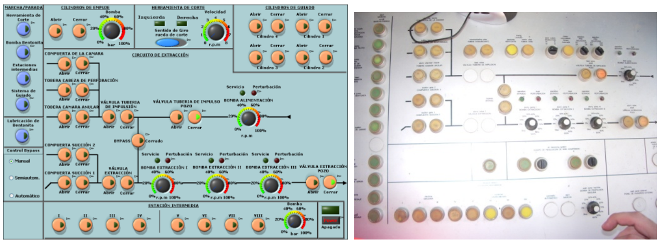


Figure 10: Virtual Control Panel (left): comparison with real one (right)

The details of the Control Panel screen are shown in Figure 11. The control panel is integrated in an industrial computer with a touch screen enabling interaction with the controls using

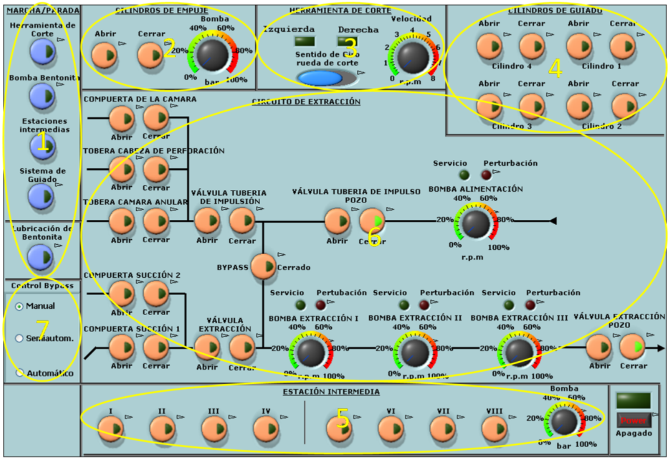


Figure 11: Virtual Control Panel: detail of functional areas

stylus or fingers. The control screen is formed by the following control areas: 1) Start/stop area: to enable the subsystems indicated on each button; 2) Jack cylinder area: to set jacking cylinder pressures; 3) Cutting head area: speed set point, indicator and clockwise/counterclockwise selector; 4) Steering cylinders area: push-buttons to increase or reduce pressure set points and actual pressure values; 5) Intermediate jacking stations area: pressure set point and activation buttons of each intermediate jack station; 6) Slurry circuit area: it contains all the controls needed to regulate the slurry circuit, extraction pumps, valves and nozzles; 7) By-pass circuit control area: where the by-pass control mode is selected.

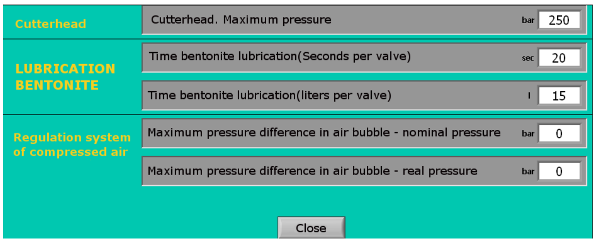

Cutterhead
Cutterhead. Maximum pressure bar 250
LUBRICATION BENTONITE
Time bentonite lubrication(Seconds per valve) sec 20
Time bentonite lubrication(liters per valve) l 15
Regulation system of compressed air
Maximum pressure difference in air bubble - nominal pressure bar 0
Maximum pressure difference in air bubble - real pressure bar 0

Figure 12: Excavation parameters setup screens

Inside the data screen, additional buttons allow access to additional windows to introduce values and ranges depending on terrain and project specifications. One of these setup parameter screens is shown on Figure 12. A global view of the system (real and simulation) can be seen in Figure 13.

6. Training process

Performance enhancements are a well-known advantage of simulator training. The key to acquiring the necessary motor skills to control complex systems, such as aircraft or heavy machinery, is hands-on and coached training. Researchers have shown that flight simulators effectively improve pilot performance related to landing skills [15] and instrument and flight control abilities [26]. Training using virtual environments has also been demonstrated to improve user performance in closely related developments as Aitemin's mining simulator [2] at Laredo Foundation, where extensive courses have been passed, or as reported by Bernold [7],[9]. They have reported that skills that were learned in a high acquisition environment (e.g. extensive training in a simulator) are retained longer than skills learned in a low acquisition environment (e.g. one-day seminars). Therefore, when measuring skill retention, it is just as important to measure skill acquisition (degree to which the skill has been effectively learned), as it is to measure post-training performance [16].

Tunneling with a MTBM, the body of the machine is held in the tunnel by one or more pairs of pipe jacking cylinders, located at the back of the machine, that push the machine to advance. After each perforation cycle, a new pipe segment is installed. Typically, steering methods are based on bending the cutter-head. The preferred method of steering consists in turning the axis from the machine by displacing the main beam with respect to the rear jacking unit, actuating the steering cylinders, shown at Figure 14, with the result that the machine pivots about the forward gripped region. To achieve a sharper turn, the pilot must hold control of the orientation of the machine. Steering data is obtained by observing the position of a laser beam spot cast on a target mounted on the machine, having a laser source being mounted on a fixed support at the rear end of the tunnel. Using this data source, the pilot must interpret the goodness of the actual guidance and modify it, if necessary, using the control panel to set the tracking error lowest as possible, getting the supervision of the expert pilot that acts as a coach.

Figure 13: Excavation data gathering phase and evaluation in the training place scenes

7. Experimental results

This framework and application has been implemented for fitting the two main objectives: for instruction, and as a predictive analysis tool. In this sense, to simulate the execution of the tunnel, the machine – terrain interaction has been completed with enough reliability to fit terrain characterization based on the rule inference motor. This, therefore, is the motor of the framework as a predictive analysis tool. Furthermore, an accelerated time scale can be applied to any of the two uses. In a real construction site, pipe jacking is a time-consuming process. The accelerated time scale allows reducing this time adjusting the simulation rate (x2, x4, x8...). Recorded data outputs can be used to check technical viability, or verify or modify cost and scheduling problems.

The performance of the simulated virtual machine can be measured with the same metrics as real ones. This can be defined entirely by 3 variables: penetration, utilization rate, and cutter costs. Additionally, for MTBM, the amount of successfully inserted tubes gives a quick overview on the performance. The average penetration rate is defined as meters per time [m/hr]. Other interesting parameters can be obtained from the average penetration, such as the instantaneous rate [mm/rev] (the depth of cutter penetration/revolution) or the mining rate [m^3/h] (calculated as volume of soil excavated per elapsed mining time) [%]. The utilization is the elapsed machine time per excavation shift time ratio. Cutter costs are measured related to cubic meters or linear advance meter [in/m^3]. The average advance rate [m/day] is obtained as penetration rate multiplied per 24h and utilization rate, or is calculated as pipes per time unit, normally per day.

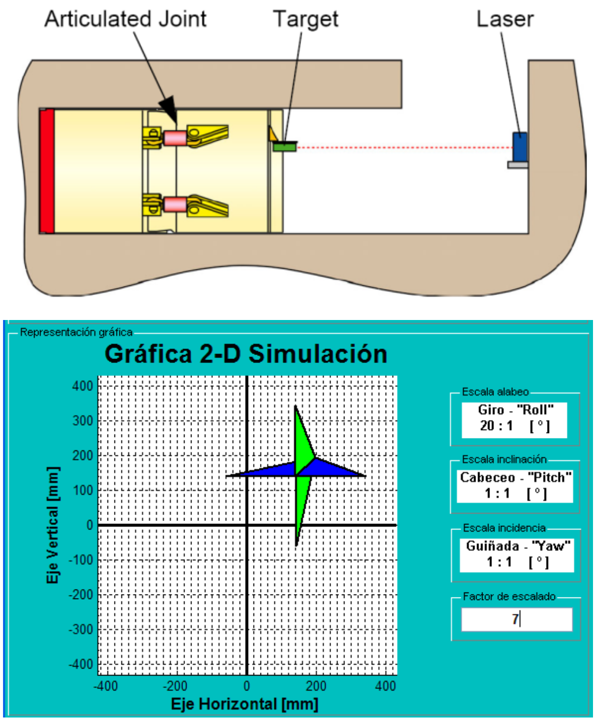


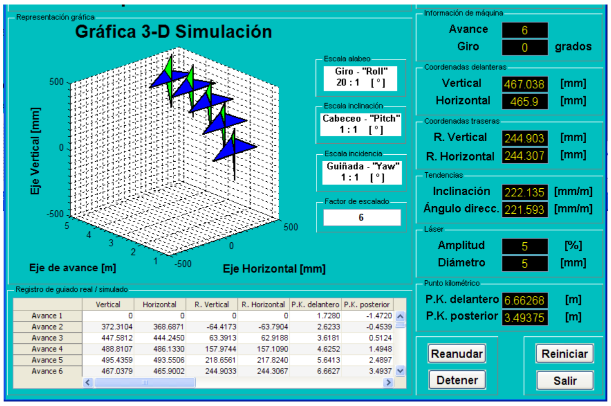
Figure 14: Schematic view of steering cylinders and detail of HMI panel for 2D vertical and horizontal deviations display

Though some additional functionalities are still under testing, such as a new module to replicate control commands previously recorded to deal the virtual machine into a known state, some results have been obtained, as summarized in Table 1. Here, five pilots of the Eurohinca company (three skilled ones, S_i; and two novices, N_i) have used the simulator in accelerated time mode. The simulator allowed them to pause the exercise and restart when they desired. Errors were displayed with messages while allowing to continue operation except in case of fatal error (e.g. due to a pump failure due overpressure). The following acronyms are used within the table: NFE (number of non-fatal errors); MTBF (mean time between failures, expressed in hours); FE (number of fatal errors); FP (number of pipes jacked at full performance); RJP (real pipes jacked); ACCR (acumulated equivalent time, to summarize each pilot experience total time with the simulator); MP (mean penetration, expressed in meters per hour). The presented results show that after several sessions the novice pilots achieved similar scores to those of senior pilots.

Pilot	NFE	MTBF	FE	FP	RJP	ACCR	MP
S1	1	48	0	6,0	5,1	48	265,6
S2	2	18	1	4,5	2,8	36	192,7
S3	4	13	1	6,6	4,7	53	221,9
N1	6	8	3	6,0	3,7	48	192,7
N2	15	3	4	6,0	4,2	48	219,6

Table 1: Preliminary trials results in heterogeneous soil

Figure 14 shows the 2D guidance display area and steering cylinders location. A partial view of results data file is shown at 15, and at the upper left side a 3D plot generated from the recorded data of an exercise. At an industrial level, steering adjustments on the machine are currently performed manually, although some research on fuzzy assistants for steering has been published [17]. At research level, automated systems are under development. Due to predictive capabilities, the developed framework supposes a first approach for designing advanced navigating systems, as it has set a reliable simulation of the machine at work. Automatic control would be focused on helping human pilots, not to remove them, and this will consequently increase productivity and steering precision for tunnel execution.



Información de máquina

Avance 6

Giro 0 grados

Coordenadas delanteras

Vertical 467.038 [mm]

Horizontal 465.9 [mm]

Coordenadas traseras

R. Vertical 244.903 [mm]

R. Horizontal 244.307 [mm]

Tendencias

Inclinación 222.135 [mm/m]

Ángulo direcc. 221.593 [mm/m]

Láser

Amplitud 5 [%]

Diámetro 5 [mm]

Punto kilométrico

P.K. delantero 6.66268 [m]

P.K. posterior 3.49375 [m]

Registro de guiado real / simulado

	Vertical	Horizontal	R. Vertical	R. Horizontal	P.K. delantero	P.K. posterior
Avance 1	0	0	0	0	1.7280	-1.4720
Avance 2	372.3104	368.6871	-64.4173	-63.7904	2.6233	-0.4539
Avance 3	447.5812	444.2450	63.3913	62.9188	3.6181	0.5124
Avance 4	488.8107	486.1330	157.9744	157.1090	4.6252	1.4948
Avance 5	495.4359	493.5506	218.6561	217.8240	5.6413	2.4897
Avance 6	467.0379	465.9002	244.9033	244.3067	6.6627	3.4937

Figure 15: 3D plot of a MTBM trajectory form simulator Guidance Area: blue and green arrows show the executed path

Initial objectives have been fully achieved with this first version of the friendly framework for modeling, interfacing and training of tunnelling machines in heterogeneous terrains. The right emulation of the machine performance has been reviewed by the evaluation of expert workmen in machine tunnelling. A “stand alone” version of the software (usable without the development kit) has been distributed to some experts and they have performed some exercises. They have all agreed that usual ranges of the main variables are reflected through the interfaces, and the responses to set points are close enough to those presented in real machines.

8. Conclusions

A framework and application has been presented for modeling and creating a machine that simulates the jack piping tunneling process in heterogeneous terrains. This replica has been integrated as a simulator which is composed by a rule-based inference engine and an HMI that closely resembles a real MTNM cockpit cabinet. These components have been physically arranged in a PC-based training desktop formed by a comfortable disposition of screens. This smart composition of data panel and touch screen allows natural interfacing. The result is a training tool which resembles the real working place of pilots operating a microtunneling machine. The simulation framework confronts the trainee not only with scenarios frequently encountered in real tunnel sites as drift-off on the machine from the planned excavation trajectory or blocking the slurry circuit, but also with serious incidents such those in which the machine can get stuck. This avoids the common lack of realism held by other construction site simulators and will allow the most rapid and effective training for future workmen by means of the following features: a) replica of all the controls and indicators; b) machine boot sequence recreation; c) slurry circuit & by-pass control mode recreation; d) clay and gravel behavior; e) pipe changing emulation; f) control system for the intermediate stations; g) implementation of frequent anomalies such as slurry circuit blocking, states of unexpected deviation and blocking and draft of the machine, implementation; h) assuring safety and low cost of the training as no real machine and excavation are needed.

Simulator training is often self-paced, allowing users to train when they are ready rather than when the skilled pilot and equipment are available. The training effectiveness using the simulator developed is still being evaluated, primarily because it is time-consuming and expensive to conduct tests on large numbers of machine jacking pipe pilots. Some enterprises have started to use this development and soon they will report with more detailed evaluation analysis.

The presented approach emphasizes simplicity and practicality, while limiting input data to that which is readily available from geological prospects or historical records. A following paper will be focused on the results of the validation study of the model developed, reporting the results of a comparison of the model's predictions with the outcome of real tunnel execution data, which must cover a wide range of project requirements, soil conditions and environmental constraints.

Acknowledgment

Special thanks to the Robotics Lab team and specially to the Eurohinca company staff, without their help it would not have been possible. This work has been funded by the Spanish government under the framework of the Multidimensional City project PSE-380000-2008-5 http://www.laciudadmultidimensional.es.

[1] Acaroglu, O. & Ozdemir L. (2008). A fuzzy logic model to predict specific energy requirement for TBM performance prediction. Tunneling and Underground Space Technology, Volume 23, Issue 5, Pages 600-608, Asbury. September.

[2] Aitemin (2005) Virtual reality mining machinery simulator, Project report. Retrieved June 08 from Web:http://www.aitemin.es/ei_sim_rv_maq_min_e.html. Last visit: June 09.

[3] Akiba, T. & Yamada, T. (1989), The Enviliner Microtunneling system: NO-Dig 89, Fourth International Conference and Exhibition on Trenchless Construction for Utilities, London, ISTT, p. 243-245.

[4] Anderson, H., Euhus, L., & Parker, D. (2002). Utilizing simulators in operator training. Retrieved June 5, 2002 from Web: http://www.pacsim.com/Articles/OpTrain/art1tr.htm. Last visit: Sept 08.

[5] Atlas Copco Rocket Boomer L2C Simulator, (2008) Atlas Copco internal report, http://www.atlascopco.es/. Last visit: Nov 08.

[6] Beaumont, T. C., (1989), UK experiences and developments of the Witte microtunneling machines: NO-Dig 89, Fourth International Conference and Exhibition on Trenchless Construction for Utilities, London, ISTT, p. 237-242.

[7] Bernold, L. E.; Lloyd, J.; Vouk, M. (2002) Equipment Operator Training in the Age of Internet2.

[8] NIST SP 989; September 2002. International Symposium on Automation and Robotics in construction, 19th (ISARC). Proceedings. National Institute of Standards and Technology, Gaithersburg, Maryland. September 23-25, 2002, 505-510 pp, 2002.

[9] Bernold, L.E. (2005). A Paradigm Shift in Construction Education is Vital for the Future of Our Profession, J. Constr. Eng. and Mgmt., ASCE, Vol. 131, No.5, April, pp. 535-542.

[10] Bourg, D.M. (2002), Physics for game developers, ISBN 0596000065, O'Reilly Media.

[11] CCITT Recommendation Z.100 (1988), Specification and Description Language SDL, Annexes A-F to Z.100, ITU, Geneva.

[12] Fonseca, P. & Casanovas, J. (2006), Using generic event for a simple reflexive intelligent agent SDL specification. European Modeling & Simulation Symposium. Barcelona.

[13] Geost (2008), Grupo Español de obras Subterráneas, Memoria general del proyecto estratégico de La ciudad Multidimensional. Web: http://www.construccion2030.org. Last visit: Nov 08.

[14] Hayashi, M. & Miyata, Y. & Okumura (1989), Markham super mini shield tunneling method: NO-Dig 89, Fourth International Conference and Exhibition on Trenchless Construction for Utilities, London, ISTT, p. 231-235.

[15] Hays, R. T., Jacobs, J. W., Prince, C., & Salas, E. (1992). Requirements for future research in flight simulation training: Guidance based on a meta-analytic review. *The International Journal of Aviation Psychology, 2*, 143-158. Web: http://www.trainingsystems.org.

[16] Healy, A. F., & Sinclair, G. P. (1994). Long term retention of trained skills (Technical Report No 94-001). Orlando, FL: Naval Air Warfare Center Training Systems Division. (DTIC No. ADB187295).

[17] Herrenknecht AG. (2008). Universal Navigation System. Technical data sheet. Web: www.herrenknecht.com. Last visit: Oct 08.

[18] Iseley, T., Gokhale, S.B., (1997). Trenchless installation of conduits beneath roadways. NCHRP Synthesis 242. Transportation Research Board/National Research Council, Washington, D.

[19] Manabe, T. & Awata, T. & Matsumoto, A. & Sako M. (1998), Automatic direction control technique for pressing type small diameter tunneling machine. In proceedings of NO DIG International Conference and Exhibition on Trenchless Construction for Utilities, Laussane.

[20] Manabe, T. & Tanno, M. & Matsumoto, M. & Yabuta, T. (1999), Automatic direction control technique for microtunneling machine. IEEE Industrial Electronics Society, 1999. IECON '99 Proceedings. The 25th Annual Conference of the IEEE. Vol. 3, p 1295-1300. San Jose, CA, USA. ISBN: 0-7803-5735-3.

[21] Martí, M. (2002), Estudio de las tuneladoras y de los sistemas de guiado, con aplicación en diversos túneles, y control dimensional de una tuneladora. Topografía y cartografía: Revista del Ilustre Colegio Oficial de Ingenieros Técnicos en Topografía, ISSN 0212-9280, Vol. 19, N 112, p. 4-9. Last visit June 2009.

[22] Martí, M (2006), Listado de referencia de obras completadas. Report from Web: http://www.eurohinca.com/pdf/listado.pdf. Last visit: June 09.

[23] Moria, H. & Sawaguchi, A. (1989), Microtunnelling system with directional control and long distance driving: NO-Dig 89, Fourth International Conference and Exhibition on Trenchless Construction for Utilities, London, ISTT, p.273-282.

[24] Narendra, K. S. & Parthasarathy, K. (1991), Identification and control of dynamic systems using neural networks, IEEE Trans. Neural Networks, vol. 1, pp. 427.

[25] Gary W. Johnson. (1997). LabVIEW Graphical Programming: Practical Applications in Instrumentation and Control (2nd ed.). McGraw-Hill School Education Group. Labview book Last visit October 2009.

[26] Pfeiffer, M. G., Horey, J. D., & Butrimas, S. K. (1991). Transfer of simulated instrument training to instrument and contact flight. The International Journal of Aviation Psychology, 219-229. http://www.trainingsystems.org Last visit July 2010.

[27] SDL (1988); ISO TC97/SC21: LOTOS, A Formal Description Technique Based on the Temporal Ordering of Observational Behaviour, ISO/IS 8807.

[28] Thoroughtec (2008), http://thoroughtec.com/products/cyberMINE/ Last visit April 2011.

[29] Travis J., Kring J, (2006): LabVIEW for Everyone: Graphical Programming Made Easy and Fun, 3rd Edition, July 27 2006, Prentice Hall. Part of the National Instruments Virtual Instrumentation Series. ISBN 0-13-185672 National Instruments Virtual Instrumentation Series Books Series.

[30] Var-trainer (2007) Collective Research project VAR-TRAINER (Versatile augmented reality simulation for training in the safe use of construction machinery). Project report form http://www.ikerlan.es/vartrainer/. Last visit October 2008.