Intelligent Control and Automation, 2011, 2, 133-143
doi:10.4236/ica.2011.22016 Published Online May 2011 (http://www.SciRP.org/journal/ica)

Identification of Issues in Predicting Multi-Robot Performance through Model-Based Simulations

Shameka Dawson, Briana Lowe Wellman, Monica Anderson
Department of Computer Science, *The University of Alabama*, *Tuscaloosa*, *USA*
E-mail: {*dawso*003, *lowe*002}@*crimson.ua.edu*, *anderson@cs.ua.edu*
Received April 28, 2011; *revised May* 4, 2011; *accepted May* 9, 2011

Abstract

Predicting the performance of intelligent multi-robot systems is advantageous because running physical experiments with teams of robots can be costly and time consuming. Controlling for every factor can be difficult in the presence of minor disparities (*i.e.* battery charge). Access to a variety of environmental configurations and hardware choices is prohibitive in many cases. With the eminent need for dependable robot controllers and algorithms, it is essential to understand when real robot performance can be accurately predicted. New prediction methods must account for the effects of digital and physical interaction between the robots that are more complex than just collision detection of 2D or physics-based 3D models. In this paper, we identify issues in predicting multi-robot performance and present examples of statistical and model-based simulation methods and their applicability to multi-robot systems. Even when sensor noise, latency and environmental configuration are modeled in some complexity, multi-robot systems interject interference and messaging latency, causing many prediction systems to fail to correlate to absolute or relative performance. We support this supposition by comparing results from 3D physics-based simulations to identical experiments with a physical robot team for a coverage task.

Keywords: Intelligent Robots, Multi-Robot Systems, Performance, Prediction, Simulation

1. Introduction

Simulations are an important component of software validation. Robot controllers are tested within a simulated environment to verify properly coded semantics. Simulation is often used to predict controller performance under a set of constraints. Specifically, the environment (placement of obstacles, walls, etc.) is varied along with the robot configuration to quantify performance under a more generalized set of parameters. Prediction is then generated by gathering average case performance data from simulation experiments. Simulations can provide both quantitative and qualitative data which are used to model robot performance in the real world.

Getting simulations to predict performance in single robot experiments is challenging. Models do not always capture important factors such as sensor to environment interactions [1,2] and hardware inconsistencies [3,4], causing simulations to perform better or worse than in the real world. This effect is magnified when using more than one robot in a team. Prediction of multi-robot performance in simulation is particularly important because of the time and cost involved in acquiring, maintaining, and running multiple robots. If simulations are to be predictive, it is important to understand when simulations accurately predict multi-robot performance.

There is a renewed interest in prediction through simulations which is evident by the emergence of new conferences such as Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR). SIMPAR was first introduced in 2008 to "identify and solve the key issues necessary to ease the development of increasingly complex robot software and to boost a smooth shifting of results from simulated to real applications" [5]. It is rare that there is a seamless migration of code from simulators to real world systems because of the complexity of modeling mechanics (sensors) and interactions with real environments.

In this article, we investigate the factors that affect the ability of statistical and model-based simulation to predict multi-robot performance. Specifically, we survey the factors that affect the accuracy of multi-robot simulations (Section 2). We discuss both statistical and model-based methods of predicting performance in multi-robot teams

(Section 3). In Section 4, we examine validation techniques in a recent robotics conference. A comparison between robotics experiments and physics-based simulations is presented in Section 5. The article concludes with Section 6.

2. Using Simulations to Predict Performance

Many researchers use a behavior-based decomposition [6] that layers task achieving behaviors such as obstacle avoidance and wander. A behavior-based robot is designed to operate in dynamic environments because its reactions are determined by what is sensed without necessarily modeling the environment structure that provides the sensor readings. As it senses the environment, it computes what it senses, and acts on what is computed (see **Figure 1**). This structure often includes higher-level behaviors that are not purely reactive that model and hold states such as mapping and path planning.

Uncertainty is introduced in the behavior-based model in several ways: sensor noise, latency, and the environment. Sensor noise is a random error that causes sensor readings to vary against the expected values. Since the robot uses sensor data directly to calculate actions, sensor noise can affect the choice of appropriate behaviors.

Latency is the delay between sensing and acting. In a mobile robot system, latency is injected because of the delay between sensing, computation time, and execution time. Latency can affect performance if robot behaviors are not executed in a timely manner. In addition, as the environment's lighting and surfaces change, sensors interact differently, often in unmodeled ways. Melhuish *et al.* [1] performed a patch sorting study in both simulation and a real multi-robot system using minimalist robots. Simulation produced better performance than real robot experiments, primarily because the actual infrared sensors encountered difficulties in changing lighting conditions.

Multi-robot experiments include additional factors such as interference, message loads and communication bandwidth that affect performance. Interference occurs when multiple robots try to occupy the same space causing robots to expend time and energy maneuvering around each other. Message passing (via a network) can also affect performance by requiring computational resources to produce and process data.

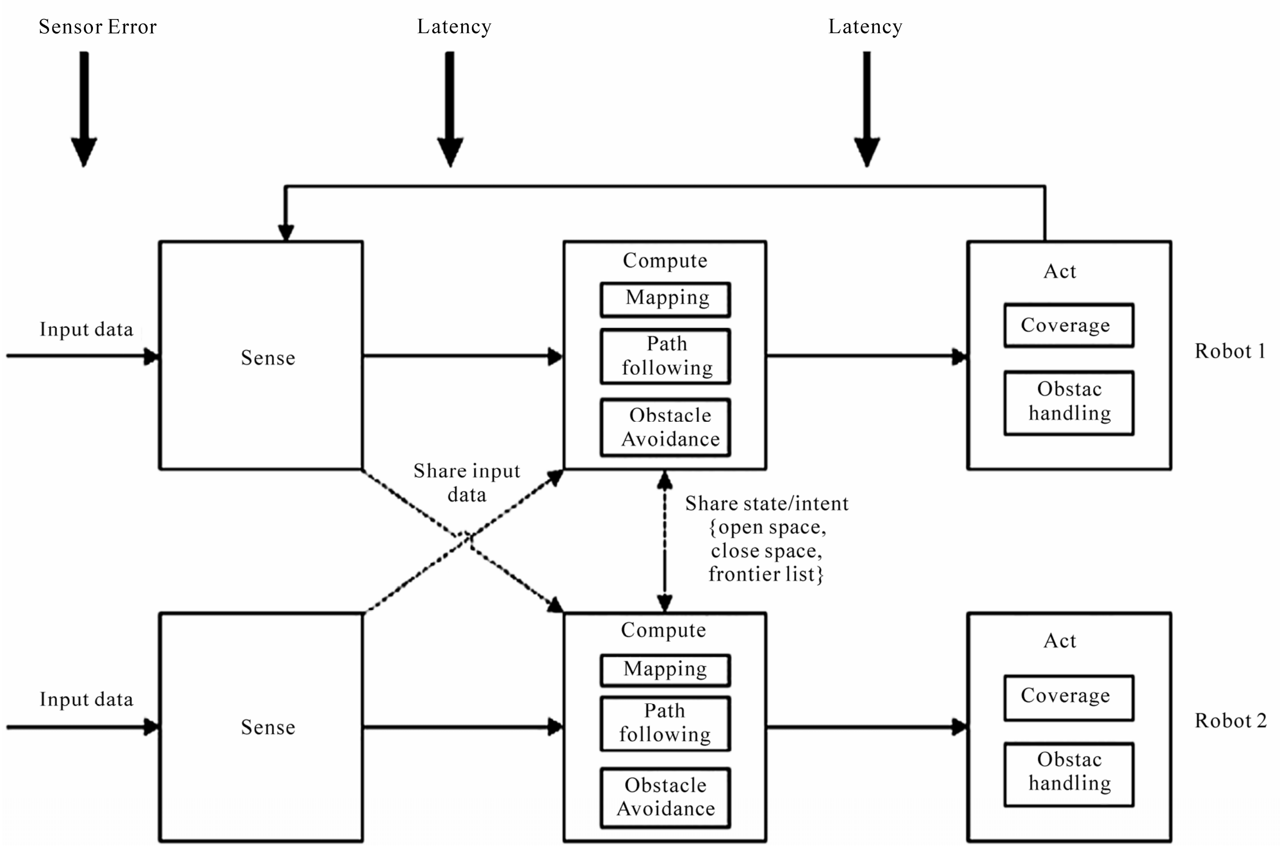

Figure 1. Intelligent mobile robots are traditionally structured using a behavior-based approach. When multiple robots interact, new sources of discrepancy, such as message production and processing, can cause simulations to vary from experiments.

2.1. Sensor Noise

Prior work indicates that researchers are concerned with how sensor noise affects the correlation between simulation and physical experiments. This is because sensor data is noisy and the range, reflectance, and other parameters of real sensors are limited [7]. For instance, Balch and Arkin [8] conducted experiments in formation control of multiple robots in simulation and with real robots. They determined that the differences between the experiments were primarily due to sensor noise and positional inaccuracies.

In a task allocation experiment [9], Mataric found that the exact behavior in a nondeterministic world is impossible to predict exactly because it is subject to real error and noise. In other studies [10,11], experiments were performed using multiple coordinating robots using different task allocation strategies focusing on noise and uncertainty. They showed that no single strategy produces the best performance in all cases and that the best strategy changes as a function of noise in the system. Other researchers found that too little, too much, or inaccurate noise in simulation creates unrealistic or non-transferable systems [12,13].

In some instances, sensors are dependent upon each other. In [14], Meeden found that there is a correlation between certain sensors. A hybrid model was developed that combined both independent and dependent sensor noise to account for different amounts of correlation. They used a Khepera robot with light sensors to test their model. They found that real world noise is not independently random for certain sensors but displays synchrony. Their results imply that the hybrid noise model transfers better from simulation to the real world than an independent noise model.

In [15,16], a description and evaluation of mobile robot motion in simulation was presented. It was found that problems in robot motion arise due to unexpected uncertainty of motion and sensor error. It was noted that some simulators pay little attention to the fact that uncertainty is inevitable.

2.2. Latency

Latency is another factor that many researchers neglect when modeling. In one study, Balch and Arkin [8] present reactive behaviors using formations in robot teams in simulation, on real robots, and Unmanned Ground Vehicles (UGVs). They determine that latency and position error in transmission of positional information can negatively impact performance. They show that in simulation there was no position error or communication latency, while experiments on real robots and UGVs experienced 1 s and 7 s communication latency, respectively.

Go *et al.* [17] present a simulation framework for vision-centric robots and conjecture that a key element of simulation is latency modeling. However, they state that the effects of latency are often ignored in simulations. They go on to state that unaccounted latency in simulation sensing and actuation leads to differences in simulated and real results. Likewise, in a recent paper, Seo *et al.* [16] mention that simulators do not consider uncertainties in latency.

2.3. Environment

The environment and robot-environment interaction are hard to accurately model in simulation. Brooks [18] states that there is a vast difference between simulated and real robots and their interaction with the environment. It is also noted that programs that work on simulated robots may fail on real robots because of differences in real world sensing and actuation because it is hard to simulate dynamics of the real world.

Gat [19] states that there is an inadequate basis for predicting the reliability and behavior of robots operating in unengineered environments. Interaction with complex environments is difficult to model because of independent variables which are often ignored. In addition, design of control systems for robots operating in complex, dynamic, uncertain environments will become more difficult as the complexity of behaviors increases [20].

In an experiment on tracking targets [21], it was discovered that the characteristics of the environment affect system performance. For instance, the shape of the environment and how obstructed it is shown to be significant. Furthermore, Smith [22] stated that robot interaction with walls in an environment is extremely hard to model accurately. This may be a result of different surface properties of walls.

Rosenfeld *et al.* [23] determined that the physical environment where robot teams operate pose challenges for robots to perform properly. They created adaptive coordination techniques and found that techniques should be adjusted to match different environmental conditions.

In experiments on cooperation strategies [24], it was determined that the environmental configuration had the greatest impact on the speed and success of robot search. Balaguer *et al.* [25] state that robots are complicated systems composed of interacting units, each characterized by its own behaviors and errors and that robots' observed behaviors depend on the environment along with the software and hardware.

Balakirsky *et al.* [26] determined that a combination of robot parameters, diverse terrain, and high variability in sensor readings and error rates create dynamic environments that are hard to accurately replicate in simulation. They state that the inherent complexities of robotic environments may introduce significant differences between real and simulated environments. They go on to say that algorithm development on a simulation assumes that information about the environment is accurate, but the complexity of the operating environment can be daunting and variables about terrain characteristics may be omitted or ignored.

2.4. Interference

In multi-robot experiments, robot interference has a major impact on performance. Gerkey and Mataric [27] suggest that a common externality in multi-robot systems is physical interference. They state that interference is often ignored or crudely modeled when estimating utilities, but have complex and unpredictable effects that may easily dominate performance. Other researchers [28,29] also determined that the number of robots has an important impact on system performance due to physical interference.

In a discussion on collective agents [9], it was found that interference increases as the size of the group grows which causes a decline in global performance. They state that the global consequence of local interaction between robots is difficult to predict.

Martinoli and Mondada [30] performed parametric simulations and real experiments for a clustering task using multiple robots. They found that the main difference is that the performance of a team of three real robots is less rapidly saturated than in simulation. In experiments with more robots, there was a substantial sublinearity because of interference and team fitness becomes saturated because of interference.

Rosenfeld *et al.* [31] studied how the productivity of robots scales with group size in a foraging experiment. They found a negative correlation between group productivity and interference using 1 to 30 robots in simulations. They show that various coordination methods affect the productivity of team performance.

An investigation on multiple robots in odor source localization in simulation was performed by Lochmatter and Martinoli [32]. In their study, they compare performance of a single robot to that of a group of 2 or 5 non-cooperating robots. While they expected the multi-robot experiments to perform significantly better than the single robot experiments, they found some of the results comparable. In addition, they found a significant difference in one algorithm where performance decreased as the number of robots increased. They conclude that the loss in performance result from interference amongst robots.

2.5. Message Passing and Communication Bandwidth

When multiple robots cooperate, message passing and communication bandwidth may impact system performance. Since robot cooperation often requires communication, the bandwidth can grow with the number of robots [9]. Rybski *et al.* [33] show that limited communication bandwidth constricts the effective team size in a surveillance task using real miniature robots. They state that performance depends on the number of robots that share the bandwidth and that the system degrades under increased loads.

Lerman *et al.* [34] determined that as the size of a multi-robot system grows, the complexity of design approaches also increases. This increase is due to increased communication bandwidth and computational abilities of robots.

In [24], coordination strategies were examined using simulation and physical experiments. They show that explicit coordination methods decrease performance as the number of robots increase because of the limited communication bandwidth and computational requirements when dealing with multiple robots. They also determined that message exchange affects performance and scales with team size. They suggest that methods that rely on reliable network connections have limited applicability in the real world.

3. Predictive Models of Behavior-Based Controllers

There are two kinds of predictive models in use: statistical analysis and simulated. Statistical models consider the transitions between states. Simulated models use representations of robots and the environment to trace execution of particular controllers. Simulations can be numerical or can use a specialized package.

3.1. Statistical Models

Lerman and Galstyan [29] presented a mathematical model of a group of robots in a foraging task. Robots searched an enclosed obstacle-free area to retrieve pucks. The foraging task consists of five states (see **Figure 2**): search for pucks, collect pucks, go home, reverse homing, and avoid collisions. For analysis, the behavior is simplified to two states: searching (including searching and collecting pucks) and avoiding.

Using the rates of detecting a puck, α_p, or another robot, α_r, the number of robots in each state could be calculated

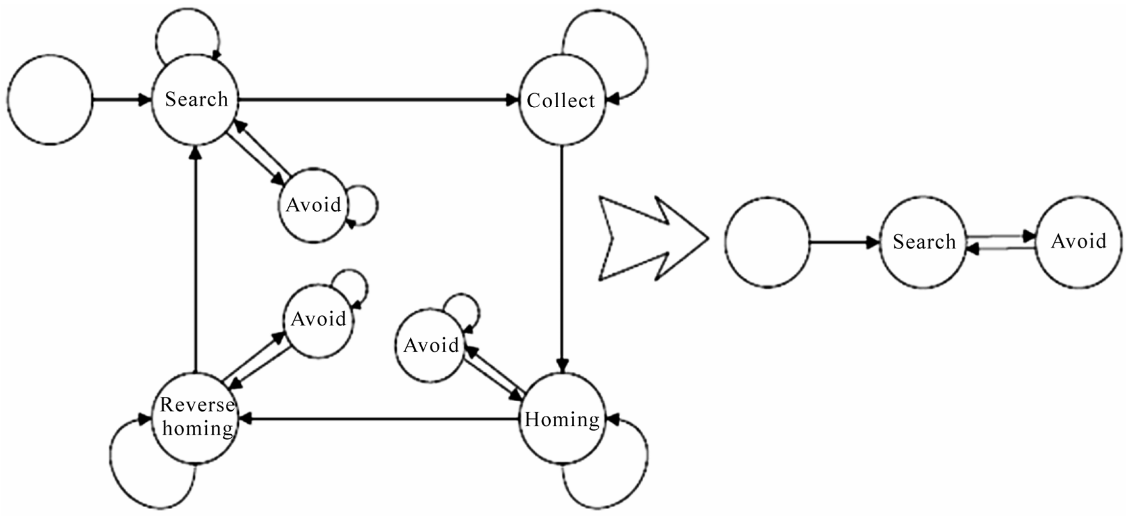

Figure 2. Lerman and Galstyan simplify the behavior-based puck collection to two states for analysis.

and the collection of pucks could be predicted. Interference, modeled as

$$\frac{dN_s(t)}{dt} = -\alpha_r N_s(t)\left[N_s(t) + N_0\right] + \alpha_r N_s(t-\tau)\left[N_s(t-\tau) + N_0\right] \quad (1)$$

describes robots that detect another robot in searching state or in the avoid state and begin the avoiding maneuver where τ is the avoiding time period if they detect an obstacle. $N_s(t)$ is the number of robots in the search state at time t, $N_a(t)$ is the number of robots in the avoid state at time t, and N_0 is the total number of robots. The collection of pucks is modeled by

$$\frac{dM(t)}{dt} = -\alpha_p N_s(t) M(t) \quad (2)$$

where $M(t)$ is number of uncollected pucks at time t.

From these equations, both efficiency (the time it takes to collect 80% of the pucks) and interference (amount of time spent avoiding) is calculated. They validate their model by comparing its predictions to results from Player/Stage [35]. They showed that while increasing group sizes reduces task completion time, the improvement is only sub-linear and the individual robot's performance is a monotonically decreasing function of group size. They found that interference can significantly affect group performance and that there was an optimal group size that maximizes team performance.

Lerman and Galstyan state that their model agrees with the simulations. However, they used probabilities for situational events specific to the environment which are best identified empirically. They note that the model depends only on present state and not past states and cannot take into account complex decision making.

3.2. Model-Based Simulation Packages

Specialized packages either rely upon kinematics or physics-based simulations. A kinematics simulation focuses on motion without reference to the force or mass that causes it. Kinematics simulators are usually fast because they use ray tracing for collision detection and sensor modeling [36]. The simulation is rendered into a grid and collisions between blocks in the grid and the sensor data are computed using ray tracking. Ray tracing involves using a line-drawing algorithm to determine if a ray intersects a block in the grid.

Physics-based simulations take force and mass into consideration and produce higher fidelity simulations. They often use a physics engine, such as Open Dynamics Engine (ODE) [37] that consists of a rigid body dynamics simulation engine and a collision detection engine. The physics engine allows the simulation of properties such as friction, velocity, and mass. While kinematics simulators are faster, an advantage of physics-based simulations is that they are thought of as more accurate and may prevent physically inaccurate situations due to inconsistencies between the real and simulated world [38]. However, physics-based simulations still include minor differences that can accrue over time and result in different behaviors from the real world [39].

Robot and sensor models are crucial to accurate simulation because they provide the basis for robot perception. Robot models detail many aspects of the physical robot that it is modeling, such as specifications about the robot (mass, friction, size, etc.). Since sensor output includes some amount of error, sensor models often include uncertainty and are adjusted to different levels of noise [16]. **Figure 3** shows how a Pioneer robot is modeled in different simulators. We present representative (not exhaustive)

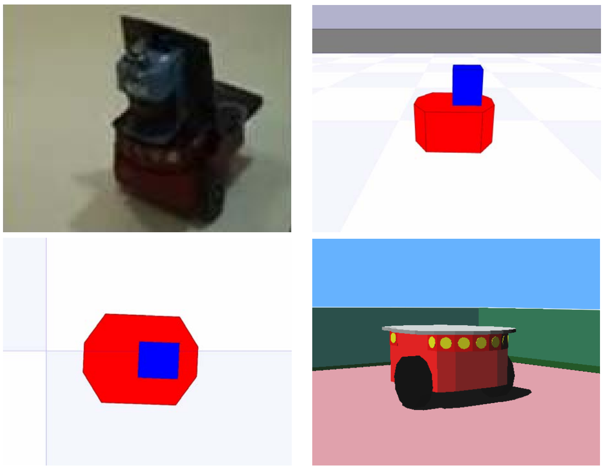

Figure 3. Images of a pioneer robot model in reality (top left), stage (top right), stage from top view (bottom left), and webots (bottom right).

set of packaged simulators and discuss their approach to modeling interaction.

3.3. Stage

Stage [36] is a 2.5D, kinematics simulator that runs on UNIX-like platforms. It is an open source, community free software that simulates large populations of robots. As of January 2010, Stage has been downloaded 60,337 times. Stage also interfaces with Player [35], a robot device server, to allow for easy transfer to real robots. It is aimed at being efficient and configurable rather than highly accurate. So, it provides simple, computationally cheap models of robots (modeled as polygon blocks) and devices, such as various sensors and actuators, that are more general than actual hardware.

Stage provides fairly coarse-grained sensor models. For instance, odometry (used to measure position based on integrating wheel movements over time) models error, E, for x, y, and Θ by choosing a value from $-E/2$ to $E/2$ at startup for use during the lifetime of the simulation. For collision detection, ray tracing is used to compute collisions between blocks and the range sensor data. It uses the velocity of a moving object and compares its current position to its next position using the *updatePose* function. It then reports a collision if ray tracing determines an intersection with another object in that range using the *TestCollision* and *Raytrace* functions. However, it does not detect collisions on the z-axis. Collision detection is reported to be accurate to 0.02 m which is the spatial resolution of the ray tracing engine.

3.4. Gazebo

Gazebo [40] is a 3D multiple robot simulator with dynamics that runs on UNIX-like platforms. Gazebo is also Player compatible. It is free software designed for outdoor environments and is capable of simulating a small population of robots with high fidelity. As of January 2010, Gazebo has been downloaded 22,329 times. It attempts to generate realistic sensor feedback based on provided parameters on lighting, surface reflectance, and friction. Rigid body physics allow robots to interact with objects based on provided robot/sensor models.

Movement error is modeled by friction and slip noise, adjusting through the *mu1* and *slip* parameters. Gazebo uses ODE to simulate collision detection. The *RaySensor* and *RayGeom* classes are used to cast rays, test for intersection, and report the range to an object. *Geoms* (types of geometries) are associated with objects to get the position and orientation from the *geom* to the object.

3.5. USARSim

USARSim [41] is an open source, high fidelity simulator intended for research in human-robot interaction and multi-robot coordination. It is platform independent and runs on Windows, Linux, and Mac OS. USARSim builds upon the widely used game engine, Unreal Engine. It provides 3D rendering and physical simulation. It features the simulation of multiple sensors and actuators.

USARSim uses the Karma physics engine for collision detection. Like other physics-based simulators, it uses mass, friction and linear and angular damping to actuate objects against external and internal forces. Karma physics engine does not document any specific parameters for interjecting error into a simulation.

3.6. Webots

Webots [42] is a 3D, physics-based mobile robot simulator that is both kinematics and physics-based. Webots is a commercial product that can run on multiple operating systems such as Windows, Linux, or Mac OS X. Webots is used by more than 700 universities and research centers worldwide. Webots allows users to define and modify robotics setup and define properties (texture, friction, mass, etc.) to objects. It also allows users to import their own 3D models and create complex environments using OpenGL technology.

The ODE library is utilized to create accurate physics simulations. The differential wheel model in Webots is used to represent any robot with two wheel differential steering. The model include *encoderNoise* which adds noise to the incremental encoders (counters that increment each time a wheel turns) and *encoderResolution* which defines the number of encoder increments per radian. The actual speed is computed from the angular speed of each wheel, the wheel radius, and the noise. Webots uses ODE for collision detection where components of a robot are associated with a bounding object. The bounding object defines the shape used for collision detection. A ray casting algorithm is used to detect collisions between a sensor ray and a *Solid* node (a group of shapes) where the intersection between two bounding objects is calculated. Force is then generated upon contact on the solids.

4. Experiments in Robotics

In order to better understand the type of experiments that are being implemented, an examination of experimental methods used in papers from the 2010 IEEE International Conference on Robotics and Automation was performed. The methodology for counting the experiments consisted of examining the papers in the conference proceedings to determine the type of experiments the researchers performed. The findings from this inquiry are presented in **Figures 4** and **5**.

The experiments were broken into four categories: real, simulation, dataset, and combination. The real experiments were those implemented with actual hardware or robot platforms, which included real robots, manipulators, etc. The simulation category consisted of experiments conducted with a simulator or theoretical or mathematical modeling. In addition, some cases used a combination of both real and simulated experiments. The last category is where datasets of real or simulated data obtained from data set repositories were used.

While there are many known issues with simulations, we found that they are still a primary means of validation. Approximately 29% of researchers still rely on only simulated results. However, more than 50% of multi-robot experiments were conducted in simulation.

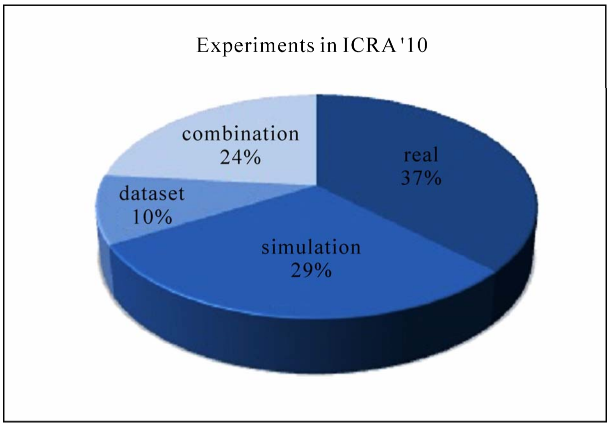

Figure 4. Validation approach in a 2010 robotics conference for all papers.

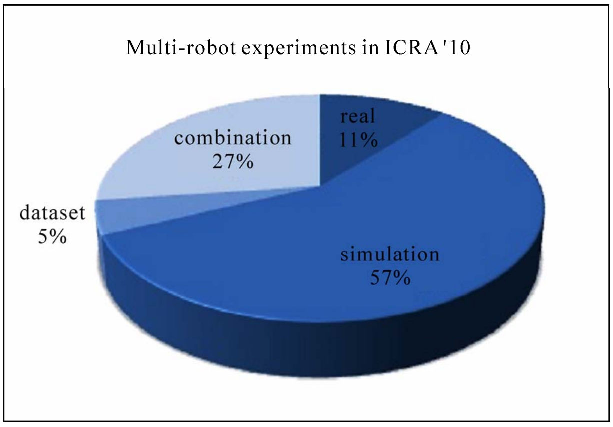

Figure 5. Validation approach in a 2010 robotics conference for multi-robot papers.

5. Using Simulations to Predict Experimental Performance in an Exploration Task

Our research focuses on exploring predictive models based on simulation results within multi-robot experiments. Certainly sensor error, latency and environment all affect team performance since team performance is an aggregation of individual performance in some ways. However, we conjecture that there are important factors specific to multi-robot systems that affect performance in real robots differently than in simulation.

The predictability of physics-based simulations for multi-robot coverage tasks was summarized in [43] and [44]. In this article, we expand the presentation of results and analysis. To understand the impact of each factor, we compared performance in simulations and real experiments using different environmental configurations and cooperation paradigms in a coverage task. Robots each perform frontier exploration [45] where none of the robots know the topology of the environment a priori. Teams can either communicate progress in the form of areas that have been explored, Direct Comm, or perform the exploration without knowledge of the actions or findings of other robots, No Comm. The control program, written in the C, was essentially the same for both the simulated and real experiments.

K-team Koala robots were used for the physical experiments. The robots were equipped with a Hokuyo URG laser range finder with a range of 2 m. Also, a Hagisonic StarGazer Localization System was used to mitigate sensor error. The robots were also equipped with a Dual Core 1.6 GHz machine running Ubuntu with 2 GB of RAM.

The simulation environment used was Webots [42], a 3-D physics-based mobile robot simulator. The robots used global positioning sensor (GPS) for localization as well as a laser range finder with a 2 m range. The simulations were performed on a Dual Core 2.33 GHz Linux machine with 2 GB of RAM. A wheel encoder noise (based on a Gaussian distribution) was added in simulation to compensate for error in the real world.

Average coverage times over five runs for the real experiments and 20 runs for simulation are presented for a three robot team (see **Table 1**) in six environments (see **Figure 6**). The environments were chosen to represent different types of outdoor areas. The robot speeds, environmental configuration and controller programs were identical between simulation and experiments. We modeled sensor latency (based on empirical testing) and sensor error (percentage determined by empirical testing).

In terms of prediction, ideally the simulations would predict the amount of time needed for coverage. The simulations completed on average 1.5 times faster than the experiments. However, prediction can be useful if we can deduce relative performance in comparisons. Unfortunately, relative performance is different between simulation and experiments. For example, in environment 2 the simulations show that the cooperation paradigm has little effect on the time-to-cover. However, in experiments, the lack of cooperation paradigm causes the time-to-cover to increase by 50%. In addition, after performing a t-test, we found that there was a statistical difference (p = 0.041) between the time-to-cover results from the simulation and experiments in the No Comm experiments. So, to really understand the predictive ability of physics-based simulations in this multi-robot task, we must consider interference and message processing individually.

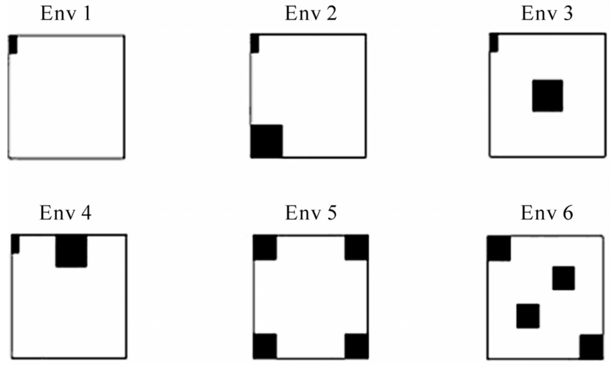

Figure 6. Six 6 m × 6 m environmental configurations used in a coverage task.

Table 1. Average time to complete 90% coverage (in sec).

Env	No Comm			Direct Comm		
	Real	Sim	Diff	Real	Sim	Diff
1	213.8	107.5	106.3	220.3	99.5	120.8
2	185.8	93.5	92.3	121.8	80.5	41.3
3	230.1	182.5	47.6	149.1	86.2	62.9
4	261.8	202.5	59.3	212.1	188.0	24.1
5	132.6	86.5	46.1	64.3	40.5	23.8
6	220.1	180.0	40.1	133.6	88.0	45.6

5.1. Interference

A summary of interference results are in **Tables 2** and **3**. Interference in real experiments occurs more frequently and lasts longer than in simulation. No Comm experiments resulted in more interference than Direct Comm experiments. Moreover, the time-to-cover difference between real and simulation experiments correlates to total time interfering (r = 0.77). In Direct Comm experiments, interference within real experiments is reduced, although simulated interference is not considerably different between the two paradigms. The difference between the time-to-cover for the real and simulated results is uncorrelated to total time interfering (r = 0.0645) when interference is managed with communications based cooperation. These findings suggest that unmodeled interference can affect how well simulations approximate performance of multi-robot experiments. If we consider environmental configuration, simulation in open environments was found to be less predictive than in more cluttered environments which has implications for experiments of outdoor environments.

5.2. Message Passing

Interference can be reduced through a cooperation paradigm that reduces the likelihood that robots will attempt to occupy the same space. However, even if interference is low (Direct Comm in environments 2 and 3), physics-based simulations are still not predictive of experimental results. An additional factor to consider is the inconsistency of message latency. **Table 4** shows the latency between two sets of experiments. Low Message Volume communicates information between robots only when no previous communication has addressed the newly found area. High Message Volume updates information on an area whenever that area is encountered, producing more messages that should adversely affect latency. Although average latency is not drastically different, the variance in latency is much larger for real experiments. Simulation does a poor job of reproducing the inconsistency in latency that can affect performance in real robots. **Table 5** shows that time-to-cover in the real robot experiments is longer and all the real experiments experience much higher variance.

Table 2. Average number of times the robots interfered with one another and the average duration of each occurrence for No Comm (in sec).

Env	Number of Occurrences				Time per Occurrence				Interference Time	
	Sim		Real		Sim		Real		Sim	Real
	μ	σ	μ	σ	μ	σ	μ	σ		
1	2.1	1.21	5.2	1.79	6.63	3.97	16.87	8.61	15.4	78.4
2	1.7	1.26	3.6	1.82	6.53	6.15	13.33	9.11	12.8	46.2
3	1.8	1.28	2.0	1.00	7.25	8.54	11.57	2.59	14.0	24.0
4	1.5	1.79	3.8	3.77	3.54	3.93	12.38	8.93	9.9	63.0
5	1.4	1.35	5.4	3.44	8.74	15.88	8.06	6.03	13.2	28.8
6	1.3	1.16	1.6	1.95	15.25	17.95	6.48	7.34	22.6	10.0

Table 3. Average number of times the robots interfered with one another and the average duration of each occurrence for Direct Comm (in sec).

Env	Number of Occurrences				Time per Occurrence				Interference Time	
	Sim		Real		Sim		Real		Sim	Real
	μ	σ	μ	σ	μ	σ	μ	σ		
1	0.15	0.37	0.6	0.55	0.9	2.99	6.6	11.52	0.9	6.6
2	0.20	0.41	0.2	0.45	0.7	2.30	0.6	1.34	0.7	0.6
3	0.90	0.97	0.6	0.89	3.9	4.45	0.5	0.71	6.3	0.8
4	0.05	0.22	0.4	0.55	0.3	1.34	7.8	11.63	0.3	7.8
5	0.45	0.51	0.4	0.55	1.8	2.44	1.4	2.19	1.8	1.4
6	0.35	0.59	0.4	0.55	0.9	3.21	18.8	28.72	0.9	18.8

Table 4. Comparision of message latency (in sec).

	Low Message Volume	σ	*High Message Volume*	σ
Sim	1.45	0.15	3.25	0.07
Real	1.79	2.02	3.62	2.72

Table 5. Comparison of average 50% and 90% coverage times for low and high message volumes (in sec).

Message Volume	Sim				Real			
	50%	σ	90%	σ	50%	σ	90%	σ
Low	12.32	2.54	24.52	7.75	48.58	22.77	168.05	77.61
HIGH	11.97	2.26	26.07	8.57	89.64	17.28	202.06	31.26

6. Conclusions/Future Work

This paper identifies issues related to predicting multi-robot performance as well as methods for predicting robot performance using both statistical analysis and simulators. Although simulations are advantageous because they are a fast and cost efficient way of performing robot experiments, we show that simulations can be affected by both interference and message passing in ways that cause simulation results to fail to predict either absolute or relative performance in physical robot teams.

For future work, we plan to propose a model that accounts for and mitigates some of the issues illustrated in this paper. In particular, we plan to better model robot interaction in simulation. We will explore the idea that significant discrepancy between simulated and real coverage experiments results from physical interference between robots. The methodology for this research is to use a frontier-based algorithm for coverage where a team of robots recursively explores an unknown area while building a cellular representation.

We anticipate that the exploration algorithm might create different forms of interaction between robots. Since each robot maintains its own individual list of frontiers to explore, robots may choose to explore the same frontier thus competing for long durations of time. Conversely, robots may choose to search adjacent areas and only interact with other robots in passing for a shorter length of time. Robots may also encounter other robots directly and try to avoid each other if they are on the same path but headed in different directions.

Therefore, we are more precisely modeling different forms of physical robot interaction. Interaction can be categorized as competing and passing. We define competing interaction as robots trying to occupy the same space when they have proximal goals. Passing interaction is when robots briefly interact with each other when trying to approach different goals. We also plan to quantify how obstacles either assist with cooperative coverage (Environment 6) or hinder cooperative coverage (Environment 4). Specifically, we plan to focus on three types of environments: open areas, convex environments, and concave environments.

7. Acknowledgements

The authors gratefully acknowledge the support of the following NSF grants: IIS-0846976 and CCF-0829827.

8. References

[1] C. Melhuish, M. Wilson and A. Sendova-Franks, "Patch Sorting: Multiobject Clustering Using Minimalist Robots," *Proceedings of 6th European Conference on Advances in Artificial Life*, Springer, 2001.

[2] A. T. Hayes, A. Martinoli and R. M. Goodman, "Comparing Distributed Exploration Strategies with Simulated and Real Autonomous Robots," *Proceedings of the 5th International Symposium on Distributed Autonomous Robotic Systems* (*DARS*'00), Springer Verlag, Berlin, 2000.

[3] A. Boeing, S. Hanham and T. Braunl, "Evolving Autonomous Biped Control from Simulation to Reality," *Proceedings of International Conference on Autonomous Robots and Agents* (*ICARA*'04), 2004.

[4] K. Easton and A. Martinoli, "Efficiency and Optimization of Explicit and Implicit Communication Schemes in Collaborative Robotics Experiments," *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2002. doi:10.1109/IRDS.2002.1041693

[5] "SIMPAR," 2008. http://www.simpar.org/.

[6] R. Brooks, "A Robust Layered Control System for a Mobile Robot," *IEEE Journal of Robotics and Automation*, Vol. 2, No. 1, 1986, pp. 14-23. doi:10.1109/JRA.1986.1087032

[7] M. Brady and H. Hu, "Software and Hardware Architecture of Advanced Mobile Robots for Manufacturing," *Journal of Experimental and Theoretical Artificial Intelligence*, Vol. 9, 1997, pp. 257-276. doi:10.1080/095281397147112

[8] T. Balch and R. C. Arkin, "Behavior-Based Formation Control for Multirobot Teams," *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 6, 1999, pp. 926-939. doi:10.1109/70.736776

[9] M. J. Mataric, "Issues and Approaches in Design of Collective Autonomous Agents," *Robotics and Autonomous Systems*, Vol. 16, 1995, pp. 321-331. doi:10.1016/0921-8890(95)00053-4

[10] M. J. Mataric, G. S. Sukhatme and E. H. Ostergaard, "Multi-Robot Task Allocation in Uncertain Environments," *Autonomous Robots*, Vol. 14, No. 1, pp. 255-263, 2003. doi:10.1023/A:1022291921717

[11] E. H. Ostergaard, M. J. Mataric and G. S. Sukhatme, "Multi-Robot Task Allocation in the Light of Uncertainty," *IEEE International Conference on Robotics and Automation*, Washington DC, 11-15 May 2002.

[12] N. Jakobi, P. Husbands and I. Harvey, "Noise and the Reality Gap: The Use of Simulation in Evolutionary Robotics," *Proceedings of* 3rd *European Conference on Artificial Life*, Springer-Verlag, Berlin, 1995.

[13] M. Mataric and D. Cliff, "Challenges in Evolving Controllers for Physical Robots," *Robotics and Autonomous Systems*, Vol. 19, No. 1, 1996, pp. 67-83. doi:10.1016/S0921-8890(96)00034-6

[14] L. Meeden, "Bridging the Gap between Robot Simulations and Reality with Improved Models of Sensor Noise," *In Proceedings of the Third Annual Genetic Programming Conference*, 1998.

[15] N. Y. Ko, D. J. Seo, G. J. Kim, Y. Moon and Y. Bae, "Simulation of Mobile Robot Motion Considering Uncertainties in Robot Model," *IEEE International Conference on Industrial Informatics*, Daejeon, 4 December 2008.

[16] D. J. Seo, N. Y. Ko, G. J. Kim, Y. Moon, Y. Bae and S.-W. Lim, "Performance Evaluation of Robot Motion Incorporating Uncertainties in Sensors and Motion," *Next-Generation Applied Intelligence*, Springer, Berlin, 2009, pp. 271-280. doi:10.1007/978-3-642-02568-6_28

[17] J. Go, B. Browning, and M. Veloso, "Accurate and Flexible Simulation for Dynamic, Vision-Centric Robots," *Proceedings of International Joint Conference on Autonomous Agents and Multi-Agent Systems*, New York, 23 July 2004.

[18] R. A. Brooks, "Artificial Life and Real Robots," *In Proceedings of the* 1*st European Conference on Artificial Life*, MIT Press, Cambridge, 1992.

[19] E. Gat, "Towards Principled Experimental Study of Autonomous Mobile Robots," *Autonomous Robots*, Vol. 2, No. 3, 1995, pp. 179-189. doi:10.1007/BF00710855

[20] P. Husbands, I. Harvey and D. Cliff, "An Evolutionary Approach to Situated Ai," *Prospects for Artificial Intelligence*: *Proceedings of the* 9*th Conference of the Society for Artificial Intelligence and the Simulation of Behaviour*, IOS Press, Amsterdam, 1993.

[21] B. Jung and G. S. Sukhatme, "Tracking Targets Using Multiple Robots: The Effect of Environment Occlusion," *Autonomous Robots*, Vol. 13, No. 3, 2002, pp. 191-205. doi:10.1023/A:1020598107671

[22] T. M. Smith, "Blurred Vision: Simulation-Reality Transfer of a Visually Guided Robot," *Proceedings of the* 1*st European Workshop on Evolutionary Robotics*, Springer Verlag, Berlin, 1998.

[23] A. Rosenfeld, G. A. Kaminka and S. Kraus, "Adaptive Robot Coordination Using Interference Metrics," *The* 16*th European Conference on Artificial Intelligence*, Valencia, 23-27 August 2004.

[24] M. Anderson and N. Papanikolopoulos, "Implicit Cooperation Strategies for Multi-Robot Search of Unknown Areas," *Journal of Intelligent and Robotic Systems*, Vol. 53, No. 4, 2008, pp. 381-397. doi:10.1007/s10846-008-9242-5

[25] B. Balaguer, S. Carpin and S. Balakirsky, "Towards Quantitative Comparisons of Robot Algorithms: Experiences with Slam in Simulation and Real World Systems," *IROS Workshop on Performance Evaluation and Benchmarking for Intelligent Robots and Systems*, San Diego, 2 November 2007.

[26] S. Balakirsky, S. Carpin, G. Dimitoglou and B. Balaguer, "From Simulation to Real Robots with Predictable Results: Methods and Examples," *Performance Evaluation and Benchmarking of Intelligent Systems*, Springer, Berlin, 2009. doi:10.1007/978-1-4419-0492-8_6

[27] B. P. Gerkey and M. J. Mataric, "Are (Explicit) Multi-Robot Coordination and Multi-Agent Coordination Really So Different?" *Proceedings of the AAAI Spring Symposium on Bridging the Multi-Agent and Multi-Robotic Research Gap*, Palo Alto, 22-24 March 2004.

[28] J. Guerrero and G. Oliver, "Physical Interference Impact in Multi-Robot Task Allocation Auction Methods," *Proceedings of IEEE Workshop on Distributed Intelligent Systems*, Beijing, 26 June 2006. doi:10.1109/DIS.2006.58

[29] K. Lerman and A. Galstyan, "Mathematical Model of Foraging in a Group of Robots: Effect of Interference," *Autonomous Robots*, Springer, Berlin, Vol. 13, No. 2, 2002, pp. 127-141.

[30] A. Martinoli and F. Mondada, "Probabilistic Modelling of a Bio-Inspired Collective Experiment with Real Robots," *Proceedings of the* 3*rd International Symposium on Distributed Autonomous Robotic Systems*, MIT Press, Cambridge, 1997.

[31] A. Rosenfeld, G. A. Kaminka and S. Kraus, "A Study of Scalability Properties in Robotic Teams," Springer, Berlin, 2005.

[32] T. Lochmatter and A. Martinoli, "Understanding the Potential Impact of Multiple Robots in Odor Source Localization," 9*th Symposium on Distributed Autonomous Robotic Systems* (*DARS* '08), Tsukuba, 17-19 November 2008.

[33] P. E. Rybski, S. A. Stoeter, M. Gini, D. F. Hougen and N. Papanikolopoulos, "Performance of a Distributed Robotic System Using Shared Communications Channels," *IEEE transactions on Robotics and Automation*, Vol. 18, No. 5, 2002, pp. 713-727. doi:10.1109/TRA.2002.803460

[34] K. Lerman, C. Jones, A. Galstyan and M. J. Mataric, "Analysis of Dynamic Task Allocation in Multi-Robot Systems," *The International Journal of Robotics Research*, Vol. 25, No. 3, 2006, pp. 225-241. doi:10.1177/0278364906063426

[35] B. P. Gerkey, R. T. Vaughan and A. Howard, "The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems," *Proceedings of International Conference on Advanced Robotics* (*ICAR*), Coimbra, 30 June-3 July 2003.

[36] R. Vaughan, "Massively Multi-Robot Simulation in Stage," *Swarm Intelligence*, Vol. 2, No. 2, 2008, pp. 189-208. doi:10.1007/s11721-008-0014-4

[37] R. Smith, "Open Dynamics Engine," 2007. http://www.ode.org/.

[38] E. Kokkevis, D. Metaxas and N. I. Badler, "User-Controlled Physicsbased Animation for Articulated Figures," *Proceedings of Computer Animation*, Geneva, 3-4 June 1996.

[39] D. Floreano, P. Husbands and S. Nolfi, "Handbook of Robotics," Chapter 61, Springer, Berlin, 2008, pp. 1423-1447.

[40] N. Koeing and A. Howard, "Player Project," 2004. http://playerstage.sourceforge.net/index.php?src=gazebo.

[41] S. Carpin, M. Lewis, J. Wang, S. Balakirsky and C. Scrapper, "USARSim: A Robot Simulator for Research and Education," *Proceedings of the IEEE International Conference on Robotics and Automation*, Roma, 10-14 April 2007.doi:10.1109/ROBOT.2007.363180

[42] O. Michel, "Cyberbotics ltd-Webots: Professional Mobile Robot Simulation," *International Journal of Advanced Robotic Systems*, Vol. 1, No. 1, 2004, pp. 39-42.

[43] S. Dawson, B. L. Wellman and M. Anderson, "The Effect of Interaction on Robotic Sensor Network Experiments," *Proceedings of* 2nd *International Conference on Sensor Networks and Applications* (*SNA*), Las Vegas, 8-10 November 2010.

[44] S. Dawson, B. L. Wellman and M. Anderson, "Using Simulation to Predict Multi-Robot Performance on Coverage Tasks," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Paris, August 2010.

[45] B. Yamauchi, "Frontier-Based Exploration Using Multiple Robots," *Proceedings of the Second International Conference on Autonomous Agents*, New York, 10-13 May 1998. doi:10.1145/280765.280773

