2015 NDIA GROUND VEHICLE SYSTEMS ENGINEERING AND TECHNOLOGY SYMPOSIUM
MODELING & SIMULATION, TESTING AND VALIDATION (MSTV) TECHNICAL SESSION
AUGUST 4-6, 2015 – NOVI, MICHIGAN

APPLYING A VIRTUAL PRODUCT DEVELOPMENT PROCESS TO MOBILE ROBOTICS INCLUDING REAL-TIME, INTERACTIVE SIMULATION

Mitchell M. Rohde, PhD; Steve M. Rohde, PhD
Quantum Signal LLC
200 N. Ann Arbor St.
Saline, MI 48176
USA
www.quantumsignal.com

ABSTRACT

Modelling and simulation together with well-defined development processes have become key approaches for creating high quality, reliable consumer products. This is particularly true for passenger cars and trucks. In parallel, the evolving area of mobile ground robotics has found a variety of useful applications over the past twenty years: helping users accomplish tasks in hospitals, warehouses, factories, and on the battlefield. While many of these robotic systems have proven highly valuable, many also lack the mobility, flexibility, reliability and/or robustness desired by the users. In this paper, the authors posit that these non-idealities are in part the result of underutilization of modelling and simulation and an appropriate processes to implement them. This underutilization is partially due to a historical lack of appropriate M&S tools. Recently, however, a new generation of real-time, highly visualized, interactive tools has emerged that has the potential to make a positive impact. In this paper, the authors argue the case for applying the virtual product development paradigm to the mobile robotics area and leveraging these new tools together with more conventional M&S to improve products and systems. An example of the newer class of tools (ANVEL) is discussed, and examples are given as to how it is being used to create complex designs and CONOPS decisions for manned and unmanned ground vehicle robotic systems. The challenges and open needs for the area are also described.

INTRODUCTION

Modern automotive vehicles are complex systems. Over the past 20 years the integration of computers, electronics, new materials, and other technologies into vehicles has resulted in considerably more functionality from a customer's perspective. Powertrain, body, brake system, and chassis controllers are now routinely used and have resulted in safer, quieter vehicles with higher performance and fuel economy, better ride and handling, considerably more features, and improved comfort. However, with these improvements the complexity and required degree of integration of the vehicle subsystems has also increased. Indeed, modern automotive vehicles are at an integration level associated with "mechatronics" – and getting more so.

Historically automotive vehicles were developed through a traditional product development process (Figure 1) with extensive testing: "the build it and break it" paradigm [1]. As the vehicles became more complex and had to satisfy more requirements, the automotive industry realized that a more sophisticated approach than that which had been used in the past was needed -- particularly to do so in a timely, cost effective manner that ensured a high degree of quality.

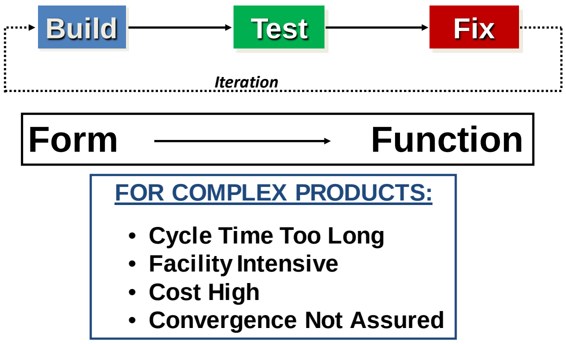


Figure 1: The "Traditional" Product Development Process.

A rational process driven approach based on a systems engineering (SE) paradigm was adopted by many manufacturers [2] and is described in the next section. Furthermore, SE provides a framework for modeling and simulation (M&S) at multiple levels, as will subsequently be discussed.

M&S tools have evolved dramatically over the past 30 years driven by the needs of various industries, particularly aerospace and automotive. Indeed, the modern automotive development process has often been coined a "virtual development process". Today, however, additional M&S tools are required in the automotive arena to deal with autonomous and semi-autonomous vehicles [3].

Similar to the growth of automotive vehicles during the last century, mobile robotics has been gaining interest and momentum over the past decade. Driven by new requirements and application opportunities, as well as the advances in technology, mobile robots have found successful applications in the hospital, the home, the factory and distribution centers, in laboratories, and on the battlefield. Certainly some of the highest profile applications have been in the defense sector, where UGVs (and UAVs) have been used to perform a variety of tasks while keeping operators out of harm's way.

While progress has been achieved, few would argue against the notion that the desire for, and envisioned application range of mobile robot based systems far outstrips the technology available and the current engineering design experience/expertise. For example, the ability to evaluate various highly different CONOPS and operations on difficult off road terrains and select/configure from a myriad of advanced sensors is markedly different from what is required in traditional automotive vehicle design. Likewise, mobility limitations hamper the ability of a ground platform to traverse both the unstructured nature of the natural world as well as the human-centric, man-made environments. Sensors, algorithms, and perception systems have difficulty comprehending scenes, and behavioural/AI software that builds upon the perception is rudimentary.

Power source issues, recharging/refuelling time, and energy management systems limit the range and duration of applications of mobile robots. Performance limitations and reliability problems plague current ground robotic systems, with even some well-recognized systems having mean time to failure measured in hours rather than years [4, 5]. This is in sharp contrast to today's passenger cars and trucks that easily attain 100,000 miles or more with no failures!

Basically, the mobile robotics industry can (and should) be considered to be in its infancy. For example, as discussed in [3], all available data indicates that, for most applications, only a limited number of robots have been produced, distributed, and sold. Indeed, the mobile robot with the largest production numbers – the iRobot Roomba and its siblings – have only been manufactured for a cumulative total of 10 million units since its introduction in 2002 [6]. As an example, compare this with another consumer appliance: the Apple iPhone, whose production in 2014 (a single year) was 170M units [7]. Also, note that the number of cars sold in the United States in 2014 was 16.5M [8]. Clearly, mobile robotics hasn't seen the volume of production that other products have, thus hasn't had the chance to evolve to as refined a level.

The authors propose in this paper that a well-structured virtual product development process built upon a systems engineering process leveraging automotive learnings would be very advantageous to the mobile robotics industry.

THE SYSTEMS ENGINEERING VEHICLE DEVELOPMENT PROCESS

Figure 2 shows the systems engineering-based vehicle engineering process for manned ground vehicles (passenger cars and trucks) that was developed at General Motors [2]. The process is driven by the customers' wants and needs, as shown at the top left hand corner of this trapezoid.

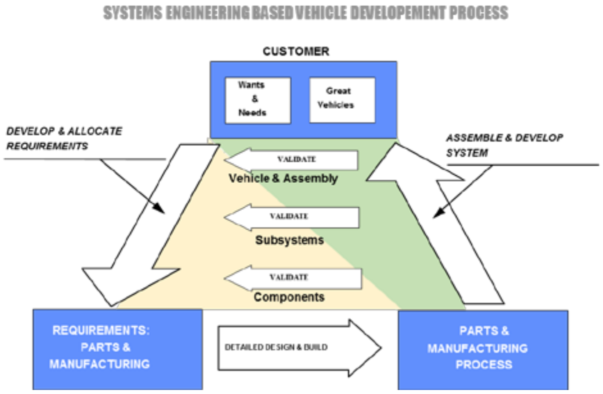


Figure 2: Systems Engineering Based Vehicle Development Process.

The left side represents requirements engineering. It includes developing and allocating the requirements for the vehicle, and for the manufacturing and assembly processes to build the vehicle. These requirements "flow" directly from, and thus are traceable to, the customers' wants and needs. The flow is from the customer to the vehicle, then to the subsystems, and then to the components. Conceptual design and **offline technology** studies are required to fill gaps that occur on this branch.

At the bottom of the trapezoid we show the detailed design of the individual parts and components, which are assembled and developed to form the vehicle as shown on the right side of the trapezoid. In the middle of the trapezoid the validation process is depicted, which includes both validation of the requirements and of the design to meet those requirements. Validation is done at the vehicle, subsystem, and component levels.

The process shown in Figure 2 is often referred to as "top down" since it starts with customer wants and needs and, via a sequence of steps, leads to the creation of a product to meet those needs. Automotive vehicle customers' needs are first **functionally** characterized, i.e., a vehicle must provide certain functions to a customer. Figure 3 shows one such characterization of vehicle functions, as well as a subsystem delineation. The extent to which these functions are met results in different products, and product differentiation. Note that neither of these representations is unique, and, particularly, the former can be viewed as a "basis" in a mathematical sense for spanning the needs of the customer. The functional characterization of the vehicle, however, is key to defining the design, including the requisite product technologies.

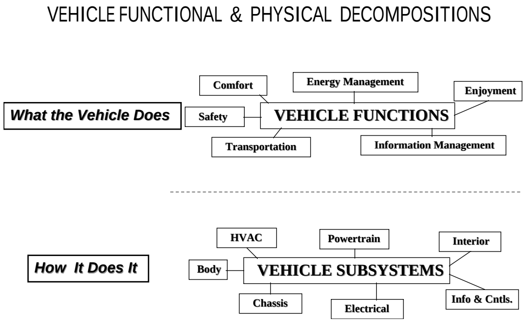


Figure 3: Vehicle Functional & Physical Decompositions.

To really be effective, this process needs to utilize math-based models extensively. Moreover, with this approach, math-models are used in all phases of the vehicle development process: from quantifying the needs of the customer to conceptual design using low order "simple" models, to validating the product using detailed, computationally intensive simulations.

M&S implemented in a systems engineering context greatly attenuated the expensive build-test cycle by enabling engineers to find and correct issues long before hardware was built -- thus initial prototypes were much more mature than with ad-hoc manual processes. A good example of the value of this paradigm was in vehicle design for crashworthiness where prototypes cost on the order of $1M each. This shortened the overall development cycle and lowered the costs of developing each new model – even in the face of increasingly complex vehicles with more sophisticated features. VPD is an integral part of the automotive industry today.

USE OF M&S IN THE AUTOMOTIVE INDUSTRY

As computers and CAD/CAE tools became available starting in the 1960s and 1970s, automakers shifted toward simulation, i.e., a math-based process more broadly and more recently termed "virtual product development" (VPD) [9]. This process, furthermore, became system engineering based in the late 1980s-1990s.

Synthesis and analysis are key concepts that fit into such a systems engineering based vehicle development process as illustrated in Figure 4. **Synthesis is a process** of designing a system in which **multiple and competing requirements** derived from the voice of the customer are balanced and allocated to the subsystems and components through a **systematic analytical process**. Synthesis relies mainly on the use of **mathematical models**. However, experience, judgment, and empirical methods are also used in the process. Synthesis, to be most effective, must lead the design.

Analysis, on the other hand, is the use of mathematical models and simulation to assess the performance of a given system, or to better understand its behavior. Increasingly, analysis is being used as a **validation** tool to replace hardware validation. Moreover, the term M&S means pure "analysis" to many people.

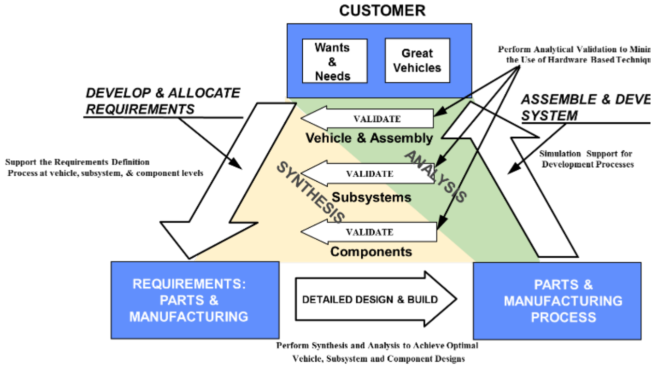


Figure 4: Synthesis & Analysis Tasks to Support the Vehicle Development Process VDP.

A key point here is the use of math-models with the **appropriate level of detail**. For example, early in the development process, conceptual models based upon regressions or algebraic equations are often used. Synthesis and analysis both play key roles in the vehicle development process as shown in Figure 4. Synthesis forms the basis for requirements engineering, i.e., for defining allocations and balancing requirements; and for design -- a synthesis process by definition. Analysis, on the other hand, is used for hardware development, e.g., debugging/tuning; as well as for validation at the component, subsystem, and vehicle levels. The latter process has been recently termed "virtual prototyping" or "paper validation" and is a very powerful approach to reducing development time and cost. Before proceeding, however, it is perhaps appropriate to illustrate the intrinsic value of synthesis and analysis conceptually. Figure 5 illustrates that at the beginning of the development process, one has a great deal of freedom, and little information about the design.

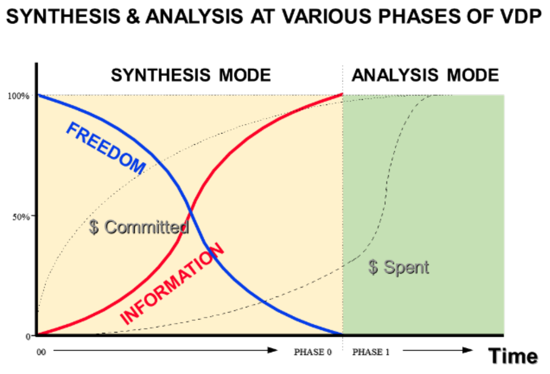


Figure 5: Synthesis & Analysis at Various Phases of the VDP.

Here is where synthesis is of great value. Conversely, at the end of the preliminary design phase, the design is known and locked in; there is little freedom. The analysis mode is the primary one at this point. Also shown on figure 5 is the fact that few resources have been committed, but that these resources are committing significant resources overall, i.e., in terms of the overall life cycle cost. Hence, we have the greatest leverage at this point, and synthesis can cast a profound "shadow" -- it drives the design process.

For example, early in the design process the tradeoffs between fuel economy, driveability, and performance can be treated from a synthesis perspective, and initial estimates of both the functional characteristics of a powertrain (e.g., engine torque curve), and discrete design variables (e.g., gear ratios) can be mathematically derived as illustrated in Figure 6.

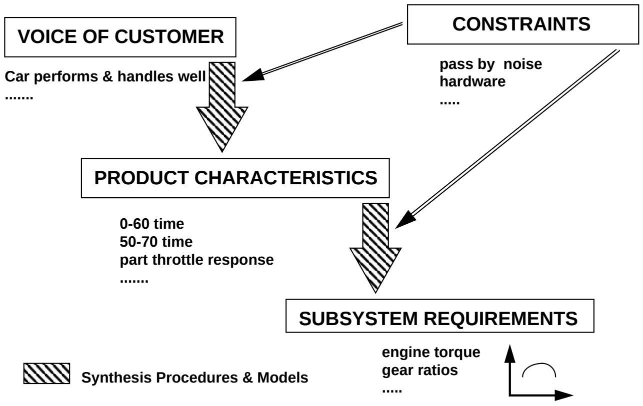


Figure 6: Powertrain Synthesis.

Other applications of this synthesis approach include body structures, and safety. In both these cases relatively simple finite element/lumped parameter models, coupled with optimization techniques, can be used to gain valuable insight into the performance of alternative concepts at a high level, and certainly to help establish technical feasibility, and overall vehicle balance early on.

Later on, in the "development" phase of the vehicle development process, math-models that have been previously generated are often used to gain a better understanding of unexpected behavior, or to "tune" the vehicle design to better meet the requirements of the target customer. Powertrain and chassis models and hardware-in-the-loop simulation are used to understand and improve the behavior of complex integrated control systems as shown in Figure 7. Here mathematical models are implemented on a digital computer, which interacts with critical system components. The latter may include controllers (including software), actuators, or the "plant" itself.

More examples of synthesis and analysis in an automotive VDP can be found in [9].

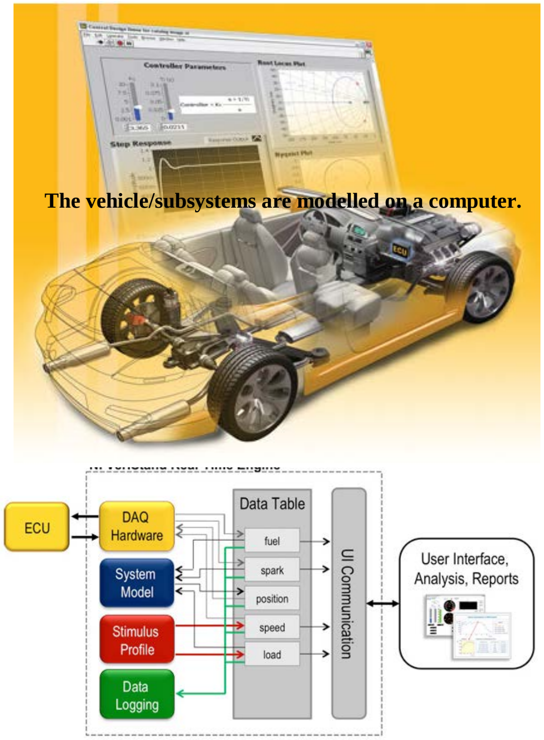


Figure 7: The Hardware-in-the-Loop Simulation Concept [10, 11].

APPLICATION OF A VIRTUAL VDP TO MOBILE ROBOTICS

It should be of no surprise, then, that a process and M&S toolset similar to that described above should be directly applicable to the mobile robotics area. Unfortunately, the practices used by and tools available to mobile robot developers have traditionally been insufficient as discussed in the introduction above. Recently the SAE UGV Reliability Task Force [12] delineated a number of best practices for UGV design[1] that apply to all robotic systems. These echo and expand upon what has evolved in the automotive arena as described above:

- **Adopt a systems engineering process**.
- **Define as completely as possible potential missions** (together with their variability and expected environments). The set of missions formulated here may be very diverse.
- **Capture the "voice of the customer** (e.g., the soldier/other robot system user)" similar to what is done with passenger vehicles. This includes expectations of the UGV as well as the OCU, communications, etc. Here, however, a more homogeneous voice is expected. Figure 8 shows a potential functional and physical decomposition for a UGV.

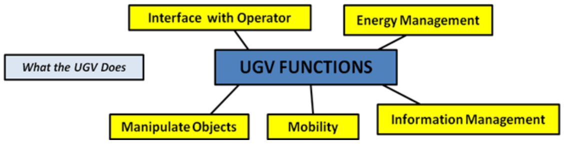


Figure 8a: Potential UGV Functional Decomposition.

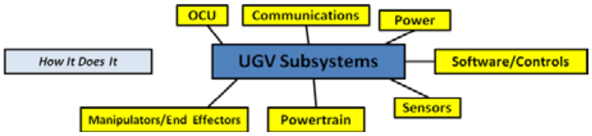


Figure 8b: Potential UGV Physical Decomposition.

Assuming the utilization of these best practices, a modelling and simulation toolset, i.e., a VPD paradigm, should be used to design the robotic system. This includes:

- **Utilize High-Level Conceptual Analytical Models/Simulation**:

A large class of engineering tools tend to be focused on simulation fidelity rather than speed. These are intended for "offline" computations and not interactive work. Conceptual models and simulations that run in real-time (or faster) would support the overall requirements flow-down process and optimization of robotic systems. Performance trade-offs could be evaluated at a high level for different CONOPS and missions; as well as the effect of more reliable componentry/redundancy, sensor/sensor fusion and different control schemes including AI.

- **Use of Modern Visualization Techniques**:

Visualization is a key part of mobile robotics. The way in which the robot perceives the environment (through its sensors) to the operators' "keyhole" view of that environment (on the OCU) and more – the systems require visualization, and being able to visualize system performance can provide rapid insight to system designers. Most engineering

---

[1] Although focused on reliability, the task force soon realized that a more encompassing perspective was required.

simulation packages have limited visualization capabilities (e.g. provide more well-rendered graphs vs. interactive virtual worlds), and those that do tend to be expensive or behind the curve when compared with even consumer-grade entertainment products.

- **Use Simulators to Define the Human-Robot System Interface**:

Once conceptual models have been defined (however high-level), these models can be used in graphics "rich" environments (e.g., a video game environment) to synthesize and evaluate operator control unit (OCU) configurations. These are traditionally key factors affecting a robotic system's ability to achieve its mission reliably.

- **Use Analytical Models to Design and Validate the Robotic System: A Virtual Robotic System**:

Throughout the systems engineering process, mathematical models should be used. This is particularly true in the preliminary and detailed design phases, where. e.g., thermal, mechanical, and electrical simulations (including battery state-of-charge) of the various subsystems and components can be performed and used to size/configure them and factored into performance and reliability predictions. For example, structural optimization can be used to reduce robot mass – a major factor in extending the range of battery powered robots.

Sensor, actuator, and control strategies/ configurations should also be modelled and used to validate system performance. Here, a combined approach where model-based validation is used together with actual hardware testing is recommended (HIL).

- **Use Hardware-in-the Loop Simulation** (HIL) to Validate the OCU-Robot Performance:

Most robotic systems involve some level of human-machine interaction (such as teleoperation), and human factors issues are key. VPD tools should thus support interactive, real-time functionalities. More specifically, having developed a library of robotic system analytical models in the development process, one should use these models (or a reduced-order subset of them) to validate the control algorithms/software including the OCU interface. Initial simulations can start with simplified mathematical models for the entire system including the hardware and software. As the design progresses, hardware (e.g., the OCU, controllers, target computers) can be interfaced with robotic system models and developed/validated including the human-robot interface performance. Again, here reduced-order models may need to be developed and used.

To implement such a VPD process robotics developers can adopt some of the traditional automotive engineering tools to help improve the development process. For example, finite element and multi-body dynamics programs are invaluable at points in the process. These tools have existed for years. However, to achieve the full value that VPD might provide one must overcome the "tools gap." Fortunately, the barriers to integration of accurate engineering models and interactive, multimedia simulation have dropped tremendously in the past 20 years. Computing platforms can be obtained at local retail stores for less than a few thousand dollars that far outperform workstations or supercomputers costing millions of dollars in the recent past. The supercomputers of Cray and rendering hardware of Evans and Sutherland gave way to the high-end workstations of Silicon Graphics, and later to PCs with advanced, multi-processor video cards. Machines with 3+ GHz processors, gigabytes of RAM, advanced graphics and sound capabilities, and 100Mb/s communications are found today in many households across the country. Moreover, the internet has made instantaneous, almost transparent, networking available at extremely low cost. The direct result of this is that most engineers have access to hardware sufficient to run simulations with intensive computational requirements.

The above evolution has led to a new class of tools over the past five years. These tools hybridize engineering simulation and video-game (or game-like) technology to yield interactive, multi-fidelic capabilities useful to the mobile robot developer. These tools are commercially available or free/open source, and have evolved out of the automotive and the robotics industries. Examples from automotive include CarSim [13], VI-grade [14], TASS Pre-scan [15], and many real-time vehicle simulators. Examples from robotics include Gazebo [16], Vortex [17], Webots [18], MS Robotics Studio [19], and JPL ROAMS/DARTS [20]. In order to understand what these tools have to offer, it is useful to examine a current one that has been developed at Quantum Signal and how it can be used to enhance UGV development work supporting the processes previously described in this paper.

ANVEL: AN EXAMPLE REAL-TIME, INTERACTIVE ROBOTICS DEVELOPMENT M&S TOOL

The Autonomous Navigation Virtual Environment Laboratory (ANVEL) [21, 22] was designed specifically to bootstrap the development of unmanned ground systems, facilitating creation, development, verification, validation, and deployment of semi-autonomous and autonomous behavior software and **conceptual studies** of the hardware prior to and during the realization of that hardware.

ANVEL delivers a unique combination of vehicle, sensor, and vehicle-terrain interaction models; a robust physics engine; and a terrain editor that enables the creation of systems and scenarios for development of semi-autonomous and autonomous behaviors. Platforms are modeled using a vehicle definition file. Creating new platforms or modifying existing platforms (Figure 9) is as simple as changing the file. Users can increase or decrease vehicle mass, drag coefficients, surface areas, wheel base, track width, tire size and stiffness, and more. Vehicles can be equipped with servos that adjust a connection between two objects (for example, a manipulator joint). Major vehicle subcomponents, such as engines and motors, can also be defined. Motors have a large number of configurable parameters, including torque constant, back EMF constant, winding resistance, and rated power. ANVEL provides a high degree of flexibility when creating a system model to enable the appropriate fidelity for the task at hand.

Figure 9: Examples of Platforms Modeled in ANVEL.

ANVEL also includes a number of sensors that are modeled for both exteroceptive and proprioceptive sensing. Notably, ANVEL models a single line scan LIDAR sensor that can be parametrically adjusted to characteristics of commonly available commercial LIDAR systems, such as the SICK LMS-5xx (see Figure 10) and the Hokuyo UTM-30LX. Multi-beam LIDARs such as the Velodyne HDL64 are also modeled. Proprioceptive sensing includes various inertial measurement unit components, including micro-electrical mechanical accelerometers and gyroscopes, ring laser gyroscopes, and fiber optic gyroscopes. ANVEL also includes a digital compass. Any of these sensors can be positioned in the vehicle through the vehicle definition file, as well as para-metrically adjusted through the sensor attributes user panel. New sensor models can be readily developed and implemented through the use of a plug-in architecture, allowing for the simulation of nearly any kind of sensor, including geometric, inertial, force-torque, or global positioning.

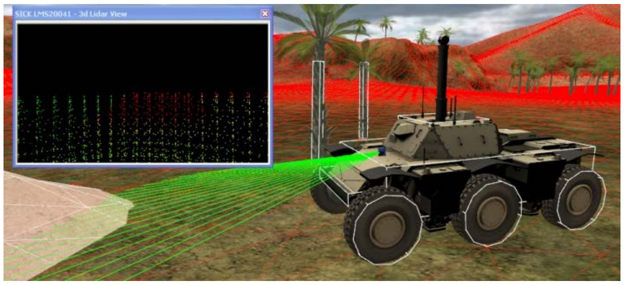
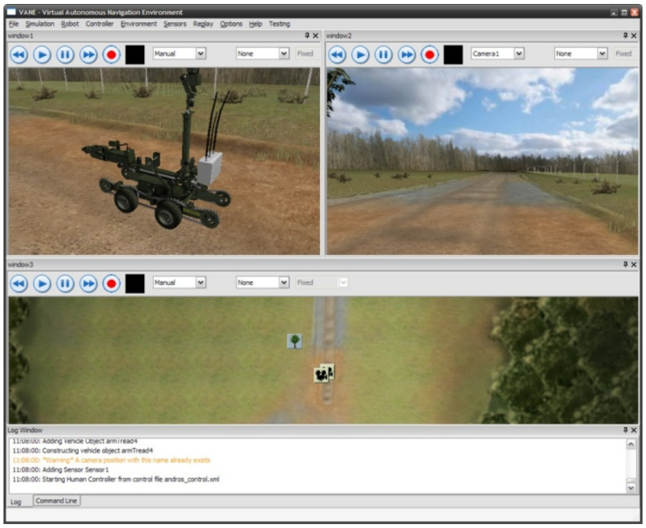


Figure 10: Example of a SICK LIDAR modeled in ANVEL.

Enabling the vehicle models and sensors to interact with the world is the core of the ANVEL simulation tool and ANVEL currently uses the Open Dynamics Engine (ODE) [23] to enable real-time simulation of the vehicle bodies in the virtual environment. ODE simulates articulated rigid body structures and the forces that act on those structures. The bodies consist of mass, position, shape, and orientation and they are articulated by joints that specify the type of motion between the bodies. Joint types include specific instances of prismatic, revolute, and spherical assemblies. The bodies can also have constraints placed upon them, such as the range of motion or force limits. In ANVEL, the physics of vehicles are modeled as a combination of shapes, and terrain is modeled using a low-resolution polygon mesh. This differs from the rendering of the vehicles and terrain, which use a higher resolution mesh representation appropriate for graphical display.

By representing the terrain as a polygonal mesh, ANVEL is able to use various vehicle-terrain interaction (VTI) models that simulate the forces between the wheel contact patch and the virtual terrain. These include an ODE VTI model, a Bekker VTI model [24], and a Pacejka model [25]. The ODE VTI model is like all other body interactions within ODE: hard contacts with a non-penetration constraint. The Bekker model fundamentally models the pressure-sinkage relationship for a track or tire in a given soil, while Pacejka models a pneumatic tire against the terrain. These VTI models allow the operator to utilize the appropriate ground or tire model in the simulation, enabling the proper fidelity ground-tire model for the required task. Additional VTI models can be created and applied to ANVEL through its plug-in infrastructure.

ANVEL ties the ODE physics engine, VTI models, vehicle models, and sensors together through the use of a world editor. The editor permits users to specify the ground contours, vegetation, man-made structures, and robot positions and orientations, allowing the creation of a number of virtual environments and scenarios for experimentation. Indoor and outdoor environments can be created and manipulated. This capability allows for virtual testing that enables rapid identification and resolution of scenarios that may prove error-prone or require a repeatable test environment for debugging, data collection, and subsequent analysis. These virtual worlds can also be used to revalidate behaviors and potential concepts of operation (CONOPS) as systems evolve throughout the course of the normal development cycle. A screenshot of ANVEL showing various views is shown in Figure 11.



```
11:08:00: Adding Vehicle Object armTread4
11:08:00: Constructing vehicle object armTread4
11:08:00: *Warning* A camera position with this name already exists
11:08:00: Adding Sensor Sensor1
11:08:00: Starting Human Controller from control file andros_control.xml
```

Figure 11: Example ANVEL Screenshot Demonstrating Third-Person View (upper left), First-Person (UGV) View (upper right), and Overhead View (bottom). The windowed ANVEL environment can be reconfigured by the user to display various combinations of views, sensors, and other data.

Example Uses of ANVEL

The description above provides a picture of the kinds of capabilities these interactive M&S tools have. Such tools are not useful in and of themselves, however: it is in how they are applied. The authors now present several examples of how ANVEL has been used[2].

Mobility Prediction

One of the earliest users of ANVEL was the Robotics Mobility Group at MIT. In 2008, ANVEL provided a platform for the development and analysis of a novel algorithm for statistical prediction of small UGV mobility. The algorithm, detailed in [23], is similar in spirit to various statistical methods for manned vehicle mobility prediction that have been co-developed by the U.S. Army over the past 50 years (i.e. the NRMM, NRMM II, and others). It exploits the fact that in field conditions, UGVs frequently have access to only sparse and uncertain estimates of important terrain and vehicle parameters (e.g. soil cohesion, vehicle center of gravity location, etc.). Thus, in order to accurately predict UGV mobility, an algorithm must explicitly consider the nature of this uncertainty, and correctly propagate it through model-based dynamic analysis.

ANVEL was used to generate “ground truth” mobility prediction results via Monte Carlo simulation. Here the probability distribution of an output metric is computed by running a simulation many times, Scripts were written off-line, containing values of sampled UGV and terrain parameters, and ANVEL was employed to run Monte Carlo simulations in silent mode and display the output metric distributions (Figure 12). ANVEL performed the task efficiently, allowing iterative study of algorithm performance. Further details on the algorithm development and performance can be found in [22].

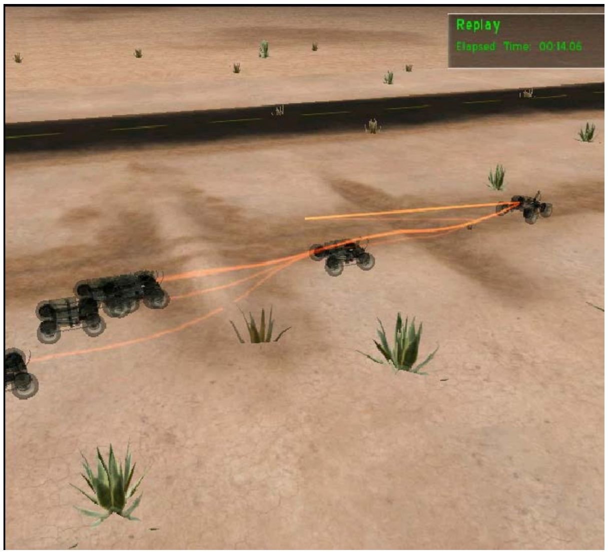


Figure 12: ANVEL Visualization Showing Repeated UGV Mobility Study. The "trails" show paths taken, with a UGV "ghost" remaining where the vehicle could

---

[2] Though a number of groups are using ANVEL, most of the examples presented are from the authors’ research team, for convenience.

progress no more given the particular constraints of that run.

ANVEL is also a part of the US Army ERDC "VANE" simulation framework, and is being used to study the impacts of weather and environment on ground vehicle mobility. The details of this activity are excluded for space here, however details are in [26-28].

Autonomy Kernel Development

ANVEL has been used in the multifaceted development of autonomy kernels. Fundamentally, an ANVEL model serves as a "stand in" virtual UGV/sensor package as well as a virtual proving ground in which algorithms can be tested and refined. In the 2014 work described in [28], the authors used ANVEL in three key ways: first, to assist in the development of an inertial navigation system. Here, the virtual ANVEL UGV/sensors provide body force measurements, rotation rates, and wheel speeds, which are then inertial frame corrected and inputted as measurements into an extended Kalman filter to produce a state estimate. The algorithm then sends back "estimated" positions that were represented as a "shadow" vehicle overlaid with the true one. The developers could thus quickly and easily visualize the difference between the estimated and true vehicle positions, and perform validation activities on the algorithms as shown in Figure 13.

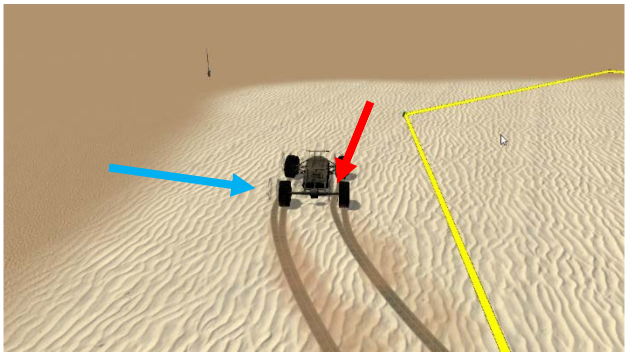


Figure 13: True Vehicle (solid, red arrow) and Shadow Vehicle (shadow, blue arrow) in ANVEL.

Similarly, ANVEL has been successfully used to develop an effective UGV system with a sensor suite that could lead or follow soldiers during operations. Figure 14 shows the simulation on the right and an actual test scene on the left. The agreement was excellent.

Figure 14: ANVEL being used to develop control/behavior models for a soldier/UGV pair.

In the second example, ANVEL is used to help in the development of a LIDAR based OD algorithm. A virtual MTRS class UGV equipped with a model LIDAR was used to feed the algorithm in real time as it traveled through a virtual test environment, and the algorithms progressed through a rapid test/refine cycle without ever having crashed a vehicle. When the algorithm was considered sufficiently tested it was transitioned to an optimized, embedded implementation, which was then validated by operating it as "hardware in the loop" with ANVEL (via a TCP/IP connection between a machine running ANVEL and the embedded box running the OD algorithm). Eventually, that embedded code/package was transitioned to a real-world platform and tested. This transition from desktop/virtual UGV to embedded/virtual UGV to real-world represents a shining example of how M&S can improve the development process. Similar uses of ANVEL for autonomy kernel development are described in [29].

Analyzing Sensor Placement and Performance

Another valuable application of ANVEL and similar tools is in the comprehension of new sensors. Often, UGV developers are faced with the challenge of assessing sensors that have come on the market and could be used as part of a system. For instance, it may be unclear whether the data coming from the sensor is of sufficient quality for the perception algorithms, or if the options available for physical placement of the sensor on the vehicle will be adequate. In [28], the QS team wanted to know whether a Fotonic E40P time of flight sensor would be sufficient for their autonomy application. Rather than going through the time, expense, and invasiveness of integrating the sensor mechanically, electrically, and software-wise, a different approach was taken. The sensor was quickly and easily tested stand-alone in an outdoor environment, using multiple targets, and data recorded. From that an empirical noise model was created, which was then used to "correct" a newly created virtual model of the same sensor. The virtual sensor was quickly and easily integrated with the virtual UGV, and tested in conjunction with existing perception algorithms. This facilitated a basic level of validation, and provided a better-than-intuitive understanding of how the system might perform under relevant circumstances (Figure 15).

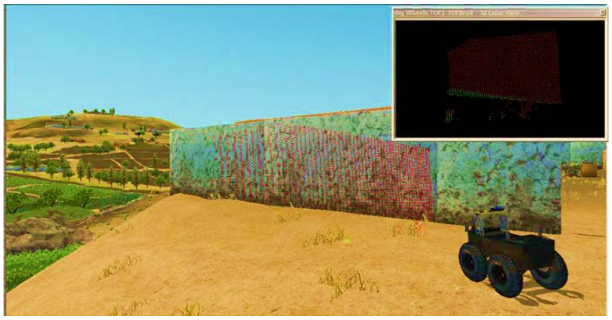


Figure 15: ANVEL screen capture taken during validation experiments of Fotonic TOF sensor. Note the visualized sensing field along the concrete wall (represented by vertical lines) and the simulated sensor data (shown in the inset). The color saturation of the image has been adjusted for visual clarity.

As a second example, in [29] ANVEL is used to test the impact of LIDAR sensor height on autonomous waypoint following algorithms. These two examples highlight the M&S capabilities that ANVEL provides for problems faced by the authors and many other engineers.

OCU and User Interface Development

One of the major advantages of tools like ANVEL is that they provide real-time interactivity and visualizations. These aspects can be used to facilitate the development of a mobile robotics man-machine interface, as alluded to in the previous section. In [29] an application is briefly described in which ANVEL is being used to test the Warfighter Machine Interface (WMI) of the Dismounted Soldier Autonomy Tools (DSAT) program. The paper shows ANVEL being used to serve as the virtual UGV and test environment, providing camera views and other relevant sensor information (Figure 16) to the WMI facilitating improvement of the HMI.

In another study [30], the QS team has been using ANVEL as part of a program to develop enhanced teleoperation systems. In that program, which focuses on using multiple types in intelligent, driving-focused compression schemes, ANVEL provides sensor data and scene visualizations with varying resolutions and frame rates. This data is used in conjunction with a rough prototype OCU. Human-in-the-loop experiments were performed to measure human performance using different scene appearances (e.g. update rates, resolutions, etc.). Man-machine interface issues reside at the heart of effective applications, and ANVEL or ANVEL-like tools can be used to support development.

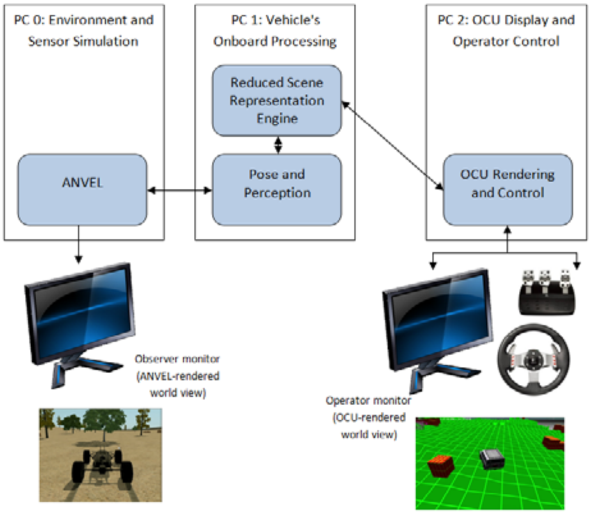


Figure 16: ANVEL being used in human-in-the-loop experiments to measure human performance using different scene appearances (e.g. update rates, resolutions, etc.) and man-machine interfaces.

These are but a few examples of how ANVEL has been used; many more exist, and new ones are being created each day. The point is that these tools, as a result of their key properties (interactivity, real-time performance, visualization), can meet a myriad of needs for the mobile robot researcher, developer, tester, and others. It facilitates virtual product development, and the transition from the virtual to the real-world.

CURRENT CHALLENGES AND NEEDS

While M&S tools for robotic systems have come a long way and have many uses, substantial challenges remain and there is a long way to go before they live up to their full potential. Some areas identified by the authors include:

- **Better integration/interoperability of M&S tools.** Today the various M&S tools tend to operate independently driven by different data inputs and providing different output formats. Some standardization of this data (and terminology) and linking of different tools would be beneficial. Unfortunately, most simulators use their own file formats and definitions, and there is little ability in the marketplace to trade models across systems. The industry would benefit greatly if M&S developers would cooperatively begin a standardization process for component models. Such a process could be driven by integrating a model requirement into acquisition processes.

Multi-physics modeling [31] is an example of integration of multiple domains for more traditional simulation. This linking would also bridge the real time vs. fidelity gap that causes consternation between those demanding accuracy and others seeking interactivity.

- **Dynamics engines** – building upon the prior point, there is a standing need for high-quality, real-time, modular dynamics simulation engines. There are issues with Open Dynamics Engine (ODE- used in ANVEL and many other simulators), and valid replacements (free and open source, commercial and closed source, or other) would be highly desirable.
- **Friendlier Tools** – while tools like ANVEL are, arguably, much easier to use than their more advanced engineering simulation brethren, they still require a substantial level of expertise to work with. The ability to use robotics simulation for a variety of tasks (including evaluation of the system in different scenarios) would grow if the tools were easier to use. Enhancements to user interfaces, simpler scripting/ configuration languages and/or visual programming could help to enhance these tools.
- **Business models** – engineering M&S tools are a traditionally difficult market in which to grow and sustain a business. This is especially true during times of constrained resources, and in an era where users have grown accustomed to open source, freeware, and similar software models. Non-traditional software sales, distribution, and support models must be adopted, at least until the market has reached a breadth and depth that can support an array of quality tools through straight commercial means.

The list above is incomplete, but reinforces the point that, like the mobile robotics industry that they support, the tools are in their youth and are ripe for improvement for those willing to pursue the challenge!

CONCLUSION

In this paper we have attempted to give the reader a flavor for the history, benefits and the potential of using modern computational methods in a systems engineering process driven framework to define and engineer automotive vehicles.

The authors advocate the use of a similar paradigm for mobile robotics. In particular the authors have attempted to make the case for greater use of modeling and simulation in mobile robotics as well as identified the gaps in current simulation toolsets.

Here, the advent of new, interactive tools has the potential to enhance a wide range of research and development activities, and push evolution of products from the traditional build-test-build paradigm into one of virtual product development.

We have used ANVEL as an example of such a tool, and referenced a number of successful application examples. Whether it be ANVEL or a similar package, the authors feel strongly that adoption and integration of such tools into a VPD cycle will bear fruit in terms of greater robustness, reduced cycle times, and lower costs.

REFERENCES

[1] "GM's Journey to Math: The Virtual Vehicle," Proceedings of the 1st MIT Conference on Computational Fluid & Solid Mechanics, V.1, Elsevier, 2001. Steve M. Rohde and Robert Ottolini.

[2] "Creating Vehicles of the Future: A Systems Engineering Approach," SAE Paper 86-02-4839 1986, Steve M. Rohde and Neil Schilke.

[3] Rohde, Mitchell M., and Rohde, Steve M. "Mobile Robotics, ANVEL, and the Need for Real-time, Interactive Simulation". Proceedings PUB REF NBR (e.g. STO-MP-IST-999), 2015.

[4] Carlson, Jennifer, Robin R. Murphy, and Andrew Nelson. "Follow-up analysis of mobile robot failures." Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on. Vol. 5. IEEE, 2004.

[5] Carlson, J.; Murphy, R.R. "How UGVs physically fail in the field", Robotics, IEEE Transactions on, page(s): 423 - 437 Volume: 21, Issue: 3, June 2005.

[6] http://www.irobot.com/About-iRobot/Company-Information.aspx

[7] http://www.statista.com/statistics/263401/global-apple-iphone-sales-since-3rd-quarter-2007/

[8] http://www.nytimes.com/2015/01/06/business/us-auto-sales-jump-for-2014.html?_r=0

[9] http://www.eetimes.com/document.asp?doc_id=1272817

[10] http://www.ecnmag.com/articles/2011/06/using-hardware-loop-simulation-create-better-cars

[11] https://www.dspace.com/en/inc/home/products/systems/ecutest.cfm

[12] “UGV Reliability Design Recommended Practices,” Steve M. Rohde and Phil Oh, minutes of the SAE UGV Reliability Task Force, 2010.

[13] http://www.carsim.com

[14] http://www.vi-grade.com

[15] https://www.tassinternational.com/prescan

[16] http://gazebosim.org/

[17] http://www.cm-labs.com/robotics

[18] http://www.cyberbotics.com/overview

[19] http://www.microsoft.com/en-us/download/details.aspx?id=29081

[20] http://dshell.jpl.nasa.gov /

[21] http://anvelsim.com/

[22] Rohde, Mitchell M., et al. "An interactive physics-based unmanned ground vehicle simulator leveraging open source gaming technology: progress in the development and application of the virtual autonomous navigation environment (VANE) desktop." SPIE Defense, Security, and Sensing. International Society for Optics and Photonics, 2009.

[23] http://www.ode.org/

[24] Bekker, G., Theory of Land Locomotion, University of Michigan Press, 1956.

[25] Pacejka, H.B., Tyre and Vehicle Dynamics, Oxford, 2002.

[26] Kewlani, G., and Iagnemma, K., “A Stochastic Response Surface Approach to Statistical Prediction of Robotic Mobility,” Proceedings of the IEEE International Conference on Robots and Systems, IROS ’08, 2008

[27] Jones, R. A., et al. "Virtual Autonomous Navigation Environment (VANE)." Proc. of ASCE Earth & Space Conference. 2008.

[28] Goodin, Christopher, et al. "High-fidelity physics-based simulation of a UGV reconnaissance mission in a complex urban environment." SPIE Defense, Security, and Sensing. International Society for Optics and Photonics, 2011.

[29] Goodin, Chris, et al. "Sensor modeling for the virtual autonomous navigation environment." Sensors, 2009 IEEE. IEEE, 2009.

[30] Cummins, C. L. "ANVEL/MATREX/OneSAF: Experiment 2." VANE Working Meeting, US Military Academy at West Point. Vol. 10. No. 29. 2008.

[31] http://www.comsol.com/landing/comsol-multiphysics?gclid=CjwKEAjw4-SrBRDP483GvreDr2ASJAD5sCIuMzyToewv80y-aI0aAM85vXhwq2AffWodykAdoIdc0xoC6Ezw_wcB