OMNITECH ROBOTICS

MARS

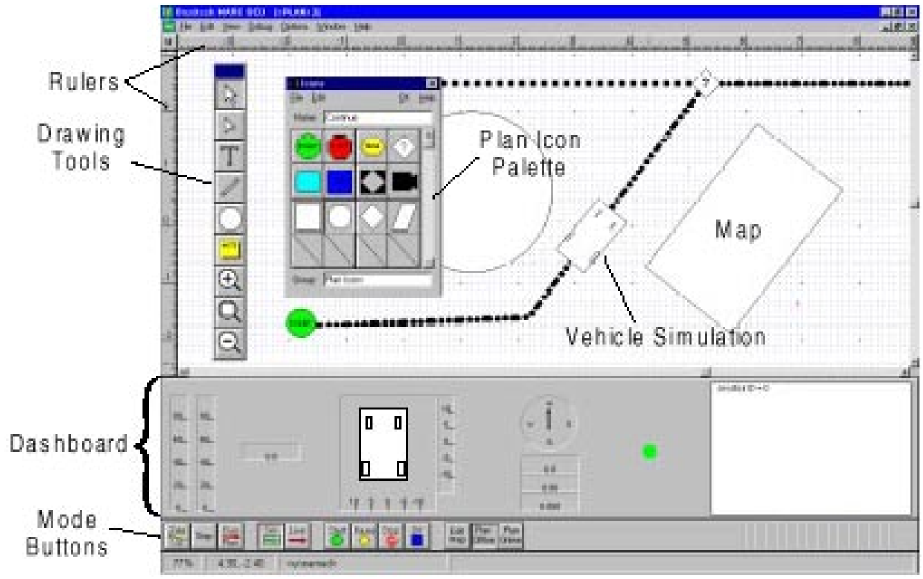
Modular Autonomous Robotic System (MARS)

Leading the way for Unmanned Ground Vehicle Controls

Contact Information

OMNITECH ROBOTICS INTERNATIONAL, LLC
2640 South Raritan Circle
Englewood, CO 80110

Tel.: (303) 922-7773
Fax: (303) 922-7775
Email: info@omnitech.com
Website: www.omnitech.com



MARS Overview

MARS is a software product that uses the STS and COMPASS as low-level control system components, providing a comprehensive autonomous vehicle control system environment.

Typically, programming mobile robots even in structured environments requires highly trained engineers. Many new application areas for mobile robots will open and current applications will be more successful when it is easier for end-users to assign mobile robots to new tasks.

Omnitech Robotics is completing a National Aeronautics and Space Administration (NASA) Jet Propulsions Laboratory (JPL) sponsored Small Business Innovative Research (SBIR) phase II effort on the MARS program. We are developing hardware and software infrastructure technologies for configuring any vehicle to use a variety of sensors and navigation strategies optimized to the specific application and operational environment of the robot. The work on MARS includes development of a graphical user interface for creating and editing mobile robot plans and supervising mobile robot operations.

"MARS" does not refer to a single robotic system, but rather a set of hardware and software components that can be integrated in a variety of ways to create automated robotic vehicles with capabilities targeted to specific tasks within differing cost and performance constraints. The software developed for MARS and hardware components, including the STS and COMPASS, support this concept.

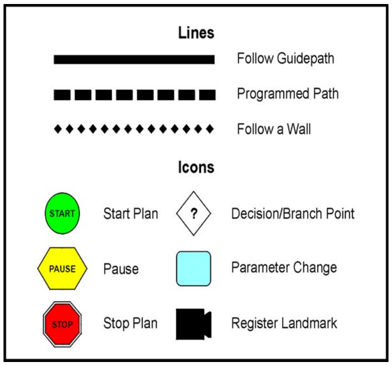
MARS OCU User Interface

The MARS OCU is designed to be used by both Robot Programmers and Robot Operators. The user interface is shown above. It supports display of plans and maps, graphical editing of plans and maps by direct manipulation, plan testing in simulation and vehicle operation. This interface design conforms to Microsoft Windows 95 conventions.

Plans in MARS are represented graphically by lines and icons. Lines represent routes to be navigated. Icons represent actions to be performed at a particular location, mode changes and decision points. The plan is overlaid on a map and the location of a plan step corresponds directly to the physical location for performing that step. We have found this to be a highly intuitive representation.

The icon graphic corresponds to the action to be taken at that location and the line dashing style corresponds to the navigation skill to be used for traversing the indicated route. The figure above shows some of the standard lines and icons.

Each line has a start node and an end node. Lines are attached to icons by placing the end of the line within the icon boundary. These attachments, along with the line directions, establish the sequence of steps in a plan. Any icon can have multiple incoming lines attached, but only decision point icons can have multiple outgoing lines. When editing a plan, lines attached to an icon move when the icon is moved.



Also associated with each plan step is a template containing parameter values for a particular plan step. The template dialog box, at left, is accessed by right-clicking on an icon or line.

The parameters displayed in the plan step template dialog box depend on the type of plan step. For example, a ProgrammedPath step might include values for nominal speed and maximum deviation from the path. Each plan step also has Preconditions, Post-conditions and Exception Handlers, which are represented using a simple macro language that is interpreted at run time. Decision points conditions include macro language statements of the conditions for taking a particular branch.

Navigation Methods

In addition to teleoperation, the MARS system is designed to support multiple navigation methods (or skills) with smooth transitions between navigation modes. Supported navigation methods include the following:

1. Continuous Guidepath Following

Autonomous vehicles often follow a continuous guidepath to navigate from point to point. This guidepath can be inductive wire, reflective tape, magnetic tape or ultraviolet paint. Continuous guidepath sensing reduces the local navigation problem to simply applying a feedback control loop to the vehicle's heading. Continuous guidepath systems are generally quite reliable. However, they can be expensive to install and are not easily modified.

2. Periodic Guidepath Following

Guidepaths do not have to be continuous if the vehicle can maintain a constant heading until the next path segment is in view. Some vehicles pilot by following walls using ultrasonic range sensors. Other vehicles may navigate via high-contrast floor markings using video. Outdoor vehicles can navigate by following roads using machine vision.

3. Programmed Motion

MARS supports several forms of programmed motion: MaintainHeading, Drive-To-Point and Programmed-Path. Commands may be in absolute (map) or relative (vehicle) coordinates. A Programmed-Path command is essentially treated as a sequence of Drive-To-Point commands. These functions depend on the availability of an accurate vehicle heading/position estimate. A variety of vehicle heading/position estimation methods can be used in conjunction with the programmed motion navigation skills, including COMPASS.

4. Landmark-Based Navigation

Structured landmarks can take a variety of forms including inductive tags, reflective tape, ultraviolet paint, bar-codes, infrared beacons, transponders, magnetic slugs or alternating black and white floor tiles. Existing environment features such as doors, walls or posts may be used as unstructured landmarks. A representation of the feature, generated using ultrasonic, vision or laser range sensors, is correlated with a previously programmed or learned map to register the robot position. Landmarks can be used in several ways depending on the nature of the landmark and associated sensors. A landmark may be used to identify the end of a guidepath segment, for periodic or continuous position updates during programmed motions, to establish stopping conditions for programmed motions (e.g. maintain heading until three feet from wall) or for special landmark-relative navigation skills (e.g. Home-To-Landmark).