2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Toward a Multi-Disciplinary Model for Bio-Robotic Systems

Rich Primerano, David Wilkie, and William Regli

Abstract— **The design of robotic systems involves contributions from several areas of science and engineering. Electrical, mechanical and software components must be integrated to form the final system. Increasingly, simulation tools are being introduced into the design flow as a means to verify the performance of particular subsystems. In order to accurately simulate the complete robotic system we propose a framework that allows designers to describe the robotic system as an interconnection of mechanical, electrical, and software components, with well defined mechanisms for communicating with each other. Through this, we form a multi-disciplinary model that captures both the dynamics of the individual subsystems, and the dynamics resulting from the interconnection of the above subsystems. As a case-study, we will apply the framework to a biologically inspired robotic snake.**

I. INTRODUCTION

A *multi-disciplinary engineering model* for a robotic system includes semantic descriptions of robotic components, behavioral and simulation models, software for robot control and navigation, as well as the tools needed to perform analysis, component surrogation and mission assessment. Such a model can play an important role in design and testing: consider the scenario in which a designer has been given the task of designing a snake-inspired robot to inspect the piping in a nuclear plant. Before building the snake robot, the designer will need to study maneuverability and power consumption of the proposed robot design to ensure mission success. Therefore, the designer will have to create a comprehensive model of the snake robot. This model will need to include (1) geometric models of the structure and components, (2) a kinematic model of the body, (3) actuator models, (4) sensor models, (5) a control logic model, (6) finite element models of the structural elements, and (7) a friction model of the structural material. Multi-disciplinary engineering models would also be useful for integration testing, which could be used to test behaviors arising from the interactions of the various sub-systems.

This paper presents a step toward the creation of a multi-disciplinary engineering model with a biologically-inspired robot as a demonstration system. Biologically-inspired robots are interesting for their ability to perform functions and traverse terrain that conventional robots cannot and for their unique features, including their high number of degrees-of-freedom and novel forms of movement. The case study presented here has engineering representations for many of its subsystems. These elements have been cosimulated to verify desired robot behavior.

The paper is organized as follows. Section 2 presents background research in the area of multi-disciplinary modeling. Section 3 describes the case study used in this paper. Section 4 presents the decomposition of the system and the implementation of the model. Section 5 discusses future work, and section 6 presents our conclusions.

II. BACKGROUND STUDY

Multi-disciplinary engineering models include engineering representations, semantic representations, and computational models. Engineering representations include various geometry-centric physical representations corresponding to appropriate formulations (both discrete and continuous) of robot models and their components. An important requirement of geometric models is that they satisfy the needs for downstream simulation and analysis. Designers of biologically-inspired robots face challenges that include the complexity of the individual components, the magnitude of component-component interactions, and the existence of flexible parts. Surrogate representations may help in achieving computationally tractable analysis, but to do so they must satisfy two conflicting criteria: (1) they must be sufficiently detailed to be useful for analysis and simulation, and (2) must be sufficiently coarse for efficient computational analysis. Surrogate modeling includes removal of non-critical features [1], [2], lumped-modeling [3], [4], exploitation of symmetry [5], dimensional reduction [6], [7], and other methods supporting multi-level and multi-resolution modeling.

The semantics representations will enable interpretation of behavioral and performance parameters at several layers. For example, global motion and locomotion constraints can be estimated from shape and parameters of the joints. Mass, mechanical stiffness and strength of components can be used to estimate allowable loads and dynamic properties of the robots. The engineering community has just begun encoding engineering knowledge in XML.

Computational models include tools and algorithms to perform geometric, dynamic, and spatially-distributed physics computations[8], [9], [10], [11], [12]. This includes algorithms for collision detection, multi-body dynamics, and mechanical simulation. The semantic and geometric models drive the computational analysis of the dynamics, behaviors and capabilities of the robot.

R. Primerano is a doctoral student in the Department of Electrical and Computer Engineering, College of Engineering, Drexel University, `primerano@drexel.edu`

D. Wilkie is a research assistant in the Department of Computer Science, College of Enigneering, Drexel University, `wilkie@drexel.edu`

W. Regli is a professor in the Departments of Computer Science, Electrical and Computer Engineering, and Mechanical Engineering and Mechanics, College of Engineering, Drexel University, 3141 Chestnut Street, Philadelphia, PA 19104, USA, `regli@drexel.edu`

978-1-4244-1647-9/08/$25.00 ©2008 IEEE.

Recent work in the area of biologically inspired robots includes [13], [14], [15], [16]. Many roboticists tend not to use elaborate and general simulations for prototyping. The development of multi-disciplinary models (models that capture the relevant electrical, mechanical, control and communications systems) would allow for true system-level simulation of devices such as these.

The simulation tools required to construct system level simulation are already in place. Many researchers have worked at developing efficient numerical simulations of arbitrary articulated rigid-body systems; a survey can be found in [17]. Simulators such as this have been used to test gaits as in [18], which used a simulation to test gaits of bipeds and quadrupeds for stability, and [19], which used a simulation to test gaits of snake-like robots for speed. The models used in these works abstracted the geometry and physical properties of the robots being simulated: instead of CAD derived models, geometric primitives were used. As mentioned in [18], previous work simulating robots to evaluate stability used simplified analytical models. These numerical simulations of articulated-body dynamics have also been of interest to the graphics community. An early example is [20], which models a cockroach out of blocks and simulates its actions under gaits. Here, the interest is in simulating motion that is visually similar to that of the real subject.

III. CASE STUDY: A BIO-INSPIRED ROBOT

We have designed a robotic snake capable of undergoing efficient rectilinear motion. Our approach, developed through the study of papers written on the topic, as well as several prototypes that were built earlier in Philadelphia-area laboratories, proposes a new robotic snake design. The differentiating feature of our design is that it propels itself using many small feet, with locomotion similar to that of a millipede. Since this robot actually walks, rather than drag itself, it is capable of navigating rough terrains. For this reason, it is expected that the proposed design would be more maneuverable that existing prototypes.

A. Mechanical Hardware

Before development of this device began, several design objectives were set forth. These included: (1) ability to perform efficient rectilinear motion, (2) small cross section, and (3) ability to easily lengthened or shortened the robot. The first of these needs arose from the observation that most current robot designs that are capable of operating in rough terrains perform forward motion by using complex gaits and as a result, that tend to move very slowly. This new design is intended to operate in these same harsh environments, but do so more efficiently. The second objective arose from the desire to develop a robot that could access confined spaces, in applications such as pipe inspection and exploration. The final objective is desirable because a robot that can easily be lengthened to suit a particular application will be more adaptable. Additionally, a design of this type means that if a segment were to malfunction or brake, it could be completely removed and the robot be placed back into service. An added benefit of this type of modular design is that robot is essentially composed of many copies of the same mechanism, simplifying the development process. To see how these issues have been addressed, we first take a system level look at the robot's mechanical structures and their interactions. Subsequent sections provide a detailed look at individual component operation and design.

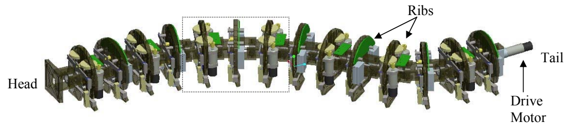

Fig. 1. ROBOTIC SNAKE.

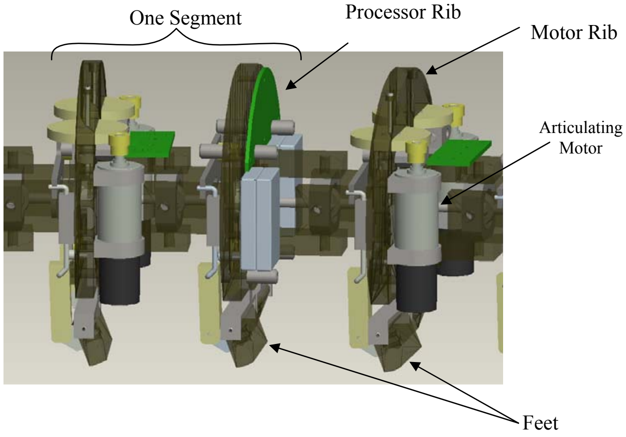

Fig. 2. A CLOSER LOOK AT SEVERAL RIBS.

Figure 1 shows a rendering of the robotic snake. The robot contains fourteen ribs and is approximately 30" long. At the bottom of each rib is a pair of feet that carry the robot forward. Forward motion is powered by a drive motor located at the tail of the snake.

Figure 2 shows a zoomed in view of three adjacent ribs. From this picture, it is clear that all fourteen ribs are not identical. The robot is composed of two types of ribs; the first contains two servo motors used for articulation while the second contains a circuit board and batteries. These are named motor ribs and processor ribs respectively. Starting from the head of the snake, ribs alternate between motor and processor type. There are a total of seven motor ribs and seven processor ribs. The design is such that (referring to Figure 1) the motors located on a particular motor segment are driven by the circuitry located on the processor board immediately behind it. For this reason, addition and removal of ribs must be done as a pair. The motor/processor rib pair will be referred to as a segment.

The robot locomotes through the use of two types of motors. The first is the drive motor (Figure 1) located at the rear of the snake. This motor transmits power to all upstream

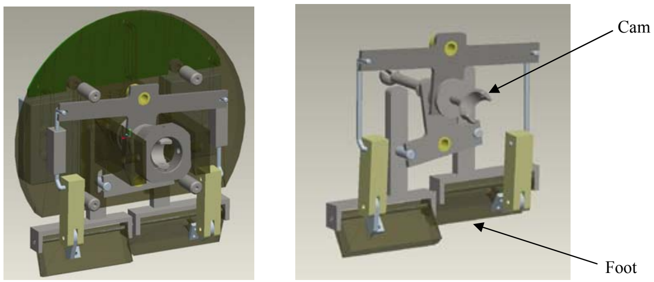

Fig. 3. COMPONENTS FOUND ON A TYPICAL RIB.

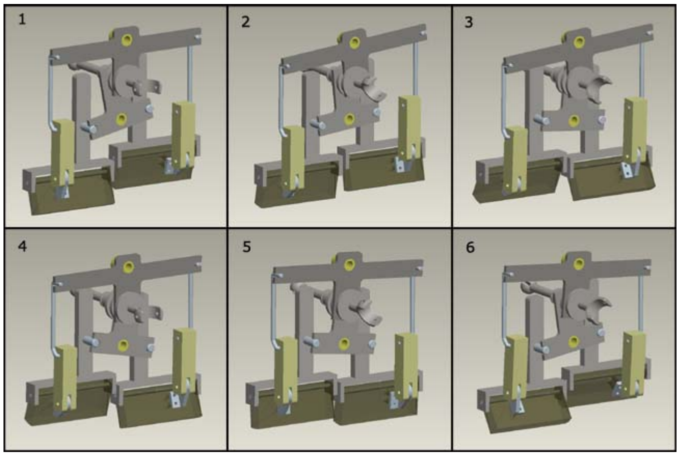

Fig. 4. SEQUENCE SHOWING OPERATION OF FOOT MECHANISM.

ribs to power a series of “feet” on the bottom of the snake. Figure 3 shows the components responsible for this and how they are positioned on the robot’s rib. The basic function of this mechanism is to take rotary motion applied to the cam and convert it to orbital motion in the feet. Figure 4 shows six sequential frames that demonstrate the operation of the mechanism.

The second type of motor found on the robot is the articulation motor (Figure 1). Each motor segment contains two such motors. These are responsible for bending (articulating) the segment on which they are located. These two motors provide two degrees of motion per segment. The prototype snake fourteen articulation motors.

B. Controller and Communications

The articulating motors located along the robot’s length are controlled by seven processor boards, as was discussed in the previous section. There is an additional processor that controls the drive motor located at the rear of the robot. Each processor board is responsible for the control of two DC brush motors and does so with a two axis PID control algorithm (except for the drive motor controller which is single axis). In addition to these distributed controllers, we also make use of a centralized controller (a PC in our case) for synchronization and user interfacing.

Figure 5 shows how the control system is decomposed. At the lowest level, distributed processors perform local motor control. These boards additionally acquire and process sensor data. Moving up one level in the control hierarchy, the centralized processor is responsible for motor coordination, sensor data fusion, and “high-level” behavior implementation (e.g. crawl over an obstacle, hug ground, etc...). All of these processors communicate with other over a single serial communication bus.

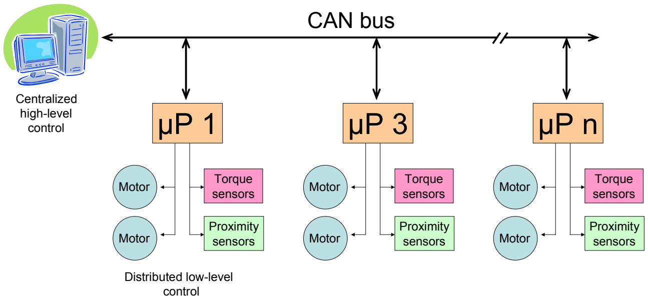

Fig. 5. SEQUENCE SHOWING OPERATION OF FOOT MECHANISM.

The bus used on this robot (Controller Area Network, CAN) provides an architecture that allows nodes to pass messages to any/all other nodes asynchronously, handling bus collisions that occur when several nodes attempt to drive the network simultaneously. The capabilities provided by this network allow for a variety of control algorithms to be developed and tested. The structure of this system accommodates both distributed and centralized control algorithms as well as various methods of fusing and acting on sensor data. Indeed, one of the main research objectives of this project is the development of a flexible platform capable of testing a myriad of control, networking, and sensor fusion algorithms.

The robot prototype is in final stages of construction and its functionality is being tested in simulation and verified in hardware.

IV. MULTI-DISCIPLINARY MODELS

Complex electromechanical systems, such as the robot described in Section III, often require simulation of their subsystems in order to verify operation or to tune performance. Partitioning such a system into its subsystems (e.g. plant dynamics, controller, communications, etc) reveals that very often, only some of these elements are tested exhaustively in simulation, and very rarely are subsystems cosimulated with one another. Simulation of the entire system (through cosimuation of all modeled subsystems) yields valuable insight into the performance of the actual device. Consider, for example, a snake-like robot that is controlled by multiple processors, all of which communicate over a single data bus. Cosimulation of the mechanism, distributed controller and data bus may reveal communications bottlenecks due to excess data traffic. This type of effect, though present in the physical system, could go undiscovered if a detailed cosimulation were not performed.

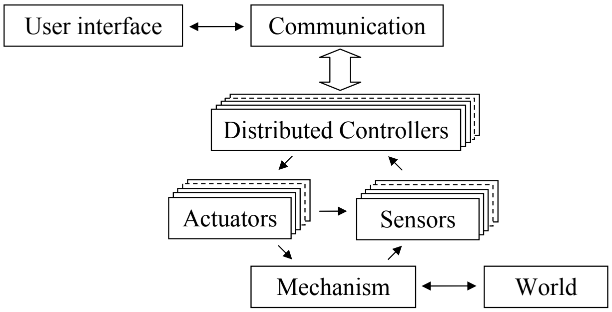

Fig. 6. DECOMPOSITION OF ROBOTIC SYSTEM INTO BLOCKS AND DATAFLOW.

A. System Decomposition

In an effort to formalize the development of a multi-disciplinary model of the snake-like robot, we first consider how we may describe the complete system as an interconnection of subsystems, each with a defined mechanism for communicating this other subsystems. To that end, we describe the robot by decomposing it into the elements shown in Figure 6. This figure represents a decomposition that is useful in understanding the operation of the physical device, and can also be applied to the development of a dynamic simulation of the complete robotic system. We now describe in greater detail, each of the subsystems shown in Figure 6.

Mechanism - represents the dynamics of the mechanical system.

Actuators - the input/output behavior of each actuator (torque and EMF constants, saturation and limiting effects, power dissipation, etc...).

Sensors - a model of each sensor's performance characteristics (noise and non-linearity, drift, saturation, etc...).

World - a description of the environment in which the system will be simulated. This includes terrain, obstacles, and environmental properties.

Distributed Controllers - the control algorithms that operate on distributed nodes and robot interface electronics.

Communications - the network over which distributed controllers communicate. This block captures channel access protocol, collision detection and bus arbitration, and bit-errors.

User Interface - describes the data acquisition and display element

B. Data Flow

The behavior of the robotic system is described by the coupled dynamics of the subsystems described above. In the physical system, coupling of these elements is manifest through interconnection of electrical signal (e.g. the controller driving a current through the motor, data moving over a network wire) and physical interaction between components (e.g. the mechanism resisting force applied from the motor, the ground colliding with the mechanism). In simulation, this interaction is accomplished by passing messages between software tools, each tool simulating a different subsystem. The simulation interval is divided into many small steps, and the simulation tools communicate with each other by passing data structures that represent the explicit and implicit interconnections in the physical system. With this method, we obtain a discrete time approximation to the coupled dynamics that arise when the physical system is operating.

The following section formally describes the interaction of the subsystems shown in Figure 6.

Distribute Controller → Actuators The controllers supply drive signals to the actuators (e.g. applies voltage to a motor) based on a desired behavior in the mechanism.

Actuators → Mechanism The drive signal received by the actuator is converted to a torque or force that acts on the mechanism.

Mechanism → Sensors Changes in the robot's configurations are passed to the appropriate sensor. For example, joint angles are passed to angular displacement sensors and contact forces are passed to tactile sensors.

Sensors → Distribute Controller The sensor's characteristics (accuracy, operating range, etc) operate on the parameters it senses from the mechanism and the sensor's output is passed to the controller.

Mechanism ↔ World While the mechanism responds to commands from the controller, its behavior is influenced by physical interaction with the world. The robot's motion may, for example, be inhibited by obstacles in the terrain.

Actuator → Sensors In some cases, parameters of the actuators may be measured directly and passed to appropriate sensors (e.g. measured motor current or torque).

Distribute Controller ↔ Communication In systems where control is distributed over several nodes, data generally must be passed between nodes to coordinate actions. This is accomplished by passing data to and receiving data from a communication network.

Communication ↔ User Interface The user interface has access to the communication network and exchanges messages with it in order to provide the operator with the ability to interface with the robot.

Data flow between subsystems occurs by passing messages from one subsystem. The message format specifies the source and destination subsystems, the type of data being passed, and the value of that data. Consider the communication that occurs in the closed loop system consisting of the controller, actuator, mechanism and sensor. Here, the following data types are defined:

Variables
voltage: $v \in [v_{\min}, v_{\max}]$
torque: $\tau \in [-\infty, \infty]$
real position: $x \in [x_{\min}, x_{\max}]$
real angle: $\alpha \in [\alpha_{\min}, \alpha_{\max}]$
measured position: $\hat{x} \in [-\infty, \infty]$
measured angle: $\hat{\alpha} \in [-\infty, \infty]$

Subsystems

controller: $c_i, i \in [1, l]$, where l = number of controllers
actuator: $a_j, j \in [1, m]$, where m = number of actuators
sensor: $a_k, k \in [1, n]$, where n = number of sensors
mechanism: m

These quantities are passed from one subsystem to another through messages. Shown below are the messages that are passed by the closed loop system consisting of controller, actuator, mechanism, and sensor. Here, *controller 1* drives *actuator 2*. The actuator applies a torque to the *mechanism*, and the mechanism's response is sensed by *sensor 5* which passes its measurement back to *controller 1*, closing the loop.

Format - {source, destination, var_type, value}

Distribute Controller → Actuators
$\{c_1$, a_2, voltage, 3 volts$\}$
Actuators → Mechanism
$\{a_2$, m, torque, 5 Nm$\}$
Mechanism → Sensors
$\{m$, s_5, real angle , 31.4 degrees$\}$
Sensors → Distribute Controller
$\{s_5$, c_1, measured angle, 30 degrees$\}$

V. MULTI-DISCIPLINARY MODEL APPLIED TO CASE STUDY

The framework outlined for creating multi-disciplinary models has been applied to the case study of Section III to yield a system level model of the robotic snake. This model has been used to perform cosimulation of the robot and controller with models for transducer saturation integrated. Figure 7 shows a simple experiment where the mechanism, whose dynamics were simulated in MSC/Adams®, is interfaced to a controller, modeled in Simulink®. By modeling the actuator and sensor non-linearities in Simulink®as well, we can integrate their effects into the framework.

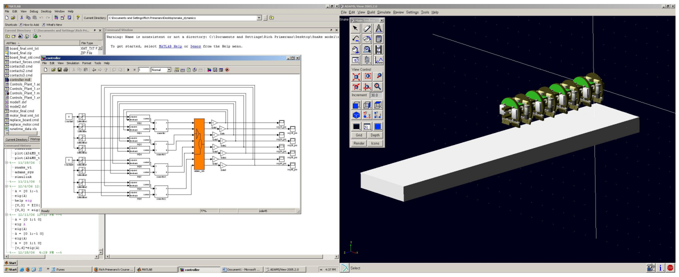

Fig. 7. COSIMULATION OF ROBOT MECHANISM AND CONTROLLER.

The elements of the robotic snake multi-disciplinary model are now discussed in greater detail.

Engineering CAD Model The starting point for simulation of the mechanism is the CAD model from which the robot is fabricated. This model, organized as a collection of assemblies, sub-assemblies and parts, contains a *complete* description of the mechanism, including full geometric detail, material properties, feature tolerances and any other information necessary to simulate, fabricate or assemble the mechanism. From this data structure, we can extract simulation models of varying levels of fidelity for use in dynamic simulation of the mechanism or cosimulation of the mechanism with other components in the framework (i.e. controller, actuators and sensors). To date, we have simulated the mechanism's dynamics using both MSC/Adams®and the Open Dynamics Engine (ODE) ^{TM}. Contact forces and friction effects are also modeled here.

Controller The control system used to actuate each of the robot's motors consists of a series of decoupled PID controllers. By interfacing this controller with the mechanism dynamics, we obtain a basic closed loop control system that captures the interaction of plant and controller. In this form, actuator and sensor non-linearities, and network traffic are not included in the model (see Figure 6). The actuator and sensor are modeled as ideal gain blocks, and the communication network is abstracted out.

Actuator Model Adding actuator dynamics and non-linearities is a simple matter of replacing the ideal gain model with a more realistic mathematical description of the device. The current model captures saturation effects of the motors used in the physical device.

A. *Comparison of Simulation and Experimental Results*

This section presents a simple experiment that compares the operation of the simulated robot and the physical robot while executing a basic maneuver. Here the simulation was carried out in ODE. At time zero, the robot is commanded to lift its head up and look to the left. The same command sequence is sent to both the physical robot and the simulated version. Figure 8 shows three frames take as the physical snake executed the maneuver.

Figure 9 shows the same three frames taken during the corresponding simulation run. We observe qualitatively similar behavior in both simulation and the physical device. The simulation correctly modeled the bending observed in segment three (the first segment touching the ground when the snake lifts its head). Though simple, this type of simulation is useful in determining if a particular configuration will put the snake on an unstable footing. The next step with this scenario will be to compare the force exerted by individual motors and feet on the real and simulated robots.

VI. FUTURE WORK

The work on this project in the coming months is divided into two areas. First, the hardware platform will be completed so that more extensive physical testing can occur. The robot will be outfitted with several types of sensors, including motor current draw and infrared object detection. These sensors give the robot the ability to perform object avoidance and stability monitoring tasks.

Second, a plug-in for Pro/ENGINEER®is under development to export a CAD model of the robot into an XML format containing joints, polygonal mesh geometries, and all relevant physical information. This model will be imported

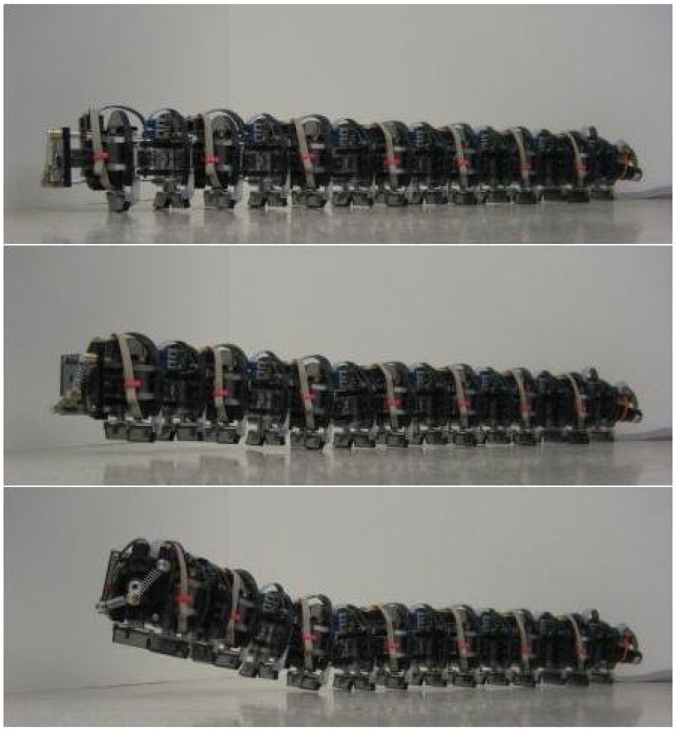

Fig. 8. Three frames taken while snake lifted head and looked left.

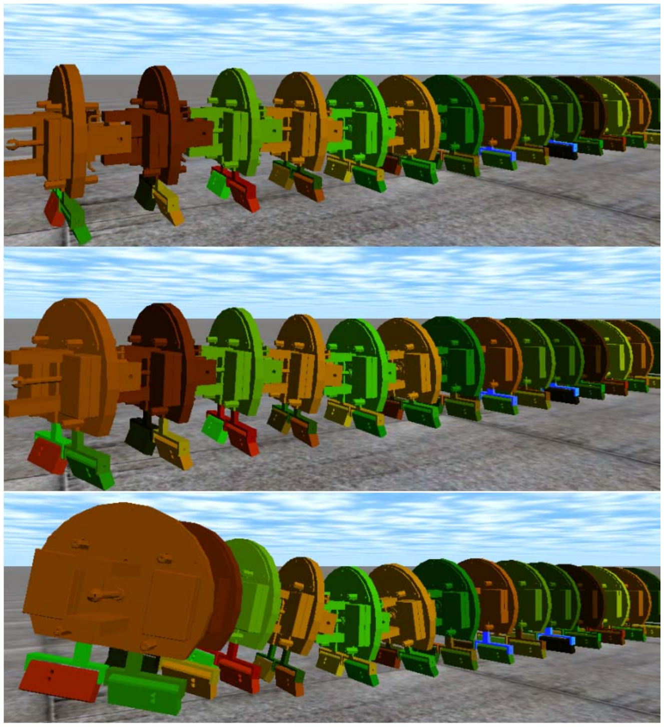

Fig. 9. Three frames taken while simulated snake lifted head and looked left.

into an open source simulation program in which models for its sensors, actuators, and controllers will be added. Simulations of various behaviors and missions will then be computed, and the simulated results will be compared to the robot's actual behavior.

VII. CONCLUSION

By applying a formal framework to the design of multidisciplinary models, we are able to develop a system level simulation of a complex electromechanical system that reflects the behavior of not just individual subsystems, but also the interaction of subsystems as they would operate in the final physical system.

VIII. ACKNOWLEDGMENTS

This work was supported in part by National Science Foundation (NSF) Office of Cyber-Infrastructure Grants OCI-0636235 and SCI-0537370. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or the other supporting government and corporate organizations.

REFERENCES

[1] e. Dabke, P., "Using features to support finite element idealizations," in *ASME Computers in Engineering Conference*. Minneapolos, MN: ASME, September 1994.

[2] A. Sheffer, T. Blacker, and M. Bercovier, "Clustering: Automated detail suppression using virtual topology," in *Trends in Unstructured Mesh Generation*. ASME, July 1997, vol. AMD-Vol. 220, pp. 57–64.

[3] e. Arora, N. D., "Modeling and extraction of interconnect capacitances for multilayer vlsi circuits," *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol. 15, no. 1, December 1996.

[4] R. e. Piertabissa, "A lumped parameter model to evaluate the fluid dynamics of different coronary bypasses," *Med. Eng. Phys.*, vol. 18, no. 6, pp. 477–484, 1996.

[5] R. D. Kangwai, S. D. Guest, and S. Pellegrino, "An introduction to the analysis of symmetric structures," *Computers and Structures*, vol. 71, pp. 671–688, December 1999.

[6] K. Suresh, "Automating the cad/cae dimensional reduction process," in *Eighth ACM Symposium on Solid Modeling and Applications*. Seattle, WA: ACM, June 2003.

[7] Suresh, K., "Generalization of the mid-element based dimensional reduction," *ASME/ACMJournal of Computing and Information Science and Engineering*, vol. 3, no. 4, December 2003.

[8] Featherstone, R., and Orin, D. E., "Robot dynamics: Equations and algorithms," in *IEEE Int. Conf. Robotics and Automation*, 2000, pp. 826–834.

[9] Featherstone, R., "Robot Dynamics Algorithms," 1987, Kluwer, Boston, MA.

[10] R. Featherstone, "A divide-and-conquer articulated body algorithm for parallel o(log(n)) calculation of rigid body dynamics. part 1: Basic algorithm." *International Journal of Robotics Research*, vol. 18(9), pp. 867–875, 1999.

[11] Featherstone, R., "A divide-and-conquer articulated body algorithm for parallel o(log(n)) calculation of rigid body dynamics. part 2: Trees, loops, and accuracy." *International Journal of Robotics Research*, vol. 18(9), pp. 876–892, 1999.

[12] Baraff, D., "Linear-Time simulation using lagrange multipliers," in

SIGGRAPH 96 Conference Proceedings, H. Rushmeier, Ed. Addison Wesley, 1996, held in New Orleans, Louisiana, 04-09 August 1996.

[13] H. Kimura, S. Hirose, and K. Shimizu, "Stuck evasion control for active-wheel passive-joint snake-like mobile robot'genbu'," *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 5.

[14] J. Ostrowski and J. Burdick, "Gait kinematics for a serpentine robot," *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, vol. 2, 1996.

[15] E. Paljug, T. Ohm, and S. Hayati, "The JPL Serpentine Robot: a 12-DOF system for inspection," *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, vol. 3, 1995.

[16] G. Migadis and K. Kyriakopoulos, "Design and forward kinematic analysis of a robotic snake," *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 4, 1997.

[17] R. Featherstone and D. Orin, "Robot dynamics: equations and algorithms," *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 1, 2000.

[18] M. Hardt and O. von Stryk, "The role of motion dynamics in the design, control and stability of bi pedal and quadrupedal robots," *RoboCup 2002 International Symposium*.

[19] I. Tanev and T. Ray, "Evolution of side-winding locomotion of simulated limbless, wheelless robots," *Artificial Life and Robotics*, vol. 9, no. 3, pp. 117–122, 2005.

[20] M. McKenna and D. Zeltzer, "Dynamic simulation of autonomous legged locomotion," in *SIGGRAPH '90: Proceedings of the 17th annual conference on Compute r graphics and interactive techniques*. New York, NY, USA: ACM Press, 1990, pp. 29–38.