4 Modeling Environment

[Objective: 2 pages, goes from motivation of modeling env, what (design) & refers to how, describe to level of verifiable believability. Pose it as a development & conversation]

The design enviroment as envisaged is based on an accurate modeling, and simulation capability. This chapter elicits the nature and composition of such a modeling environment. The proposed implementation of the modeling environment is then covered in detail. Also discussed are issues of modeling fidelity and accuracy, and time scale issues. This is followed by a description of the proposed agents. The descriptions in this chapter, and referred appendices are intended to provide sufficient detail for implementation.

4.1 Framework design

Conversation flow: all needed properties. Then why component level, why blackboard, why agents,...

To enable design environment properties of rational configuration, design, user defined robot, the primary needs of a modeling environment are:

- Comprehensive: covers and integrates several aspects of a mobile robot's behavior
- Fidelity: sufficient modeling accuracy to enable rational configuration and design
- Systemic: constraint monitoring for mission verification of configuration
- Fast: in built modeling, component library for low instantiation time
- Flexible: robot structure defined by user
- Component encapsulation: to enable component level changeout for investigating multiple configurations
- Extensible: input parser, with component templates to enlarge domain of robot models
- Visual: adaptable user interface, universal peek-poke for visual feedback of output
- Refinability: ability to refine models for accuracy when presented with a reference prototype
- consistency: a single repository for robot design, and parameters from which all models and simulations are generated

The primary needs of comprehensive fidelity, flexibility, and system level analysis lead to a blackboard approach to simulation, where the mobile robot is represented by a set of components residing in the blackboard.The blackboard provides the information and structure necessary to create the models, and also holds all the simulation results, acting as a repository and conduit for propagation of these results.The robot model is created and simulated by agents which interact with the blackboard. This representation also allows rapid design iterations while maintaining consistency, as each parameter is represented in one location only. Comprehensive robot simulation is enabled by a set of knowledge agents which generate and simulate models for aspects of geometry, dynamics, graphics, thermal, power, software, constraint, resource, computing, and communication. The proposed modeling fidelity of these agents is intended to allow both functional analysis and preliminary mission performance analysis.

An integrated modeling approach attempts to simultaneously simulate the facets of mechanical performance, thermal behavior, power distribution, and other factors essential to a mobile robot. However, these facets share a common mobile robot and need to exchange the results to model systemic behavior. A natural and appealing representation is to assemble the mobile robot from interconnected components, where each component is aware of its interactions and constraints with others. Such a system would be modular and reconfigurable. In addition, it would enable high bandwidth information exchange inside a component and lesser exchange between components. This bandwidth model is consistent with the physics of most real systems.

4.1.1 Overview

The environment can be broken down into the following modules of work, each well decoupled from the other, and the inter-module interactions will be specified and user accessible.

The purpose of the various modules is as follows:

1. Blackboard: This blackboard generates a robot representation using the database of components, and a specified structure. The blackboard is responsible for book-keeping the component contents, current simulation values, intra-system relationships (constraints, and violations), and in addition provides interfaces to the constraint manipulators, modeling agent, user, and the learning module.
2. Component database: a repository of basic mobile robot components which map the basic elements of current mobile robots. These components are parameterized with respect to component functionality for multiple uses. In addition, abstract templates are provided to enable designers to develop specialized components. Since a component can contain other components, subsystems are similarly defined.
3. Constraint analyser: evaluates the constraints periodically or according to user rules and places the results in the blackboard. Both static and dynamic constraints are evaluated identically, and ones labelled static may be handled with more efficiency.
4. Modeling agents: Each modeling agent is responsible for generating and simulating the model of one aspect of the mobile robot's behavior. The modeling agent interacts only with the blackboard, to access relevant robot structure, parameters, and to convey the periodic results of simulation. Each modeling agent is triggered at a frequency independent from the others, to reduce computing time while attempting unform modeling fidelity. The behavior aspects identified for implementation as agents are geometry, dynamics, graphics, thermal, electrical, software, sensing, constraint, resource, computing, and communication.

4.2 Physics and algorithms of modeling

The modeling agents are designed to generate the model, and simulation of one specific physical aspect of mobile robot behavior (eg dynamic, thermal). This approach allows using existing methods of physical/numerical analysis, and testing against highly detailed analysis packages for model fidelity estimates. The choice and prioritization of these aspects is based on the focus of existing simulators used for configuration, and design of mobile robots. Descriptions of these agents follow. The modeling techniques relevant to each agent have been the subject of vast, and deep research. The approach selected and adopted here focus on simplicity, and completeness. The rationale being that configuration design needs understanding of the broad interactions, more than precise measurement of such interactions. In addition, the user definitions of robot configurations tend to be approximate, which inherently renders high accuracy modeling, and simulation superfluous.

1. Dynamic: given the external, and actuator forces, and fields, generates the dynamic behavior along with measures of internal forces. Soil contact, and terrain interaction is modeled[Ref Bekker], while excavation effects are ignored.
2. Geometric modeler: Evaluates global geometric quantities like center of mass, inertial dyadics, tolerances, and locations of body points specified by the designer.
3. Graphical modeler: Uses the component layout and locations of bodies as evaluated by the geometric engine to display the rover in terrain. Shadows, and specular effects are ignored.
4. Power: given vehicle state, determine power usage, including voltage & current properties. Common circuit elements like resistors, battery, capacitors, transistors, and amplifiers are modeled in a piecewise linear fashion. The simulation performs transient analysis.
5. Thermal: determine thermal state given power usage, and external environment. Thermal system is modeled as a capacitive electrical network with transient behavior. Specular radiation, and radiosity effects are ignored.
6. To be completed???

4.3 Modeling Fidelity and Simulation Accuracy

This section covers issues of modeling fidelity with respect to estimated reality, variation of modeling fidelity between components, and bounds on needed fidelity. The relation between simulation accuracy, and modeling accuracy is then established.

The fidelity of the simulation resulting from the environment is the lesser of modeling fidelity, and simulation accuracy. The fidelity required depends on the mission, performance metrics, and the configuration in consideration. It is observed that most configuration, and systemic design analyses needs are satisfied by modeling the components at an engineering level (define??).

The importance of any component and sub-system is determined by its role in the robot system, and current knowledge. Hence, a particular component may need to be modeled in great detail while a whole sub-system can be abstracted into simple functionality. Modeling and simulating each component to equal resolution will increase the number of variables enormously and obfuscate the primary relationships in the system.

Modeling fidelity beyond a certain accuracy yields no benefit. The upper bound on this modeling accuracy is determined by an order of magnitude analysis of knowledge (to what accuracy do i expect to know parameters), perception (to what accuracy do i expect to measure a quantity), and engineering (to what accuracy will the artifact be built).

Model accuracy is determined by accuracy & tolerance of physical parameters, tolerance of artifact parameters, and the bounds placed by the analysis technique.Since the model is accurate with respect to reality only to a limited extent, any simulation accuracy below the model accuracy is superfluous. Bounds are also placed on the desired accuracy by the phenomenon(s) to be observed. A simulation can be generated to any arbitrary accuracy subject to the bounds set by representation of real numbers. With the given techniques, the accuracy specified determines the computational time needed. Careful specification of desired accuracy is needed since the computational time is usually proportional to ??? of the desired accuracy. Another bound on the accuracy is the model accuracy.

A modeling fidelity analysis approach will be developed to quantify the upper bounds of modeling detail based on engineering factors like domain knowledge, manufacturing ability, and performance impact.

4.4 Time scale issues

4.4.1 Multi resolution

In general, different agents are triggered at different frequencies. This can cause the input to agents being triggered at higher frequencies to be inaccurate due to static or incorrect output of agents triggered at lower frequencies. For example, if the software agent is triggered at 100 Hz while the dynamics agent is triggered at 30 Hz, a PID wheel controller running at 100 Hz will see static position, constant velocity & acceleration. This effect can be reduced by operating agents in two modes; the full up mode, and the interpolation mode. The full up mode for the dynamics simulator in this case is triggered at 10 Hz, while the interpolation mode is triggered at 100 Hz (if the full up is not triggered). This change however, only causes reasonable values of velocity and position to be generated. The controller still does not see any effect of its output during interpolation. This causes spiraling of the controller output during interpolation. One way to fix such a problem would be to latch all inputs to the controller when all the relevant full-up modes are active. To implement such a latch, an algorithm to identify the relevant agents is needed.

4.4.2 Agent synchronization

The input model parameters of one agent can consist of output from another. The ideal resolution of this issue would be to generate simultaneous models, iterate to stabilize, or reduce the time steps used sufficiently. Simultaneous modeling is complex and not considered at this time. Among the other two options, the one that needs lesser computing for a desired accuracy is more suitable. Initially, time steps will be chosen sufficiently small, so that the difference in input to agents causes acceptable loss in accuracy. Additionally, interpolation agents will reduce the difference and may increase the time step for a given accuracy.

4.4.3 Step size

Computational expense considerations cause time step size changes when the user wants to observe different phenomenon. A particular simulated phenomenon is relevant or observable only in a given range of time-steps. For example, turning behavior is observable in the range below O(1/Omegamax), while electrical ripple due to pulse modulation of motors is observable below half pulse size. Usually the two differ in scale by two orders of magnitude. The model needed at any given step size is in general different from models accurate at other step sizes. Except for the software agents, all other agents would need the same modeling level or transformation at a given time step. The agent would then decide on the modeling approach for a given step size.

The software agent is a collection of software elements, each operating at a different frequency, and receiving data from agents operating at their own frequencies. The software agent would element would have to be designed to take the possible modes into account and ensure the most rational output.

4.5 Model Testing

A series of tests have been composed to measure the accuracy, and fidelity of the modeling agents both individually, and while they interact in a system. Table?? lists the agents with the proposed tests. In addition, comparisons of integrated simulations of robots with known behavior will give preliminary validity to the nevironment.

Agent				
Geometric				
Graphic				
Dynamic				
Thermal				
Electrical				
Software				
Sensing				
Constraint				
Resource				
Computing				
Comm				