NISTIR 4596

Task Decomposition and Algorithm Development for Real-Time Motion Control of a Continuous Mining Machine

Hui-Min Huang
John Albert Horst
Richard Quintero

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
Unmanned Systems Group
Robot Systems Division
Bldg. 220 Rm. B124
Gaithersburg, MD 20899

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
John W. Lyons, Director

NIST

NISTIR 4596

Task Decomposition and Algorithm Development for Real-Time Motion Control of a Continuous Mining Machine

Hui-Min Huang
John Albert Horst
Richard Quintero

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards
and Technology
Unmanned Systems Group
Robot Systems Division
Bldg. 220 Rm. B124
Gaithersburg, MD 20899

May 1991

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
John W. Lyons, Director

Task Decomposition and Algorithm Development for Real-Time Motion Control of a Continuous Mining Machine[1]

Hui-Min Huang
Mechanical Engineer

John Albert Horst
Electronics Engineer

Richard Quintero
Supervisory Electronics Engineer

Robot Systems Division
National Institute of Standards and Technology

[1] This work is sponsored by the U. S. Bureau of Mines under Interagency Agreement (Number J0189027).

CONTENTS

ABSTRACT

The drive toward increased safety for coal miners has led to the development of computer-assisted methods of underground coal mining. The development of control architectures and accompanying code for the control of the movement of continuous mining machines (tramming control) is an important part of this overall effort. The tramming control algorithm design described is in concert with hierarchical architecture design principles developed at National Institute of Standards and Technology (NIST), referred to as the Real-Time Control Systems (RCS) methodology. The algorithm design with accompanying code allows for the control of both cutting and free-space movement by a continuous mining machine and allows for a high degree of human operator interaction.

1. INTRODUCTION

The National Institute of Standards and Technology's (NIST) Robot Systems Division has been involved in the support of the United States Bureau·of Mines' computer-assisted underground coal mining research effort since 1988. The Mining Automation Standard Reference Model (MASREM) [Hu 90-1, Al 89][2] has been developed to define the overall conceptual framework of a hierarchical architecture for a mining automation control system. MASREM adapts the Real-Time Control System (RCS) [Al 82] architecture to coal mining. A task decomposition methodology has been developed [Hu 90-2, Hu 90-3] to describe the design procedure for a hierarchical computer-assisted coal mining control system.

Based on this previous work, this paper describes our effort to develop a software control algorithm for the motion control of a Joy 14CM[3] continuous miner (a continuous mining machine, see figure 1) [Jo 82]. This implementation is referred to as the "tram controller" for the machine and the corresponding control activity is referred to as "tram control." The controller provides closed loop control for all the movement (or "tramming") related tasks that the machine performs.

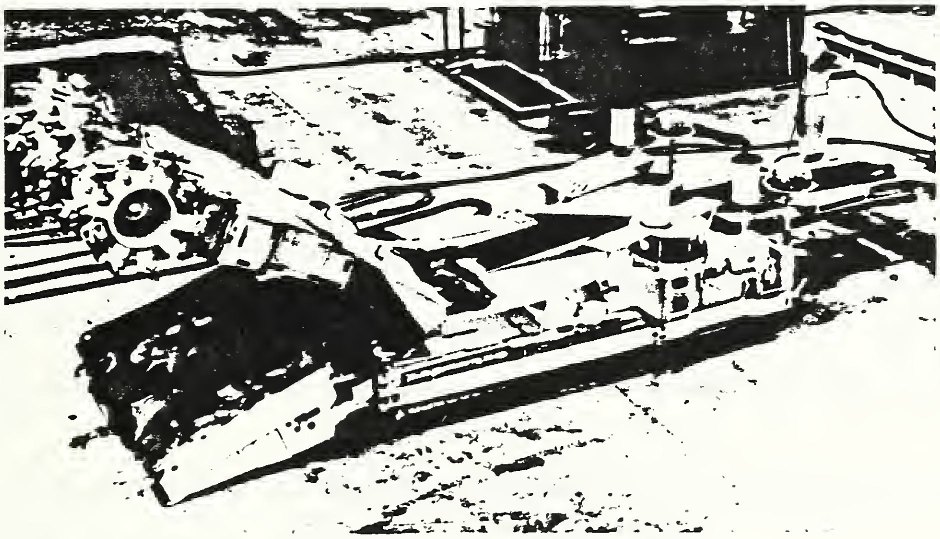


Figure 1: A Joy 14CM is shown as configured by the U. S. Bureau of Mines to perform computer-assisted closed loop tramming. All the tramming functions are computer controlled onboard the Joy 14CM. The loop is closed with measurements of the position and orientation made by several sensor systems (courtesy of the U. S. Bureau of Mines Pittsburgh Research Center).

A careful study of the above-mentioned mining machine and what might be typical conditions in a mine was undertaken. The current testbed being implemented by the U. S. Bureau of Mines (BOM) researchers [Schi 90] was also studied. An overview of the mining environment along with term definitions is presented (in appendix I) for those who are not familiar with them. A brief description of BOM's testbed is also given. The RCS design method, on which the development of this tram control algorithm is based, is briefly described in which the establishment of the context for the tram controller is the first step.

[2] Denotes the references listed at the end of this paper.

[3] References to product or company names are for identification only and do not imply government endorsement.

Since error control is the main objective for this tram controller, the sources of errors which may occur during machine motion are elaborated, and the consequent main control strategy is described.

A brief description is given with an attempt to map this control algorithm to a previously defined RCS architecture for mining automation. The tasks to be performed at each level of the tram control architecture are defined. The activities at each level are described using state transition diagrams. The corresponding program design is described, from which computer programs have been generated. These programs can be referenced through the authors.

2. THE JOY 14CM CONTINUOUS MINER

Continuous mining machines are typically used in room-and-pillar mining as well as in the development of entries for long-wall mining. The major components of a Joy 14CM continuous miner (CM) that directly relate to tramming control include (figure 2):

* A Tramming Subsystem: A tram motor driving a tread set exists on both sides of the machine.

* A Cutter Drum: A hydraulic actuated cutter boom extends out at the front of the machine. Attached to the front of the boom is an electrically operated cutter drum. Replaceable cutting bits are installed at the surface of the cutter drum which fracture the coal as the drum is pushed into the coal face while turning.

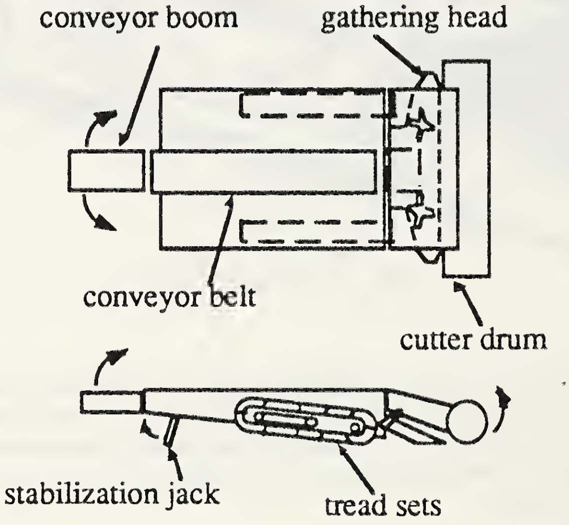


Figure 2: The Major Components of a Continuous Miner

* A Gathering Head Subsystem: This subsystem is located at the bottom of the front end of the machine. A gathering pan can be set to float on the floor. The gathering head, using a rotary motion, scoops the coal inward onto the gathering pan. A conveyor belt is behind the gathering head and moves the coal to the rear of the machine.

* A Conveyor Subsystem: The conveyor extends from the gathering head to the rear of the machine. An adjustable position conveyor boom forms the end of the

conveyor system. It can move from right to left as well as up or down. Coal is dumped from the conveyor boom onto a haulage unit behind the CM.

* A Stabilization Jack: This hydraulic jack provides a stabilizing force to counterbalance the cutting force.

The continuous mining machine has ten tram control commands: slow/fast speed forward, slow/fast speed reverse, pivot left/right, turn left/right forward, and turn left/right reverse. These are open-loop commands. Execution of any of these commands can be terminated by either a stop command (implying the tram control loop is closed at a higher level where the sensory information is processed), or by a condition that some maximum time has expired (a safety time-out condition associated with this command).

3. THE U. S. BUREAU OF MINES TESTBED -- BOM/NET

The U. S. Bureau of Mines has been implementing a computer control system testbed. This testbed, referred to as BOM/NET [Schi 90], is a distributed network with a maximum capacity of sixteen nodes. The continuous mining machine, various sensor systems (the laser, the gyro, the linear cable transducer, etc.), and an operator console are all nodes on the network. The tram controller will also be a node. The tramming commands conform to a pre-established BOM/NET communication protocol [Schi 90].

4. TASK DECOMPOSITION AS A METHOD FOR DEVELOPING THE TRAM CONTROL ALGORITHM

The RCS methodology developed at NIST specifies a three-part functional breakdown horizontally at each hierarchical level into sensory processing, world modeling, and task decomposition. Generally, the task decomposition function in an RCS control system refers to the system's capability to execute predefined tasks at each level of the hierarchy.

When the tram control algorithm was designed, answers to the following questions were searched for:

* how does the machine behave given its mechanical/electrical/hydraulic design,

* what are the tasks that the machine must perform, and

* what is a natural hierarchical organization of those tasks?

This entire activity may also be referred to as the task decomposition method or process.

Internal to an intelligent RCS, hierarchical and heterarchical (at a same level) task decomposition (including temporal and spatial task decomposition) occur. Task decomposition between any two successive levels may be described as follows: the higher level sends down "what needs to be done" and the lower level generates "how it is to be done."

The task decomposition method can involve an iteration of the following steps:

* Establish the context. In designing an RCS, the definition of context is the first step. This design step includes the establishment of the system's objectives, the problem domain, the constraints, and the assumptions. This design step must also include a narrative description of the approach selected to achieve the system goals. Scenarios are often used to develop typical operational descriptions.

* Develop an organizational hierarchy. A first sketch of the system's architecture is developed to serve as a foundation for design work. Such a hierarchy takes into account the system's goals, the environment, the existing facility, and other factors such as the pre-defined functional requirements for each of the RCS levels [Hu 90-3]. Figure 3, developed earlier [Hu 90-3], shows the lower levels of an entire mining automation control system hierarchy. The tram controller is a part of the "vehicle piloting" control module[4].

* Perform task analysis. Knowledge necessary to perform tasks, including the machine capability and the operational environment, is acquired and assimilated along the following lines: what are the activities each control module can perform, what are the associated constraints, and what is the information required to perform a given task? Task commands for each module at each level are developed in this design step.

* Develop RCS plans. Task commands defined above are used to develop RCS plans using state transition diagrams. These RCS plans describe how higher level tasks are decomposed into lower level tasks, and how the constraints for the commands are implemented as transition requirements among the different states.

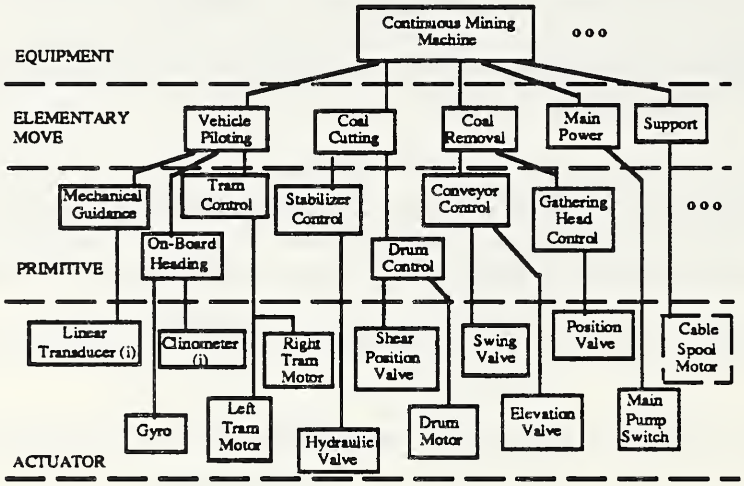


Figure 3: Mining Automation System Hierarchy (Lower Levels)

5. THE CONTEXT FOR THE TRAM CONTROLLER

The primary responsibility for the tram controller is to move the machine to desired locations. During the performance of tram control, problems including tread slippage along the lateral or rotational directions may be encountered. Tram control is a fundamental problem in that it interacts with other aspects of the machine control/mining process (such as the control of the conveyor boom and the cutter drum). Therefore, issues such as problem scope and developing a typical scenario all have to be established before task analysis can be performed.

[4] The pilot/guidance module has additional capability such as obstacle avoidance.

5.1 Scope for Tram Control

Closed loop tramming for continuous mining machines can generally be categorized into the following two types:

* Free-Space. Free-space tramming can allow for the machine to navigate to a different panel or a different section at far distances. The performance of such tasks typically requires multiple levels of path planning. The current focus, however, is in the vicinity of the face area. Obstacle avoidance is assumed to be a human operator function.

* Cutting. The width of entries usually requires the machine to make two passes to make a single cut. The current focus for cutting tasks is on the first pass of the cut. The length of a cut can be a variable (see appendix I) to be specified by operators. However, for convenience, a 5 meter cut is used throughout this paper.

The scope of the tram controller is further defined by existing constraints and assumptions (they are also important references for the tram controller to be integrated into the overall system [Schi 90]). The following pre-conditions apply in this tram control design:

* A coal haulage unit is available to transport coal to a main coal transportation conveyor system.

* No obstacle avoidance is involved.

* No Coal Interface Detection (CID) system [Sch 89] is employed. The formation of the coal seam to be cut is assumed to be regular and in extractable condition.

* The machine will pivot only to adjust its orientation (the term "yaw" is used interchangeably with "orientation"). Assume that the pivot action does not change the position of the machine.

* The pivotal point for the machine is used as a reference for the machine position.

* This tram controller must be integrated into the existing BOM/NET system. The BOM/NET system includes the computer commands for tramming control. Therefore, the lower level constraint for the tram controller is that its output should be compatible with the BOM/NET command structure. The tram control commands in the primitive level plans will be sent down through the BOM/NET to its destination nodes. Therefore, these commands should conform to the BOM/NET specification.

* The operation of the stabilization jack is crucial to tramming control and is included as a part of the tramming control.

The tram controller assumes additionally the existence of the following conditions if it is commanded to perform a cutting task:

* The coal seam has been surveyed and the cutting location identified. A cutting path coordinate system has been defined (see section 5.3).

* The machine is assumed to be located at the origin of the cut coordinates.

* Adequate machine power for the electrical, hydraulic and mechanical systems

exists.

- * The cutter drum is at the cut-height position.
- * The gathering head is in a floating position.

5.2 Breakdown and General Description of Machine Movement Control

In order to accomplish free-space tramming tasks, the machine will be given goal position coordinates. The machine will then tram to the position through sensory interactive, closed loop control. Typical machine movements include pivoting and pointing the machine towards the position, and tramming-forward to approach the goal. Maximum errors allowed for pivot and tram-forward/reverse will be given. Error boundaries for the goal positions and yaw will also be given. Sensory data acquisition rates (sampling rates) must be fast enough to prevent position overshoot in machine motion. Since the speed of the machine is comparatively slow, the sampling rates are not anticipated to be a problem. Section 6 describes the error control issue.

For cutting tasks, the machine will receive a desired cut distance and repeat as many times as necessary a so-called sump-and-shear cycle, which includes the following primitive functions: approach the face, sump into the coal face by tramming forward, shear coal by moving the cutting drum down to the floor, gather the cut coal and move it to the rear of the machine, then cut the remaining coal on the floor while tramming in reverse (see sections 7 and 8 for definitions). Similar error control (as in free-space tramming) is used. However, more frequent correction activity is expected (due to the cutting process).

A combined scenario may include first of all a free-space-tramming task to move the machine to the coal face followed by a cutting task to extract the coal.

5.3 Cutting Task Coordinate Frame

As described in the previous work [Hu 90-3], successive transitions in coordinate frames or resolutions can be seen among different levels in the hierarchical control architecture. Higher levels are concerned with larger areas but with coarser resolution, and a global coordinate frame is generally used. At the lower levels, machine centered local frames are used. In a global frame, further subclassification in terms of resolution typically is required for different levels[5].

At a higher level, a mine map might refer to objects such as panels. At a lower level, a map typically has a finer resolution. Objects within a panel would be referenced, such as a specific pillar or entry. At another level down, pillars may be represented in even finer detail. The shapes of objects (e.g., pillars) may be represented in vectors, arrays (polygons), or a quad-tree spatial data structure format [Sa 89].

The tram controller uses a local path coordinate frame with origin at the starting point of either a cutting or a free-space tramming task. The vector from the starting point to the goal is used as the forward (or Y) axis (figure 4). The starting point for a cut is set at a point such that the cutter drum is about one meter away from the coal face (instead of where the drum touches the face). The reasons are:

[5] A similar transition in time domain can also be seen. The upper levels deal with plans or events that cover greater periods of time but with less detail, while the lower levels deal with specific tasks or events that cover shorter periods of time.

- * a simpler criteria -- monitoring the cutter motor current instead of the CM position -- can be used to determine when the CM reaches the face;
- * necessary (albeit minor) adjustment of the yaw of the machine can be made to allow the machine to come in line with the direction of the cut.

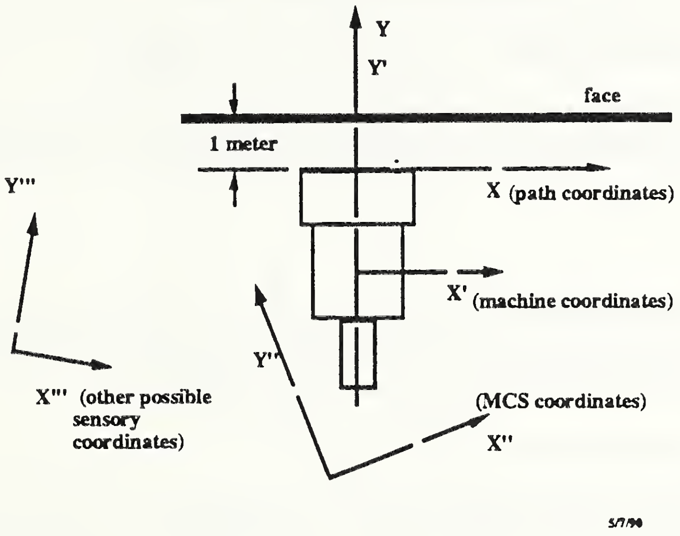


Figure 4: Different Local Coordinate Frames

Data manipulation can be simplified since the references for the yaw and the lateral (X) directions are both zero during the performance of tramming control, i.e., the goal for a five meter cut can be represented as (0, 6, 0). One might require that sensory system coordinates coincide with the path coordinates. If not, transformations are needed (figure 4). Surveys can be made in advance to determine the transformation formula (a translation plus a rotation). The transformation is a world model function, but may be performed at the sending nodes before information is sent to the receiving nodes on the BOM/NET.

The Bureau of Mines intends to include in its testbed a mobile control structure (MCS) [Sch 89] which can be used as a reference during a cut (figure 4). The MCS coordinates may be made to coincide with the path coordinates, otherwise a transformation would be needed.

The above-mentioned starting position for a cut is the goal position for another navigation task which uses its own path coordinates to move the machine to the face area. It is assumed that the free-space tramming control is able to position the machine fairly accurately at the cutting path coordinate origin.

6. ERROR ANALYSIS AND CONTROL

In order to more fully characterize the performance of the tram controller, some understanding of the source and kinds of errors affecting control must be acquired. The basic control strategy and some related issues during the performance of the error control of the machine motion must be disclosed as well. These issues are discused in this section.

6.1 Source and Kinds of Errors Affecting Tram Control

Problems affecting tram control are grouped into the following categories:

(a) Environmental. Coal seams are generally not flat throughout the mine. Irregularities of concern in the nature of the coal seams include:

- oblique grade of the floor, which can be as steep as 9 degrees [Ho 90-2] either sideways or in the front direction;
- anomalies, such as faults in the earth, rocks, etc.;
- varying height or width[6].

The floor can also cause control errors for other reasons such as a wet or muddy surface which would make the CM motion unpredictable (the machine may slip or slide).

(b) Machine.

- Mechanical/electrical delay between issuance of a stop command and the actual stoppage of the tram motors [Sch 90], and similarly, the issuance of a motion command and the actual movement of the machine;
- Uneven speeds between two tram motors;
- Uneven wearing conditions between two tread sets.

(c) Sensory or communication fault. Control errors may be introduced by problems in:

- sensory data query time;
- measurement error propagation [Ho 90-1];
- communication failures in the BOM/NET.

All these problems, uncorrected, can result in deviations from the intended cutting or free-space tramming paths.

6.2 Error Control Strategy and Some Related Issues

The problems identified in section 6.1 are rich enough (they suggest that the underlying system is non-linear) to warrant the consideration of applying more advanced techniques such as on-line system identification [Ey 74] or adaptive control [As 84] as control solutions. For example, one system identification technique uses the on-line iterative Auto Regression Moving Average (ARMA) model to linearly approximate the motion pattern. Error can be predicted and control can be applied.

However, for now, a simpler control strategy is chosen which will merely keep a tight control on the yaw and the lateral deviation. In other words, the machine will be stopped any time that the yaw or lateral error gets beyond the specified value for that task. A new course would be drawn from the current position to the goal and the corresponding commands will be issued and executed. This previously incurred error will be corrected as the machine proceeds along the new course (figure 5).

[6] Excavation of coal becomes uneconomical if the height becomes too thin.

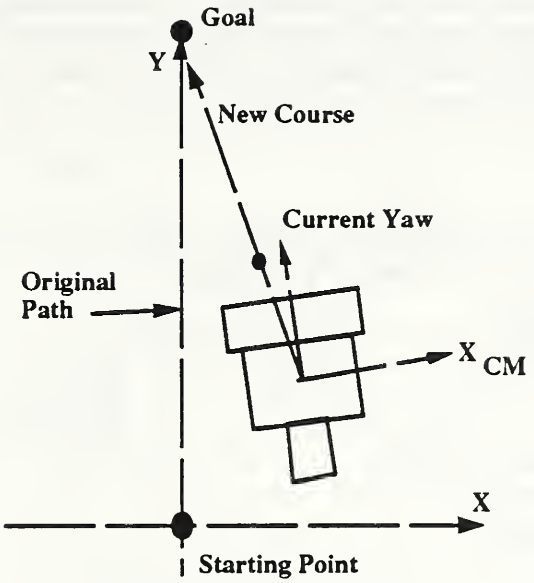


Figure 5: Path Revectoring During Free-Space Tramming

One interesting characteristic in this tramming control algorithm is that some "non-linear" factors may override the established control strategy in some situations. This simple control strategy was selected in the interest of balancing the need for accurate path planning against coal production efficiency (a trade-off between coal produced per hour and cutting precision).

There may be one exception to the above control strategy. When error continues during corrections, this raises a flag signaling that corrections may be too difficult or uneconomical. The tram controller (or the operator in an interactive operation mode) can elect to allow large error tolerances or a large goal region (up to a meter, for example), and deviation from the original path is acceptable. This exception is more acceptable in the first pass of a cut with a hope that replanning for the second pass of the same cut can compensate for most of the large deviation resulting from the first pass.

Given the nature of the errors and uncertainties affecting the tram controller and given the simple control strategy outlined above, certain issues arise as one seeks to minimize the adverse effect of the errors:

(a) Anticipative Control. As a first step approach, an anticipative type of control will be used [An 90] as the machine approaches the goal position or orientation. Tests were performed and estimated lag-times for tram forward, reverse, pivot-left and pivot-right during start-up and stop were obtained [Sch 90]. When the machine is close to a goal, the Tram-Off command is sent with sufficient lead-time to allow the desired machine stop distance to be achieved, as can be seen in figure 6.

(b) Acceptance Region for Goals (either position or orientation). In an ideal and predictable operational environment, the size of acceptance region (see figure 7) may not need to be large, since one may easily quantify the mechanical delay for the CM. By using the anticipative control method mentioned above alone one may be able to handle error control. However, a safe requirement (in a non-ideal environment) is that the whole acceptance region be larger than the worst case stop distance. Once the machine is started in order to correct an out-of-region error, the acceptance region should be large enough to enclose the stop distance to allow the machine to physically stop after the tram-off command is received. Otherwise, corrective action might cause oscillation (an infinite back-and-forth cycle).

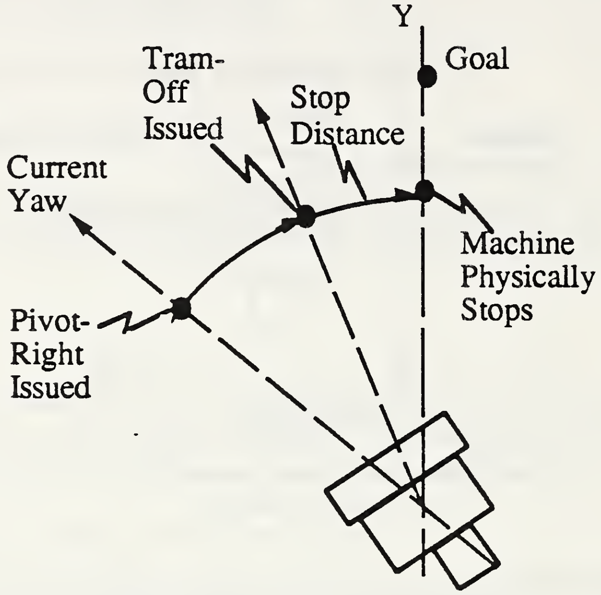


Figure 6: Approaching Desired Orientation During Pivot

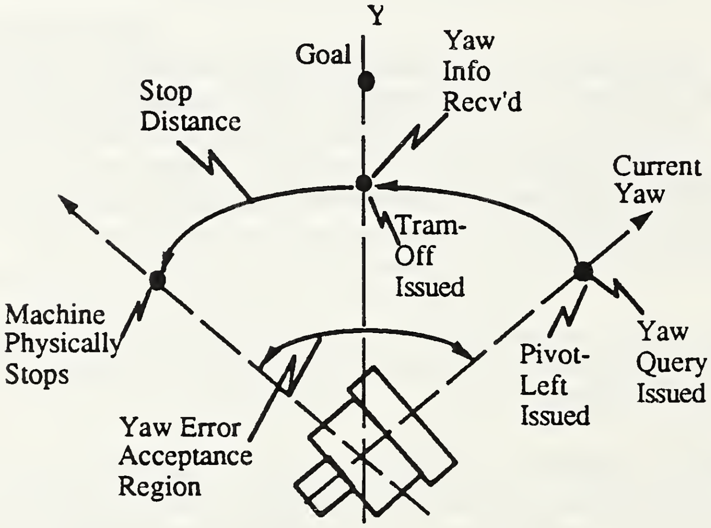


Figure 7: Acceptance Region, Sensory Data Sampling Period, and Mechanical Stop Delay

(c) Sampling Frequency/Overshoot. The minimal requirement for the sensory data query period, including data processing time at both ends (the requester and the provider) and communication time, is that it be smaller than the above-mentioned acceptance region to prevent the machine from overshooting the goal point and bouncing back-and-forth beyond both sides of the error boundary (unstable control). To reduce the number of maneuvers (numbers of different error correction commands) is of concern. The acceptance region may be set at least twice as large as the stop distance. Meanwhile the sampling period may be set not to exceed the stop distance. In such cases the machine position will be detected once it falls within the acceptance region and the region is large enough to allow the machine to stop (figure 7). However, such a large acceptance region may conflict with factor (e) below, namely "operational requirement for yaw error." Also, same as in factor (b) above, an acceptance region may not need to be this large in an ideal operation environment.

Sampling frequency may need to be higher if error grows at larger rates. For example, when a forward motion carries a lot of side slip.

(d) Yaw-Error Boundary during Tramming Forward. When the current orientation of the machine exceeds the error specification (threshold), forward commands will cease and the machine will pivot. There is a trade-off between the path-following accuracy that the controller can achieve and frequency of tram, stop, and pivot commands required. Frequent error correction is highly undesirable from the standpoint of machine maintenance, power consumption, and coal production efficiency. A very tight yaw tolerance in a poor (slippery) operational environment may result in an inefficient "forward - stop - pivot-left - stop - forward - stop - pivot-right - stop" loop without advancing the machine significantly.

(e) The Operational Requirement for the Yaw Error. The error specification also has to take into account the remaining goal distance. When the machine is farther from the goal, yaw error is less critical since the machine has a longer time to correct for it. However, one must also realize that a yaw error accumulating over larger distances translates into a larger lateral offset at the goal (arc equals angle times distance). When the machine is closer, the yaw error boundary must be tighter because the controllability for the machine is lower. Figure 5 illustrates this effect. A lateral deviation at a position closer to the goal means a larger pivot correction which in many situations may not be feasible since the machine must maneuver in very tight quarters.

Non-symmetrical error boundaries may be required due to the environment constraints as well as the machine's mechanical conditions (e.g. one motor may be faster than the other).

(f) X-Directional Error. When the CM position slips laterally during a cut, the tram controller will not attempt to bring the machine back to the original path -- coordinates (0,0,0) to (0,6,0). Rather, it will "replan" for a new path. The new path will be the vector from the current position to the final goal position, (0,6,0), as shown in figure 5.

The horizontal component of the yaw deviation contributes to the lateral error during a forward motion.

7. TASK ANALYSIS

Tram control tasks are defined in this section. They are categorized according to the previously developed architecture (as described in section 7.1 below). The terms "tasks" and "commands" are used interchangeably. They also correspond to the term "plans" in section 8.

7.1 Architectural Consideration

As shown in figure 3 [Hu 90-3], the tram control is a part of the "vehicle piloting" control module at the elementary move (e-move) level and the "tram motion" control module at the primitive level (these control modules assume additional duties including obstacle avoidance). The output of the tram controller conforms to the existing lowest level tram control commands (BOM/NET commands) implemented by the U. S. Bureau of Mines' (BOM) researchers [Schi 90]. These BOM/NET commands are treated as the input to the actuator control level.

The tram controller is also designed so that human operators can interact with the system at each level. The operator can elect to have the CM perform a complete cut task (involving operator interaction at the e-move level), or perform any primitive task (involving operator interaction at the primitive level).

7.2 E-Move Level Tasks

The following e-move commands have been defined:

CUT

A cut plan receives a goal. The current algorithm will only receive a command to make the first pass of a 5 meter cut. This goal is translated to the goal coordinates, (0, 6, 0), in the path coordinate system. The machine then cyclically performs a series of primitive commands (defined later in this section) in order to do the work of cutting coal. A clean-up task to clean up loose coal on the floor may be required but is currently not included in the plan. This series of operations executes in cycles until the 5 meter cut distance has been reached. The operator needs to be able to suspend this plan at any stage, have the status recorded, and resume the operation as desired. One occasion to suspend the operation is when the haulage unit is full. The tram controller can perform a back-out operation at the end of a cut if the operator elects to do so.

FREE-SPACE-TRAM

This is a plan allowing for the movement of a CM machine in non-cutting situations. However, two of the major planning functions normally performed at the e-move level, namely kinematics computation and obstacle avoidance, are performed by human operators at this stage. In other words, when an operator sends a goal position to the tram controller, he must make sure that this position can be reached by the mining machine and that intermediate planning is neither needed nor intended.

At the beginning of this task, the operator will be asked to verify, and correct if necessary, the current position information stored in the computer. He will also be asked to enter the goal position. The controller then computes a vector from the machine's current position to the goal (assuming there is no obstacle in between). The angle between the current machine yaw and the goal vector will be computed and used as a transformation angle. The reference for the machine will be transformed to a new coordinate system that uses the goal vector as the Y-axis. Such planning activity normally results in a plan consisting of a pivot action to point the machine to the goal position and a forward motion to move the machine to the goal.

A future enhancement to this plan is the ability to determine whether to use the forward or the reverse mode in approaching each goal position. Criteria for making such a decision may include power/time efficiency, physical constraints along the path, and the need to set up a convenient orientation for the next following task.

7.3 Primitive Level Tasks

The e-move level commands defined in section 7.2 will be decomposed into the primitive level commands defined below:

INITIAL-APPROACH-TO-THE-FACE

The first step the machine performs during a cut is to approach the coal face coincident to the direction of the cut. The precision of the direction is important to the straightness of the whole cut. The stabilization jack is lowered (before the machine trams forward) to provide more stable platform during cutting operations. Because of precision requirements the yaw error and the lateral deviation of the machine are tightly monitored and controlled. The operator types in an error specification after considering factors such as floor condition, since, for example, on a slippery floor it may not be possible to achieve path accuracies that are possible under drier conditions. The completion of this task is defined when the cutter motor current exceeds a pre-defined threshold which will, under normal conditions, occur soon after the cutter drum makes contact with the coal seam.

SUMP

A sump is accomplished when the cutter drum is pushed into the coal face by the force of forward tramming while maintaining the height of the turning drum. The first increment of the sump distance (approximately 15 centimeters, per discussion with the BOM researchers [Ho 90-2]) is critical to the straightness of the whole sump and is the only time during the execution of a SUMP command that the yaw of the machine is controllable. Afterwards the machine is allowed to sump in, without control of yaw, its entire desired distance (see section 8.2).

SHEAR

After the cutter drum is sumped into the coal, the coal can be excavated by shearing down the rotating drum. A cusp may be left on the floor due to the shape and the angular motion of the drum. Lateral or yaw deviation may be severe due to the large cutting force involved, yet no attempt will be made to correct the error so that static (rather than a smaller dynamic) friction force is maintained on the treads. An exception exists when the cutting reaction force is large enough to push the machine out away from the face. During such circumstances forward motion may be applied to re-engage the machine in the coal so that shearing can resume.

Normally the cutter shears down to the floor. However, under certain conditions the cutter may shear down lower or higher in order to follow the coal seam, to make more room for machine maneuver, or because rock may exist at the bottom.

CUSP-REMOVAL

The cusp left on the floor due to the geometry of the cutter drum must be removed. The machine performs a reverse motion while the cutter is turning at roughly the floor height (see the command above). The yaw error is not monitored for the following reasons:

* One can expect that (with fairly high probability) the machine will tram back to where that last sump started since the amount of coal to cut is much less, the reacting force is in turn much smaller, and, as a result, the machine is less likely to slip.

* From a production efficiency standpoint, there is no point applying a lot of corrective action for relatively small amount of coal.

If the machine is indeed stuck, the tram controller will not attempt to remove remaining coal any further. In either case, the controller will perform a pivot motion in an attempt to re-orient the machine to prepare for the next sump-and-shear cycle.

RE-APPROACH-TO-THE-FACE

This command is similar to the INITIAL-APPROACH-TO-THE-FACE command except that :

* the completion criteria is that the forward distance be equivalent to that of the last cusp-removal, instead of a condition that the cutter motor current reach a threshold;

* the tram controller will wait for the coal cutting subsystem [Hu 90-3] to have the cutter drum raised to the cutting height.

BACKOUT

This command is used primarily when the machine has finished its five meter cut and needs to get back to the origin of the path coordinate frame. This command is designed to be highly user-interactive at this stage expecting that problems may arise during the course of long-distance reverse tramming. It is desirable that such user interactions become part of the machine's planning capability in the future. The cutter drum will be raised before the machine trams in reverse.

A new command, CLEAN-UP, may be defined by simply modifying this BACKOUT command to include the action of keeping the cutter drum at the floor while turning. This new command is used to pick up remaining loose coal on the floor by continuing tramming in reverse after a cusp-removal operation has been completed.

LOCATE-FORWARD

This command is used in free-space tramming to move the machine to a specified location. This command will refer to a path coordinate system computed in the e-move FREE-SPACE-TRAM command. Future enhancements to this program include a decision making process, either through human interaction or rule-based inference, to determine whether to use the forward or the reverse mode of operation to approach the goal.

LOCATE-PIVOT

This command is used when pivot is needed during free-space tramming tasks to point the machine towards the goal. Future enhancements to this plan can include a post-pivot position verification and correction, when necessary.

8. RCS PLANS

State transition diagrams have been developed which essentially utilize the above defined commands to graphically represent RCS plans. The constraints for the commands may be represented as transition requirements for the states.

8.1 E-Move Level Plans

The following two state transition diagrams are used to describe the two e-move commands defined earlier. Each of the states in the diagrams corresponds to a pre-defined primitive command.

The CUT Plan (figure 8)

The cut plan starts with checks on the machine's position and the readiness of all the related subsystems (coal removal, etc., as in figure 3). An initial approach to the face operation moves the machine forward. Sump, shear and cusp removal

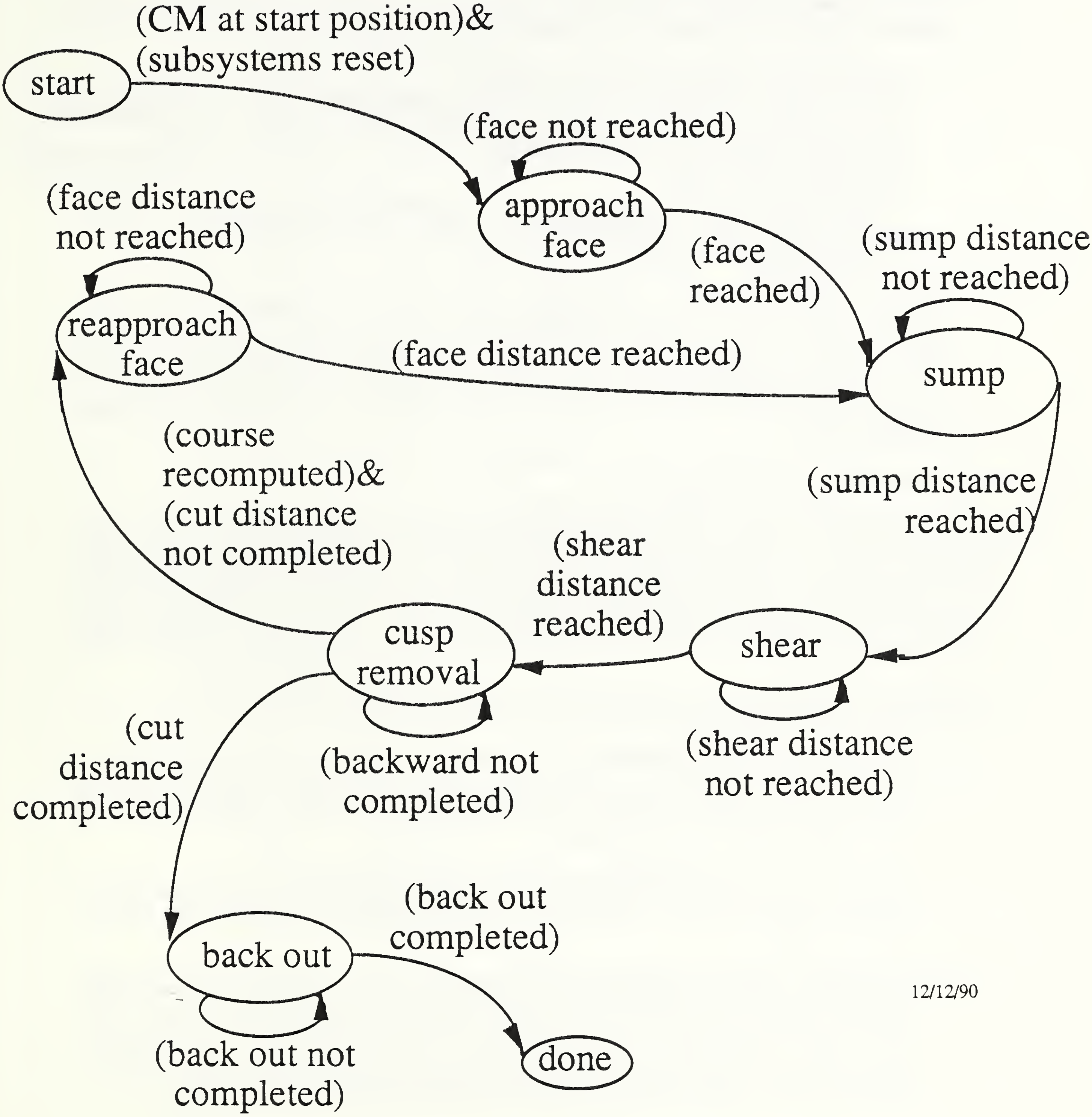


Figure 8: A Cut Plan

operations follow. Further back-out to clean up loose coal on the floor may be required but is currently not included in the plan. This series of operations cycles until a 5 meter distance has been reached.

The FREE-SPACE-TRAM Plan (figure 9)

The tram controller has to use reset to prepare for free-space tramming tasks. The following items are involved in such a reset:

- request sensory data for the current position;
- verify whether the computer storage of the current position data is consistent with what the sensory information shows, correct if not;
- verify that the goal position can be reached;
- compute the goal vector;
- compute the yaw difference from the above and use this difference as the coordinate transformation angle to establish the path coordinate frame of reference;
- do coordinate transformation.

LOCATE-PIVOT will be executed first after the reset is completed. The yaw for the machine will be brought within a specified range of the goal vector. The machine then enters the LOCATE-FORWARD state, within which the machine will reach the goal position. The orientation of the machine at the goal position can be treated as the first sub-task for the next FREE-SPACE-TRAM plan which may or may not contain actual forward motion.

8.2 Primitive Level Plans

The following RCS plans describe the primitive commands defined earlier. Each of the states corresponds to a pre-defined lowest level CM tramming command [Schi 90], except that the required tram-off commands between any two successive different tramming motion commands are not shown for simplicity. In other words, a transition from a TRAM-FORWARD command to a PIVOT-LEFT command implies a TRAM-OFF command in between.

The INITIAL-APPROACH-TO-THE-FACE plan (figure 10)

- * Operator Interface: The operator will be asked to enter or modify (optionally) the maximum errors allowed along the yaw and the X axes. Since vehicle accuracy is crucial to the whole cutting task, the maximum yaw deviation allowed should be kept feasibly small (see section 6.2).

- * Prerequisites: The tram controller needs to verify that the machine is at the origin of the path coordinates before this primitive command can be executed (see section 5.1 for more detail).

- * Execution: Upon verification of its position, the machine will lower the stabilization jack[7] to the floor in order to increase the support to the machine. The machine will then perform a forward motion along the goal vector to approach the face. Pivot motion will be required if the yaw deviation gets larger than the specified limit. In such cases, the forward command will be stopped and a

[7] The extent of the use of the jack seems to vary, depending on floor conditions. The authors assume in this paper that the floor can be slippery and therefore the jack is to be lowered at the beginning of the approach to the face plans, and raised at the end of the SHEAR plan.

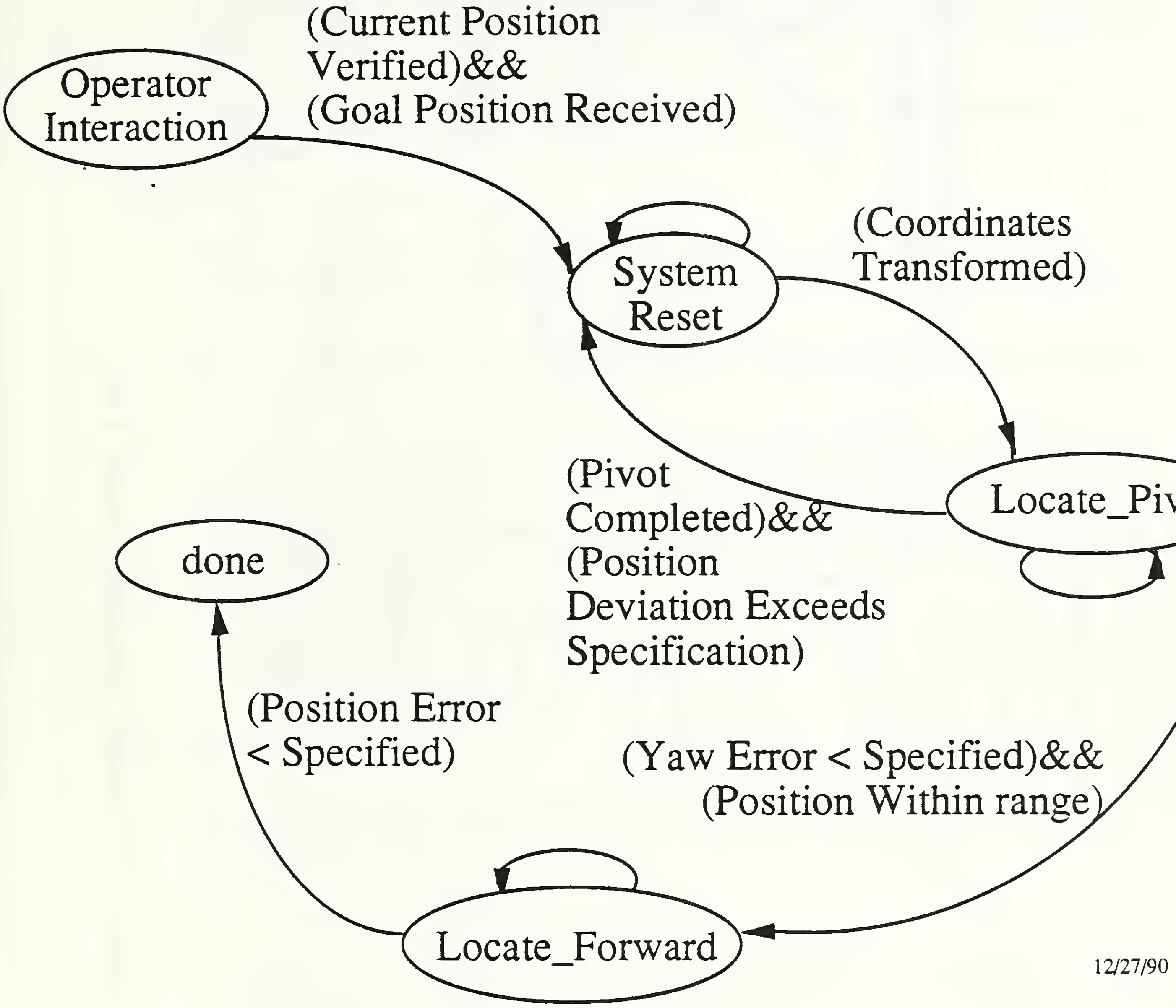
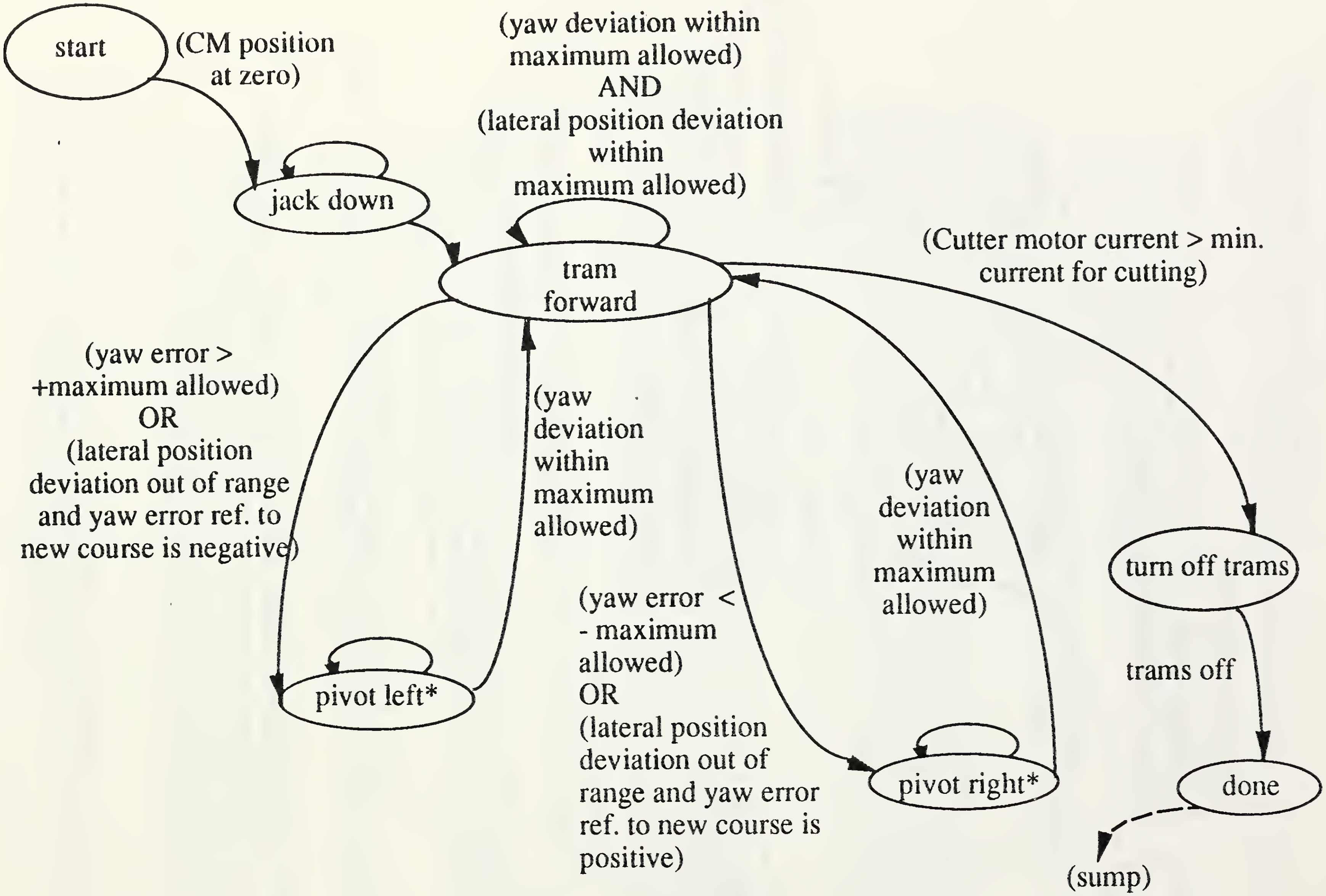


Figure 9: Tram Control for a Free Space Tramming Task


start
(CM position at zero)
jack down
(yaw deviation within maximum allowed) AND (lateral position deviation within maximum allowed)
tram forward
(Cutter motor current > min. current for cutting)
(yaw error > +maximum allowed) OR (lateral position deviation out of range and yaw error ref. to new course is negative)
(yaw deviation within maximum allowed)
pivot left*
(yaw error < - maximum allowed) OR (lateral position deviation out of range and yaw error ref. to new course is positive)
(yaw deviation within maximum allowed)
pivot right*
turn off trams
trams off
done
(sump)

2/13/91

Figure 10: Initial Approach to the Face

corrective pivot command will be issued. PIVOT-LEFT and PIVOT-RIGHT are two distinct states in the diagram. The machine will come back to the forward motion once the yaw correction is completed and the pivot command has been stopped. The same corrective activity will also be required when the lateral deviation exceeds the specified amount. However, in this situation a re-vector algorithm will be used first to compute the new goal vector and the accompanying required amount of yaw correction in order to re-position the machine toward the goal.

* Completion: This command is completed when the cutter motor's current exceeds the pre-specified threshold.

The SUMP plan (figure 11)

* Operator Interface: The operator will be asked to enter or modify (optionally) the maximum errors allowed and the sump completion criteria (see the "completion" factor below).

* Prerequisites: The cutter drum must be at the cutting height, in contact with the coal, and turning.

* Execution: The machine will perform a tram forward motion. As described in section 7.3, during the first 15 centimeters of a sump, the orientation of the mining machine will be closely monitored. A forward command will be stopped and a pivot command will be issued to correct for the yaw error once the error exceeds the specified limit. No corrective action will take place once the drum sumps in for more than 15 centimeters.

* Completion: One can conclude a sump when the gathering pan butts against the coal face.

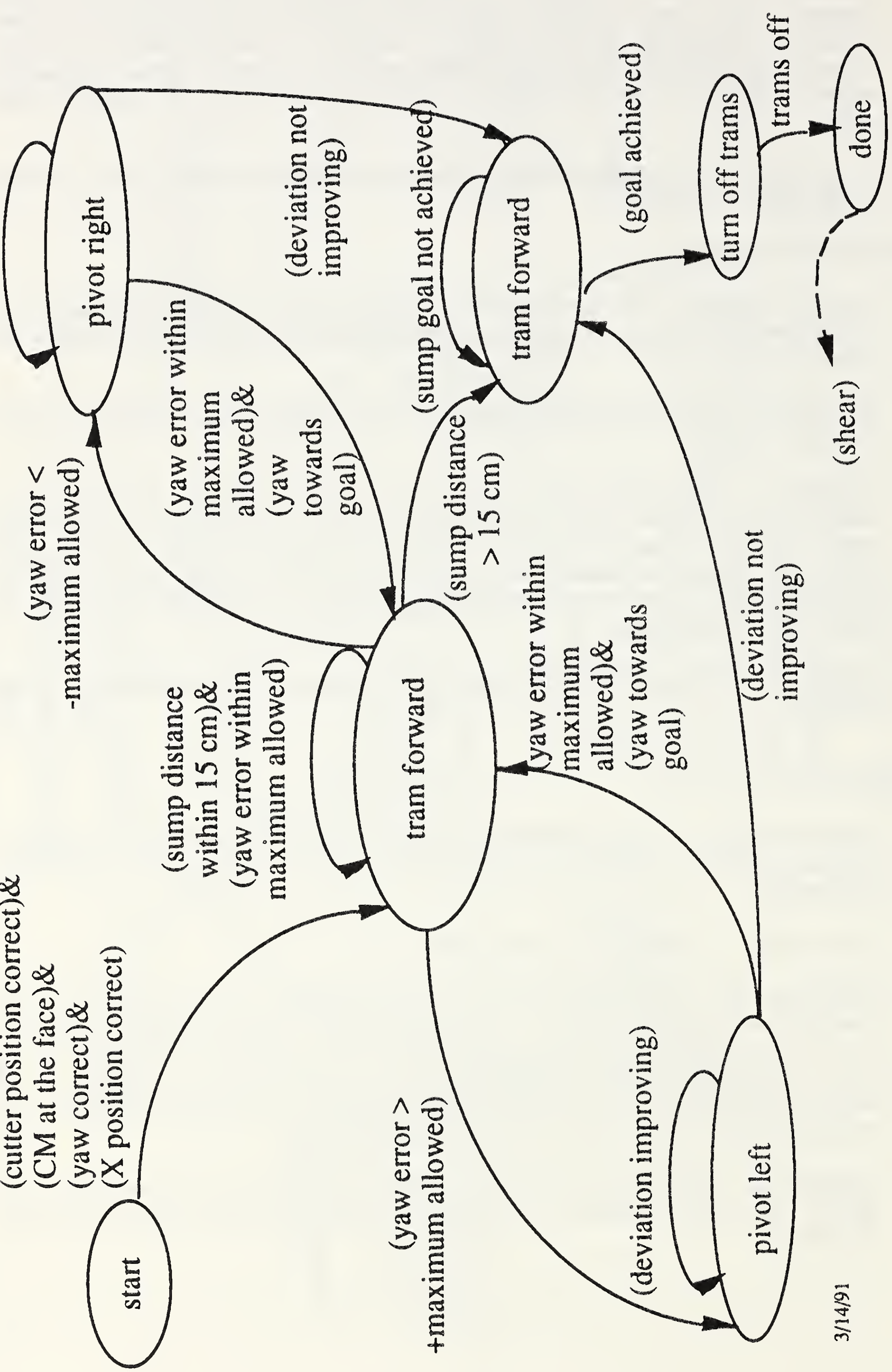
The SHEAR plan (figure 12)

* Operator Interface: None. The maximum errors allowed in the X and Y directions may be specified in advance in a data file since they are not expected to be changed frequently.

* Prerequisites: A sump has been completed.

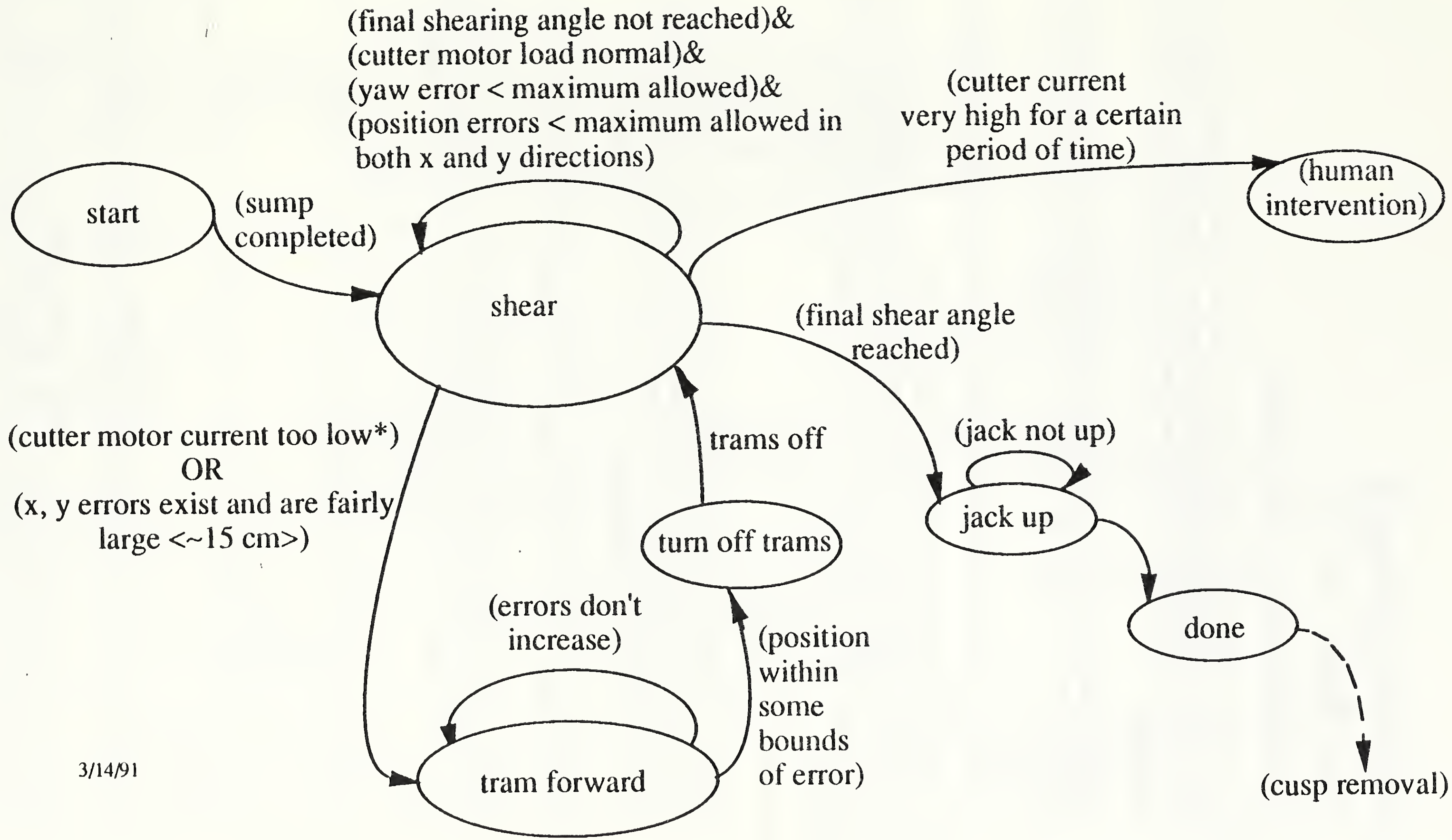
* Execution: The cutter drum will shear down to remove coal from the face. The yaw error will not be monitored. A large error in the Y direction may indicate that the mining machine has been pushed out of the coal face by the counter reacting force. In such case a tram-forward command will be issued to re-engage the cutter drum to the face in order to complete the execution of the shear command. The stabilization jack will be raised after a shear is completed.

* Completion: Generally the cutter drum will shear down to the floor height (see section 7.2 for other situations). The tram motors will be turned off once a sump is completed.



3/14/91

Figure 11: A Sump Plan



* May be a sign of CM popping back.

Figure 12: A Shear Plan

The CUSP-REMOVAL plan (figure 13)

- Operator Interface: None.

- Prerequisites: A shear has to be completed.

- Execution: The machine will tram in reverse for a distance equivalent to the last sump distance without monitoring the yaw error. Cusp left on the floor from the last sump-and-shear cycle can be removed by the turning cutter drum, set at the floor height.

  After the machine is back to the desired position or gets stuck before achieving that position, a pivot command will be issued to attempt to point the machine to the goal for the cut.

- Completion: This command is completed after the machine points itself to the goal.

The RE-APPROACH-TO-THE-FACE plan (figure omitted)

This command is similar to the INITIAL-APPROACH-TO-THE-FACE command except that a tramming distance equivalent to the last cusp-removal distance (instead of the cutter motor current threshold) will be used to determine whether this task has been completed.

The BACKOUT plan (figure 14)

- Operator Interface: An operator will be asked to enter the error tolerances in the yaw and the X directions at the beginning of the execution of this command. He will also be asked to enter a pivot direction and the accompanying pivot amount when the machine is stuck during the course of tramming backwards.

- Prerequisites: The machine has completed a cusp removal operation, the stabilization jack has been raised, and the cutter drum has been raised.

- Execution: The basic operation is tramming in reverse. Errors in the X and yaw directions will be corrected according to the operator specification. As described above, the operator will be involved in error correction when the machine gets stuck.

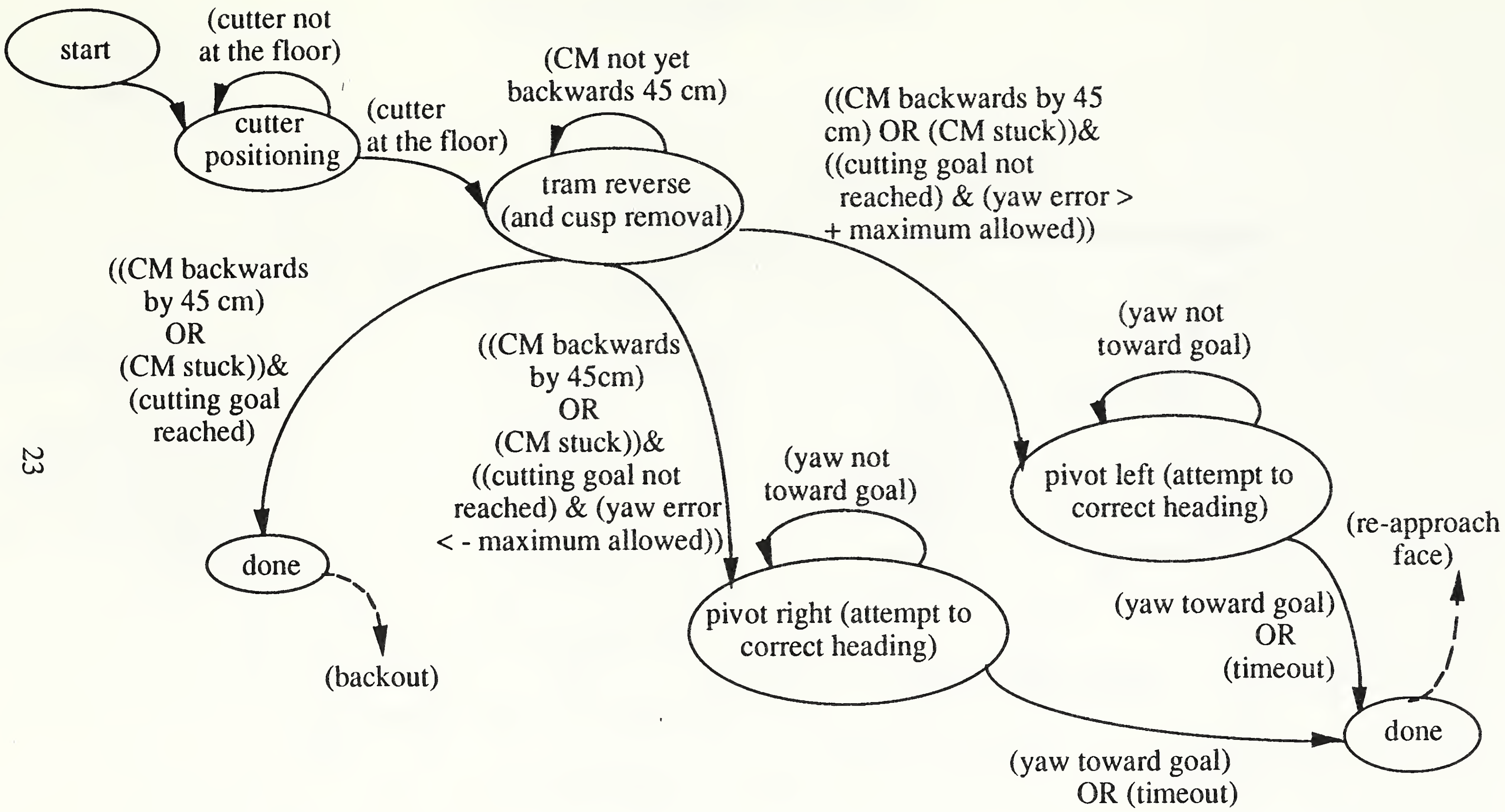
- Completion: The command is executed successfully when the machine is back to the origin of the path coordinate reference frame.

The LOCATE-FORWARD plan (figure 15)

- Operator Interface: Error specification will be entered as in most other primitive commands.

- Prerequisites: The machine is up and operational. The stabilization jack is raised. The cutter drum is not at the floor height. There is no obstacles along the path.

- Execution: This command will receive a goal position from the e-move level. Forward motion and error checking are the main activities in this command. All the error checking will be referring to the new coordinates, which also implies that the sensory data must be transformed before being used.



3/14/91

Figure 13: A Cusp Removal Plan

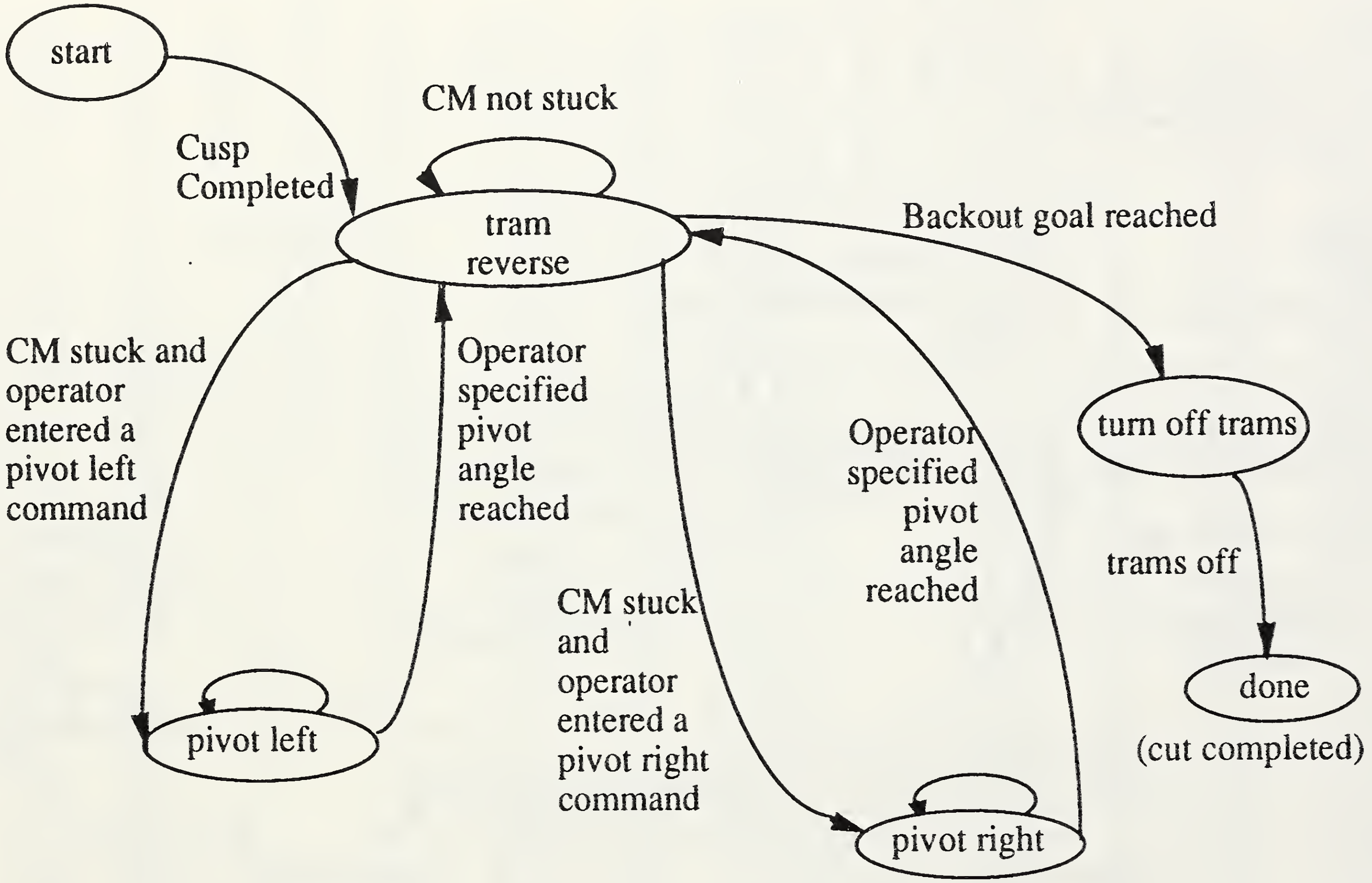

start
Cusp Completed
CM not stuck
tram reverse
Backout goal reached
CM stuck and operator entered a pivot left command
Operator specified pivot angle reached
CM stuck and operator entered a pivot right command
Operator specified pivot angle reached
turn off trams
trams off
done
(cut completed)
pivot left
pivot right

Figure 14: A Back Out Plan

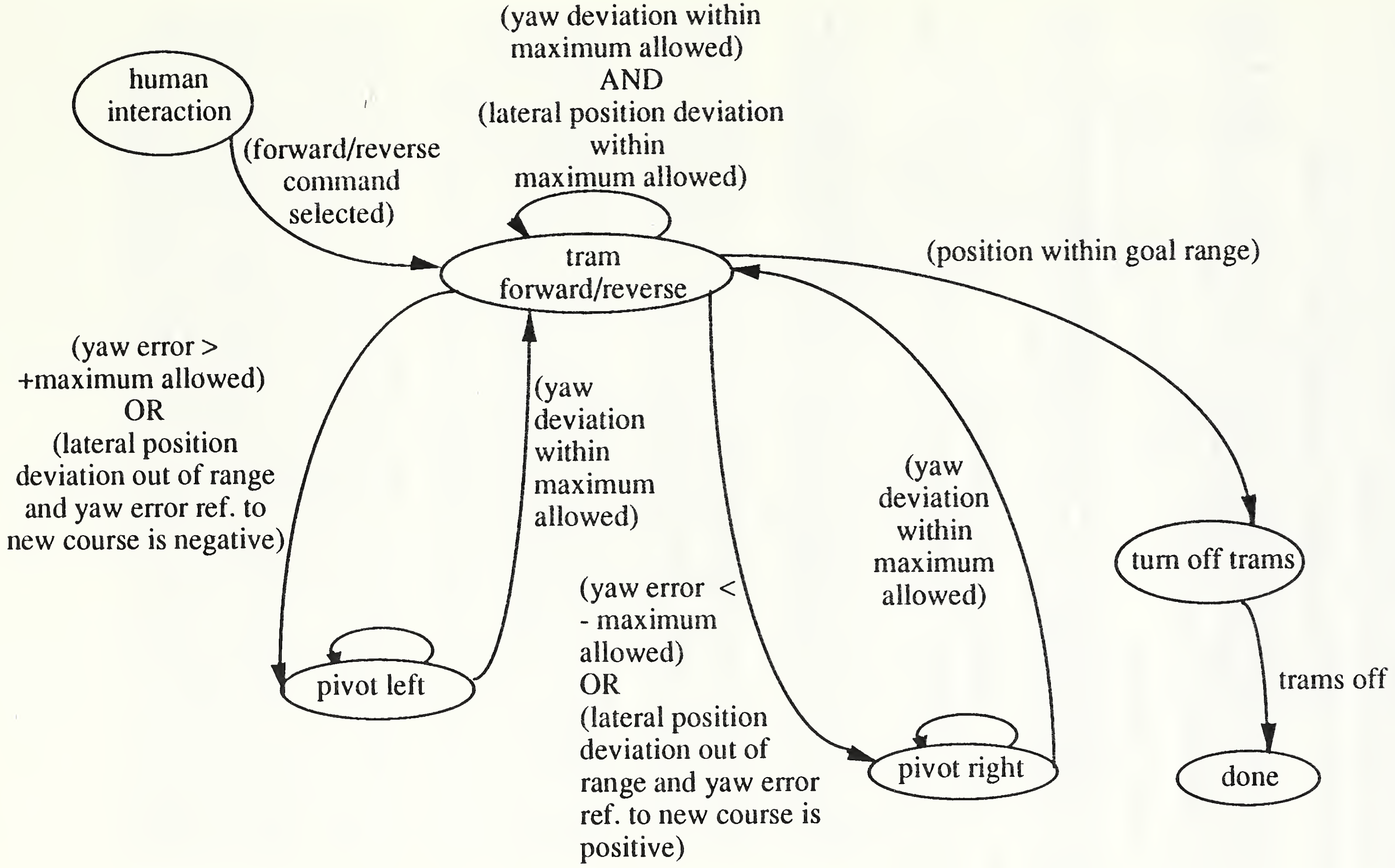

human interaction
(forward/reverse command selected)
(yaw deviation within maximum allowed) AND (lateral position deviation within maximum allowed)
tram forward/reverse
(position within goal range)
(yaw error > +maximum allowed) OR (lateral position deviation out of range and yaw error ref. to new course is negative)
(yaw deviation within maximum allowed)
(yaw deviation within maximum allowed)
(yaw error < - maximum allowed) OR (lateral position deviation out of range and yaw error ref. to new course is positive)
pivot left
pivot right
turn off trams
trams off
done

Figure 15: A Locate_Forward Plan

- * Completion: This command is completed when the machine is within the specified range of the goal position.

The LOCATE-PIVOT plan (figure 16)

This command is similar to the above command except that its primary activity is to point the machine toward the goal position.

9. COMPUTER PROGRAM DESIGN AND IMPLEMENTATION

This section describes the software implementation for the controller. Basically each command is implemented as an independent software module. A main program is capable of executing any tasks (e-move or primitive) selected by an operator (see sections 7.1 and 9.1). The implementation is done using the C language.

9.1 Command Structure

The primitive commands can be either individually activated by human operators (through AMREDS [AM 90] if desired), or automatically executed by the e-move commands. The command structure is given in figure 17 to show such effect.

9.2 The Generic Program Structure for Plans

The following attributes describe the generic structure of the tram control software modules:

- * **Finite State Plan:** Each plan has a finite number of states. Except for the dummy 'start' and 'done' states, all the other states correspond to the pre-defined commands. An example is given in appendix II which uses structured English to describe the INITIAL-APPROACH-TO-THE-FACE plan.

- * **Checking Pre-Conditions:** When a plan is activated, it is at the 'start' state (the state pointer, see below, has a value of zero). If all the preconditions are met, the pointer will be assigned a number corresponding to the first state. Otherwise an error message ("Can not start the task") will be printed on the screen.

- * **SWITCH/CASE Statements** The skeleton of a plan is represented by using a 'switch' statement in the C language. Each state is a 'case'. During the execution of a plan, transition requirements are checked inside each case to determine whether the system should stay at the same state (thus performing the same activity) or switch to another state. For example, if the yaw error exceeds the limit, the system activity will switch from tramming forward to pivoting. The values for the state pointers (see below) are updated consequently at the end of a execution cycle.

- * **State Pointers:** Each plan has a pointer pointing to the system's current state or the state that the system is due to enter in light of the existence of the triggering events. Two additional auxiliary pointers are also used, one stores the information of the system's state for the last pass and the other stores that for the next pass. The pointers are reset to be zero initially. If the execution of a plan shows that the value for the current state pointer is different from that for the last pass state pointer (meaning new command is required), this new state value is sent to a BOM/NET command processor, and a corresponding BOM/NET command is issued. Otherwise no different action will be taken (meaning the execution of a same command as in the last pass continues). The pointer for the next cycle is used because a tram-off command is required between a tram straight command and a

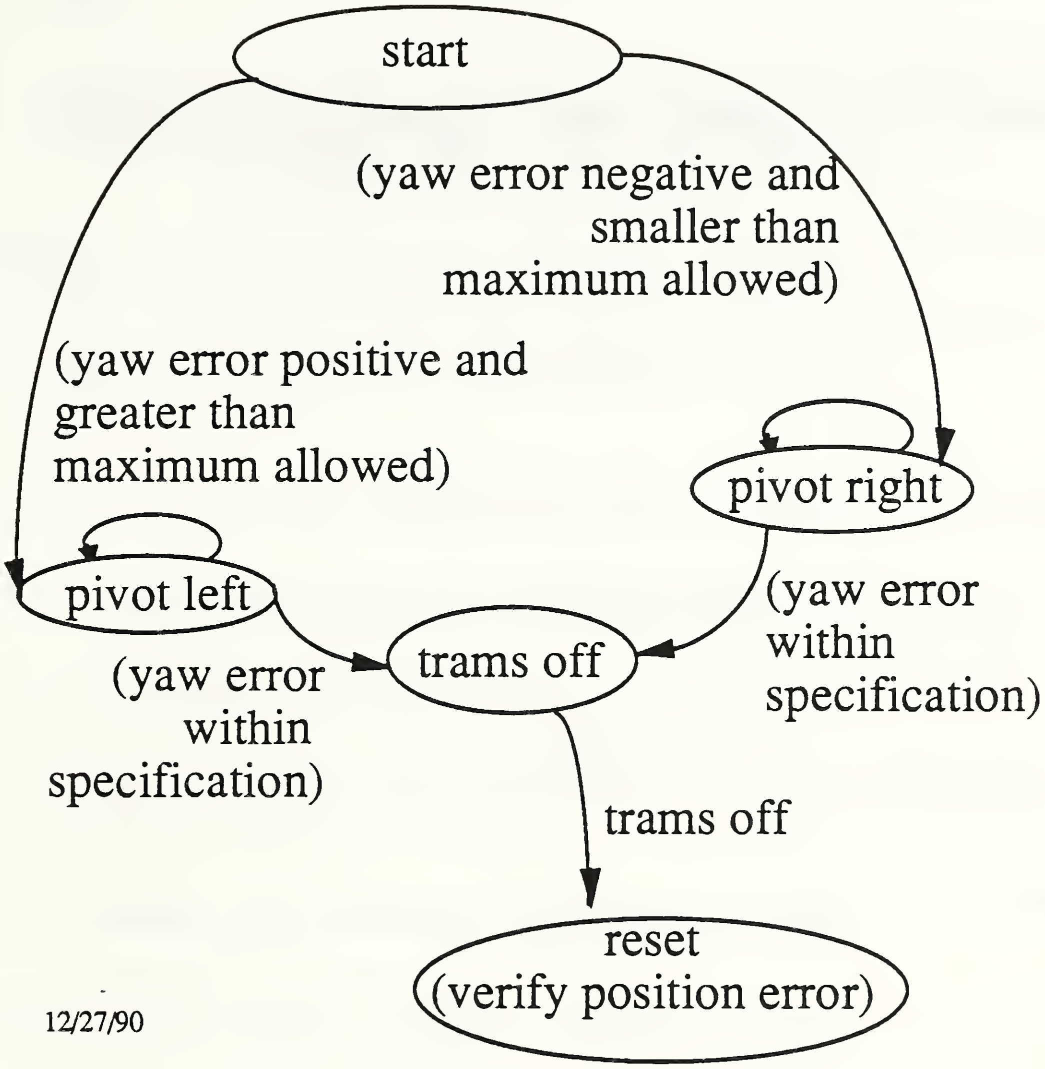
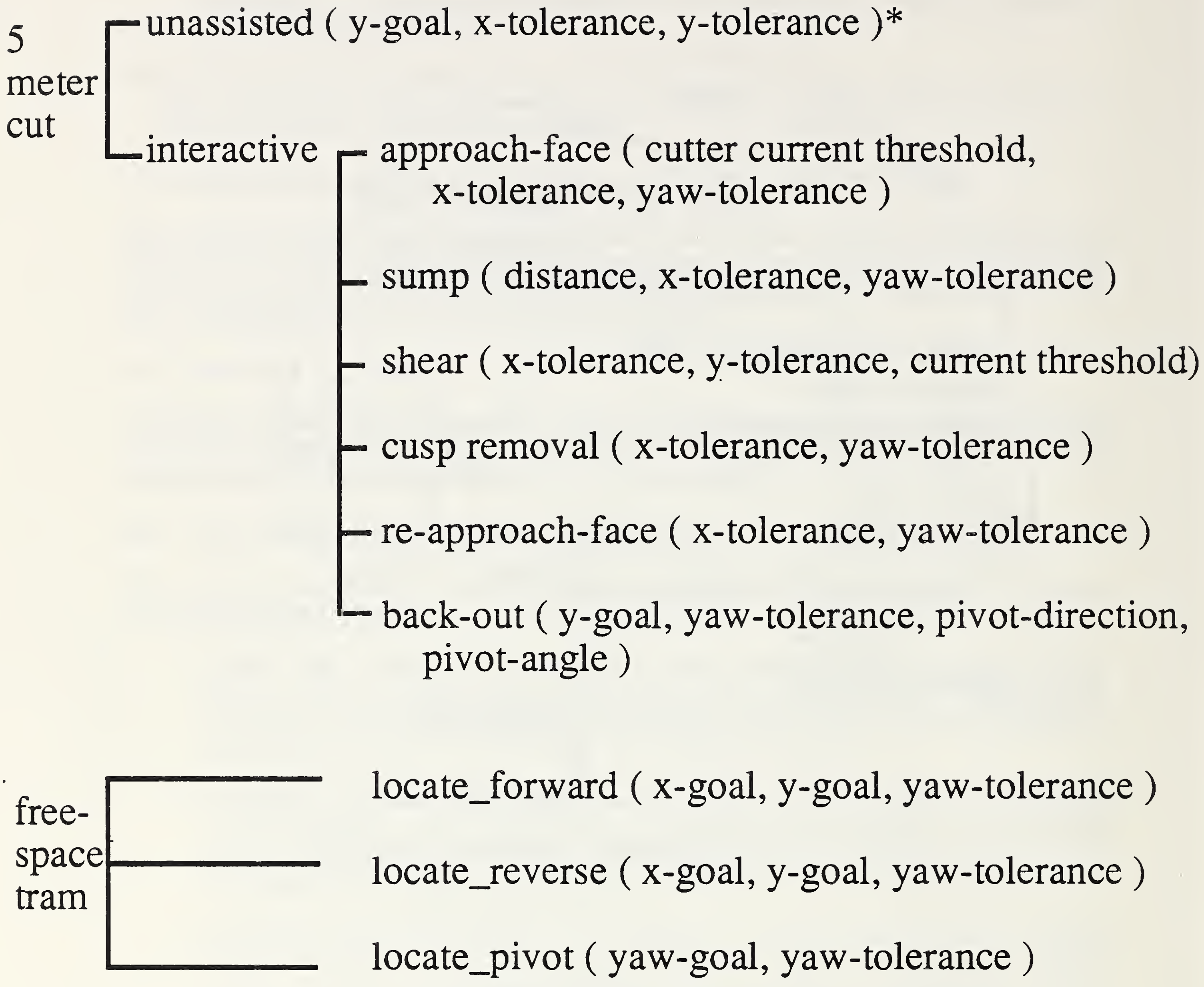

start
(yaw error negative and smaller than maximum allowed)
(yaw error positive and greater than maximum allowed)
pivot right
pivot left
trams off
(yaw error within specification)
(yaw error within specification)
trams off
reset (verify position error)
12/27/90

Figure 16: A Locate_Pivot Plan



5 meter cut

unassisted (y-goal, x-tolerance, y-tolerance)*

interactive

approach-face (cutter current threshold, x-tolerance, yaw-tolerance)

sump (distance, x-tolerance, yaw-tolerance)

shear (x-tolerance, y-tolerance, current threshold)

cusp removal (x-tolerance, yaw-tolerance)

re-approach-face (x-tolerance, yaw-tolerance)

back-out (y-goal, yaw-tolerance, pivot-direction, pivot-angle)

free-space tram

locate_forward (x-goal, y-goal, yaw-tolerance)

locate_reverse (x-goal, y-goal, yaw-tolerance)

locate_pivot (yaw-goal, yaw-tolerance)

*Note: The unassisted mode and the interactive mode of operations execute the same primitive tasks.

Figure 17: Tram Control Command Structure

pivot command. Therefore two future states have to be resolved during the evaluation of the state transitions (e.g., do a TRAM-OFF follow by a PIVOT-LEFT).

- **Completion Flags** A plan completion flag is issued when the criteria are met. It will be reported to the upper level and/or the human operator, so that the next command can be activated. This flag is reset to zero thereafter.

The sensory data query and world model aspects of the program modules are described in section 9.4.

Appendix II illustrates a program for a tram control primitive command using structured English. The actual programs can be referenced through the authors.

9.3 BOM/NET Interface

BOM/NET commands can be grouped into two categories: tramming commands and sensory data request commands. They are handled in the tram control software as follows:

- Tramming Command Processing. A command processing routine is implemented to simplify the tram controller-BOM/NET interface problem. This routine receives the pointer values, recognizes new commands (as opposed to the repetitive ones) and generates corresponding BOM/NET commands.

- Sensory Data Request Command Processing. These commands are included in each individual world model module (see, as well, the following section). Currently the data request commands are sent out to the network at the beginning of each cycle of execution.

9.4 World Model and Global/Local Variable Declarations

Currently the world model handles the following functions:

- Query for and maintain the mining machine position/orientation data.

- Query for and maintain the status of the appendages. The status will be summarized as status flags. The tram controller will look at the flags to determine whether certain tasks can be performed.

A world model module is currently placed at the beginning of each of the command programs. This implies that the programs are executed sequentially. Concurrent processing of the world models might be desirable at later stages.

Each software command module has an associated data declaration file containing globally accessible data (for data potentially required by multiple modules). Specific data declarations were implemented as the ".h" files in the computer programs (could be referenced through the authors).

9.5 Emulation

For debugging purposes a simple emulator has been implemented which essentially translates the program execution cycle counts into CM positional and orientation increments. Error conditions such as angular deviation can be generated to facilitate software testing.

9.6 Program Structure

The program structure for the tram controller is shown in appendix III (which is different from the command structure, shown in figure 17). This chart describes the software module organization for the tram controller. Program names and their associated operator input parameters are specifically listed. The actual programs can be referenced through the authors.

10. SUMMARY AND FUTURE WORK

A tram control algorithm for continuous mining machines has been developed and described. It includes commands necessary to perform underground coal cutting as well as free-space motion tasks. This tram control algorithm has been mapped to a previously developed RCS architecture for coal mining automation. Computer programs have been created and tested. However, due to a project re-direction encountered at the implementation and testing stage, some issues discovered in this paper may not be fully explored or implemented. The following are some issues for future consideration:

- A position verification after each pivot (since the performance of a pivot does not guarantee that the assumed pivotal point does not move).

- A concurrent processing environment for the algorithm, including the world model modules.

- A decision process in free-space tramming to determine whether forward or reverse motion is to be used to approach a given goal position.

- Integration of this tram control algorithm into the existing U. S. Bureau of Mines' operator interface software [AM 90].

The following summarizes this tram control design and implementation work:

- This work is another demonstration showing that the NIST RCS task decomposition method can be an effective approach for solving complex real-time system control problems. This method utilizes a generic approach to describe system behavior. In addition, system configuration, integration, and modification are facilitated.

- It was discovered that the implementation procedure for this method could be described in more detail and generic software templates could be developed to facilitate implementation. Such improvements are ongoing at NIST. A formal theory of intelligent machine systems and an RCS textbook with implementation examples are expected to be developed as a result of this effort.

ACKNOWLEDGEMENTS

The authors extend their appreciation to Mr. Timothy Matty, Dr. George Schnakenberg, Dr. Christopher Jobes, and Mr. Edward Fries of the U. S. Bureau of Mines for their insightful input, especially regarding coal mining practice. The authors also extend their acknowledgement to Mr. Don Orser of NIST for his participation in this project, and Cheryl Hutchins for editing this manuscript.

REFERENCES

[Al 82] Albus, J.S., McLean, C., Barbera, A., and Fitzgerald, M., "An Architecture for Real-Time Sensory-interactive Control of Robots in a Manufacturing Environment," 4th IFAC/IFIP Symposium on Information Control Problems in a Manufacturing Technology, Gaithersburg, MD, Oct. 1982.

[Al 87] Albus, J.S., McCain, H.G., and Lumia, R., "NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)", NBS Technical Note 1235, National Bureau of Standards, U. S. Department of Commerce, July 1987.

[Al 88] Albus, J.S., "System Description and Design Architecture for Multiple Autonomous Undersea Vehicles", NIST Technical Note 1251, National Institute of Standards and Technology, U. S. Department of Commerce, September 1988.

[Al 89] Albus, J., Quintero, R., Huang, H., and Roche, M., "Mining Automation Real-Time Control System Architecture Standard Reference Model (MASREM)", NIST Technical Note 1261 Volume 1, National Institute of Standards and Technology, U. S. Department of Commerce, May 1989.

[AM 90] AMREDS, U. S. Bureau of Mines Internal Computer Program.

[An 90] Anderson, D.L., "Laser485.c", Computer Program, Bureau of Mines, 1990.

[As 84] Astrom, K. J. and Wittenmark, B., Computer Controller Systems, Theory and Design, Prentice-Hall Information and System Sciences Series, Prentice-Hall, Inc., Englewood Cliff, N. J., 1984.

[Ey 74] Eykhoff, P., System Identification, Parameter and State Estimation, John Wiley & Sons Ltd., New York, N. Y., 1974.

[Ho 90-1] Horst, J. A., "An Application of Measurement Error Propagation Theory to Two Measurement Systems Used to Calculate the Position and Heading of a Vehicle on a Flat Surface", NIST Interagency Report NISTIR 90-4434, National Institute of Standards and Technology, U. S. Department of Commerce, 1990.

[Ho 90-2] Horst, J. A., "John Horst's Phone Conversation of March 23, 1990 with Tim Matty, a CM 'Expert' at BOM PRC", Telephone Conversation Notes, March, 1990.

[Hu 90-1] Huang, H., "A Hierarchical Real-Time Control System for Use with Coal Mining Automation", Fourth Conference on the Use of Computers in the Coal Industry, the West Virginia University, Morgantown, West Virginia, June 1990.

[Hu 90-2] Huang, H., "Hierarchical Real-Time Control Task Decomposition for a Coal Mining Automation Project", NIST Interagency Report NISTIR 90-4271, National Institute of Standards and Technology, U. S. Department of Commerce, March, 1990.

[Hu 90-3] Huang, H. and Quintero, R., "Task Decomposition Methodology for the Design of a Coal Mining Automation Hierarchical Real-Time Control System", The Fifth IEEE International Symposium on Intelligent Control, Philadelphia, PA, September 5 - 7, 1990.

[Jo 82] Joy Machinery Company, Technical Service Manual, Joy 16CM Miner, 1982.

[Mi 68] U. S. Bureau of Mines, a dictionary of mining, minerals, and related terms, 1968.

[Pa 88] Penn State Mining Engineering Program, Course Notes on "Elements of Underground Coal Mining", The Pennsylvania State University, University Park, Pennsylvania, Oct., 1988.

[Sa 89] Samet, H., The Design and Analysis of Spatial Data Structures, Addison-Wesley Publishing Company, Inc., New York, New York, 1990.

[Sch 89] Schnakenberg, G., "Underground Coal Mining Automation Research (Room and Pillar Mining) Justification and Research Program Plan", Ver. 1.1, Bureau of Mines, United States Department of the Interior, August 24, 1989.

[Sch 90] Schnakenberg, G., and Sammarco, J., "Joy 14 CM Machine Test Data," U. S. Bureau of Mines Internal Document, September 28, 1990.

[Schi 90] Schiffbaur W. H., et al., "Computer Network for Continuous Mining Machine (Joy 14CM) Control -- Documentation of Node Functions and Communications," U. S. Bureau of Mines Internal Document, April 1990.

[St 83] Stefanko, R. Coal Mining Technology, Theory and Practice, Society of Mining Engineers of the American Institute of Mining, Metallurgical, and Petroleum Engineers, Inc., New York, New York, 1983.

Appendix I: Brief Review of Underground Coal Mining Environment

A typical underground coal mining environment is described for those readers less familiar with the common terminology. Only aspects of the underground mining environment relating to tramming control will be covered. Such description is done through a graphic illustration (figure I.1) and some brief discussion on each term [St 83, Mi 68, Pa 88]:

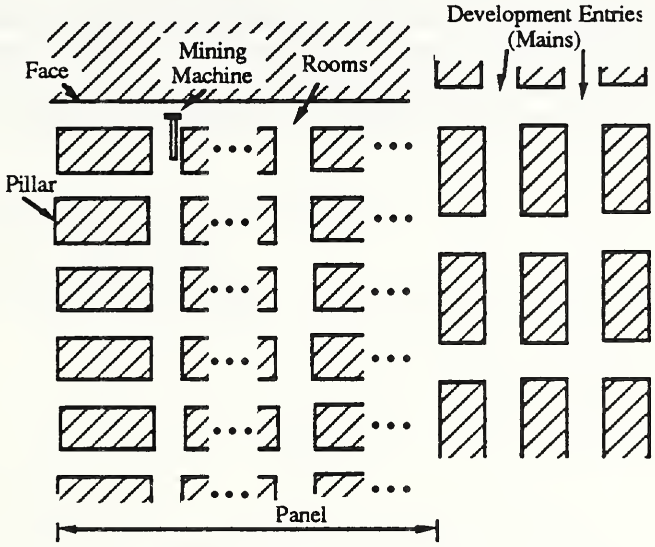


Figure I.1: A Room-and-Pillar Mining Environment

* Panel: a large block of coal (usually rectangular) to be extracted which is separated from the next panel by leaving a long rectangular pillar of unextracted coal between panels. The long unextracted pillar is a safety precaution to prevent the collapse of the coal roof over any more than one panel in the event of a cave in.

* Pillar: generally means a small (compared to panel) block of unextracted coal, 36 meters or less in length and 4.5 to 18 meters in width. As long as they are of sufficient size and separation from one another, pillars (along with roof bolts) help keep the roof of the mine from caving in.

* Face: the front of the coal seam where cutting operations occur.

* Entry: a passage way in a panel where coal has been extracted [Mi 68], typically 9 to 14 meters in width. Entries can be used as haulage roads, transportation roads, or ventilation paths. Entries are formed by removing coal and leaving pillars that are nominally in rectangular in shape.

* Cut: The action that the mining machine takes to excavate a block of coal. The maximum length of a cut is regulated by Mine Safety and Health Administration (MSHA) from the viewpoints of roof support, ventilation, and methane level requirements. The width of a cut (typically the width of an entry) requires the mining machine to cut in two passes.

* Room-and-Pillar Mining: a mining method that also features the development of main entries at both sides of a panel. Coal is extracted forming rooms [St 83][8] with

[8] P. 45 in the reference.

pillars left. Pillars may be extracted at a later stage in retracting operation.

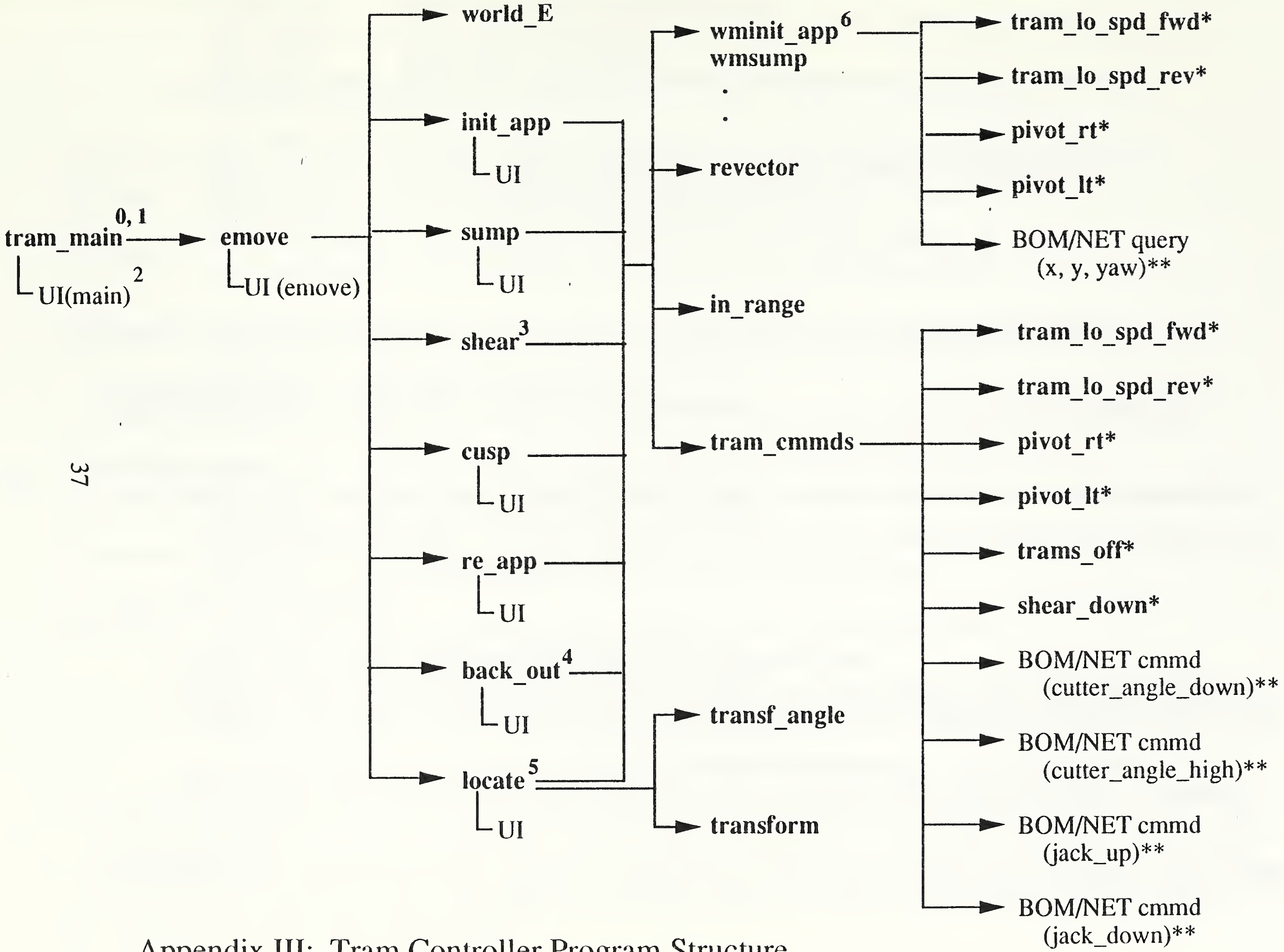
* Long-Wall Mining: a more recent mining method in which multiple entries (three, five or seven) are developed at both sides, so that a large panel of coal can be entirely extracted (leaving no pillars) using long-wall mining equipment.

Sizes of pillars and width of entries may vary depending on the roof support and the transportation support requirements. Main entries are more permanent and may hold more traffic, they typically have larger pillars (18 meters by 36 meters). In the major production area in room-and-pillar mining, smaller pillars (4.5 meters by 36 meters) are kept to maximize coal output. Size may also vary due to the existing softer strata above.

Appendix II: An Example of Tram Control Primitive Program

Structured English is used to describe this program. This program performs initial approach to the face when the cutting task starts. The program is described as follows:

```
BEGIN PROGRAM

IF ( x direction deviation exceeds tolerance )
    re-compute course;

yaw error = new yaw goal - current yaw;

IF ( first entering the plan )
    IF ( position checked && appendage checked )
        {   reset position checked flag;
            reset appendage checked flag;
            set state pointer to 'jack-down';  /** 'state pointer' abbr. as 'pointer' **/
        }
    ELSE
        print error message "can not start init_app";
ELSE
    {   CASE pointer = 'jack up':
            IF ( jack is down )
                pointer = 'tram forward';

        CASE pointer = 'tram forward':
            IF (cutter current > cutter current at no load )
                trams off;
            ELSE IF ( yaw error > yaw tolerance for pivot left in init_app )
                {   pointer = 'trams off ';
                    next pointer = 'pivot left';
                }
            ELSE IF ( yaw error < - yaw tolerance for pivot right in init_app )
                {   point = 'tram off';
                    next pointer = 'pivot right';
                }
            ELSE
                <increment tram forward counter> /** for simulation only **/

        CASE pointer = 'pivot left':
            IF ( yaw error < yaw tolerance for pivot left in init_app )
                {   pointer = 'trams off ';
                    next pointer = 'tram forward';
                }
            ELSE
                <increment pivot left counter>  /** for simulation only **/

        CASE pointer = 'pivot right':
            IF ( yaw error > - yaw tolerance for pivot right in init_app )
                {   pointer = 'trams off ';
                    next pointer = 'tram forward';
                }
            ELSE
                <increment pivot right counter>  /** for simulation only **/
        CASE pointer = 'trams off':
            IF ( trams off command completion acknowledgement received )
                IF ( next pointer = non-zero ) /** typically when changing
                                    between forward & pivot **/
                    {   pointer = next pointer;
                        reset next pointer to zero;
                    }
                ELSE        /** plan execution completed **/
                    {   reset pointer set;   /** include last pass pointer and
                                    next pointer **/
                        set cm_at_face flag;
                        <reset all CM motion counter>  /** simulation only
                                                **/
                    }
                reset command ack. to zero;
    }

IF (( pointer for the last pass != pointer ) && ( CM not at face ))
    {   update pointer for the last pass;
        send pointer value to the command processor;
    }

END PROGRAM
```



Appendix III: Tram Controller Program Structure

0: All bold-faced labels are program names.

1: Each program has an associated .h data files. All program parameters must be entered either in these .h files or as UI as shown below.

2: UI: User Interface. A program having a "UI" mark means that it contains a set of paramters which an operator will be asked to enter values. He will do it either through the input data file or on-screen during execution.

The specific UI for each program is listed in the "UI Tables" shown below. Note that the status variables to be obtained the other CM subsystems are listed as UI here (such as g_head_float). They will become global variables once the tram controller is integrated into the existing system.

3: **shear** Does not call **revector**.

4, 5:Do not call **revector** and **in_range.**

6: Each primitive has its own 'wm' module: **wminit_app, wmsump, wmshear, wmcusp, wmre_app, wmbackout,** and **wmlocate**.

*: These programs are for the simulation of the CM motion only. They are to be replaced by the corresponding BOM/NET tramming or appendage control commands.

**: These are program statements which are to be replaced by BOM/NET sensor data request commands.

UI Tables:

UI (main):	auto_cut_on:	to select whether to perform a 5 meter cut or an interactive cut
	bk :	to select whether to do a backout after the completion of a 5 meter cut
	loco_plan_node:	to select a primitive funtion for execution during an interactive cut

The following initial conditions must be established before the init_app program can be executed:

cutter_angle":	initial cutter drum position
cutter_motor_on":	initial cutter motor status
g_head_on":	initial on/off status of gathering head

g_head_float": initial status of whether the gathering head is in a floating position
jack_up": initial status of whether the jack is up
tram_motor_on": initial status of whether the tram motors are on
perturbation parameters": for simulation only and to be ignored in real situations

UI (emove): Users are asked to verify the values of the following parameters in an interactive-cut operation:
CM_at_face_init_E": pre-condition before a sump can be started
sump_completed_E": pre-condition before a shear can be started
shear_completed_E": pre-condition before a cusp removal can be started
cusp_completed_E": pre-condition before a backout or re_app can be started

UI (init_app), UI (sump), UI (re_app):
change_yaw: to select whether or not to change the max. yaw errors allowed
dyaw_pvl: max. yaw error allowed for a pivot-left
dyaw_pvr: max. yaw error allowed for a pivot-right

UI (backout): change_yaw: to select whether or not to change the max. yaw errors allowed
dyaw_pvl: max. yaw error allowed for a pivot-left
dyaw_pvr: max. yaw error allowed for a pivot-right
backout_plan_node_next: pivot direction that the operator desires when the CM gets stuck during motion
yaw_pivot: the pivot amount when the CM is stuck (always positive)

UI (locate): change_yaw: to select whether or not to change the max. yaw errors allowed
dyaw_pvl: max. yaw error allowed for a pivot-left
dyaw_pvr: max. yaw error allowed for a pivot-right
tmp1: to select whether or not the current CM position needs to be changed because it is incorrect
x_current, y_current: the correct CM position to be entered
x_read, y_read: goal position ref. to the original coordinates

": Values to the parameters such marked will be provided or derived based on the information obtained from the other nodes on the bitbus once the controller is intrgrated (no longer UI). In other words, these parameters either need a BOM/NET sensor data query commands (the tram controller may have a wm module to preform the queries and keep records of all the status) or to be ignored asumming that the status is checked by other higher level sub-systems.

NIST-114A
(REV. 3-90)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION OR REPORT NUMBER
NISTIR 4596

2. PERFORMING ORGANIZATION REPORT NUMBER

3. PUBLICATION DATE
[illegible]

4. TITLE AND SUBTITLE
Task Decomposition and Algorithm Development for Real-Time Motion Control of a Continuous Mining Machine

5. AUTHOR(S)
Hui-Min Huang, John A. Horst and Richard Quintero

6. PERFORMING ORGANIZATION (IF JOINT OR OTHER THAN NIST, SEE INSTRUCTIONS)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
GAITHERSBURG, MD 20899

7. CONTRACT/GRANT NUMBER

8. TYPE OF REPORT AND PERIOD COVERED

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP)

U.S. Bureau of Mines
Cochrans Mill Road
P.O. Box 18070
Pittsburgh, Pa. 15236

10. SUPPLEMENTARY NOTES

11. ABSTRACT (A 200-WORD OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY OR LITERATURE SURVEY, MENTION IT HERE.)

The drive toward increased safety for coal miners has led to the development of computer-assisted methods of underground coal mining. The development of control architectures and accompanying code for the control of the movement of continuous mining machines (tramming control) is an important part of this overall effort. The tramming control algorithm design described is in concert with hierarchical architecture design principles developed at National Institute of Standards and Technology (NIST), referred to as the Real-Time Control Systems (RCS) methodology. The algorithm design with accompanying code allows for the control of both cutting and free-space movement by a continuous mining machine (CM) and allows for a high degree of human operator interaction.

12. KEY WORDS (6 TO 12 ENTRIES; ALPHABETICAL ORDER; CAPITALIZE ONLY PROPER NAMES; AND SEPARATE KEY WORDS BY SEMICOLONS)
algorithm; computer-assisted; continuous mining machine; control; error analysis; plan; real-time; state transition diagram; task; task decomposition; tram; underground coal mining;

13. AVAILABILITY

☒ UNLIMITED

□ FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS).

□ ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402.

☒ ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.

14. NUMBER OF PRINTED PAGES
45

15. PRICE
A03

ELECTRONIC FORM