Intelligent Behavior Generation for Autonomous Mobile Robots: Planning and Control

- CSOIS Autonomous Robotics Overview -

Kevin L. Moore, Director

Center for Self-Organizing and Intelligent Systems

Utah State University

Logan, Utah

February 2004

Outline

- Background
- ODIS – An ODV Robot for Physical Security
- USU UGV Architecture (Computing Hardware and Sensors)
- Mission Planning and Control System
  - – Multi-Resolution Approach
  - – Epsilon-Controller
- Intelligent Behavior Generation
  - – Delayed-Commitment Concept
  - – MoRSE: a Grammar-Based Command Environment
  - – Software Architecture
- Reaction via Feedback in the Planner
- Conclusions

Utah State University

Located in Logan, Utah, USA
80 miles North of Salt Lake City

18,000 students study at USU's Logan campus, nestled in the Rocky Mountains of the inter-mountain west

CSOIS is a research center in the Department of Electrical and Computer Engineering

CSOIS
CENTER FOR SELF-ORGANIZING AND INTELLIGENT SYSTEMS

CSOIS Core Capabilities and Expertise

- Center expertise is robotics, automation, control, and AI
- Control System Engineering
  - Algorithms (Intelligent Control)
  - Actuators and Sensors
  - Hardware and Software Implementation
- Intelligent Planning and Optimization
- Real-Time Programming
- Electronics Design and Implementation
- Mechanical Engineering Design and Implementation
- System Integration

We make real systems that WORK!

Center for Self-Organizing and Intelligent Systems

- Utah Center of Excellence graduate (formed in 1992)
- Horizontally-integrated (multi-disciplinary)
  - Electrical and Computer Engineering (Home dept.)
  - Mechanical Engineering
  - Computer Science
- Vertically-integrated staff (20-40) of faculty, postdocs, engineers, grad students and undergrads
- Average over $2.0M in funding per year since 1998
- Three spin-off companies since 1994
- Major commercialization in 2004
- Primary focus on unmanned ground vehicles and control systems

CSOIS Projects

- Since 1992: Approximately
  - 15 automation and control projects
  - 15 robotics/autonomous vehicle projects
  - Funding from both private industry and government
- Current focus on vehicle automation and robotics
- Major US Army Tank-Automotive Command (TACOM) program, 1998-present

Representative CSOIS Projects

- Intelligent Irrigation Systems (Campbell Scientific Inc.)
- Exercise Machines (Icon Inc.)
- Automated Wheelchairs (Marriner S. Eccles Foundation)
- Red Rover Educational Product (Visionary Products Inc.)
- NN Coin Recognition Device (Monetary Systems)
- Secondary Water Meter (Design Analysis Associates)
- Internet Telepresence Control
- Potato Harvester Yield Monitor
- Flat Panel Multi-Agent Interface Software (Driver Tech Inc.)
- Computer-Controlled Autonomous Wheeled Platforms for Hazardous Environment Applications (INEEL/DOE)
- Computer-Controlled Advanced Farm Systems (INEEL/DOE/Commercial)
- "Hopping" Robots
- Foundry Control Systems
- Small- to Mid-Scale Robotic Systems (US Army)

Current CSOIS Projects

- Intelligent Mobility Project (Moore/Flann/Wood, funded by TACOM)
- Distributed Sensor Nets (Moore/Chen, funded by SDL)
- Gimbal Control via ILC and Vision (Moore/Chen/Fulmer)

Recently-Completed CSOIS Projects

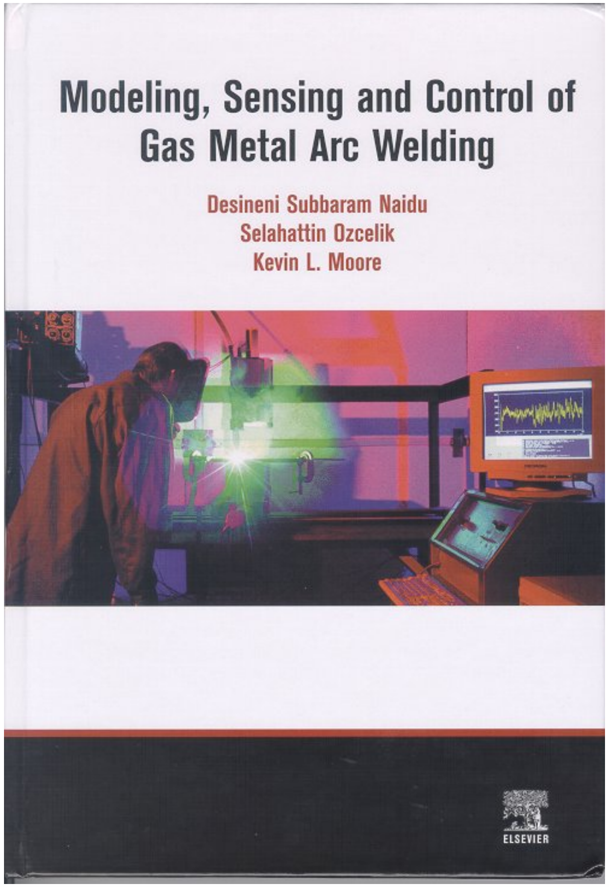
- Packing Optimization Project (Flann, funded INEEL)
- Automated Orchard Spraying Project (Moore/Flann, private funding)
- Vehicle Safety Project (Moore/Flann, funded by TACOM)
- Welding Control Project (Moore, funded internally)
- Shape-shifting robot (funded by VPI through a DARPA SBIR)
- WATV robot (CSOIS internally funded)
- Radar sensor project (private funding)
- Large tractor automation project (private funding)
- USUSAT (CSOIS internal funding of one student)
- Foundry Control Project (Moore, funded by DOE)
- Hopping Robot Project (Berkemeier, funded by JPL/NASA)
- Swimming Robot Project (Berkemeier, funded by NSF)

Cupola Control Project

- Cupola Furnace:
  - Charged with coke, metal, and other materials
  - Hot air blast with oxygen added
  - Diameters from 2’ to 15’, melt rates from 1 to 200 tons per hour
  - An essential part of most cast iron foundries
- Project Goal:
  - Develop intelligent control of meltrate, temperature, and carbon composition
  - Develop less reliance on operator experience and develop tools for automatic control

Welding Research

- <u>Goal</u>: achieve a “good” weld by controlling
  - Torch travel speed
  - Electrode wire speed
  - Torch height
  - Power supply
- Research led to a book



CSOIS Automated Vehicle Projects

- Rover Ballast Tail
- Marshod Rover Telepresence Control
- JPL Rocky Rover Fuzzy-Logic Navigation
- Red Rover
- Arc II Mini-Rover
- Arc III
- Triton Predator
- Yamaha Grizzly
- Tractor Automation Projects: 8200, 5510
- Seed Projects: WATV (Chaos) Robot, MANTS Robot
- TARDEC: T1, T2, T3, ODIS-I, ODIS-T, ODIS-S, T4, ODIS-T2

Center for Self-Organizing and Intelligent Systems

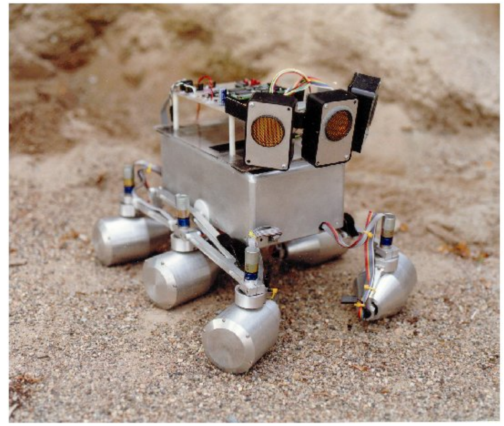
- Utah Center of Excellence graduate (formed in 1992)
- Horizontally-integrated (multi-disciplinary)
  - Electrical and Computer Engineering (Home dept.)
  - Mechanical Engineering
  - Computer Science
- Vertically-integrated staff (20-40) of faculty, postdocs, engineers, grad students and undergrads
- Average over $2.0M in funding per year since 1998
- Three spin-off companies since 1994
- Major commercialization in 2004
- Primary focus on unmanned ground vehicles and control systems

Utah State University

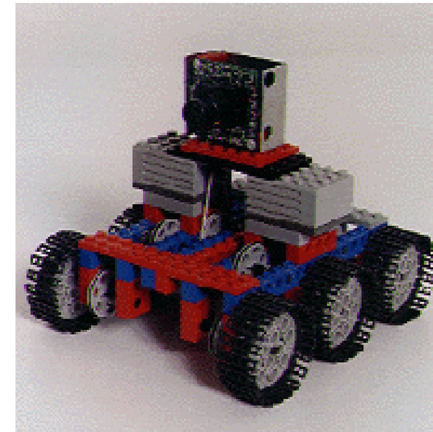
Some Robots Built At USU

CSOIS
CENTER FOR SELF-ORGANIZING AND INTELLIGENT SYSTEMS

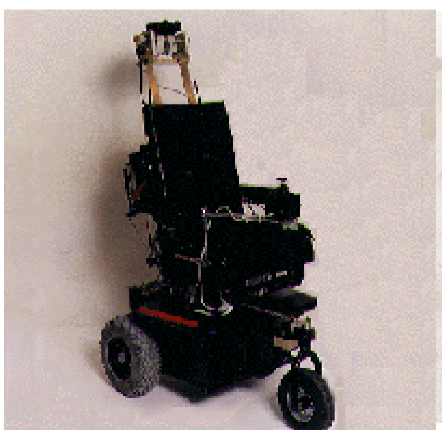
1994-1995 Rocky Rover



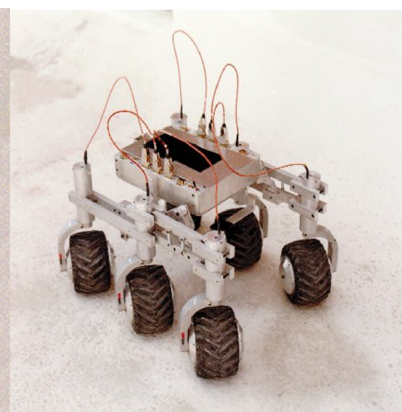
1995-1996 Arc II



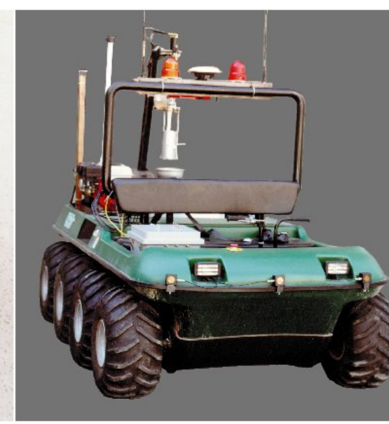
Red Rover-Red Rover VPI Spin-Off



Autonomous wheelchair



1996-1998 Arc III



1997-1998 Predator

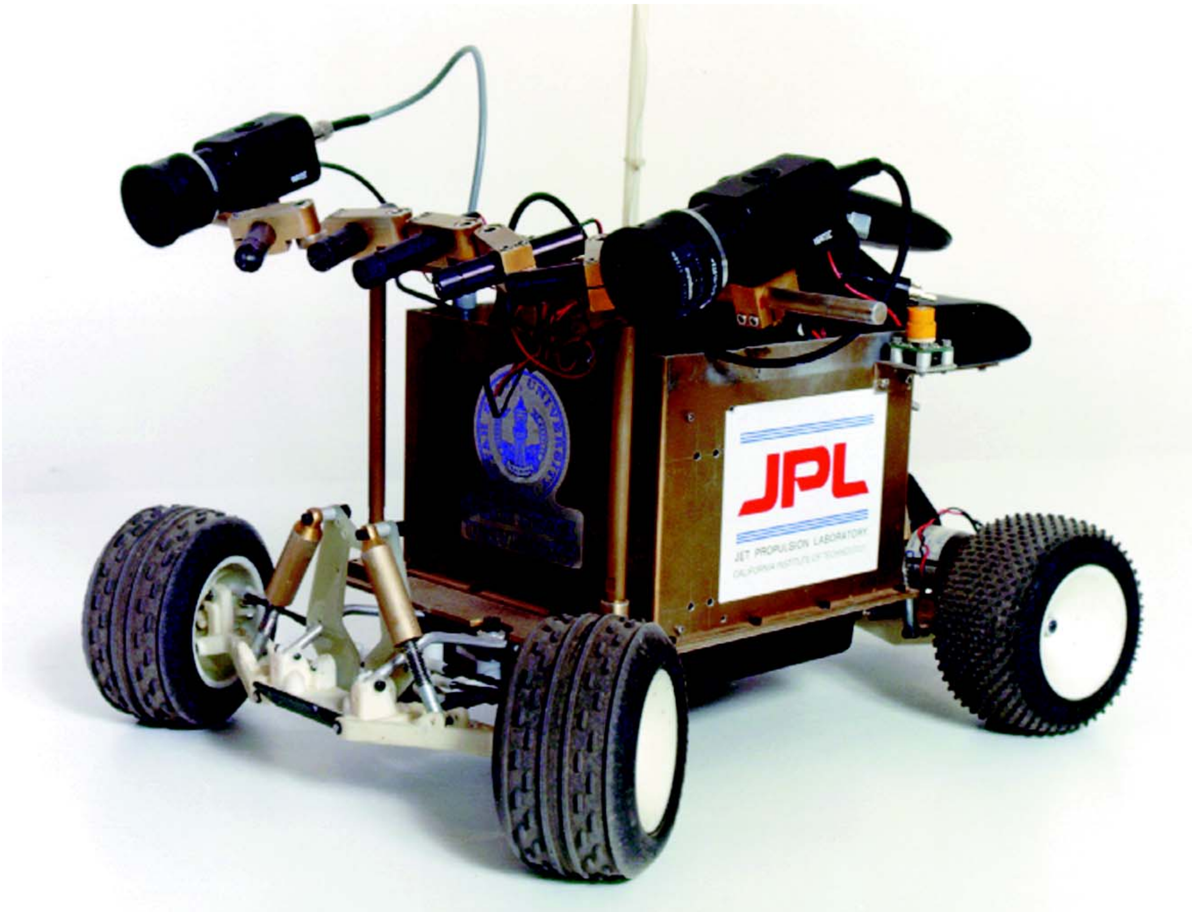
Predator with ARC II

Utah State University

1994-95: JPL Rocky Rover

CSOIS Center for Self-Organizing and Intelligent Systems

Mars Exploration Fuzzy-Inference Backup Navigation Scheme

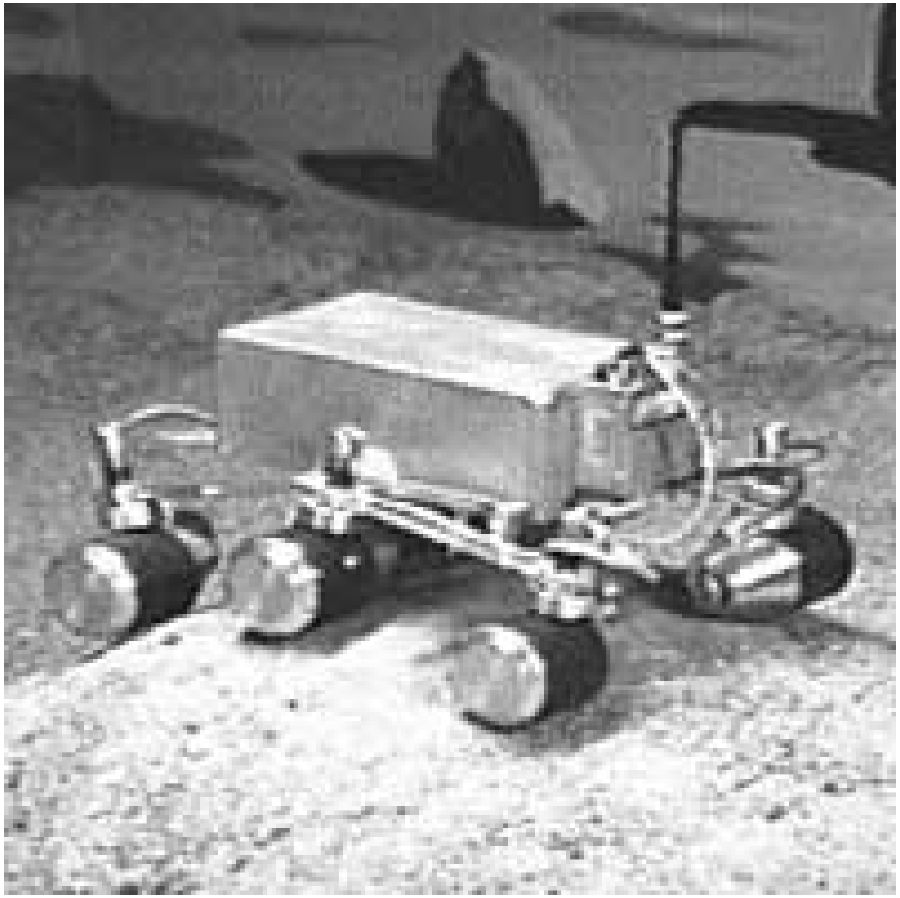


Rocky Rover Striping Laser Detector Array

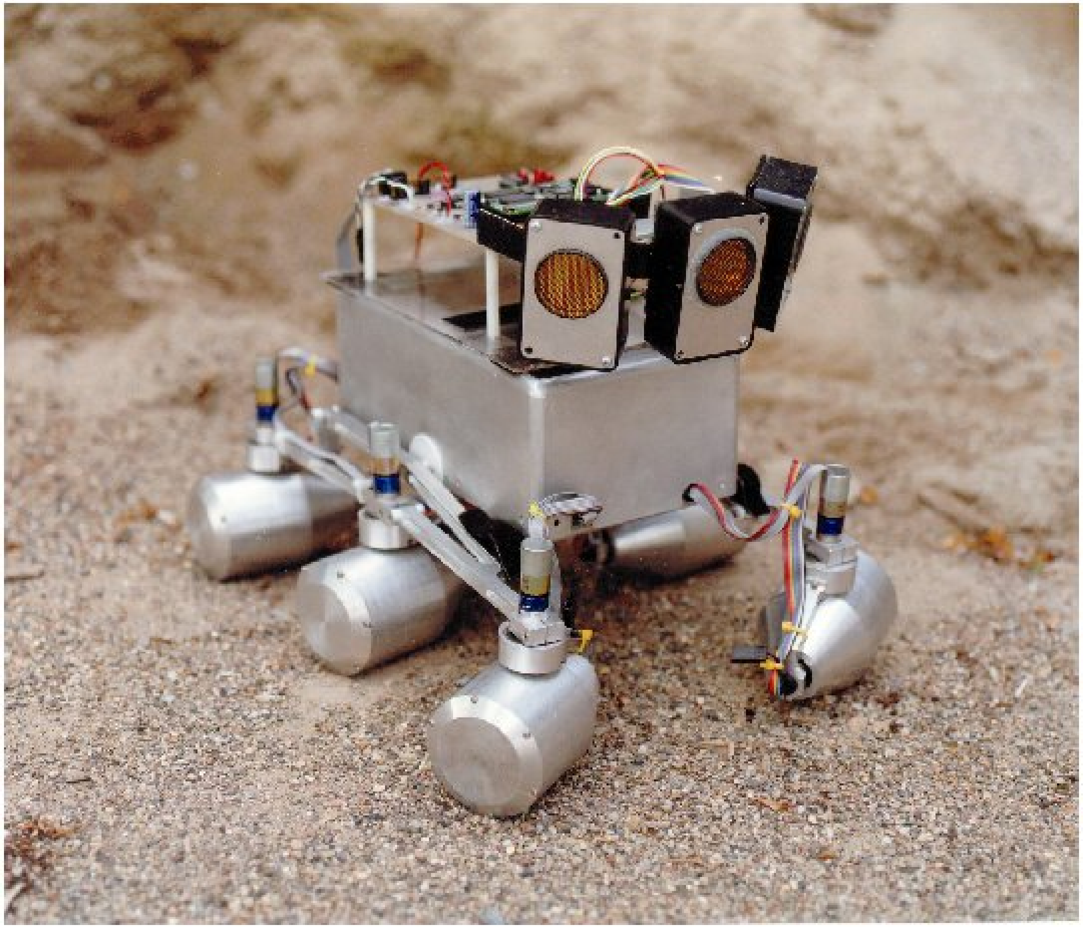
Red Rover, Red Rover Educational Project - 1995

- Collaboration with Lego and The Planetary Society
- Produced by CSOIS spin-off company, VPI
- Students build Rover and Marscape
- Other students drive Rover over the internet
- 500-600 were sold

ARC

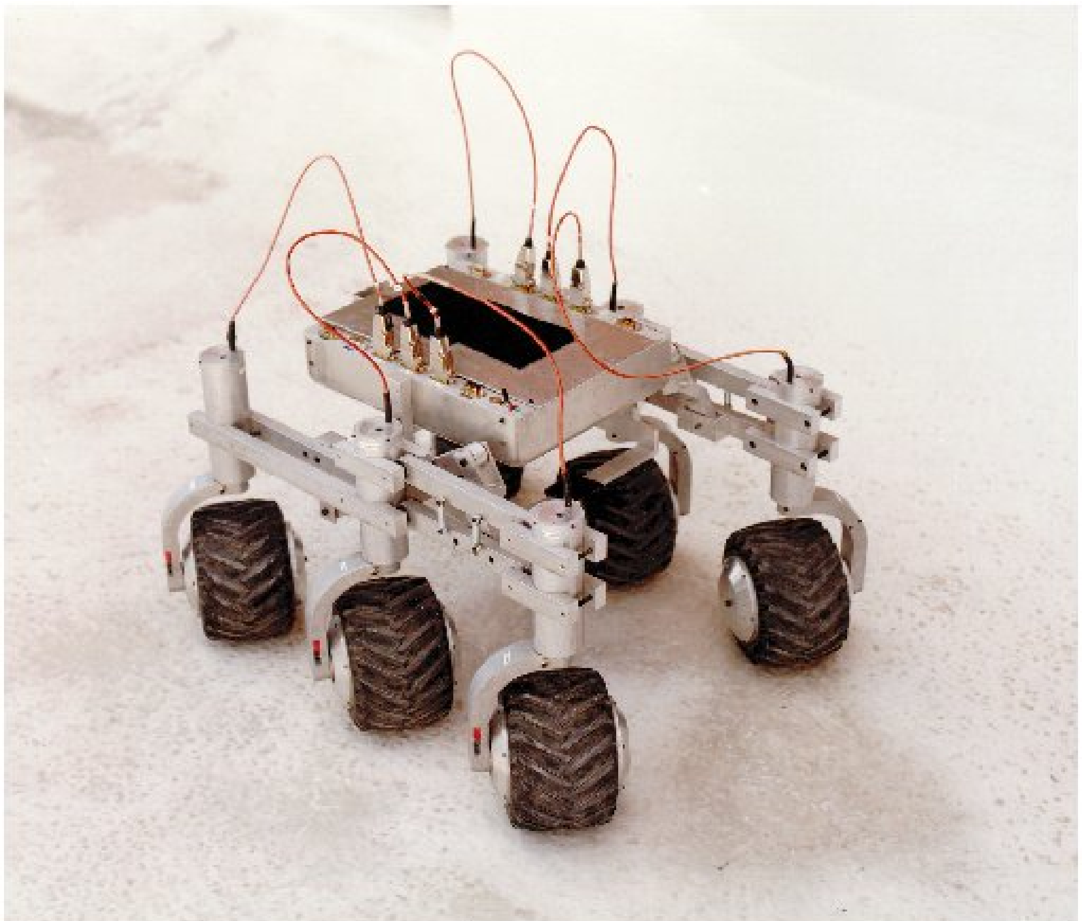


1995-96: ARC II Mini-Rover
Test for navigation and control



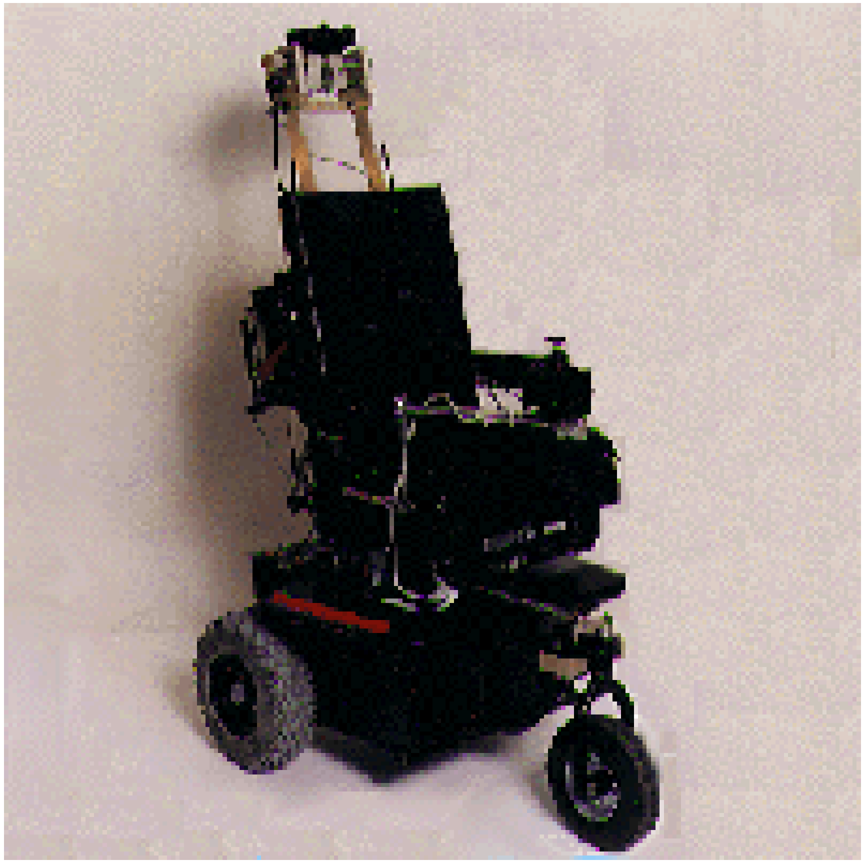
- Passive suspension
- Independent drive & steering motors
- In-wheel power
- Distributed controls

1996-1998: ARC III



- Practical size
- Multi-agent path & mission planning
- IR slip-ring
  - In-wheel controller & batteries

Autonomous Wheelchair Project



1997-98: Autonomous ATV-Class Computer Controlled Earth Rovers

- INEEL dual use
- CSOIS multi-agent path and mission planning
- dGPS (3-5 cm XYZ accuracy)
- 8-wheel track-type Triton Predator (1000 lb. unloaded)

Utah State University

CSOIS Center for Self-Organizing and Intelligent Systems

Triton Predator (Transport) with ARC III (Explorer)

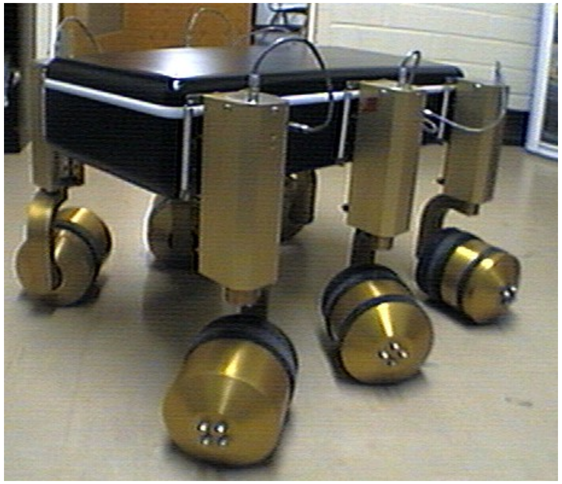
Yamaha Grizzly

Utah State University

CSOIS Center for Self-Organizing and Intelligent Systems

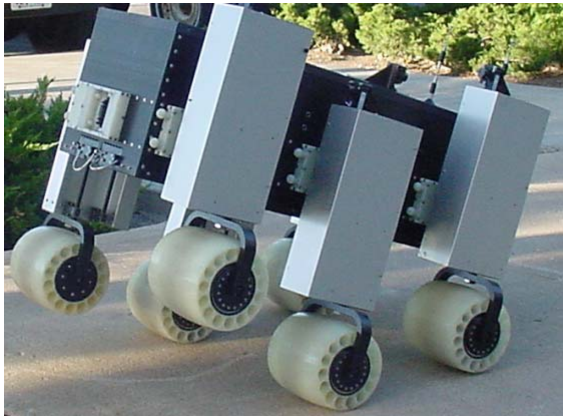
Some More Robots Built At USU

T1 -1998



T2 -1998

ODIS I -2000



T3 -1999

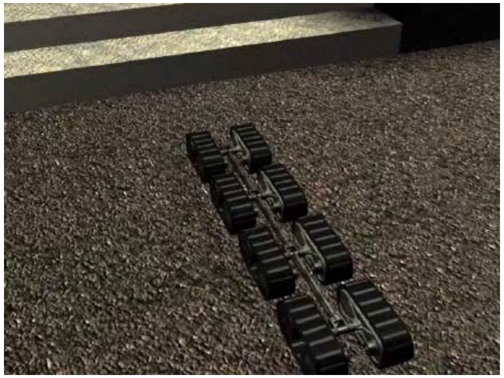
Utah State University

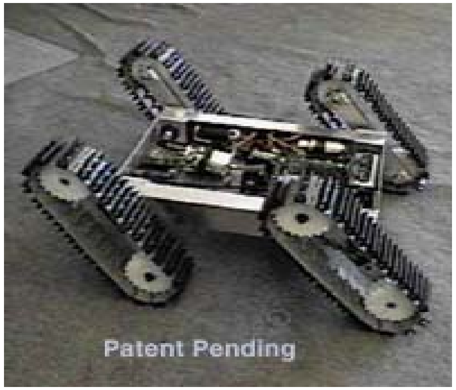
CSOIS CENTER FOR SELF-ORGANIZING AND INTELLIGENT SYSTEMS

Automated Tractor Projects

(CSOIS Spin-Off, Autonomous Solutions, Inc.)

Unique Mobility Robots





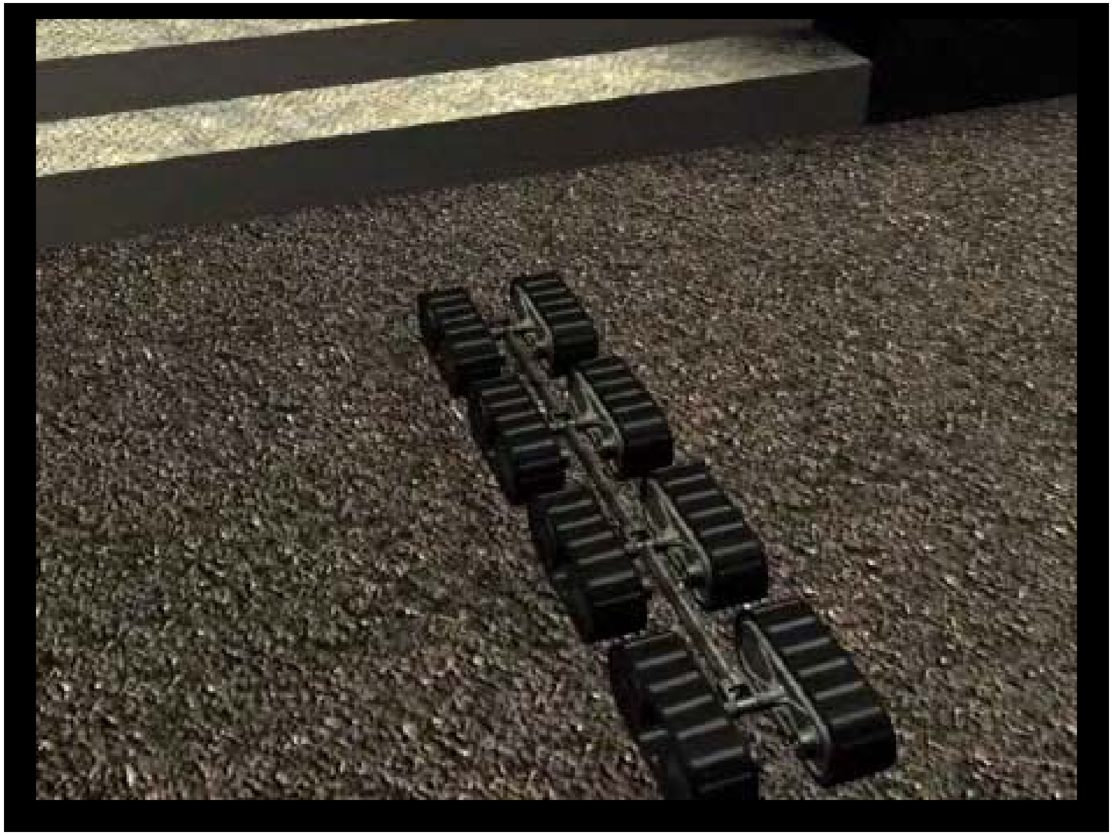
Automated Tractor Projects

(CSOIS Spin-Off, Autonomous Solutions, Inc.)

JD 8200

JD 5510N

DARPA SBIR with VPI



Walking Articulated Vehicle

Utah State University

CSOIS
CENTER FOR SELF-ORGANIZING AND INTELLIGENT SYSTEMS

Mote-Based Distributed Robots

Prototype plume-tracking testbed - 2004

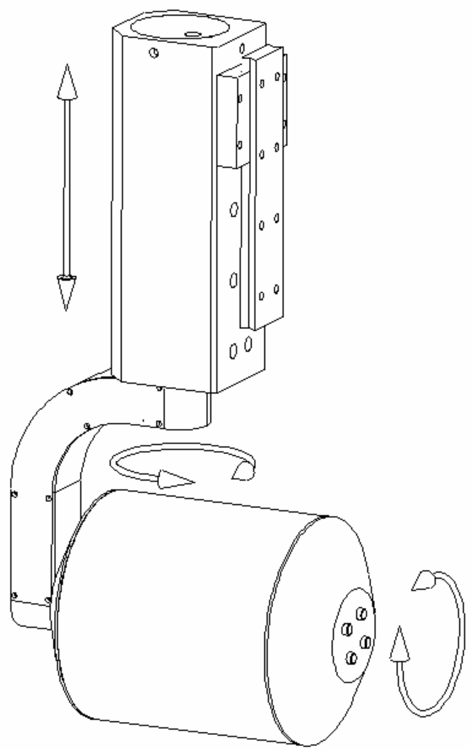
\$2000 2nd Place Prize in 2005 Crossbow Smart-Dust Challenge

Autonomous Vehicle Technology

- Autonomous vehicles are enabled by advances in:
  - Vehicle concept and mechanical design
  - Vehicle electronics (vetronics)
  - Sensors (e.g., GPS) and perception algorithms
  - Control
  - Planning
- We consider two key aspects of autonomy:
  - **Inherent mobility capability** built into the vehicle
  - Mobility control to exploit these capabilities

USU ODV Technology

- USU has worked on a mobility capability called the “smart wheel”
- Each “smart wheel” has two or three independent degrees of freedom:
  - Drive
  - Steering (infinite rotation)
  - Height
- Multiple smart wheels on a chassis creates a “nearly-holonomic” or omni-directional (ODV) vehicle



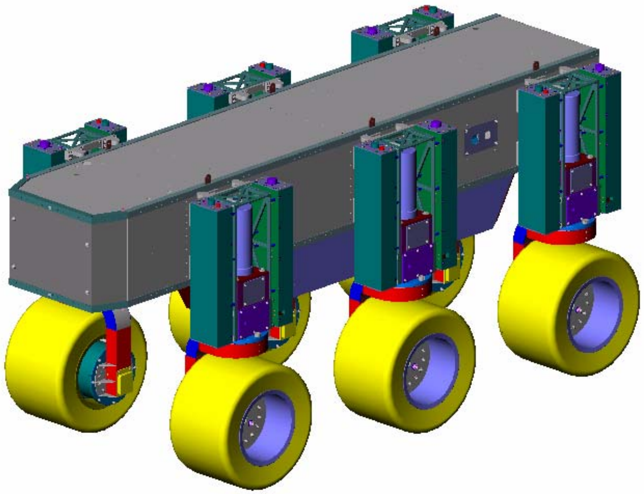

CSOIS

T1 Omni Directional Vehicle (ODV)

ODV steering gives improved mobility compared to conventional steering

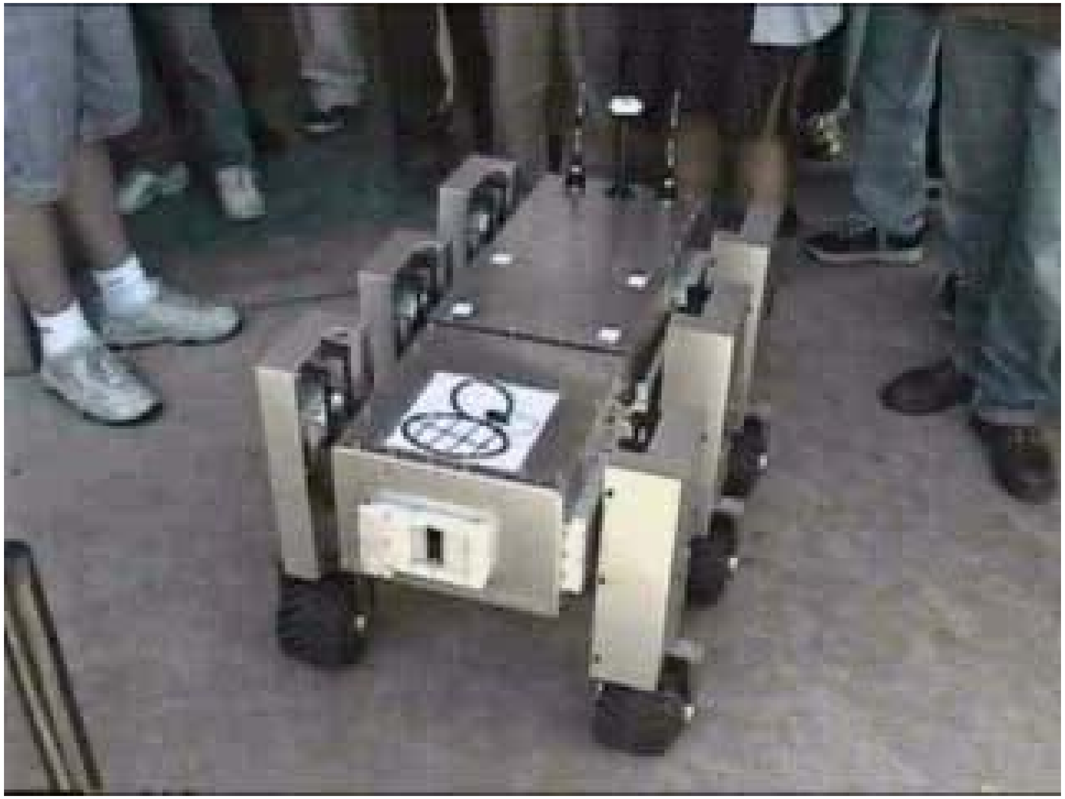
Smart wheels make it possible to simultaneously
- Translate
- Rotate

T2 Omni Directional Vehicle

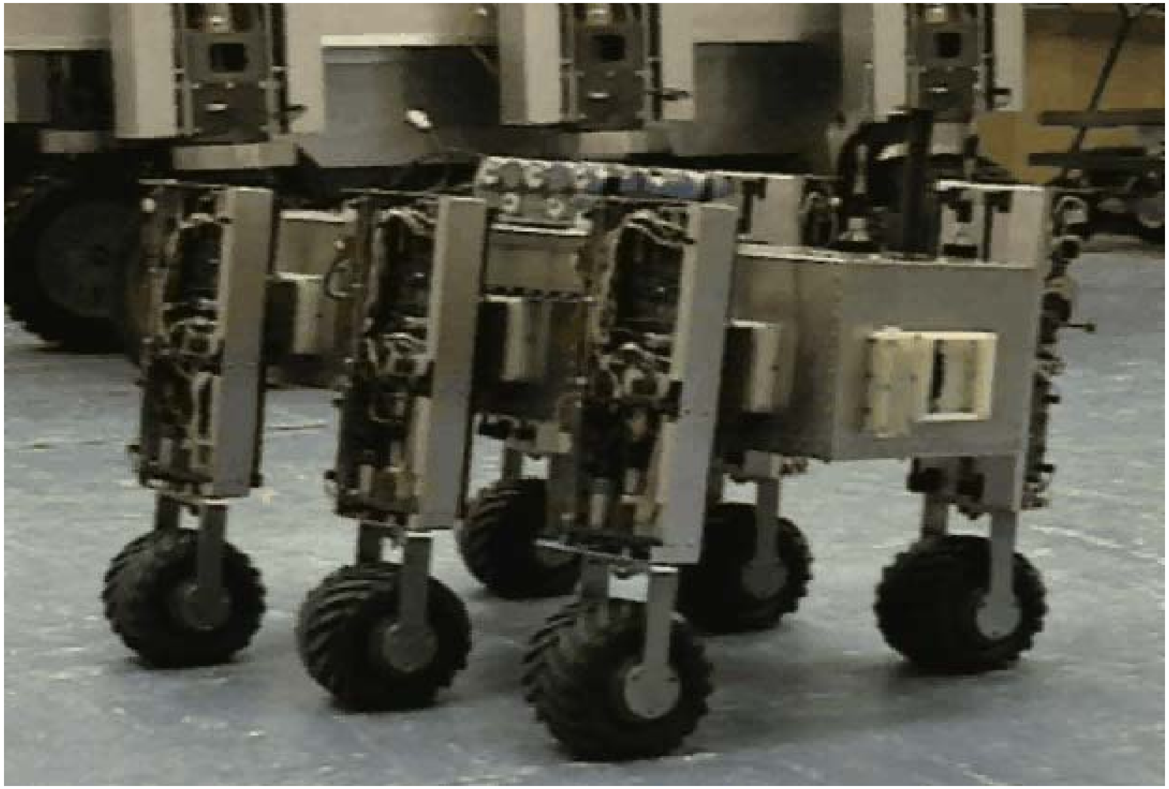


T2 can be used for military scout missions, remote surveillance, EOD, remote sensor deployment, etc.

T3 ODV Vehicle

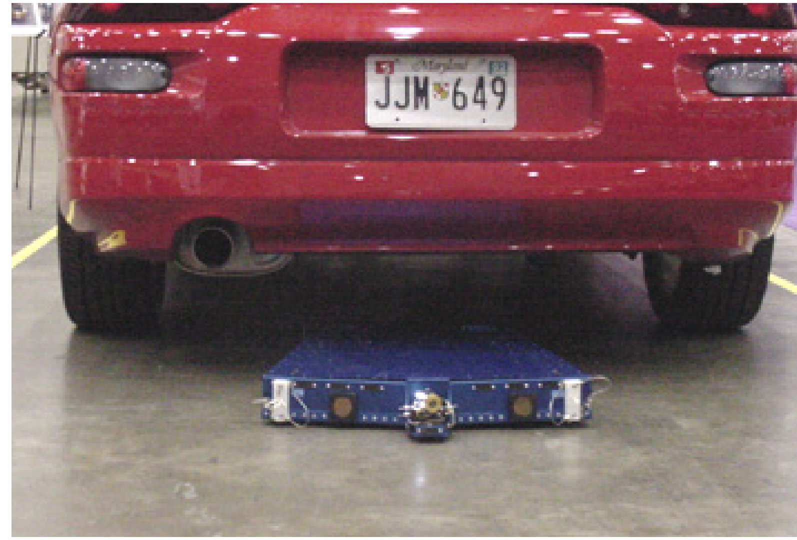


T3 Step-Climb Using a Rule-Based Controller



“Putting Robots in Harm’s Way So People Aren’t”

An ODV Application: Physical Security



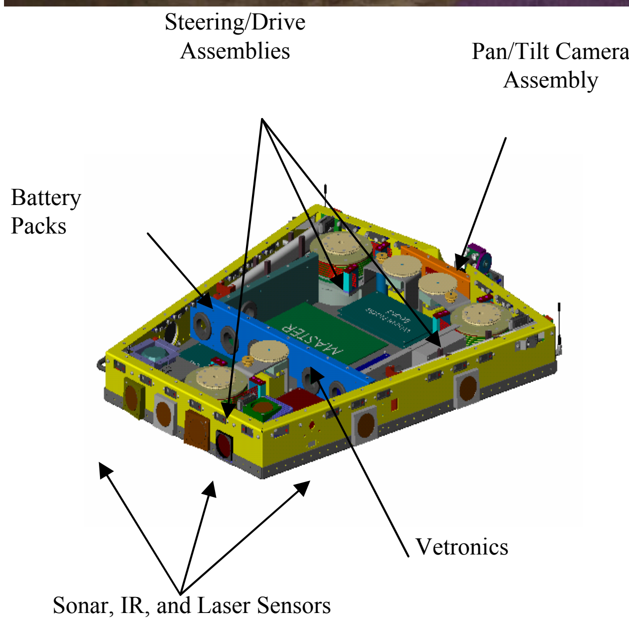
Omni-Directional Inspection System (ODIS)

- First application of ODV technology
- Man-portable physical security mobile robotic system
- Remote inspection under vehicles in a parking area
- Carries camera or other sensors
- Can be tele-operated, semi-autonomous, or autonomous

ODIS I – An Autonomous Robot Concept

ODIS I Description

- Laser Rangefinder
- IR Sensors
- Sonar
- FOG Gyro
- 3 Wheels



ODIS-T – A Tele-operated Robot

- Replaces traditional "mirror on a stick" at security checkpoints
- Joystick-driven; video/other sensor feedback to operator
- Ideal for stand-off inspection, surveillance, hazard detection

ODIS Under Joystick Control

(ODIS was designed and built in about four months)

“Mirror-on-a-Stick” vs. ODIS

Security, Law Enforcement, and Counter-Terrorism ODIS Applications

- Under vehicle inspection at security check points
- Parking lot and other surveillance
- Embassy protection
- Federal courthouse and other federal building protection
- Secret Service personnel protection activities
- Military physical security and force protection
- Customs/INS entry point inspection
- Public safety contraband detection
- Large public venue security – i.e. Olympics, etc.
- DoT vehicle safety applications
- Marsupial deployment by a larger platform

ODIS-T Sensor Suites

- Visual – pan/tilt imaging camera
- Passive & active thermal imaging
- Chemical sniffers – i.e. nitrates, toxic industrial chemicals
- Night vision sensors
- Acoustic sensors
- Radiation detectors – i.e. dirty bombs
- Biological agents detection
- MEMS technology – multiple threats
- License plate recognition

Can't Detect IED's, but …
Some Mission Packages Actually Deployed

1. LCAD Chem "Sniffer"

2. Radiation Detector (not shown)

3. IR Thermal Imaging Camera (recently driven vehicle)

•Continuous, real-time detection of CW Agents.

•Enhanced IMS technology using a non-radioactive source.

•Communication port for use with computer, ear piece or network systems.

• Small and lightweight

• Audio and / or visual alarm

•40 + hours on AA type batteries

•Data logging capabilities

• Detection of TIC'S (Toxic Industrial Compounds)

Mission Packages - IR

IR Image – Warm Brake

IR Image – Recently Driven Vehicle

ODIS Commercialization Status

- Field tested the ODIS-T:
  - in a Limited Objective Experiment (LOE) at the Ft. Leonard Wood (Mo.) Military Police School
  - At the Los Angeles Port Authority, with CHP cooperation
- Based on tests, have designed improved versions, the ODIS-S and the ODIS-T2
- A commercial license for ODIS-T2 has been negotiated between USU and Kuchera
- 20 ODIS-T2 robots have been built and will be deployed in Afghanistan and Iraq in Feb, with additional acquisition expected
- The ODIS-T2 technology can be considered COTS
- USU and Kuchera are working to develop other types of robotic mobility platforms for sensor payload delivery systems, both UGV and UAV

ODIS Robot Family

Utah State University

CSOIS
CENTER FOR SELF-ORGANIZING AND INTELLIGENT SYSTEMS

ODIS in Theatre

- 10 ODIS-T2 robots in Theaters since last March
- Additional 250 in production

ODIS in Theatre

- 10 ODIS-T2 robots in Theaters since last March
- Additional 120 in production

ODIS in Theatre

Utah State University

CSOIS
CENTER FOR SELF-ORGANIZING AND INTELLIGENT SYSTEMS

ODIS in Theatre

Utah State University

CSOIS Center for Self-Organizing and Intelligent Systems

ODIS in Theatre

Utah State University

CSOIS CENTER FOR SELF-ORGANIZING AND INTELLIGENT SYSTEMS

Stand-off is the main benefit

Security and Counter-Terrorism Applications for Larger Automated Vehicles

- Larger automated vehicles (tractors, construction equipment) can be used by security and law enforcement personnel for
  - Fire-fighting
  - Road-block and debris clearing
  - Building breaching
  - Crowd control
  - Explosive ordinance disposal

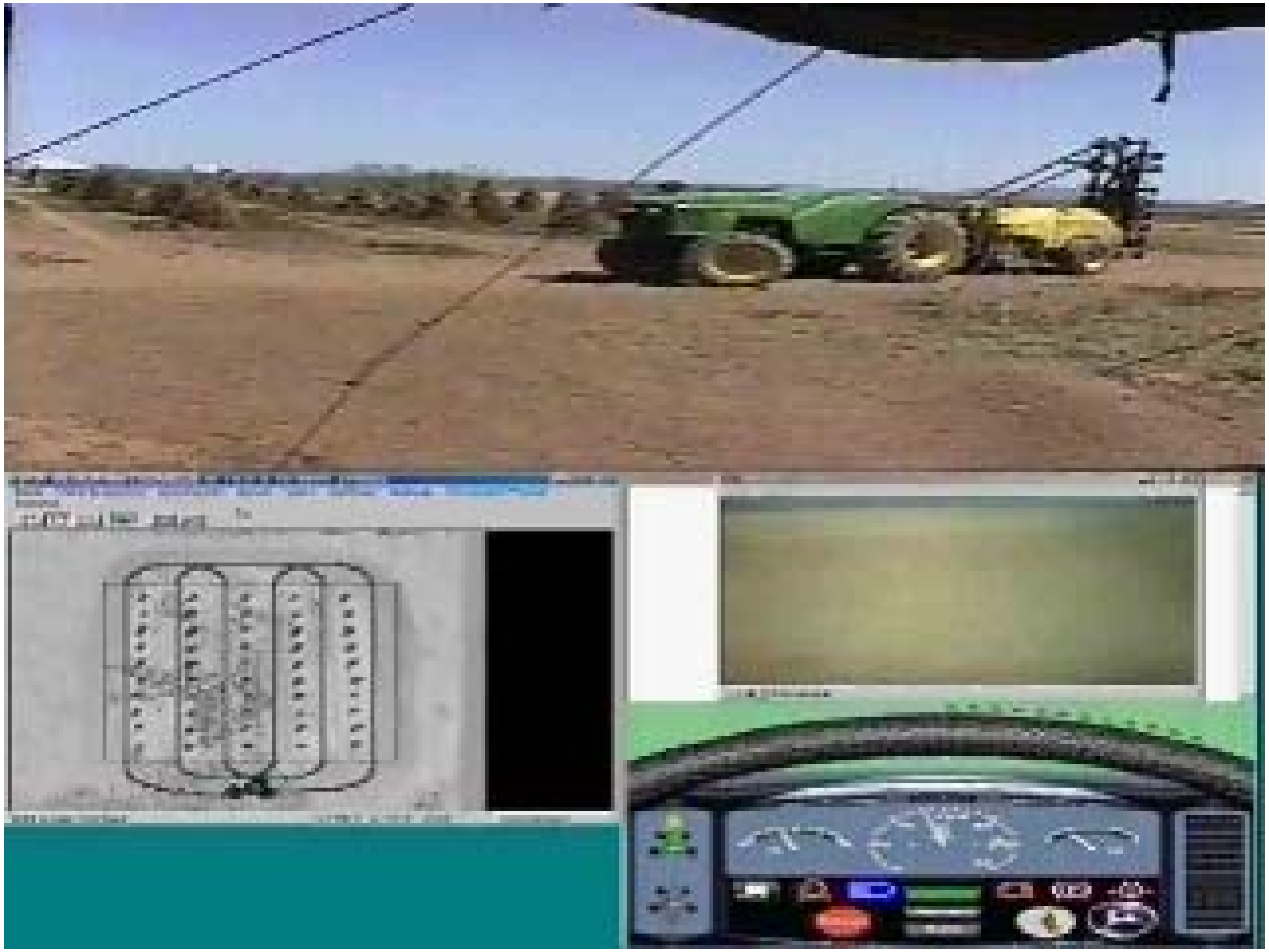
Automated Gator ATV developed by Logan-based CSOIS spin-off, Autonomous Solutions, Inc.

Automated Tractor Project

Utah State University

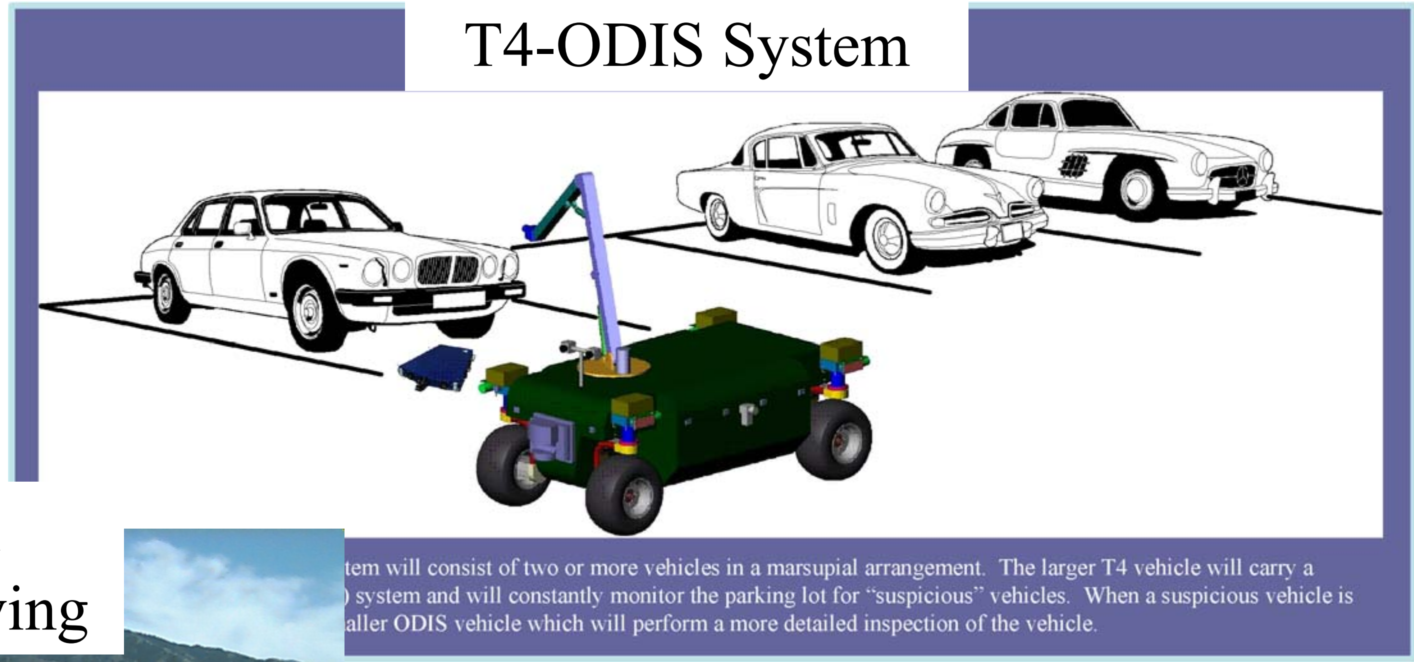
CSOIS
CENTER FOR SELF-ORGANIZING AND INTELLIGENT SYSTEMS

Automated Tractor Project



USU Multi-Vehicle Systems

T4-ODIS System



Coordinated Sampling/Spraying

Both the systems shown have been successfully demonstrated

UtahState UNIVERSITY

CSOIS CENTER FOR SELF-ORGANIZING AND INTELLIGENT SYSTEMS

T4 Parking Lot Surveillance Robot

- Omni-directional
- Hydraulically driven
- Gasoline Powered
- Designed to work in cooperation with ODIS

T4 Parking Lot Surveillance Robot

- Omni-directional
- Hydraulically driven
- Gasoline Powered
- Designed to work in cooperation with ODIS

UtahState UNIVERSITY

CSOIS CENTER FOR SELF-ORGANIZING AND INTELLIGENT SYSTEMS

T4 – Almost Done

- The T4 will be a “one-of-a-kind” hydraulic-drive, gasoline-powered ODV robot

T4 Hydraulic Smart Wheel

Drive Motors

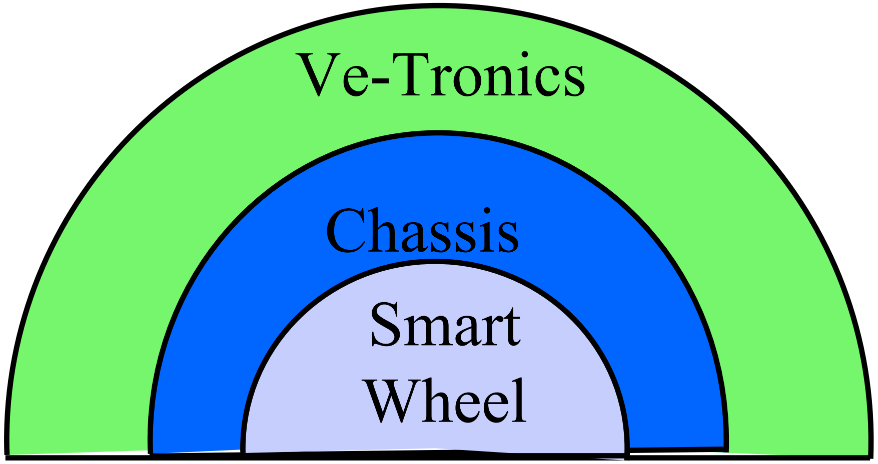
Drive and Steering Motors

Utah State University

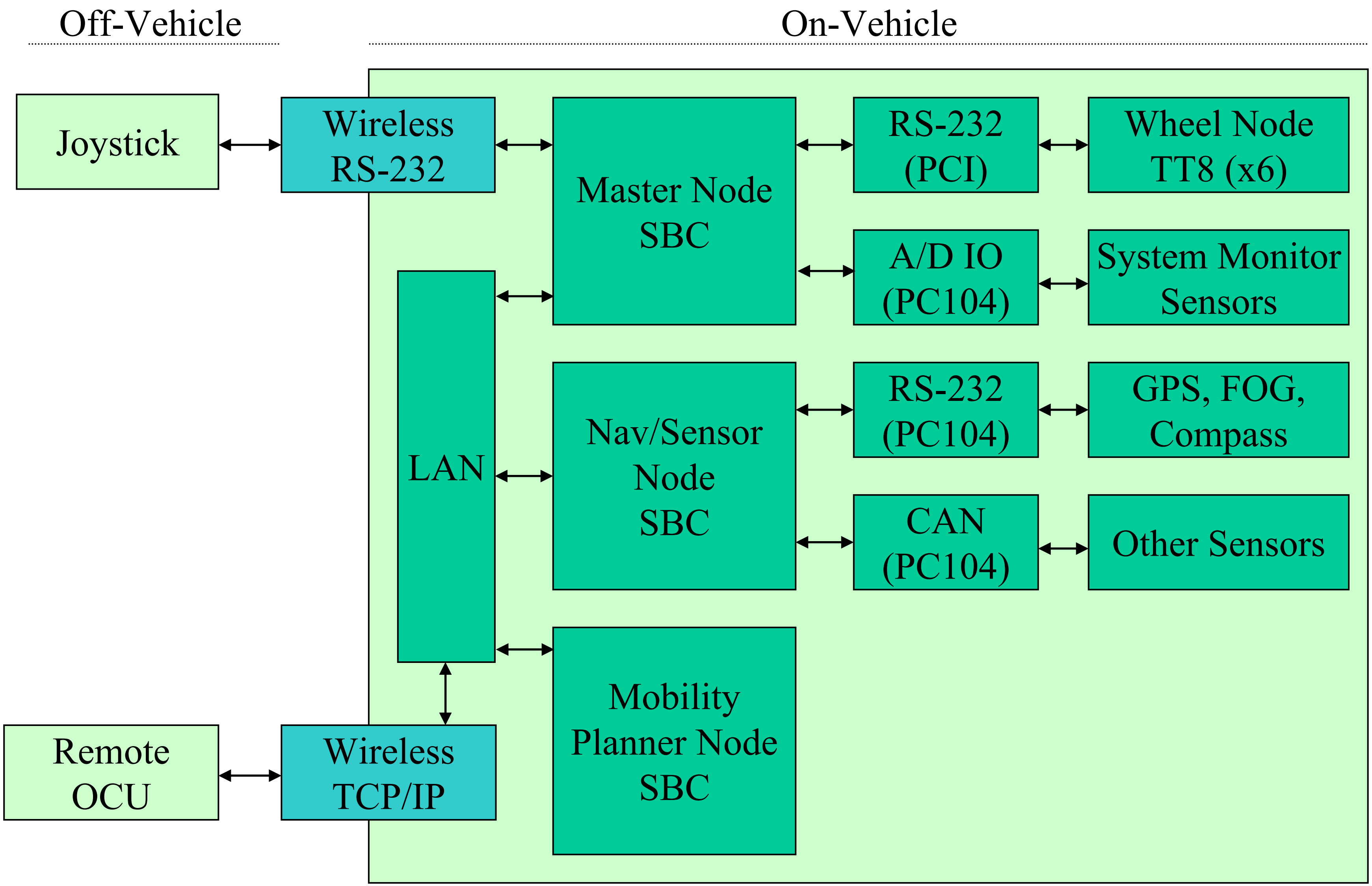
CSOIS Center for Self-Organizing and Intelligent Systems

T4-ODIS Cooperative Behavior

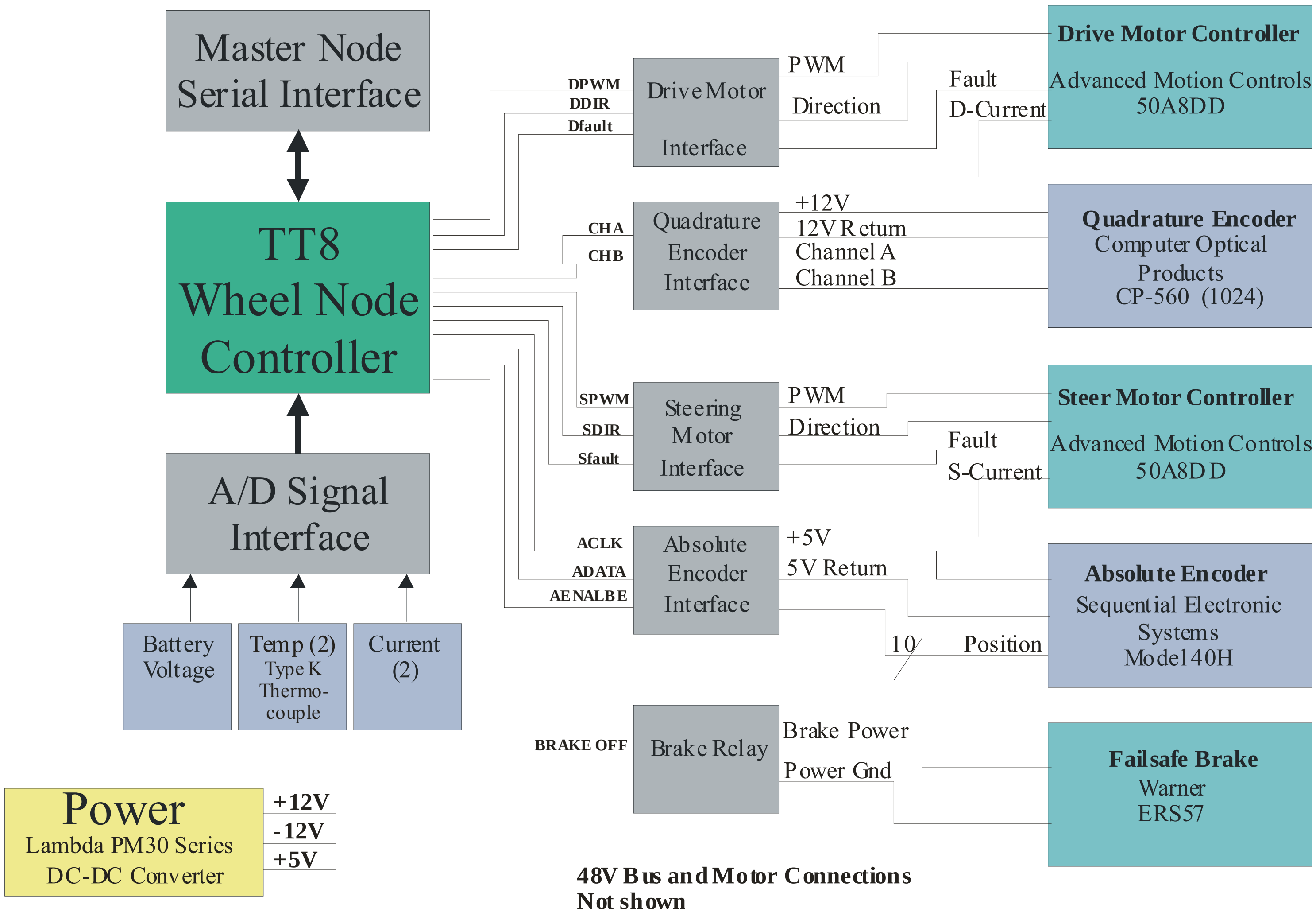
USU's UGV Technology



T2 Vetronics Architecture

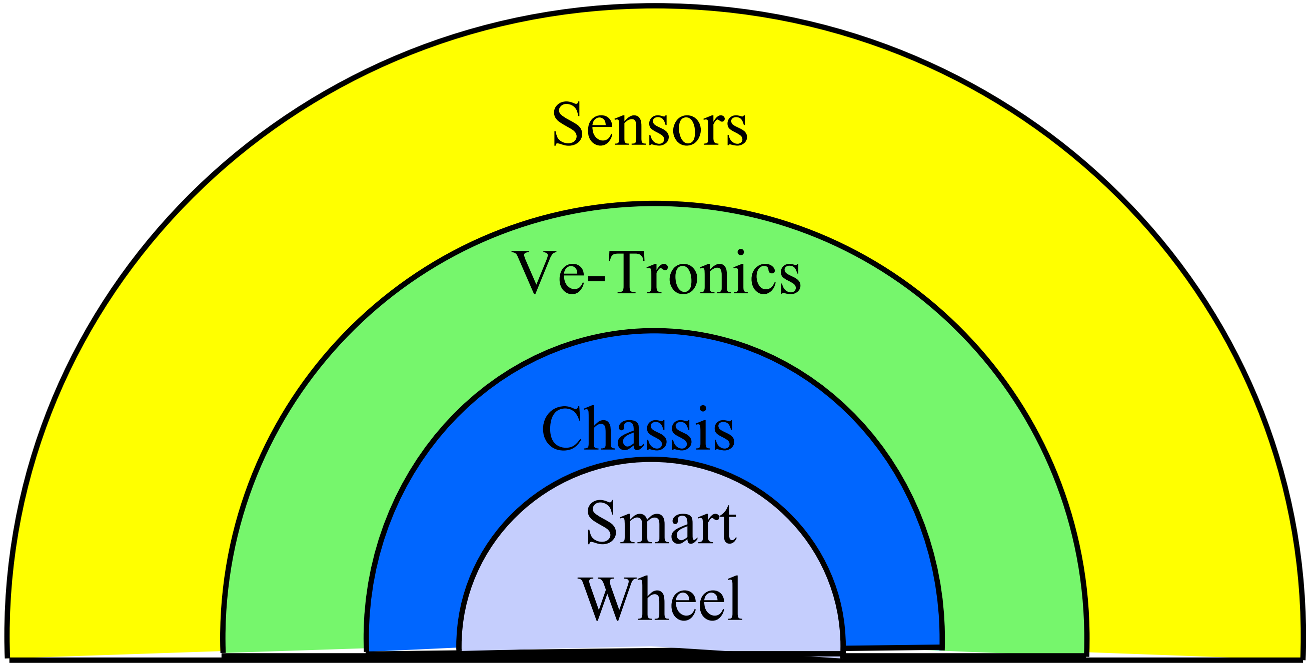


T2 Wheel Node
(Hardware Diagram)



48V Bus and Motor Connections Not shown

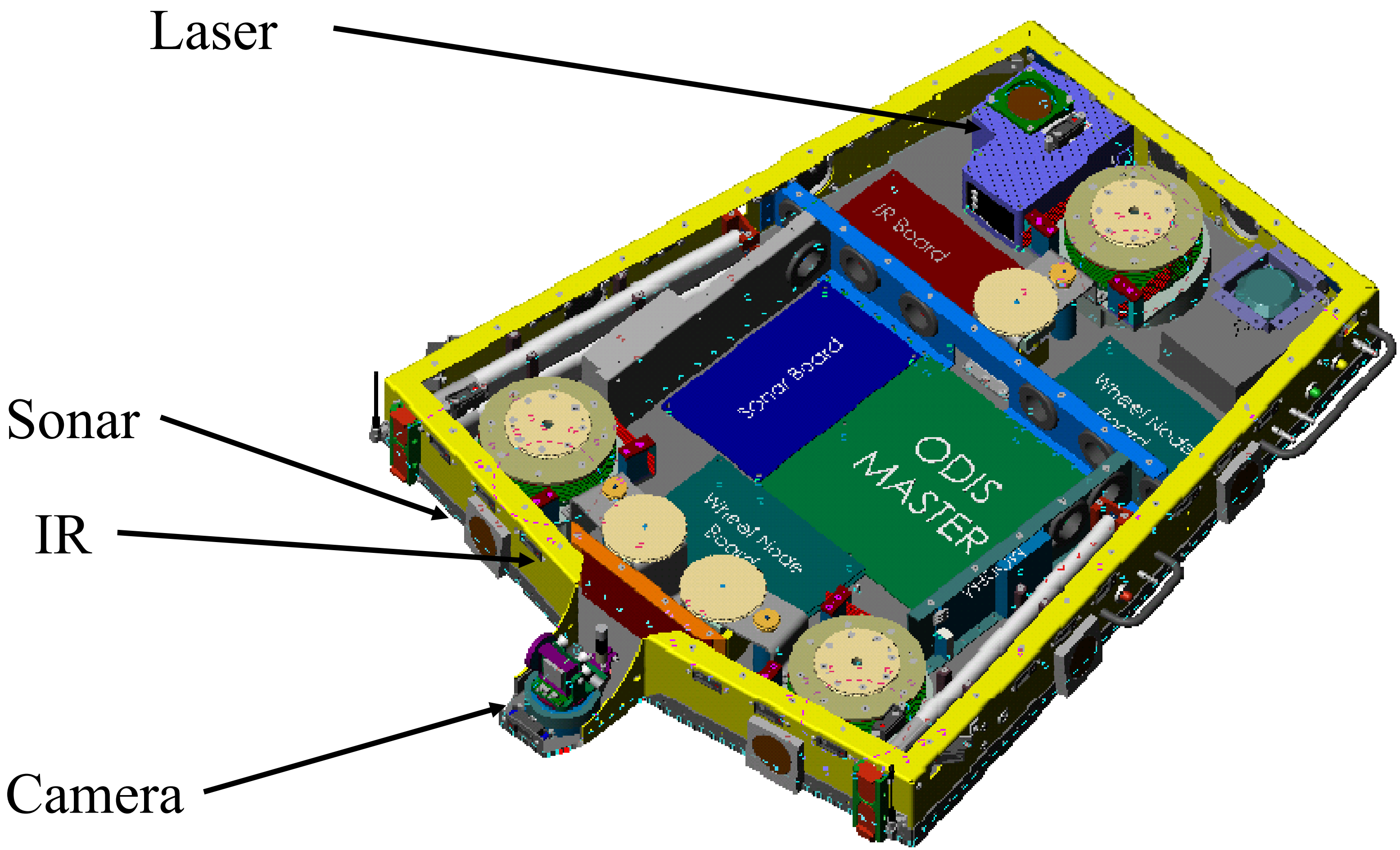
USU's UGV Technology



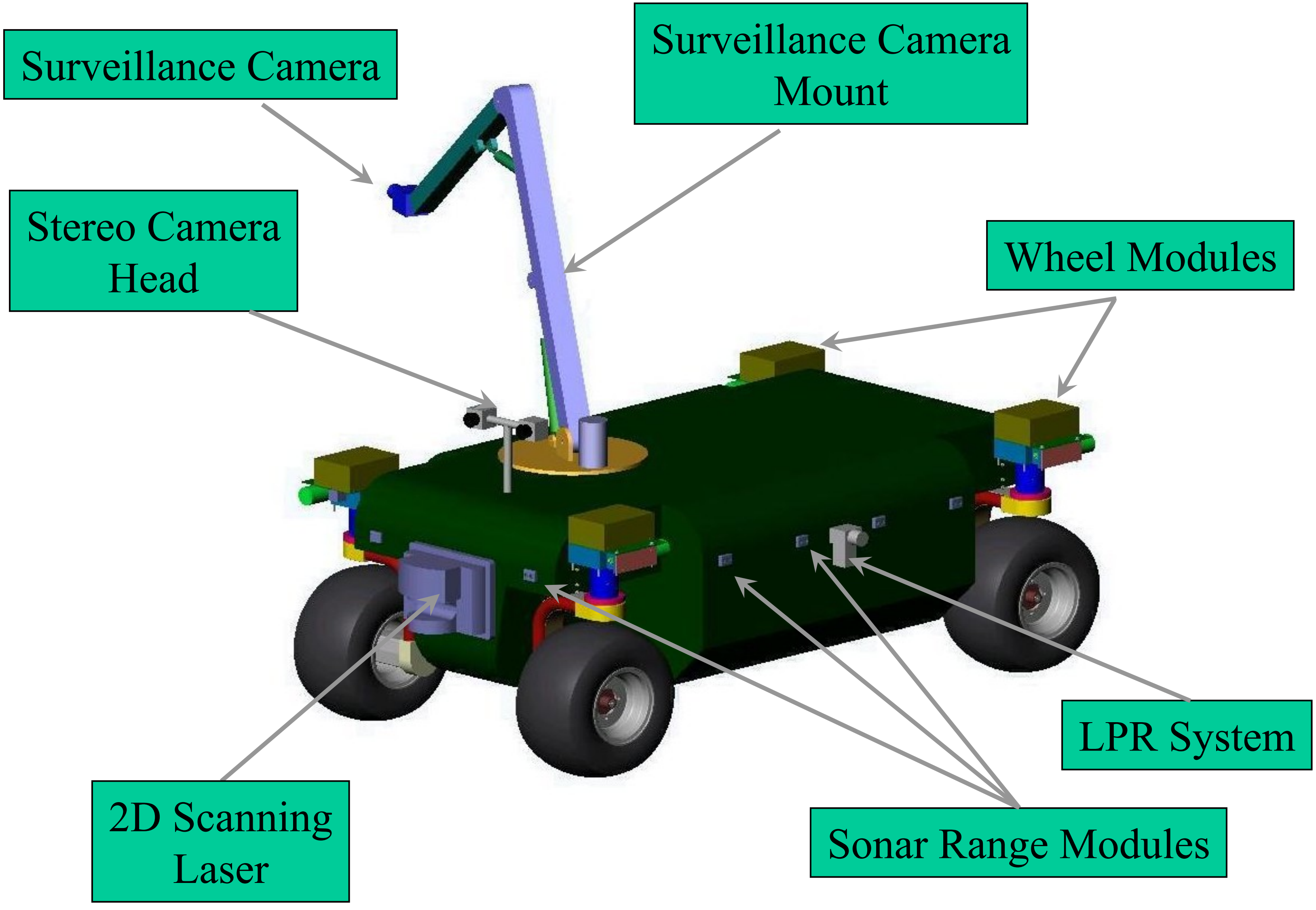
Utah State University

CSOIS
CENTER FOR SELF-ORGANIZING AND INTELLIGENT SYSTEMS

ODIS-I Sensor Suite



T4 Sensors - Artist's Rendition



T2e – A Testbed for T4 Behaviors

- The T2 was equipped with the sensors and vetronics that will be found on T4, to enable testing of intelligent behavior generation strategies; call it the T2e

Autonomous Vehicle Technology

- Autonomous vehicles are enabled by advances in:
  - Vehicle concept and mechanical design
  - Vehicle electronics (vetronics)
  - Sensors (e.g., GPS) and perception algorithms
  - Control
  - Planning
- We consider two key aspects of autonomy:
  - Inherent mobility capability built into the vehicle
  - **Mobility control** to exploit these capabilities

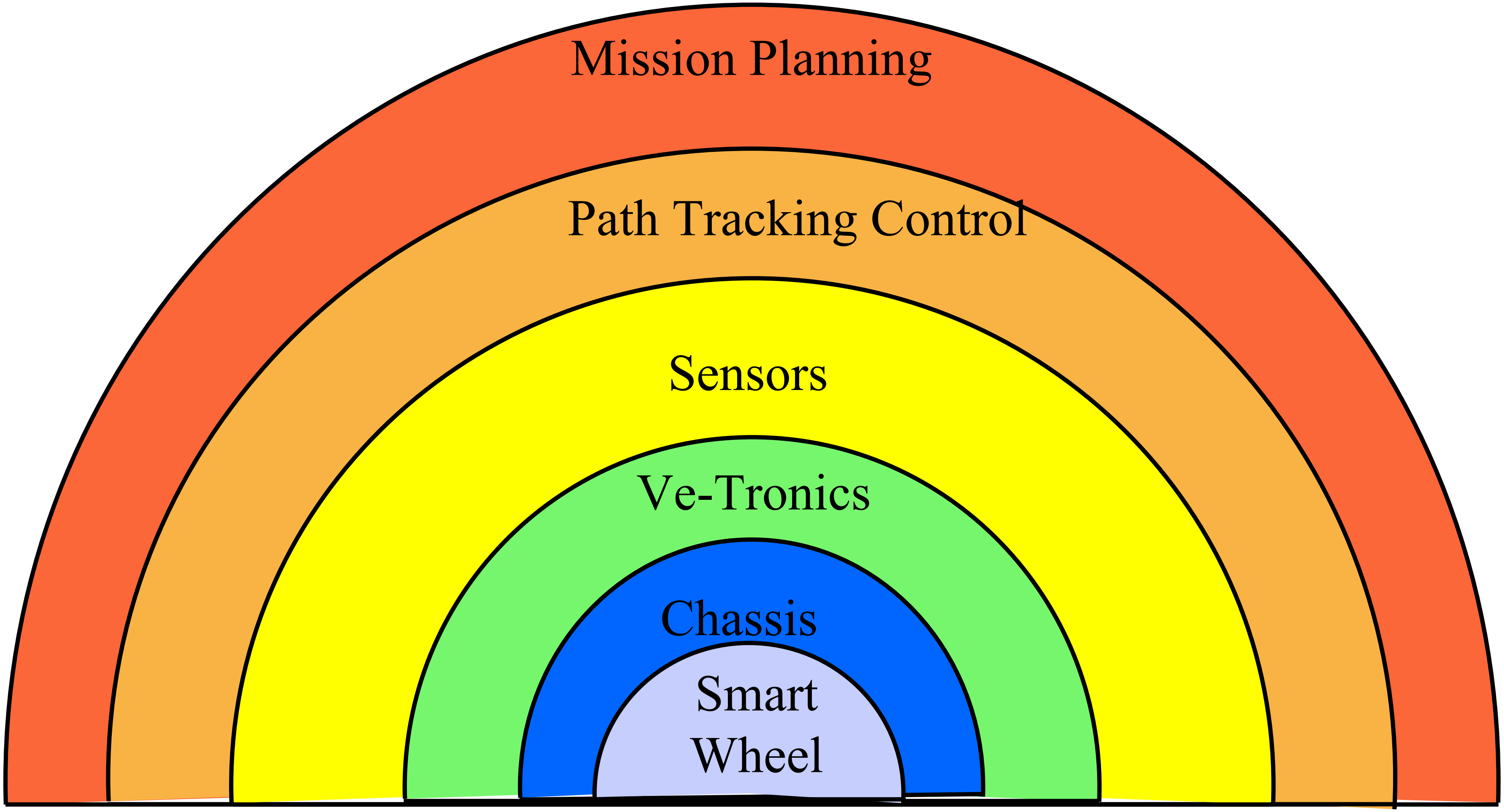
Just for Fun

CSOIS

Utah State University

CSOIS Center for Self-Organizing and Intelligent Systems

USU's UGV Technology



Mission Planning and Control System

•Transforms a collection of smart wheels into a smart, mobile vehicle

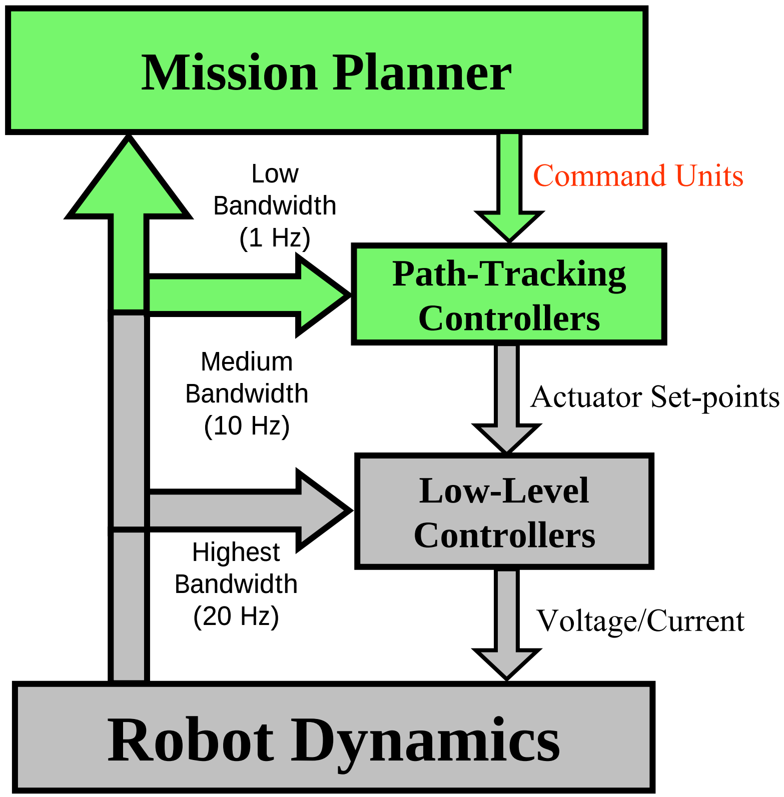
•Smart mobility is achieved by coordinating and executing the action of multiple smart wheels:

- –Wheel drive and steering: ARC III, T1, T2, ODIS, T4
- –Active height control: T3 concept

•Philosophy is to use a multi-resolution system to implement a “task decomposition” approach

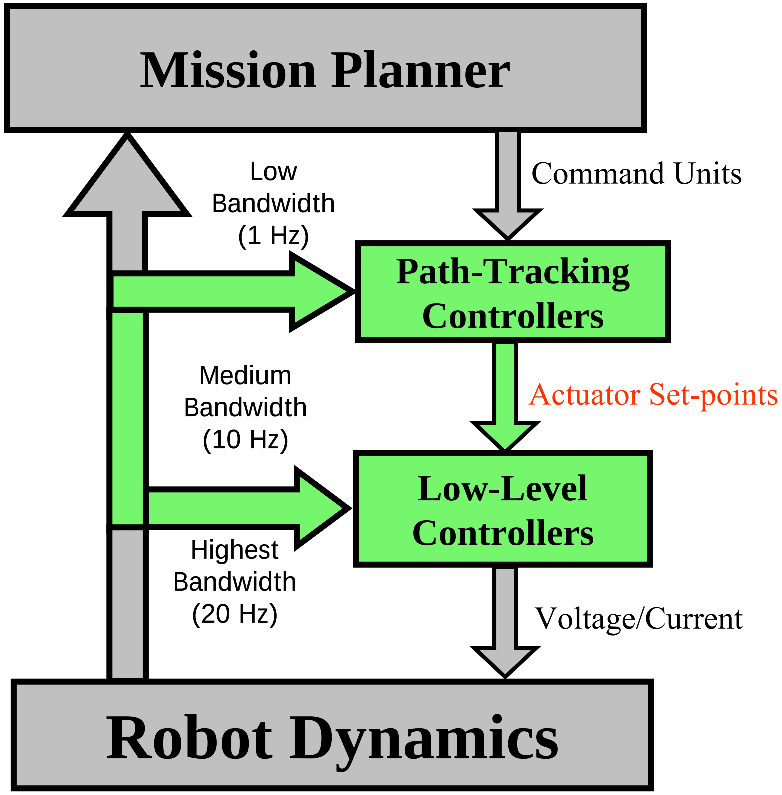
Multi-Resolution Control Strategy

- At the highest level:
  - The mission planner decomposes a mission into atomic tasks and passes them to the path tracking controllers as command-units



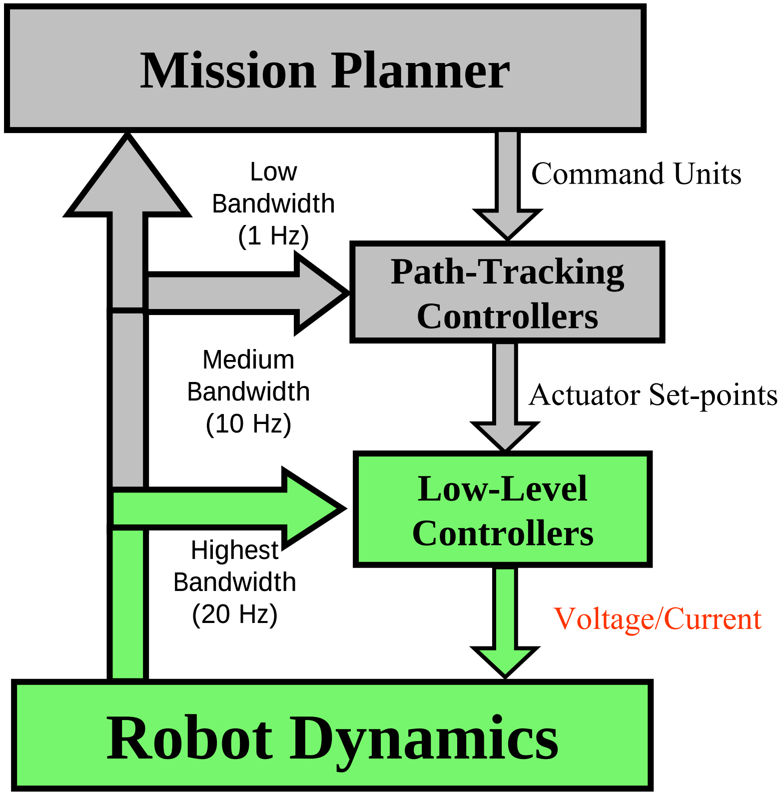
Multi-Resolution Control Strategy

- At the middle level:
  - The path tracking controllers generate set-points (steering angles and drive velocities) and pass them to the low level (actuator) controllers



Multi-Resolution Control Strategy

- At the lowest level:
  - Actuators run the robot

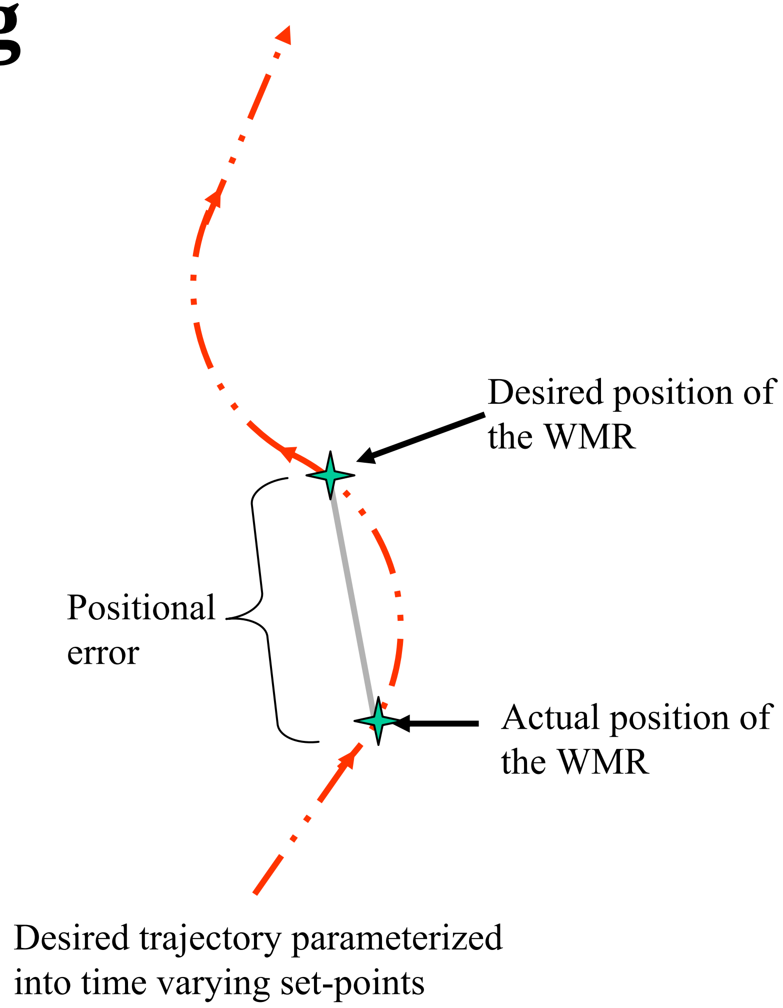


Path Tracking Strategies

- Fundamental for behavior generation
- Can be broadly classified into two groups
  1. Time trajectory based (temporal)
     – Desired path is parameterized into time-varying set-points
     – Locus of these set-points follow (in time) the desired trajectory (in space)
  2. Spatial
- We have implemented a variety of each type of controller on our robots

Disadvantages of Time Trajectory Path Tracking

- Indirect path tracking approach
- Can generate unexpected results, especially in presence of external disturbances and actuator saturation
- Positional errors due to this approach may cause the robot to "cut corners"
- Not suited for real time changes in desired speed along the path



Spatial Path Tracking Control Law: The ε-Controller (C_ε)

- Based completely on static inputs – the geometry of the desired path
- All desired paths are composed of either arc or line segments
- Real time variations of the desired speed (V_d) along the paths are allowed
- Uses only the current position (χ) of the robot as the feedback variable
- References to time are avoided in the controller development

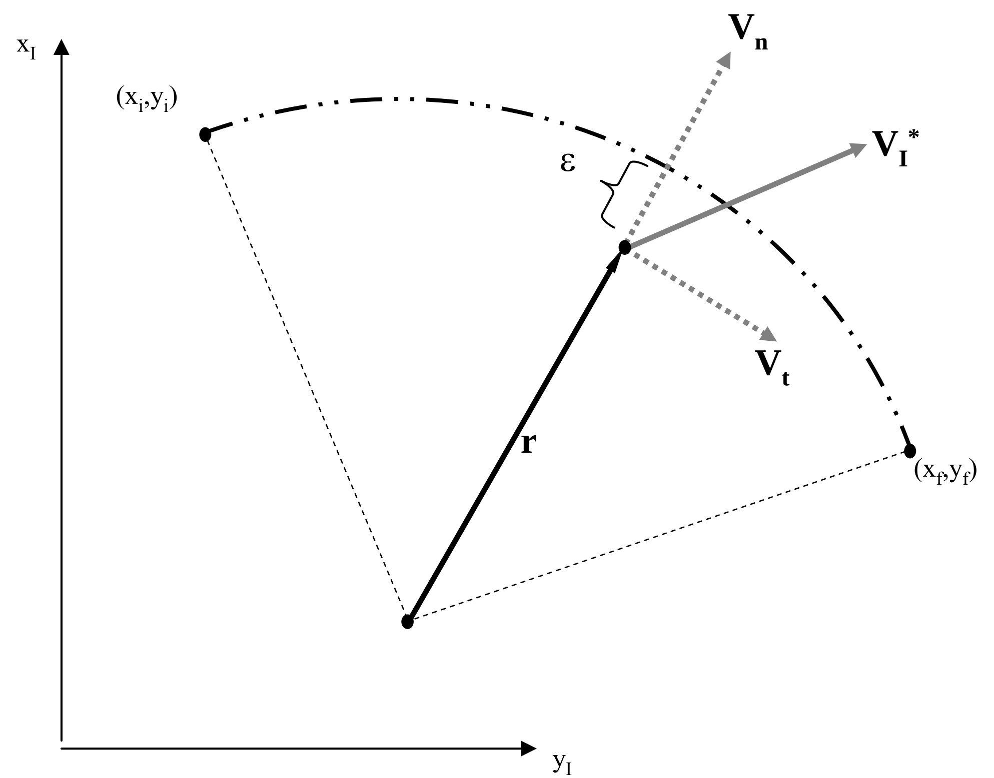
The Concept

- Definition of path:

  $U = [\chi_i, \chi_f, R, V_d]$

- Error is distance to the path:

  $\varepsilon = |R| - ||r||$

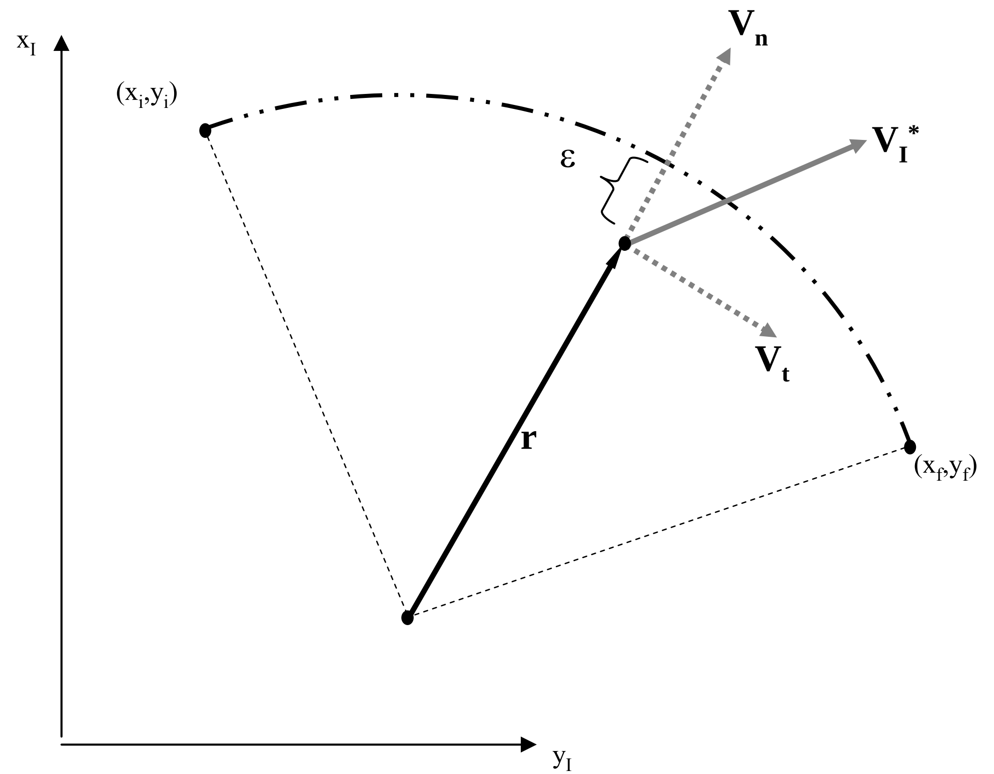


The Control Strategy

Compute separately the normal and tangential velocities:

$$\|\mathbf{V}_n\| = f(\varepsilon)$$

$$\|\mathbf{V}_t\| = V_d - \|\mathbf{V}_n\|$$

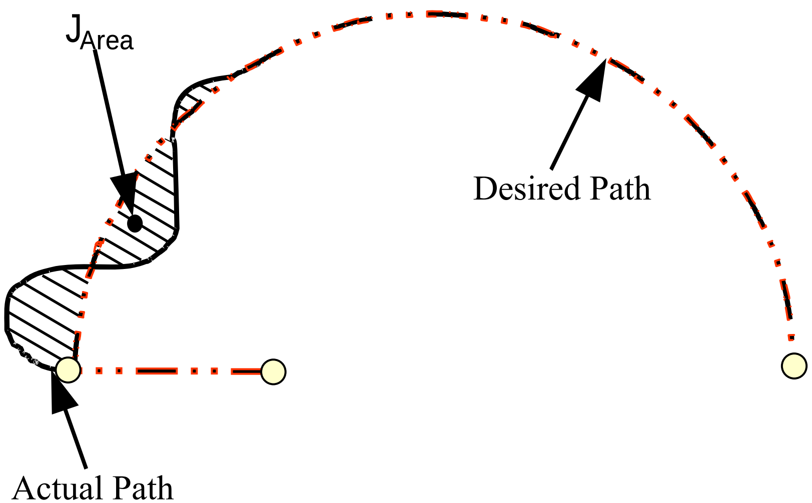


C_ε Control Laws

- Proportional control was the baseline regulator for C_ε :

$$U_r = K_p\, \varepsilon$$

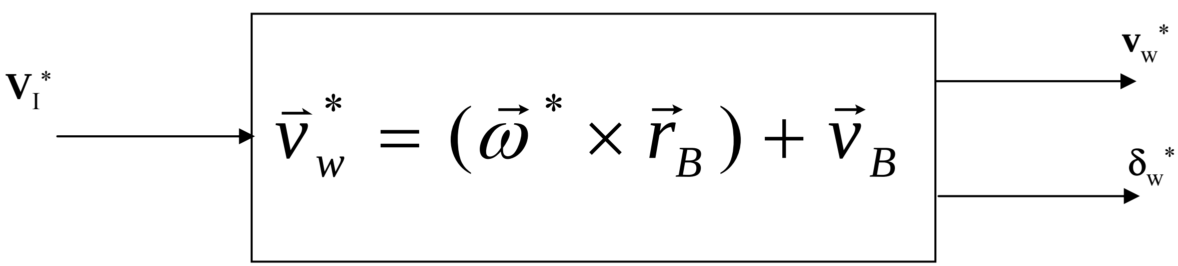
- Another interesting concept we have introduced is the idea of a **spatial** Proportional-Integral controller:



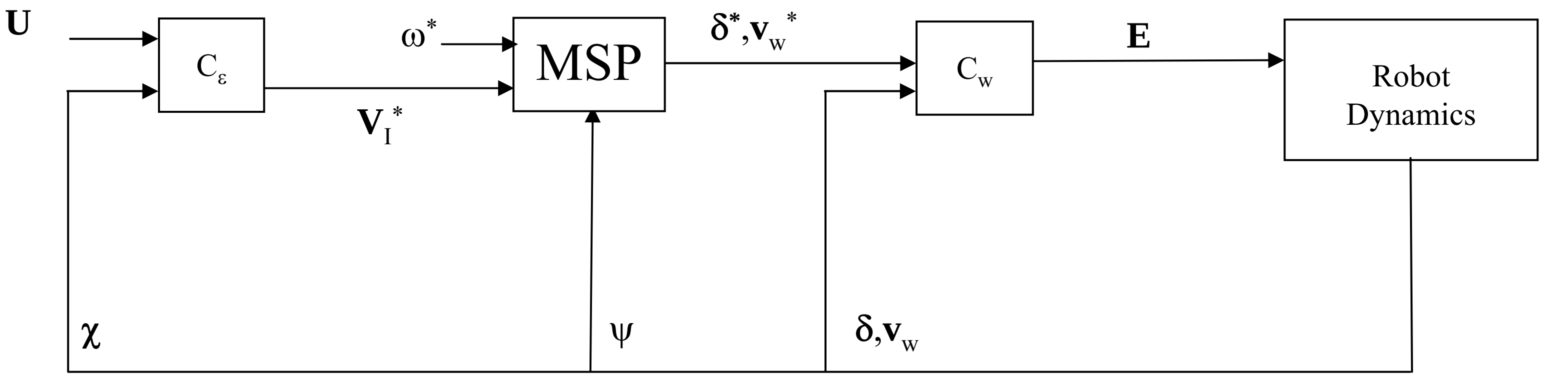
$$U_r = K_p\, \varepsilon + K_I \int \varepsilon(s)\,ds$$

After the ε-Controller: MakeSetPoints (MSP)

- The ε-controller defines desired vehicle velocities for tracking the path in inertial coordinates
- Next, these velocities must be translated into drive and steering commands
- The kinematics to do this are embodied in an algorithm we call "MakeSetPoints"

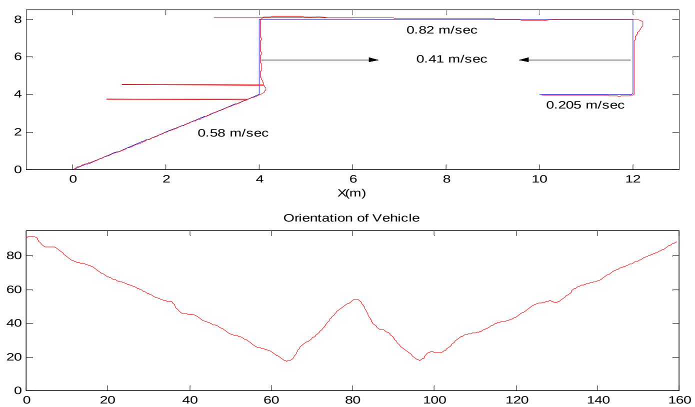


Cascade Control Architecture

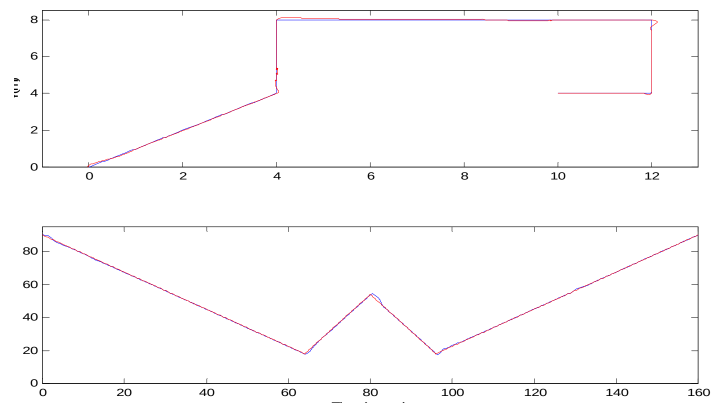


- This basic architecture has been implemented on all our robots for both:
  - Computer-control of the vehicle
  - Joystick-control of the vehicle
- The architecture has also been developed and applied for:
  - ODV steering with any number of wheels
  - Track (skid)-steer vehicles
  - Ackerman-steer vehicles

Modeling and Control (Epsilon Controller – on T1)



Experimental Results



Dynamic Model Validation

T2 Path-Tracking Control

Path Tracking

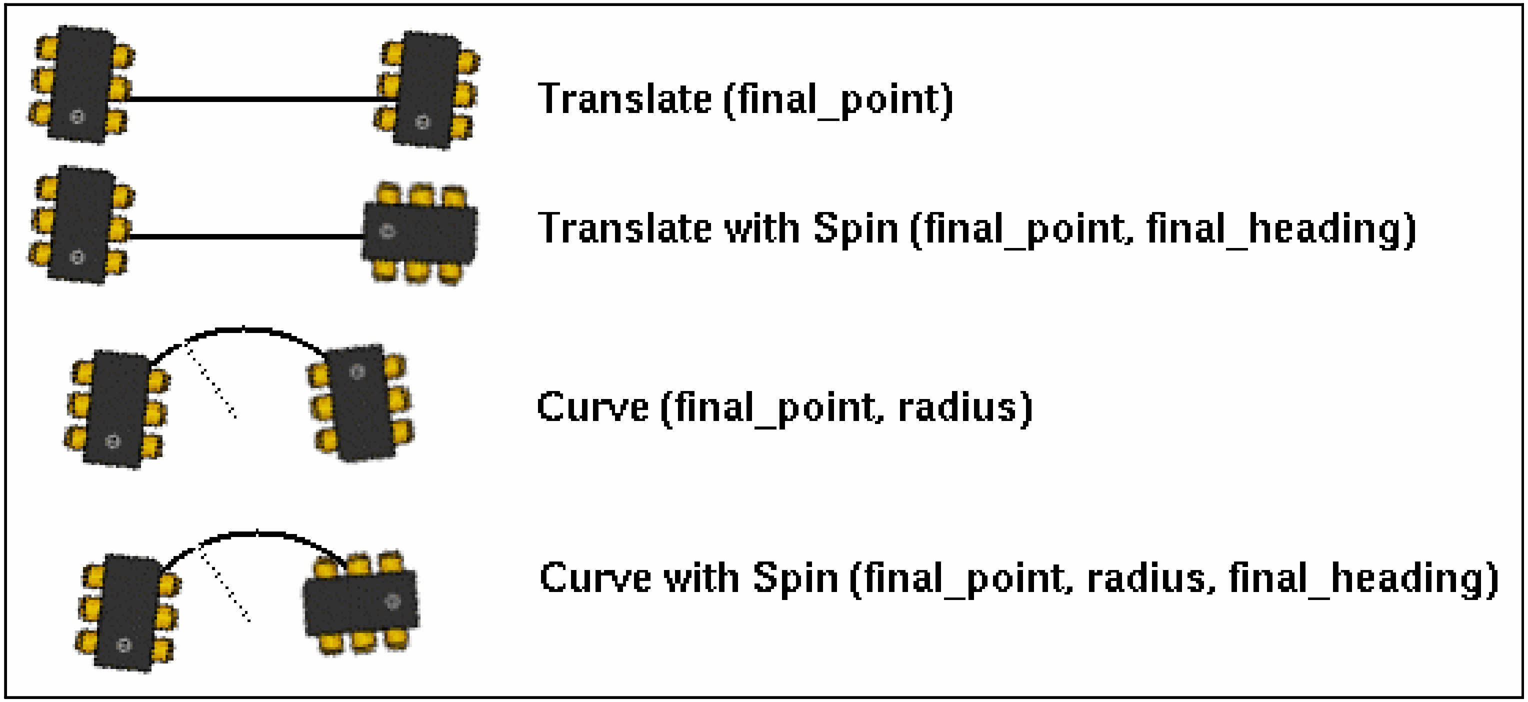
Intelligent Behavior Generation

- To enable autonomous behaviors ODIS is equipped with:
  - Vehicle mechanical design and vehicle-level control
  - Suite of environmental sensors
  - Command language based on a grammar, or set, of low-level action commands
  - Software architecture
  - Mechanisms for reactive behavior
- Approach can be used for the complete multi-robot parking security system (will mostly describe application to ODIS)

Behavior Generation Strategies

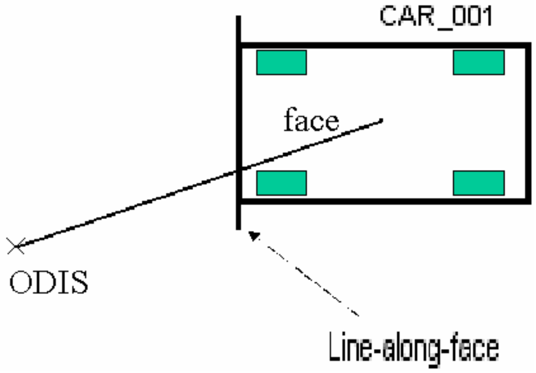
- First Generation: pre-T1
  - Waypoints fit using splines for path generation
  - **User-based path generation**
- Second Generation: T1, T2
  - decomposition of path into primitives
  - fixed input parameters
  - **open-loop path generation**
- Third Generation: T2, T3, ODIS
  - decomposition of paths into primitives
  - variable input parameters that depend on sensor data
  - **sensor-driven path generation**
- Fourth Generation: ODIS, T2e, T4
  - Deliberative behavior via exception control
  - reactive behavior via interacting threads (agents)
  - **closed-loop path generation** (goal)

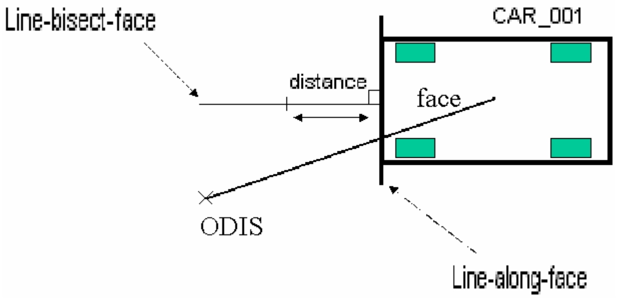
2nd Generation Maneuver Grammar: z-commands

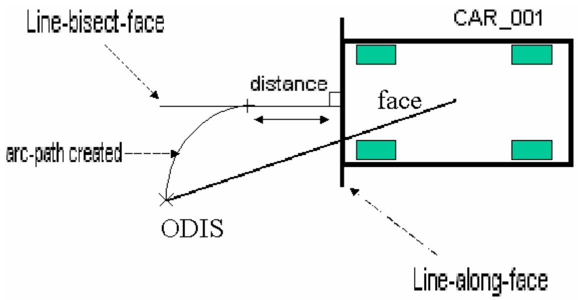


3rd Generation Maneuver Command: Sensor-Driven, Delayed Commitment Strategy

(ALIGN-ALONG (LINE-BISECT-FACE CAR_001) distance)







ODIS Command Environment - 1

- Developed to implement our delayed commitment approach
- Called MoRSE (Mobile Robots in Structured Environments)
- Has a high degree of orthogonality:
  - a number of small orthogonal constructs
  - mixed and matched to provide almost any behavior
  - effectively spans the action space of the robot
- Initial implementation was an actual compiled language that we wrote to use a familiar imperative programming style, with looping constructs, conditional execution, and interpretive operation
- Later changed to a set of C libraries

ODIS Command Environment - 2

- *Variables* include standard integer and floating point data types, as well as specialized geometric data types, such as:
  – Points, lines, arcs, corners, pointsets
  – Data constructs for objects in the environment, which can be fit and matched to data
- *Geometric computation* functions:
  - Functions for building arcs and lines from points
  - Functions for returning points on objects
  - Functions for extracting geometry from environment objects
  - Functions to generate unit vectors based on geometry
  - Fitting functions to turn raw data into complex objects
  - Vector math

ODIS Command Environment - 3

- A key feature of MoRSE is the command unit:
  - Set of individual commands defining various vehicle actions that will be executed in parallel
- *Commands for XY movement*:
  - moveAlongLine(Line path, Float vmax, Float vtrans = 0)
  - moveAlongArc(Arc path, Float vmax, Float vtrans = 0)
- *Commands for Yaw movement:*
  - yawToAngle(Float angle_I, Float rate = max)
  - yawThroughAngle(Float delta, Float rate = max)
- *Commands for sensing:*
  - SenseSonar
  - SenseLaser
  - SenseIR
  - Camera commands
- A set of rules defines how these commands may be combined

Rules for Combining Commands to Form a Command-Unit

- At most one command for XY movement
- At most one command for yaw movement
- Only one Rapid-stop command
- At most 1 of each sense command (laser, sonar, IR)
- At most 1 command for camera action
- No XY, yaw movement, and senseLaser commands allowed with Rapid-stop command
- No yaw movement command when a senseLaser command is used

Example Macroscript - 1

`findCar()` script

- If there is a car, find bumper and move closer.
- Fit the open-left tire.
- Fit the open-right tire.
- Move up the centerline of car.
- Fit the closed-left tire.
- Fit the closed-right tire.
- Fit the entire car and prepare for inspection.

Example Macroscript - 2

The detailed structure of the first two steps is as follows:

If (car) fit bumper and move in

fire sonar at rear of stall

if there is something in the stall

fire sonar at front half of stall

fit bumper_line

move to ∩ of bumper_line with c.l. of stall

fit tire_ol

coarse scan of ol and or_quadrants

move to the line connecting two data centroids

arc and detail scan around the ol data centroid

fit tire_ol with the resulting data

else go to next stall

Example Macroscript - 3
Actual Code

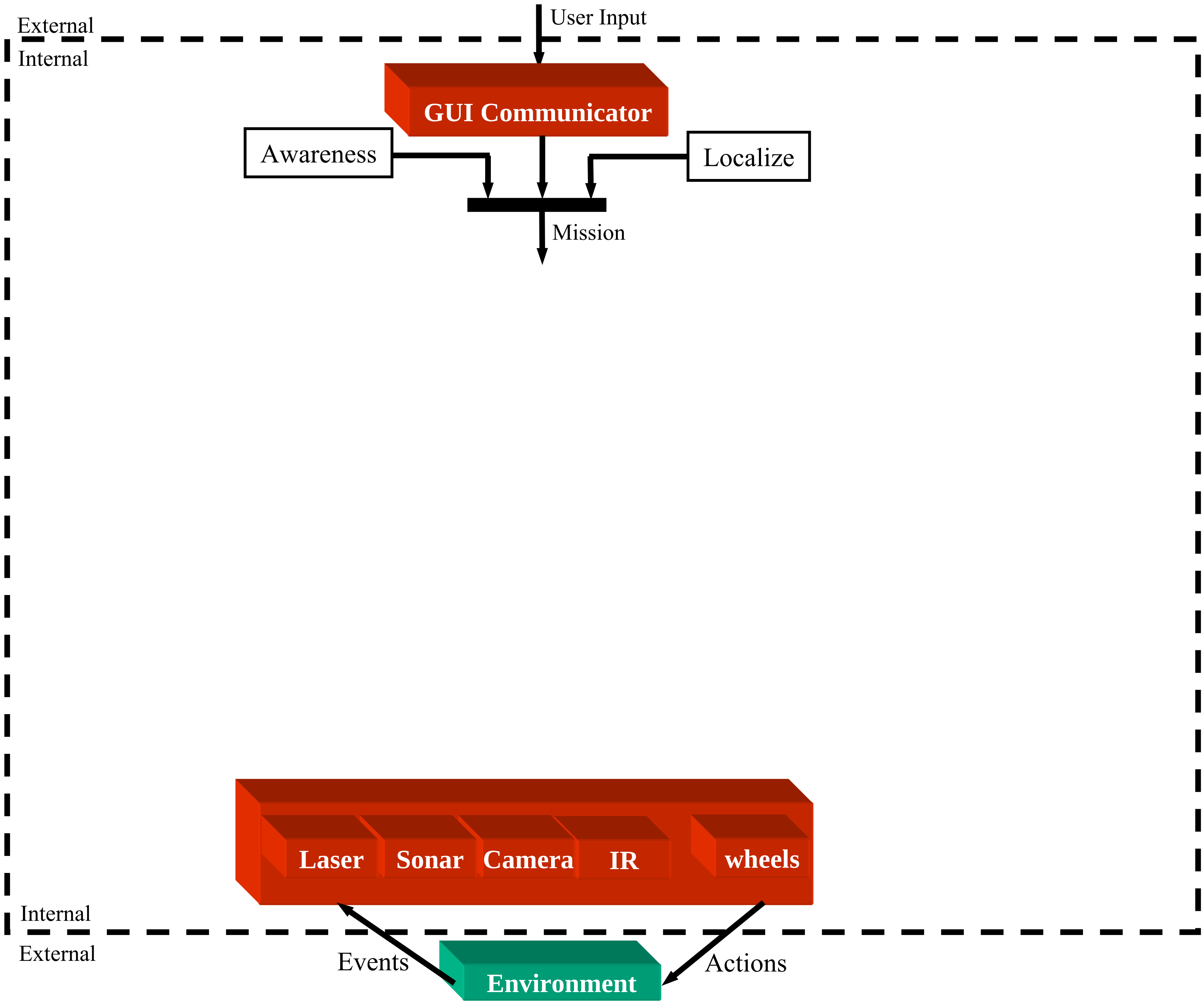
```
If (car) fit bumper and move in
sense_sonar_duration = 1.0;
sense_radius = max_sonar_range;
<<<
// fires sonar to see if there is a car in the stall.
senseSonar( my_stall.p_cl, my_stall.p_cr,
            sonar_cutoff_radius, sense_sonar_duration );
>>>
sonar_data = getSonarData();
// If there is a car.
if ( sonar_data.size > 5 &&
      pointIsInsideStall ( sonar_data.mean(),my_stall ))
{
    Line  stall_centerline;
    Line  line_to_bumper;
    Line  bumper_line;
    Vector  stall_x_axis;  // Unit vector pointing toward
                             //the face_c of stall.
    Vector  stall_y_axis;  // Unit vector 90 degrees from
                             //stall_x_axis.
    Point  stall_cline_and_bumper_inter;
    sense_sonar_duration = 4.0;
    sonar_cutoff_radius = dist_from_stall +
                      my_stall.face_r.length() * 0.5;
    if( fitLineSegLMS( sonar_data, bumper_line ) <=
                      minimumConfidence )
    {
     // Fit is not good.
     ?return 0;
    }
    stall_centerline=makeLine(my_stall.face_o.midPoint()
                                  ,my_stall.center() );
    stall_x_axis = stall_centerline.unitvec();
    stall_y_axis = rotateVec( stall_x_axis, 90 );
    stall_cline_and_bumper_inter = LineIntersection(
                         bumper_line,stall_centerline );
    line_to_bumper = makeLine( entry_point,
                      stallcl_and_bumper_int );
    <<<
    // moves in to the intersection of the bumper line with
    // the stall centerline.
    moveAlongLine( line_to_bumper, max_velocity );
    >>>
…
}
```

Example: ODIS FindCar() Script

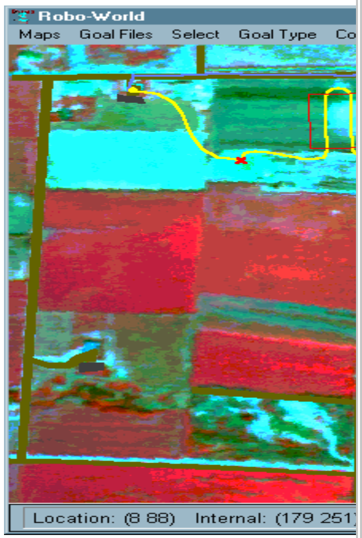
Software Architecture

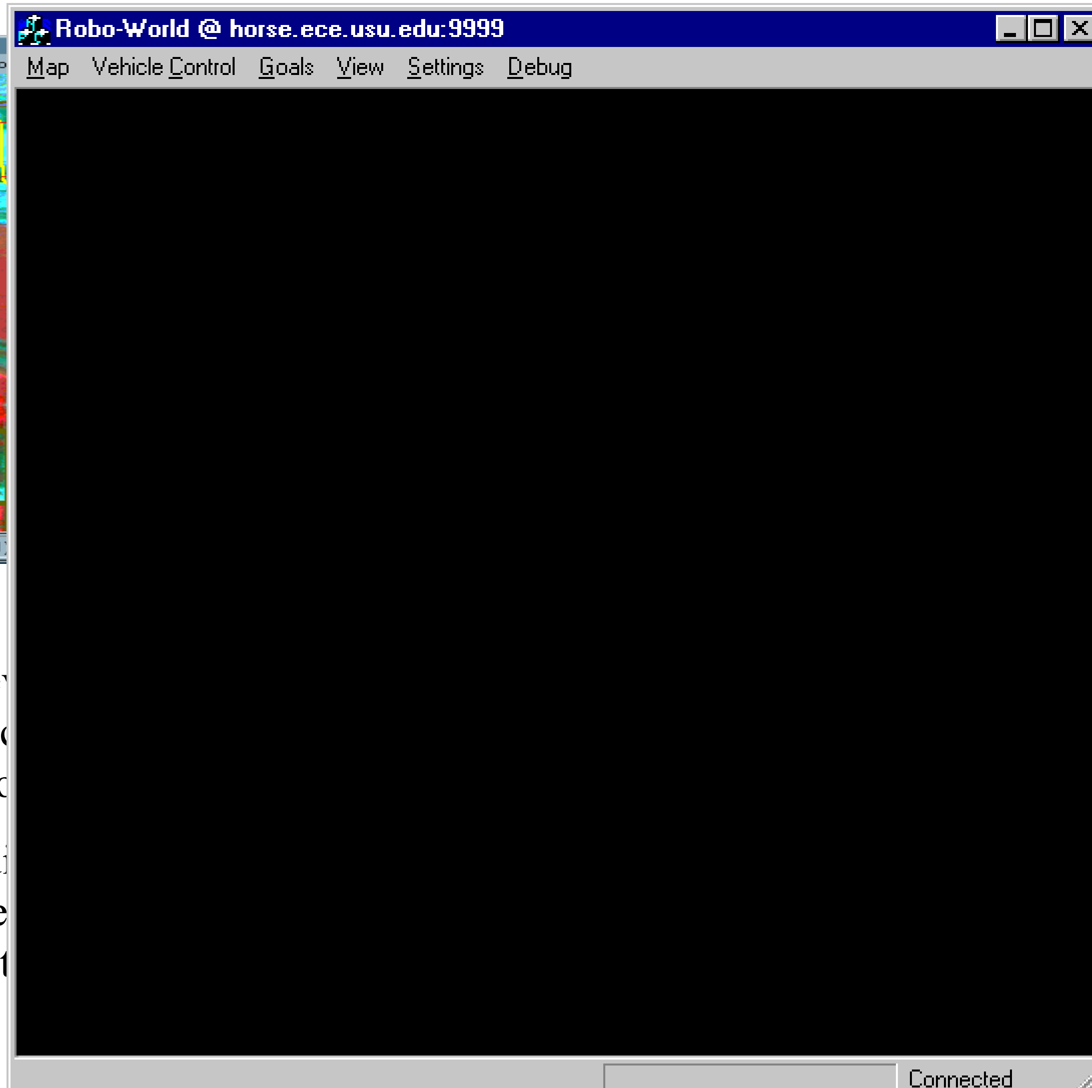
- Command actions are the lowest-level tasks allowed in our architecture that can be commanded to run in parallel
- For planning and intelligent behavior generation, higher-level tasks are defined as compositions of lower-level tasks
- In our hierarchy we define:

Mission	User-defined
Tasks Subtasks	Variable (planned)
Atomic Tasks (Scripts) Command Units Command Actions	Hard-wired (but, (parameterized and sensor-driven)


User Input
External
Internal
GUI Communicator
Awareness
Localize
Mission
Laser
Sonar
Camera
IR
wheels
Internal
External
Events
Actions
Environment

Farming Automation Projects





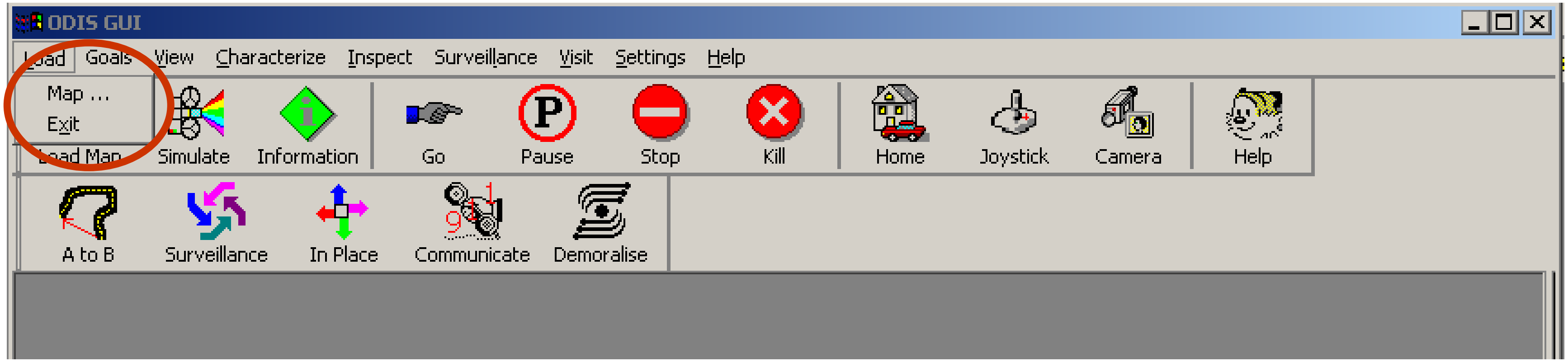
Co-operative
ng

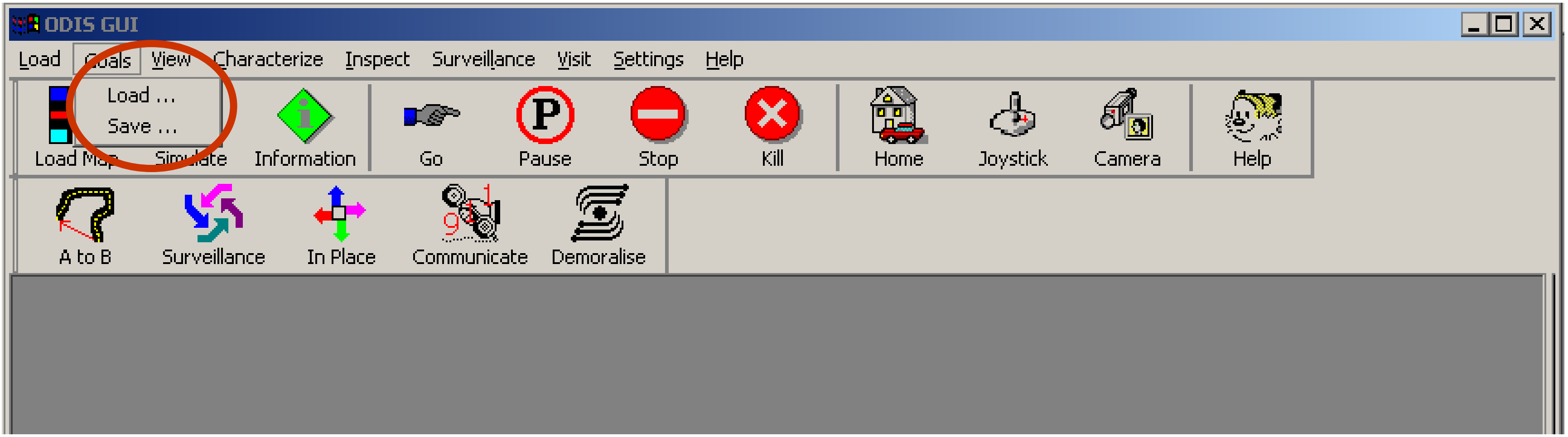
llite map of the
sks are
lligent path and

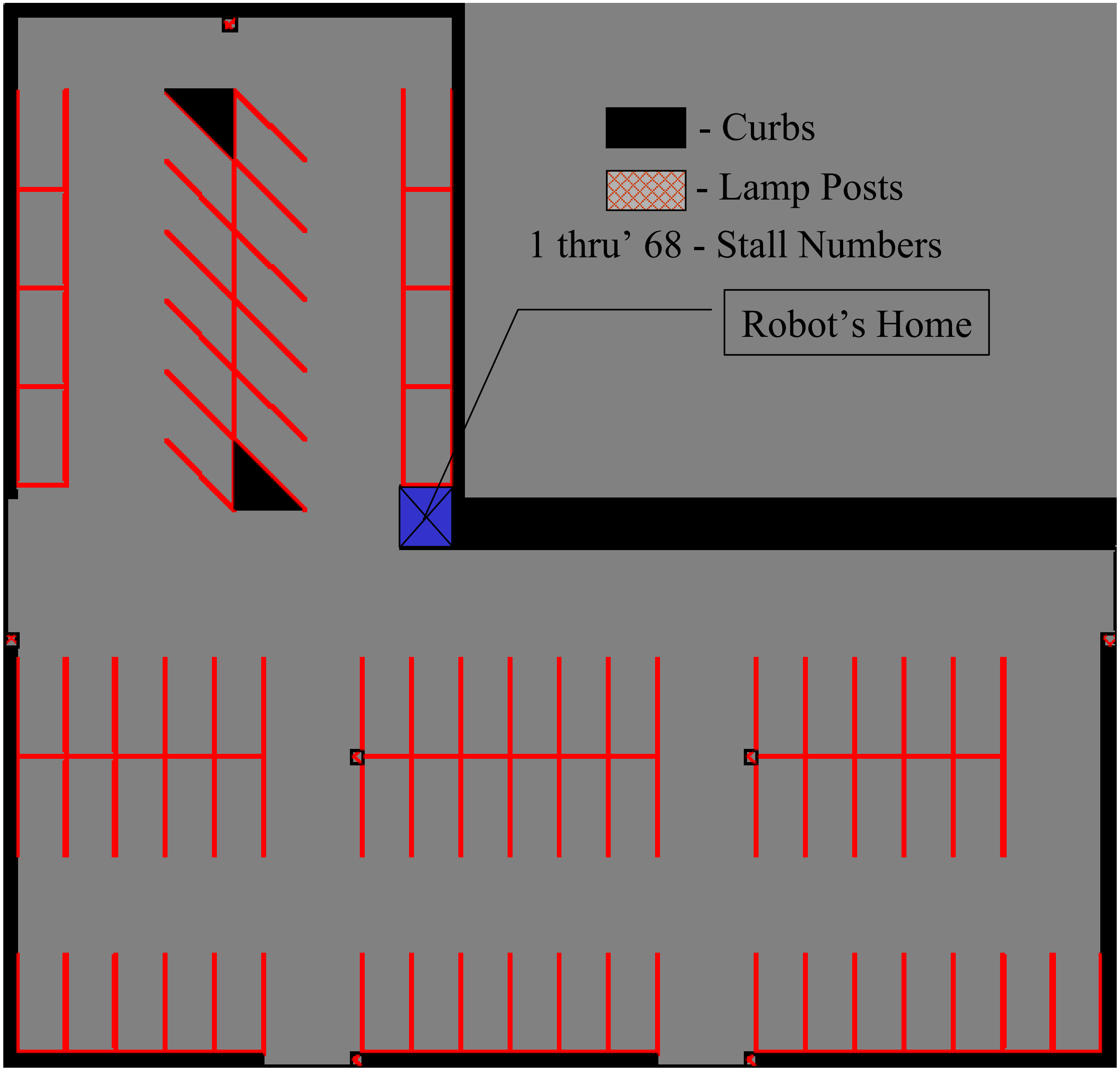
expected
ures by re-
n and path

- Technology de
autonomously
dGPS navigatio
- Prototypes equi
equipment, che
radiation detect

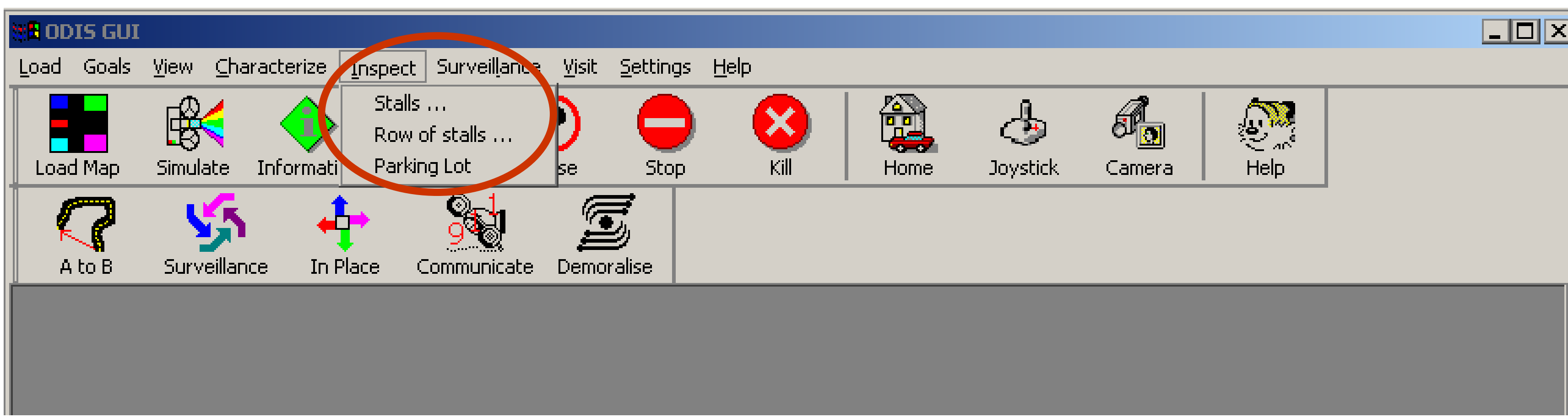
Commanding the Robot

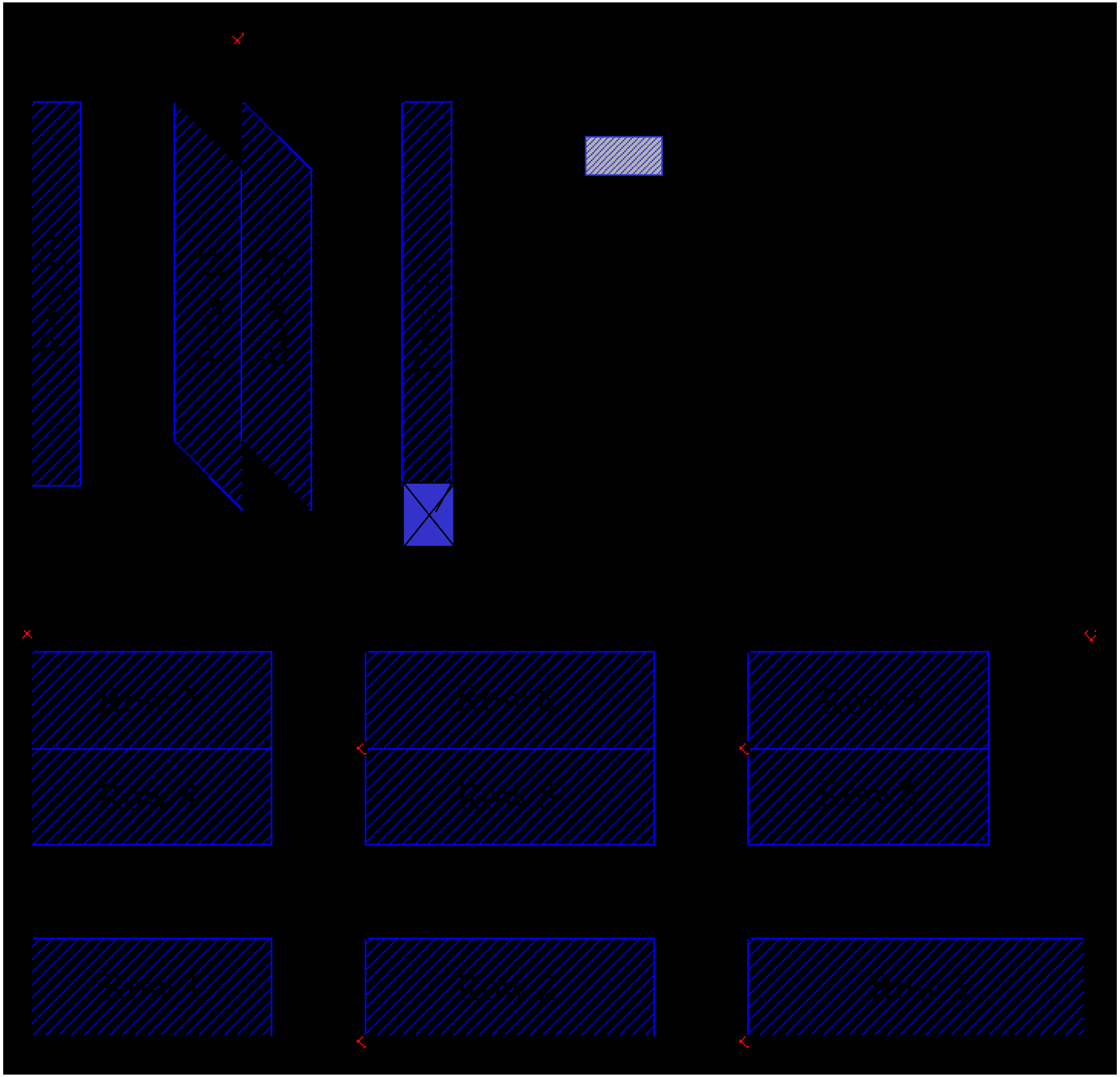





- Curbs
- Lamp Posts
1 thru’ 68 - Stall Numbers
Robot’s Home

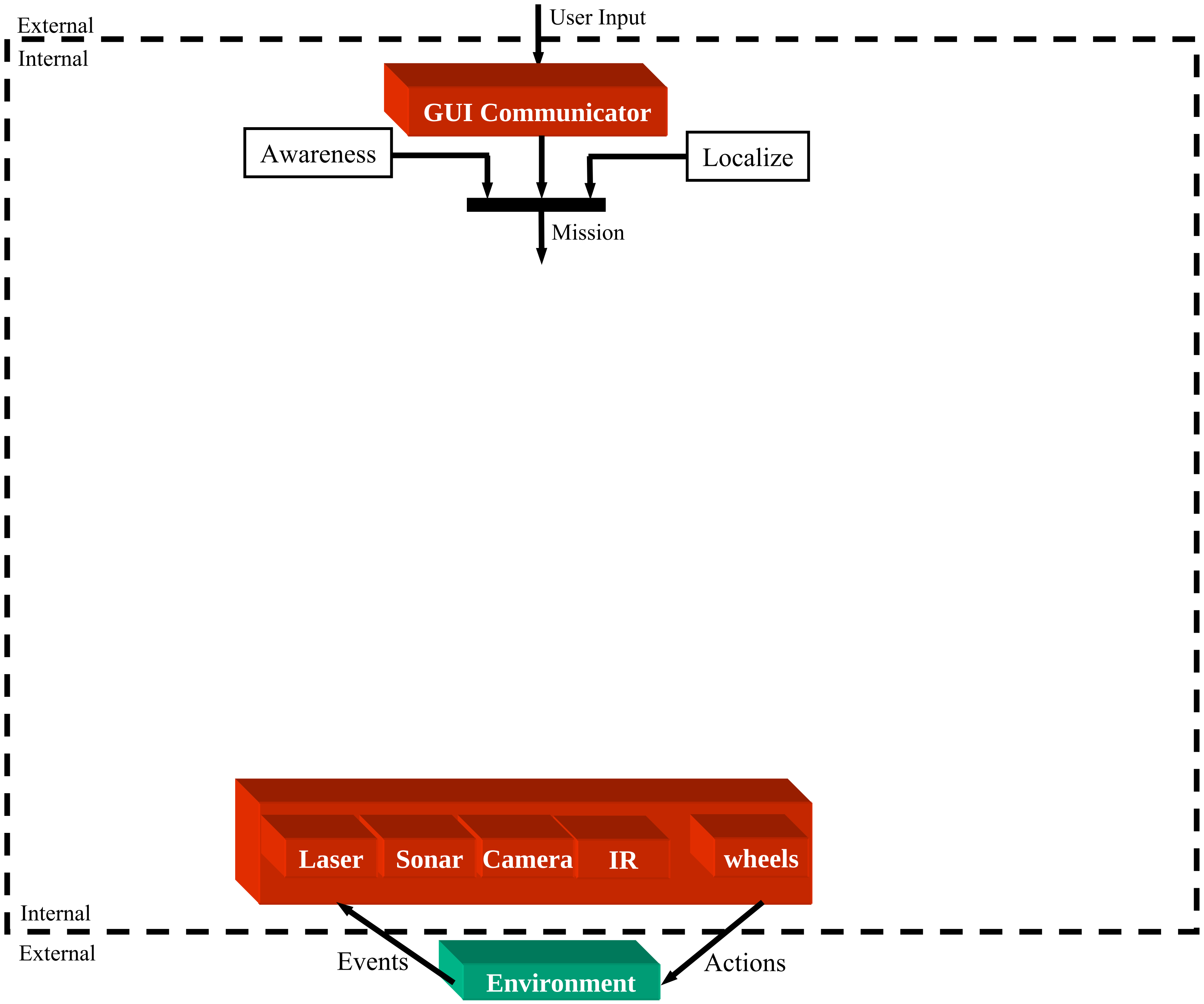
Issuing a mission for the robot

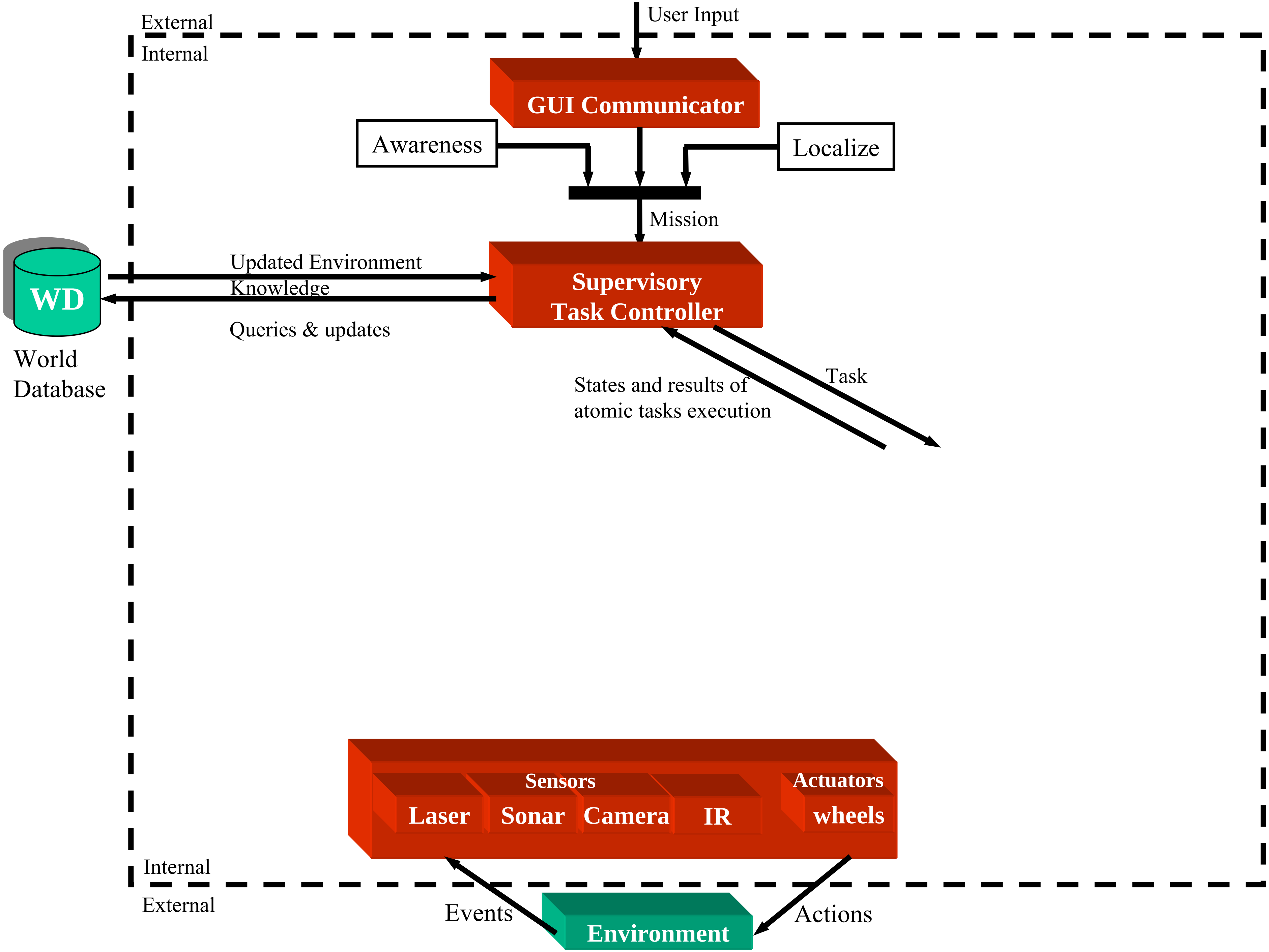




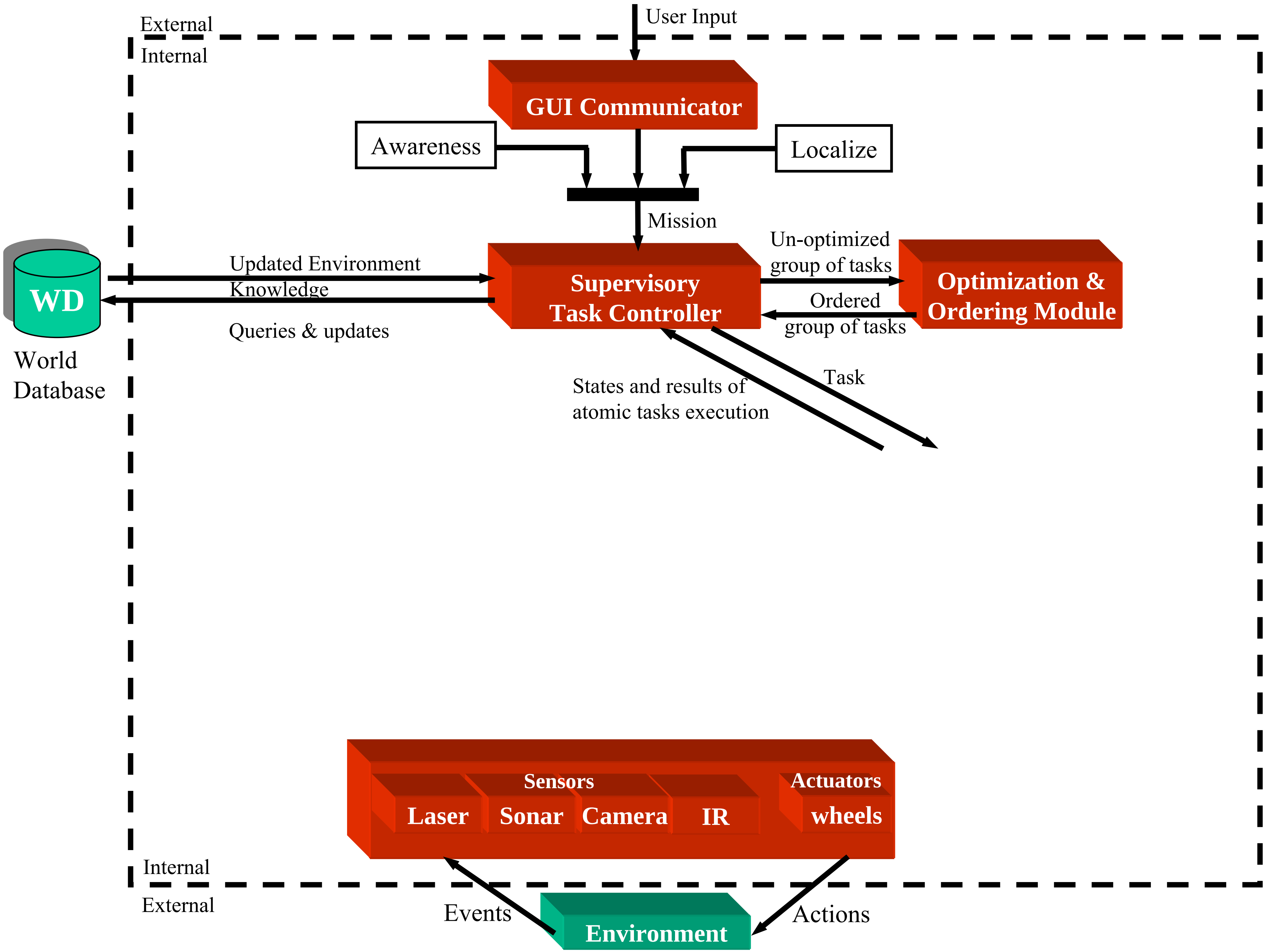
User-tasks in the environment

- {MoveTo Point}
- {Characterize a stall}
- {Inspect a stall}
- {Characterize a row of stalls}
- {Inspect a row of stalls}
- {Localize}
- {Find my Car}
- {Sweep the parking lot}
- {Sweep Specific area of the parking lot}

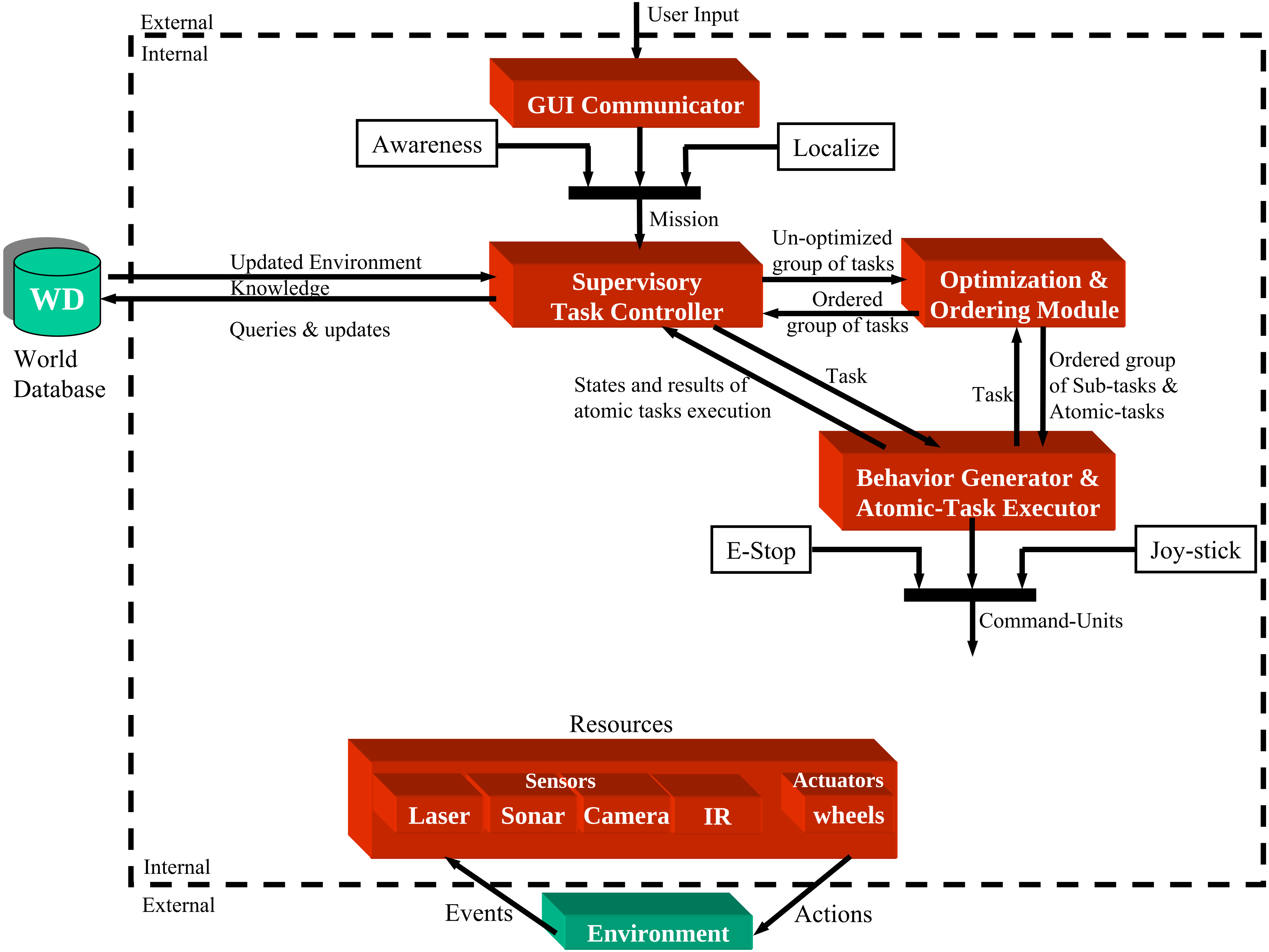

External
Internal
User Input
GUI Communicator
Awareness
Localize
Mission
Laser
Sonar
Camera
IR
wheels
Internal
External
Events
Environment
Actions



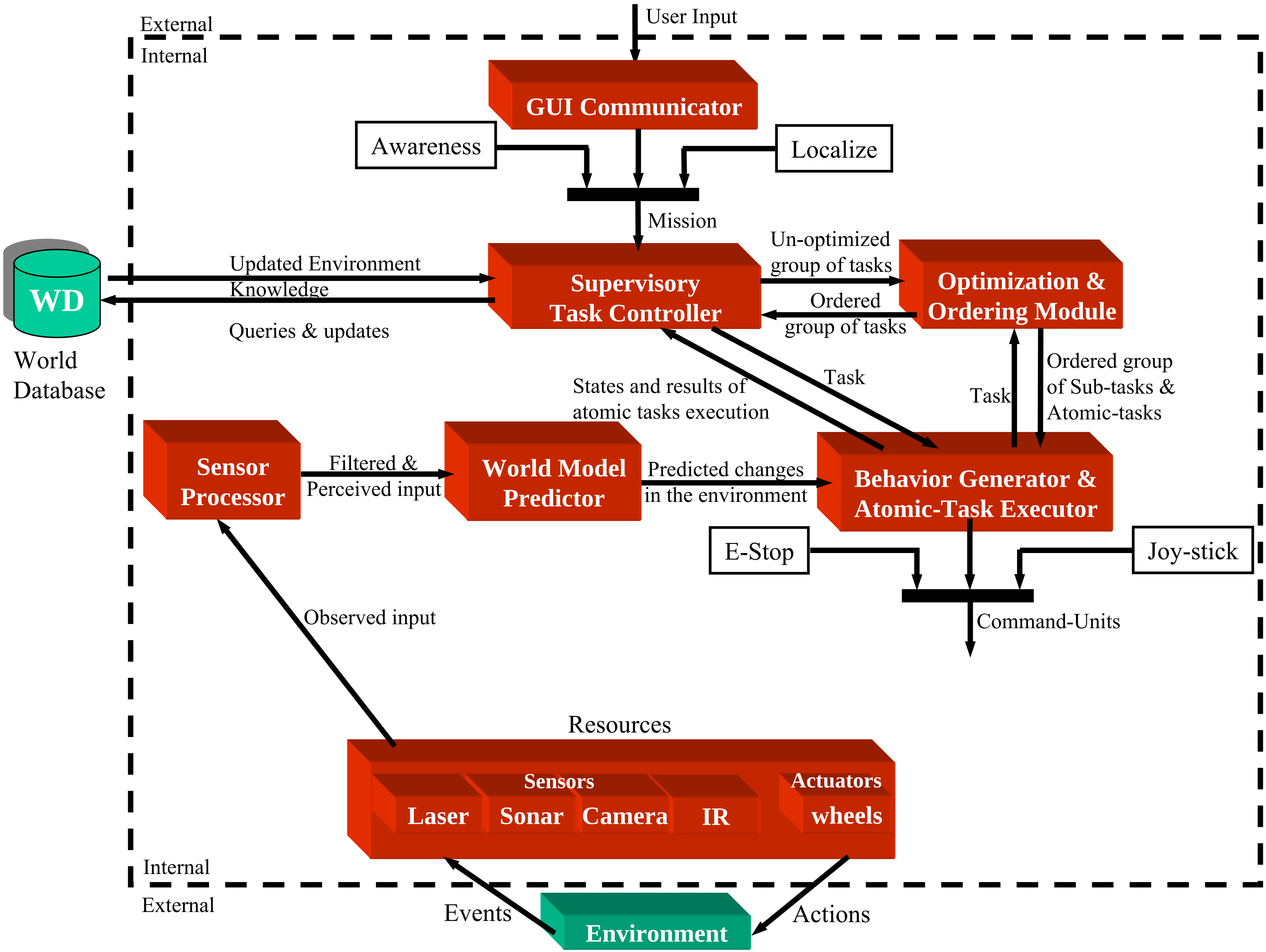
External
Internal
User Input
GUI Communicator
Awareness
Localize
Mission
Updated Environment
Knowledge
Queries & updates
WD
World
Database
Supervisory
Task Controller
Task
States and results of
atomic tasks execution
Sensors
Actuators
Laser
Sonar
Camera
IR
wheels
Internal
External
Events
Environment
Actions



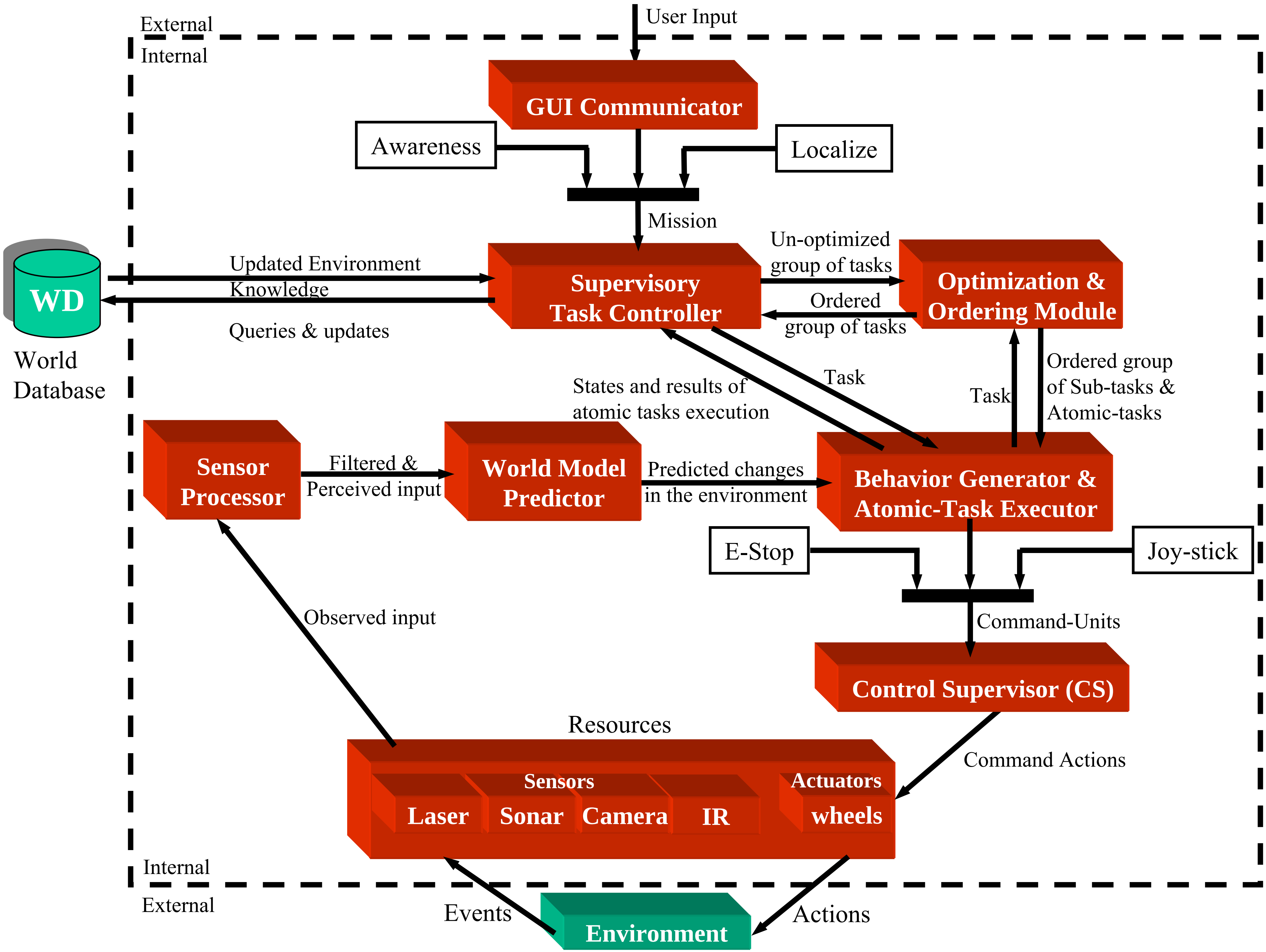
User Input
External
Internal
GUI Communicator
Awareness
Localize
Mission
WD
World
Database
Updated Environment
Knowledge
Queries & updates
Supervisory
Task Controller
Un-optimized
group of tasks
Optimization &
Ordering Module
Ordered
group of tasks
States and results of
atomic tasks execution
Task
Sensors
Laser
Sonar
Camera
IR
Actuators
wheels
Internal
External
Events
Environment
Actions



User Input
External
Internal
GUI Communicator
Awareness
Localize
Mission
WD
World Database
Updated Environment Knowledge
Queries & updates
Supervisory Task Controller
Un-optimized group of tasks
Ordered group of tasks
Optimization & Ordering Module
States and results of atomic tasks execution
Task
Task
Ordered group of Sub-tasks & Atomic-tasks
Behavior Generator & Atomic-Task Executor
E-Stop
Joy-stick
Command-Units
Resources
Sensors
Laser
Sonar
Camera
IR
Actuators
wheels
Internal
External
Events
Environment
Actions

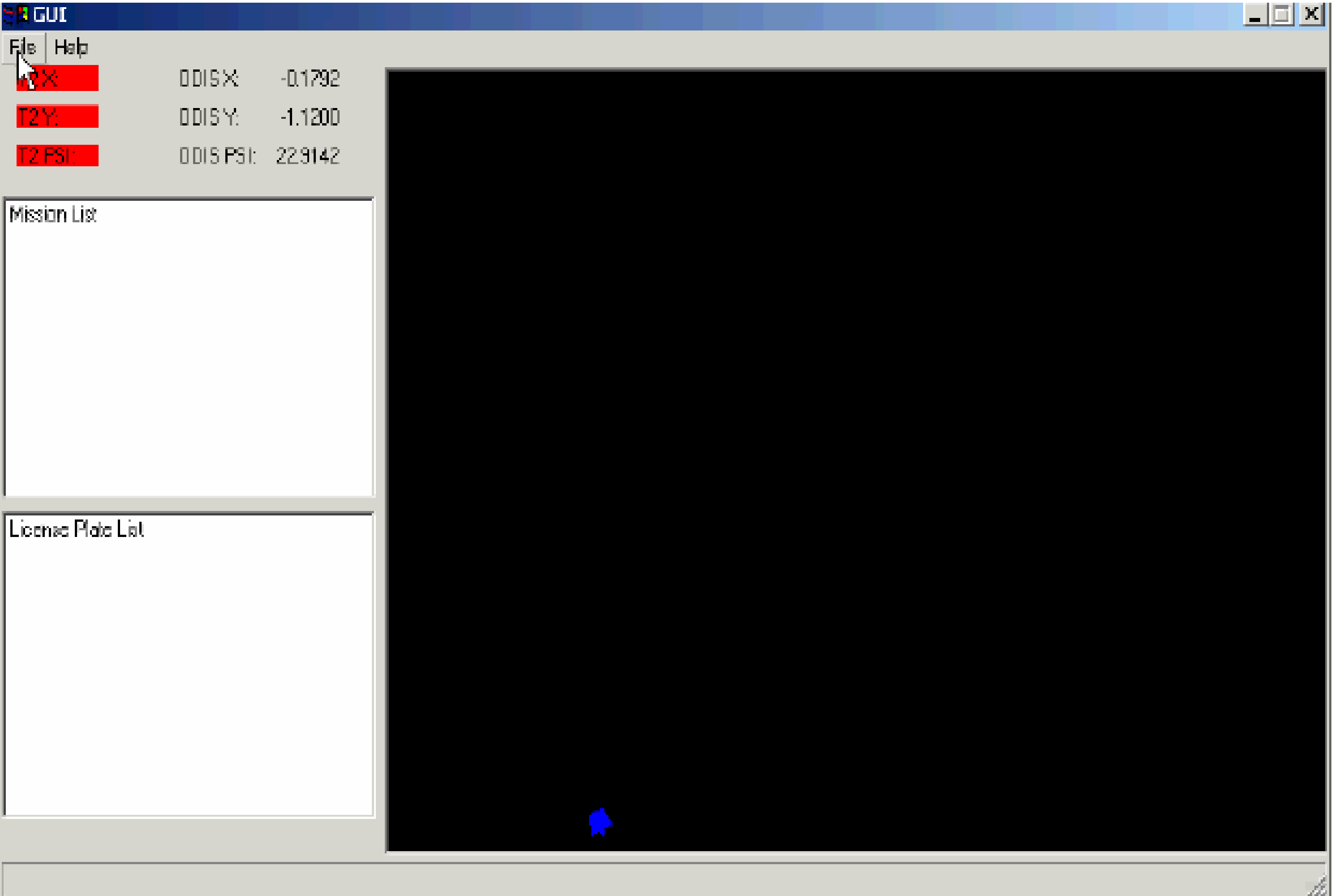


User Input
External
Internal
GUI Communicator
Awareness
Localize
Mission
WD
World Database
Updated Environment Knowledge
Queries & updates
Supervisory Task Controller
Un-optimized group of tasks
Ordered group of tasks
Optimization & Ordering Module
States and results of atomic tasks execution
Task
Task
Ordered group of Sub-tasks & Atomic-tasks
Sensor Processor
Filtered & Perceived input
World Model Predictor
Predicted changes in the environment
Behavior Generator & Atomic-Task Executor
E-Stop
Joy-stick
Command-Units
Observed input
Resources
Sensors
Laser
Sonar
Camera
IR
Actuators
wheels
Internal
External
Events
Environment
Actions



External
Internal
User Input
GUI Communicator
Awareness
Localize
Mission
Updated Environment
Knowledge
Queries & updates
WD
World
Database
Supervisory
Task Controller
Un-optimized
group of tasks
Optimization &
Ordering Module
Ordered
group of tasks
Task
Task
Ordered group
of Sub-tasks &
Atomic-tasks
States and results of
atomic tasks execution
Sensor
Processor
Filtered &
Perceived input
World Model
Predictor
Predicted changes
in the environment
Behavior Generator &
Atomic-Task Executor
E-Stop
Joy-stick
Command-Units
Control Supervisor (CS)
Observed input
Command Actions
Resources
Sensors
Actuators
Laser
Sonar
Camera
IR
wheels
Internal
External
Events
Environment
Actions

Intelligent Behavior Generation (Cross-Platform/Multi-Platform T2e/T4/ODIS-I, ODIS-S)



Demonstration Example

Reactive Behaviors

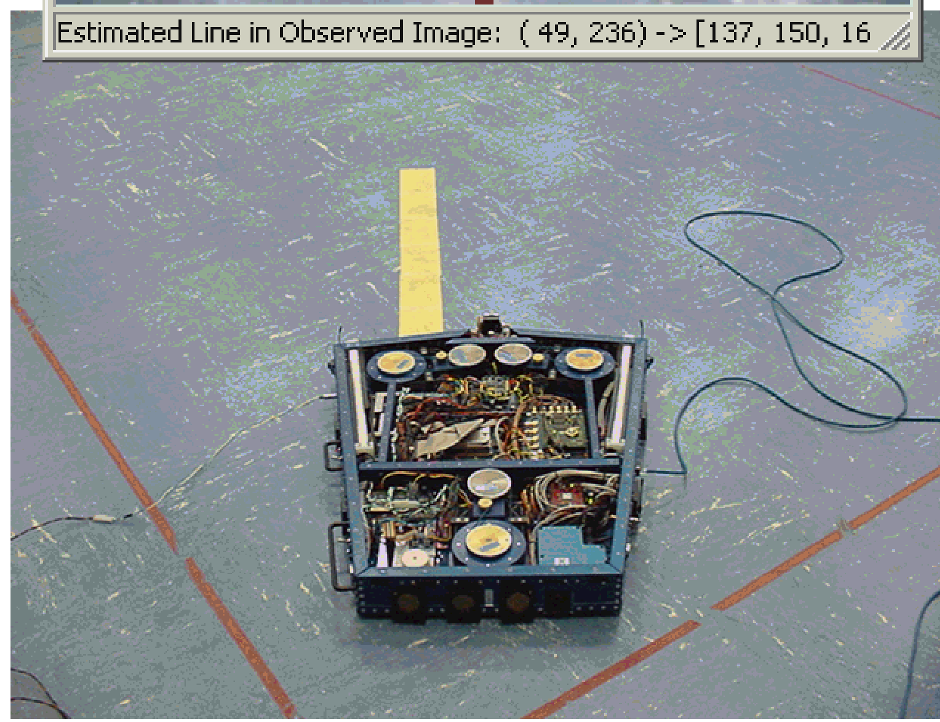
Reactive behaviors are induced via:

1. Localization thread
   – Compares expected positions to actual sensors' data and makes correction to GPS and odometry as needed

Localization to Yellow Lines

• Periodically the fiber-optic gyro is reset:

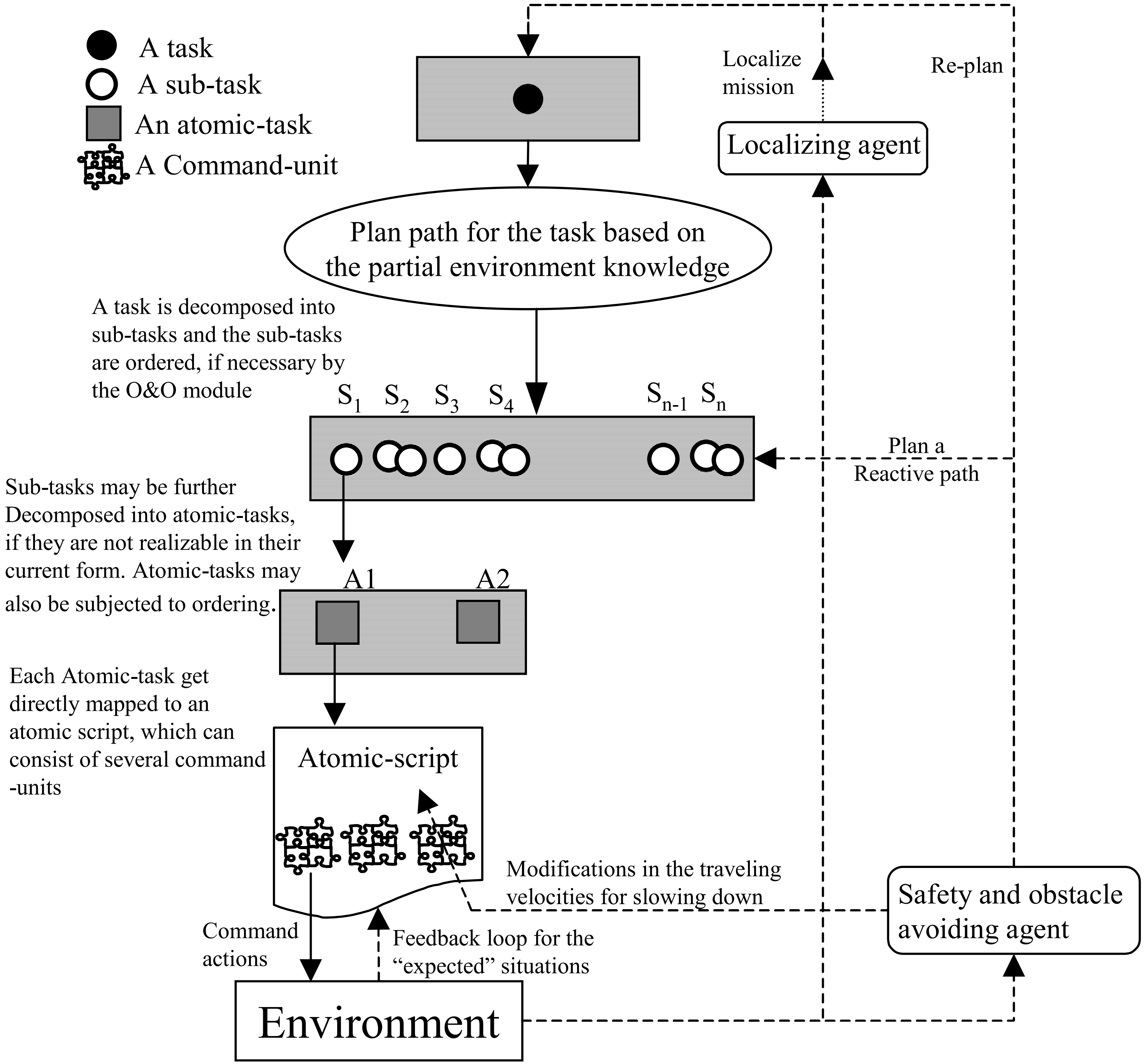
- Yellow line is identified in camera image
- Vehicle is rotated to align its body-centered axis with identified line
- Process repeats iteratively



Reactive Behaviors

Reactive behaviors are induced via:

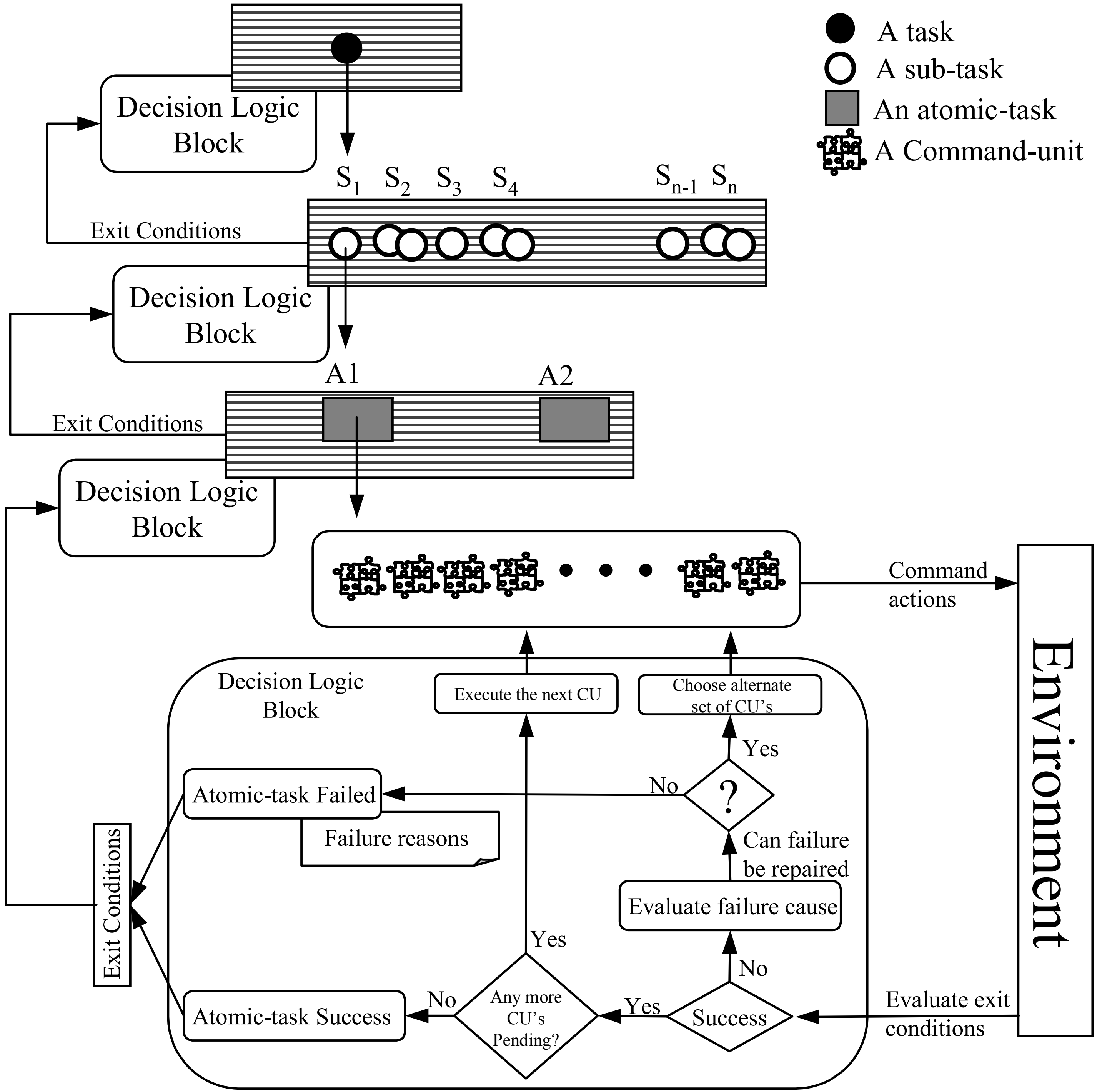
1. Localization thread
   - Compares expected positions to actual sensors data and makes correction to GPS and odometry as needed
2. Awareness thread
   - Interacts with the execution thread based on safety assessments of the environment


A task
A sub-task
An atomic-task
A Command-unit
Localize mission
Re-plan
Localizing agent
Plan path for the task based on the partial environment knowledge
A task is decomposed into sub-tasks and the sub-tasks are ordered, if necessary by the O&O module
S_1 S_2 S_3 S_4 S_{n-1} S_n
Plan a Reactive path
Sub-tasks may be further Decomposed into atomic-tasks, if they are not realizable in their current form. Atomic-tasks may also be subjected to ordering.
A1
A2
Each Atomic-task get directly mapped to an atomic script, which can consist of several command -units
Atomic-script
Modifications in the traveling velocities for slowing down
Safety and obstacle avoiding agent
Command actions
Feedback loop for the "expected" situations
Environment

Reactive Behaviors

Reactive behaviors are induced via:

1. Localization thread
   - Compares expected positions to actual sensors data and makes correction to GPS and odometry as needed
2. Awareness thread
   - Interacts with the execution thread based on safety assessments of the environment
3. Logic within the execution thread
   - Exit conditions at each level of the hierarchy determine branching to pre-defined actions or to re-plan events


A task
A sub-task
An atomic-task
A Command-unit
Decision Logic Block
S_1 S_2 S_3 S_4 S_{n-1} S_n
Exit Conditions
Decision Logic Block
A1
A2
Exit Conditions
Decision Logic Block
Command actions
Environment
Decision Logic Block
Execute the next CU
Choose alternate set of CU's
Yes
No
?
Atomic-task Failed
Failure reasons
Can failure be repaired
Evaluate failure cause
Exit Conditions
Yes
No
Evaluate exit conditions
Atomic-task Success
No
Any more CU's Pending?
Yes
Success

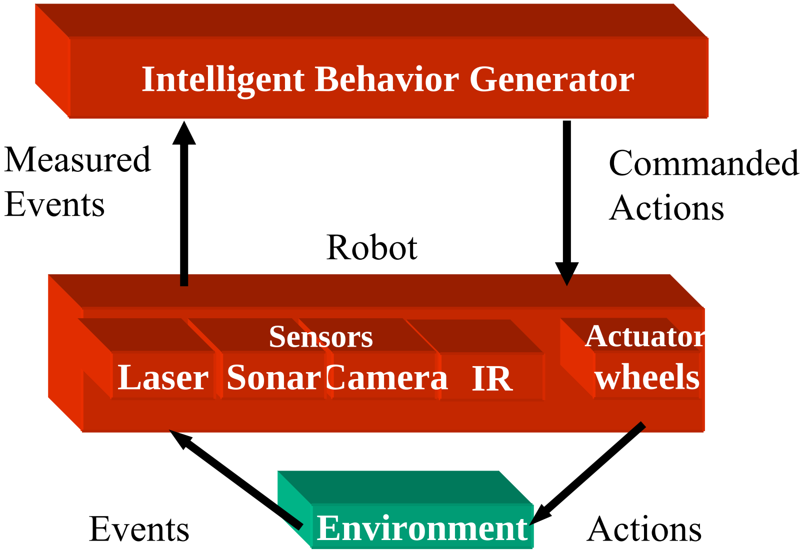
T2 Adaptive/Reactive Hill-Climbing

Conclusion

- A variety of ODV robots have been presented
- System architecture for enabling intelligent behaviors has been presented
- The architecture is characterized by:
  - A sensor-driven, parameterized low-level action command grammar
  - Multi-level planning and task decomposition
  - Multi-level feedback and decision-making
- Architecture enables adaptive, reactive behaviors
- Longer-range goal is to incorporate automated script generation via discrete event dynamic systems theory

DEDS Approach

- The mobile robot behavior generator can be interpreted as a discrete-event dynamic system (DEDS)

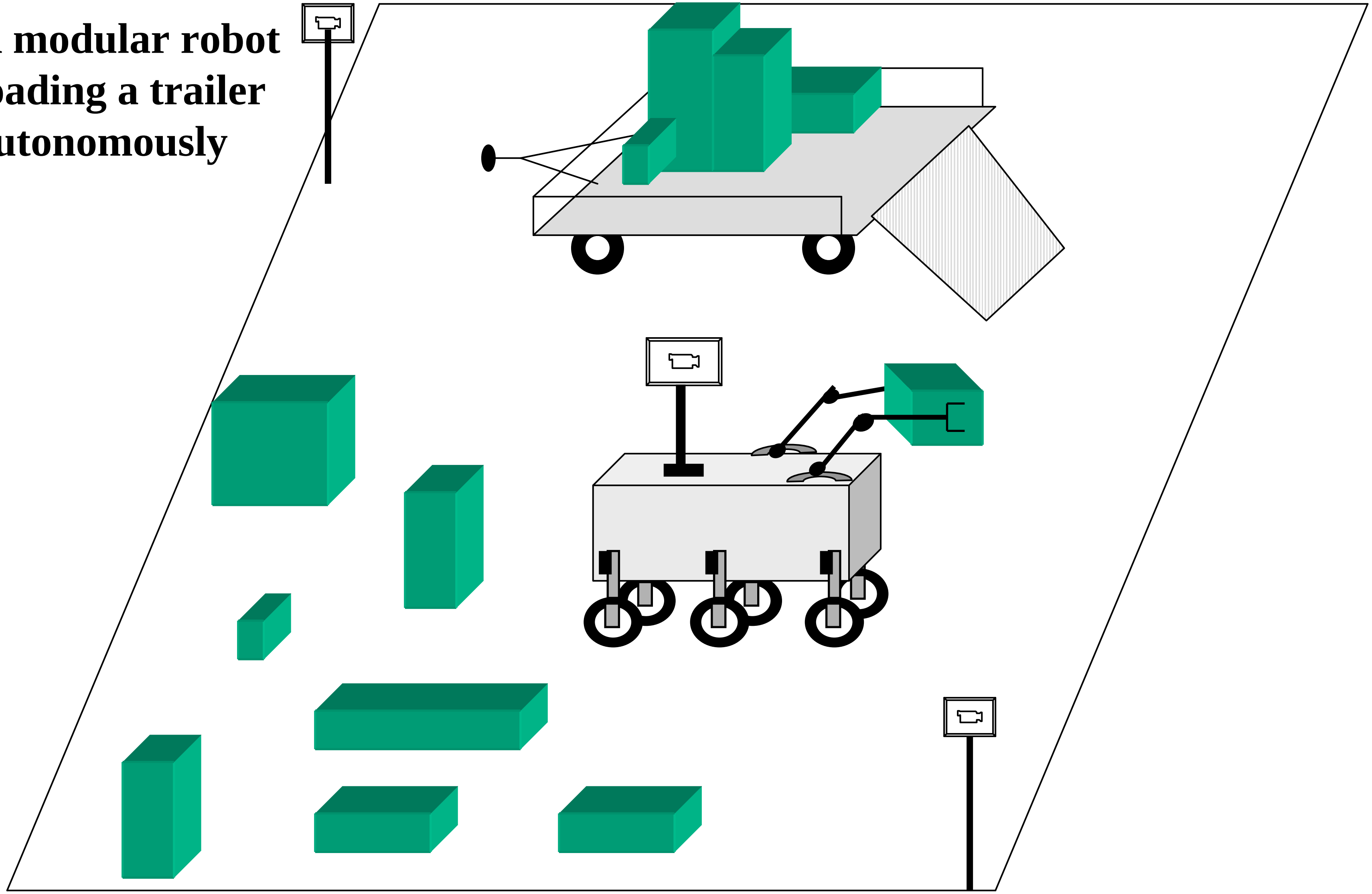


- In this interpretation commands and events are symbols in an alphabet associated with a (regular) language
- This formalism can be used for synthesis of scripts
- Other suggested approaches for synthesis include Petri nets and recent results on controller design for finite state machine model matching

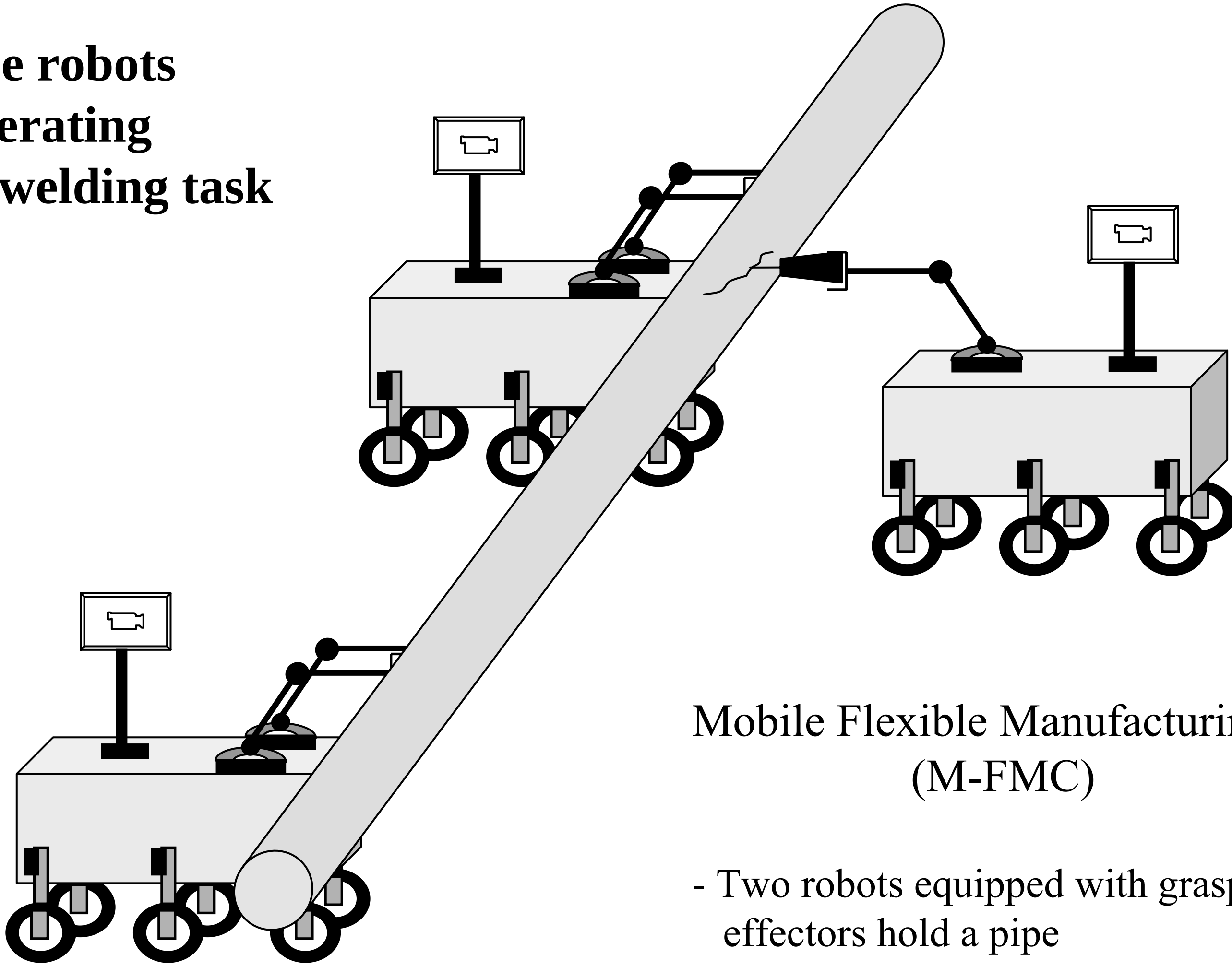
Conclusion

- A variety of ODV robots have been presented
- System architecture for enabling intelligent behaviors has been presented
- The architecture is characterized by:
  - A sensor-driven, parameterized low-level action command grammar
  - Multi-level planning and task decomposition
  - Multi-level feedback and decision-making
- Architecture enables adaptive, reactive behaviors
- Longer-range goal is to incorporate automated script generation via discrete event dynamic systems theory
- Future applications are planned

A modular robot loading a trailer autonomously



Three robots cooperating on a welding task



Mobile Flexible Manufacturing Cell (M-FMC)

- Two robots equipped with grasping end effectors hold a pipe
- Third robot equipped with welding end effector lays a bead

Conclusion

- A variety of ODV robots have been presented
- System architecture for enabling intelligent behaviors has been presented
- The architecture is characterized by:
  - A sensor-driven, parameterized low-level action command grammar
  - Multi-level planning and task decomposition
  - Multi-level feedback and decision-making
- Architecture enables adaptive, reactive behaviors
- Longer-range goal is to incorporate automated script generation via discrete event dynamic systems theory
- Future applications are planned
- More details are available:
  - Software architecture
  - Control systems
  - Visual servoing work

Robotics Research in General

- Technical aspects (non-arm-based):
  - Wheels
  - Legs
  - Wings
  - Fins
  - Scales
- Applications
  - Military
  - Homeland security
  - Industrial/Commercial/Agriculture
  - Consumer
  - Medical
- Groups/People
  - Academics (MIT/CMU Robotics Institute)
  - Companies (I-Robotics, Remotech)
  - Government Labs (Sandia, DoD)
  - Countries (Europe, Japan/Asia)

Utah State UNIVERSITY

Some Links

CSOIS CENTER FOR SELF-ORGANIZING AND INTELLIGENT SYSTEMS

Misc. Resource

- http://www.geocities.com/roboticsresources/index.html

Military/Government

- DoD OSD Joint Program Office http://www.redstone.army.mil/ugvsjpo/
- DARPA Grand Challenge http://www.darpa.mil/grandchallenge/index.htm
- SPAWAR http://www.spawar.navy.mil/robots/
- AFRL http://www.ml.afrl.af.mil/mlq/g-robotics.html
- UAVs http://www.va.afrl.af.mil/

CMU

- http://www.ri.cmu.edu/

Companies

- http://www.remotec-andros.com/
- http://www.irobot.com/home/default.asp

Other Cool Stuff

- Assistive technologies http://www.independencenow.com/ibot/index.html
- Humanoid robots http://world.honda.com/ASIMO/
- Lego Mindstorms http://www.lmsm.info/
- Bartender http://www.roboyhd.fi/english/drinkkirobotti.html
- WorkPartner http://www.automation.hut.fi/IMSRI/workpartner/index.html