

Compensating for Model Uncertainty in the Control of Cooperative Field Robots

by

Vivek Anand Sujan

Master of Science in Mechanical Engineering
Massachusetts Institute of Technology (1998)

Submitted to the Department of Mechanical Engineering in partial fulfillment of the requirements for the degree of

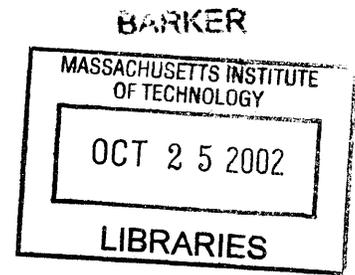
Doctor of Philosophy in Mechanical Engineering

at the

Massachusetts Institute of Technology

June, 2002

© 2002 Vivek Anand Sujan.
All rights reserved.



The author hereby grants to MIT permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole or in part.

Signature of Author _____

Vivek Anand Sujan
Department of Mechanical Engineering
April 17, 2002

Certified By _____

Steven Dubowsky
Professor of Mechanical Engineering
Thesis Supervisor

Accepted By _____

Ain A. Sonin
Professor of Mechanical Engineering
Chairman, Departmental Graduate Committee

Compensating for Model Uncertainty in the Control of Cooperative Field Robots

Submitted to the Department of Mechanical Engineering
on April 17, 2002, in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Mechanical Engineering

by

Vivek Anand Sujan

Abstract

Current control and planning algorithms are largely unsuitable for mobile robots in unstructured field environment due to uncertainties in the environment, task, robot models and sensors. A key problem is that it is often difficult to directly measure key information required for the control of interacting cooperative mobile robots. The objective of this research is to develop algorithms that can compensate for these uncertainties and limitations. The proposed approach is to develop physics-based information gathering models that fuse available sensor data with predictive models that can be used in lieu of missing sensory information.

First, the dynamic parameters of the physical models of mobile field robots may not be well known. A new information-based performance metric for on-line dynamic parameter identification of a multi-body system is presented. The metric is used in an algorithm to optimally regulate the external excitation required by the dynamic system identification process. Next, an algorithm based on iterative sensor planning and sensor redundancy is presented to enable field robots to efficiently build 3D models of their environment. The algorithm uses the measured scene information to find new camera poses based on information content. Next, an algorithm is presented to enable field robots to efficiently position their cameras with respect to the task/target. The algorithm uses the environment model, the task/target model, the measured scene information and camera models to find optimum camera poses for vision guided tasks.

Finally, the above algorithms are combined to compensate for uncertainties in the environment, task, robot models and sensors. This is applied to a cooperative robot assembly task in an unstructured environment. Simulations and experimental results are presented that demonstrate the effectiveness of the above algorithms on a cooperative robot test-bed.

Thesis Supervisor: Dr. Steven Dubowsky
 Professor of Mechanical Engineering

Acknowledgements

I would like to thank my advisor, Prof. S. Dubowsky, for providing me with the opportunity and guiding me during my time with the Field and Space Robotics Laboratory. I would also like to thank Professors George Barbastathis, Sanjay Sarma and Alexander Slocum for their insights and contributions as members of my Ph.D. committee. I would like to thank NASA/JPL for providing the opportunity and resources necessary to carry out the research presented here.

My thanks to Professors Barbara Andereck and L.T. Dillman, of Ohio Wesleyan University, Professors Erik K. Antonsson and Joel Burdick, of Caltech, that helped guide and set me on the track of academic research.

Special thanks to those friends who have had a real positive impact in my life: Marco “Come on” Meggiolaro, Mike “Bull Dog” Mulqueen, Christina “Joy” D’Arrigo, Leslie Reagan, Matt Whitacre, and the gym crew. To Vince McMahon and the WWF, thank you for all those wonderful Thursday evenings!

And yet none of this could have ever been done without my wonderful parents. It is through their love and optimism for life that I have learnt never to feel weak, never to feel down, never to be boastful, and never to accept defeat. My thanks to my kick-ass sister, Menka, and her sons, Kaarin and Yuveer, after whom the experimental robots developed in this work, have been named.

Finally, I would like to thank the Almighty, for blessing me with these opportunities. This thesis is dedicated to the memories of my grandmother Kishni J. Ramchandani.

Table of Contents

Chapter 1. Introduction.....	13
1.1. Problem Statement And Motivation.....	13
1.2. Purpose Of This Thesis.....	15
1.3. Background And Literature Review.....	18
1.3.1. Control and planning cooperative mobile robots.....	18
1.3.2. Identification of dynamic system parameters.....	19
1.3.3. Environment modeling.....	20
1.3.4. Task modeling—visual sensing strategy.....	22
1.4. Outline Of This Thesis.....	23
Chapter 2. Dynamic Parameter Identification.....	25
2.1. Introduction.....	25
2.2. System Dynamic Model.....	26
2.3. Estimating The Dynamic Parameters.....	29
2.4. A Metric For Observability.....	30
2.4.1. Classical observability metric.....	30
2.4.2. Mutual information based metric.....	32
2.5. Formulation Of Exciting Trajectories.....	36
2.6. Results.....	39
2.6.1. Simulation studies.....	39
2.6.2. Experimental studies.....	43
2.7. Conclusions.....	47

Chapter 3. Environment Modeling.....	48
3.1. Introduction.....	48
3.2. Algorithm Description.....	49
3.2.1. Overview.....	49
3.2.2. Algorithm initialization.....	51
3.2.3. Data modeling and fusion.....	51
3.2.4. Vision system pose selection.....	56
3.2.5. Camera motion correction.....	61
3.3. Results.....	65
3.3.1. Simulation studies.....	65
3.3.2. Experimental studies.....	72
3.4. Conclusions.....	76
Chapter 4. Task Modeling.....	77
4.1. Introduction.....	77
4.2. Algorithm Description.....	78
4.2.1. Overview.....	78
4.2.2. Algorithm initialization and target identification.....	80
4.2.3. Optimum camera pose identification.....	80
4.2.3.1. Depth of field (DOF)	80
4.2.3.2. Resolution of target.....	81
4.2.3.3. Target field visibility.....	81
4.2.3.4. Target angular visibility.....	84
4.2.3.5. Alternate/secondary targets.....	85

4.2.3.6. Camera motion calibration.....	85
4.3. Results.....	86
4.3.1. Simulation.....	86
4.3.2. Experiments.....	91
4.4. Conclusions.....	93
Chapter 5. Cooperative Task Execution.....	94
5.1. Problem Overview.....	94
5.2. Experimental Setup.....	97
5.3. Marker Identification.....	99
5.4. Task Execution Results.....	104
5.5. Conclusions.....	108
Chapter 6. Conclusions and Suggestions for Future Work.....	109
6.1. Contributions of This Thesis.....	109
6.2. Suggestions for Future Work.....	110
References.....	113
Appendix A – Cooperative Mobile Robot System Dynamic Model.....	124
Appendix B – Newton-Euler Equations of Motion of Mobile Robot.....	136
Appendix C – Loss-less Image Compression.....	142
Appendix D – Experimental Cooperative Robot System.....	149
Appendix E – Lightweight hyper-redundant binary mechanisms.....	155

List of Figures

Figure 1-1: Solar panel assembly by cooperative robots.....	14
Figure 1-2: Rocky 7 inspecting rock sample.....	14
Figure 1-3: Representative physical system.....	15
Figure 1-4: Conventional control architecture.....	15
Figure 1-5: Physical model based sensor fusion.....	16
Figure 2-1: Representation of a general mobile field robot.....	26
Figure 2-2: Representation of the simplified mobile robot.....	27
Figure 2-3: Flow-diagram of a Kalman Filter.....	30
Figure 2-4: Constant parameter arm motion.....	40
Figure 2-5: Variable parameter arm motion.....	40
Figure 2-6: Mutual information metric for Stiffness θ_p	41
Figure 2-7: Example of identification converge curves.....	43
Figure 2-8: Experimental mobile manipulator.....	43
Figure 2-9: Constant parameter arm motion.....	44
Figure 2-10: Variable parameter arm motion.....	45
Figure 2-11: Inclinometer pitch reading (radians).....	45
Figure 2-12: Example of identification converge curves.....	46
Figure 3-1: Cooperative mapping by robots.....	49
Figure 3-2: Outline of model building and placement algorithm.....	50
Figure 3-3: Flowchart of the initialization of environment mapping algorithm.....	51
Figure 3-4: 3-D range measurement fusion with sensor uncertainty.....	52
Figure 3-5: Stereo imaging error evaluation.....	53

Figure 3-6: Sample of merging two Gaussian probability distributions (2-D case)..	55
Figure 3-7: Flowchart for data fusion using known vision system motion.....	56
Figure 3-8: Evaluation of expected new information.....	58
Figure 3-9: Flowchart for vision system pose selection of environment mapping algorithm.....	60
Figure 3-10: Relationship of camera and target frames.....	61
Figure 3-11: Flowchart for vision system motion identification using scene fiducials.....	65
Figure 3-12: Unstructured planar environment.....	66
Figure 3-13: Results of single vision system modeling an unknown environment...	67
Figure 3-14: Single vision system path.....	68
Figure 3-15: Single vision system area mapped (gray=empty space, black=obstacle, white=unknown)	68
Figure 3-16: Results of two vision systems modeling an unknown environment....	69
Figure 3-17: Two cooperating vision systems path.....	70
Figure 3-18: Two cooperating vision systems area mapped (gray=empty space, black=obstacle, white=unknown).....	70
Figure 3-19: Unstructured indoor-type planar environment.....	71
Figure 3-20: Results of single vision system modeling an unknown environment...	72
Figure 3-21: Single vision system path.....	72
Figure 3-22: Experimental mobile vision system modeling an unstructured environment.....	73
Figure 3-23: Identification and tracking of 6 fiducials (□--tracked with previous	

image, O--tracked with next image).....	74
Figure 3-24: Accumulated r.m.s. translation error of vision system.....	74
Figure 3-25: Number of mapped environment points as a function of scan number	75
Figure 3-26: Environment mapped/modeled- Environment mapped/modeled- Sequential camera pose selection.....	75
Figure 3-26: Environment mapped/modeled- Environment mapped/modeled- Maximum information based camera pose selection.....	76
Figure 4-1: Cooperative assembly by robots.....	78
Figure 4-2: Task directed optimal camera placement.....	79
Figure 4-3: Target reduction and occlusion expansion.....	82
Figure 4-4: Projection of expanded object.....	83
Figure 4-5: Computing the TFV_{RF} (shaded regions).....	83
Figure 4-6: Difference between true and expanded object occluded regions.....	84
Figure 4-7: Target angular visibility.....	85
Figure 4-8: Optimal camera placement.....	87
Figure 4-9: Cooperative assembly concept.....	88
Figure 4-10: Simulated planar environment.....	89
Figure 4-11: Optimal camera placement with moving object (known CAD model)	90
Figure 4-12: Experimental test.....	92
Figure 4-13: Experimental test-Camera view of target from optimum pose.....	93
Figure 5-1: Cooperative insertion task layout.....	94
Figure 5-2: Cooperative task target model- representaive problem.....	96
Figure 5-3: Cooperative task target model- extrapolation of true viewing target.....	96

Figure 5-4: Cooperative task target model- relating the coordinate frames of the cooperating robots.....	97
Figure 5-5: White circular markers to locate object and insertion site.....	98
Figure 5-6: Model based sensor fusion from a sensor suite.....	99
Figure 5-7: Histogram of images.....	101
Figure 5-8: Example of image reduction process.....	101
Figure 5-9: Circle defined by three points.....	102
Figure 5-10: Identification of marker (centermost hole with 0.25” diameter)	104
Figure 5-11: Intermediate steps during task execution (success)	105
Figure 5-12: Camera view (stereo pairs)	106
Figure 5-13: Optimal camera placement—successful task execution.....	107
Figure 5-14: Random camera placement—unsuccessful task execution.....	107
Figure A-1: Block diagram of linear feed-forward compensation for dynamic disturbance rejection.....	124
Figure A-2: Representative physical system.....	125
Figure A-3: Cooperative robot modeling.....	125
Figure A-4: Dynamic tip-over stability.....	130
Figure A-5: Environment contact states.....	131
Figure A-6: Relation of interacting forces, contact points and measured forces/torques.....	132
Figure A-7: Error of contact point location w.r.t. sensor noise.....	133
Figure A-8: Decentralized cooperative control architecture using surrogate sensing.....	133

Figure A-9: Physically cooperating mobile robots.....	134
Figure A-10: Endpoint position and force of master robot and slave robot.....	134
Figure A-11: Arm joint and base position of master robot and slave robot.....	135
Figure A-12: Arm joint and base forces of master robot and slave robot.....	135
Figure B-1: Force/moment balance.....	136
Figure B-2: Force/moment balance of compliance module.....	137
Figure B-3: Representation of the simplified mobile robot.....	139
Figure B-4: Force/moment balance.....	139
Figure B-5: Force/moment balance of compliance module.....	140
Figure C-1: LZW compression flowchart.....	146
Figure C-2: Comparison of 2-D image compression methods.....	148
Figure D-1: FSRL Experimental cooperative rovers.....	149
Figure D-2: Experimental system: 4 DOF manipulator kinematics.....	150
Figure D-3: PWM motor control circuit.....	151
Figure D-4: Overview of experimental system hardware.....	152
Figure E-1: BRAID design concept.....	157
Figure E-2: i^{th} parallel link stage.....	159
Figure E-3: Projection of section ABCD from Figure E-3.....	160
Figure E-4: Projection of section EFGH from Figure E-3.....	161
Figure E-5: Workspace of 5 stage BRAID element.....	163
Figure E-7: SMA power and control bus.....	164
Figure E-8: SMA power bus address decoding and latching electronics.....	166
Figure E-9: Experimental platform of BRAID.....	166

List of Tables

Table 2-1: Simulation parameter identification.....	41
Table 2-2: Experimental parameter identification.....	45
Table 4-1: Results of changing task difficulty, occlusion density and task execution success.....	91
Table 5-1: Task execution success for varying task difficulty levels.....	105
Table C-1: Comparison of compression methods on 2-D images.....	147
Table D-1: Inclinator calibration data at room temperature.....	153
Table D-2: Camera specifications.....	154

1.1. Problem Statement And Motivation

Human exploration and development of the planets and moons of the solar system are stated goals of NASA and the international space science community [Huntsberger-3]. These missions will require robot scouts to lead the way, by exploring, mapping, seeking or extracting minerals and eventually constructing facilities in complex terrains. Multiple cooperating robots will be required to set up surface facilities in challenging rough terrain for in-situ measurements, communications, and to pave the way for human exploration of planetary surfaces (see Figure 1-1). Tasks may include building permanent stations and fuel generation equipment. This will require the handling of relatively large objects, such as deploying of solar panels and sensor arrays, anchoring of deployed structures, movement of rocks, and clearing of terrain. Robots will also assist future space explorers.

Such future robotic mission scenarios suggest that current planetary rover robots, with their limited functionality, such as simple rock sampling (see Figure 1-2), will not be sufficient for such missions [Baumgartner, Huntsberger-2, Parker]. A new generation of planetary worker robots will be essential for future missions [Baumgartner, Huntsberger-1, Huntsberger-2, Huntsberger-3, Schenker]. In addition to the exploration and development of space, such robotic systems could prove vital in earth-based field applications including environment restoration, underground mining, hazardous waste disposal, handling of large weapons, and

assisting/supporting humans in field tasks [Huntsberger-1, Huntsberger-2, Khatib-2, Osborn, Schenker, Yeo, Walker, Shaffer].

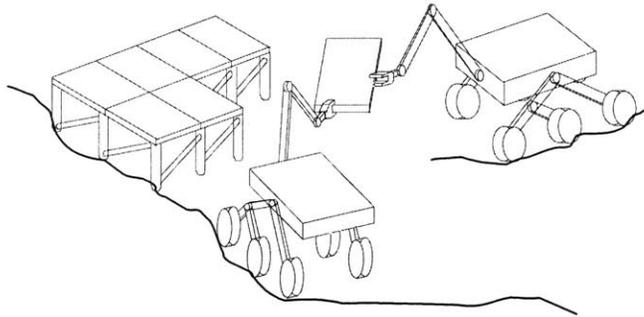


Figure 1-1: Solar panel assembly by cooperative robots

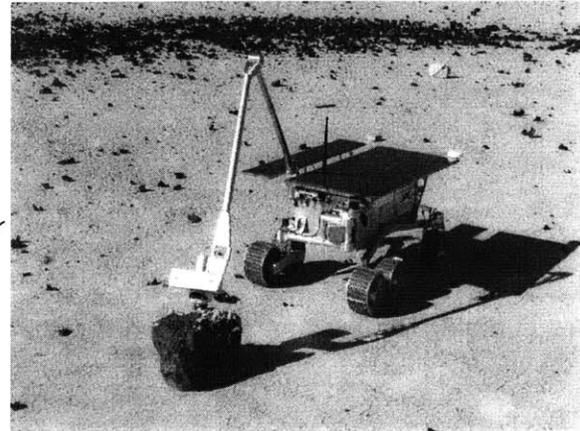


Figure 1-2: Rocky 7 inspecting rock sample

Substantial previous research has been devoted to control and planning of cooperative robots and manipulators [Choi, Khatib-1, Khatib-2, Marapane, Parker, Pfeffer, Takanishi, Veloso-1, Veloso-2, Yeo, Donald, Mataric, Gerkey, Alur]. However, these results are largely inapplicable to mobile robots in unstructured field environments. In simple terms, the conventional approach for the control of robotic systems is to use sensory information as input to the control algorithms. System models are then used to determine control commands. The methods developed to date generally rely on assumptions that include: flat and hard terrain; accurate knowledge of the environment; little or no task uncertainty; and sufficient sensing capability. For real field environments, a number of these assumptions are often not valid.

For example, Figure 1-3 shows two physically interacting cooperative robots working in an unstructured field environment. The mobile robotic systems have independently mobile cameras and other onboard sensors, and are working together to assemble a large structure. However, visual sensing is limited due to target occlusions by the object being handled and

objects in the environment (e.g. rocks, building supplies, drums of materials, debris). There is significant task uncertainty in relative pose between the robots and the target, the grasp points, etc. Due to these limitations and uncertainties, classical robot control and planning techniques break down (see Figure 1-4).

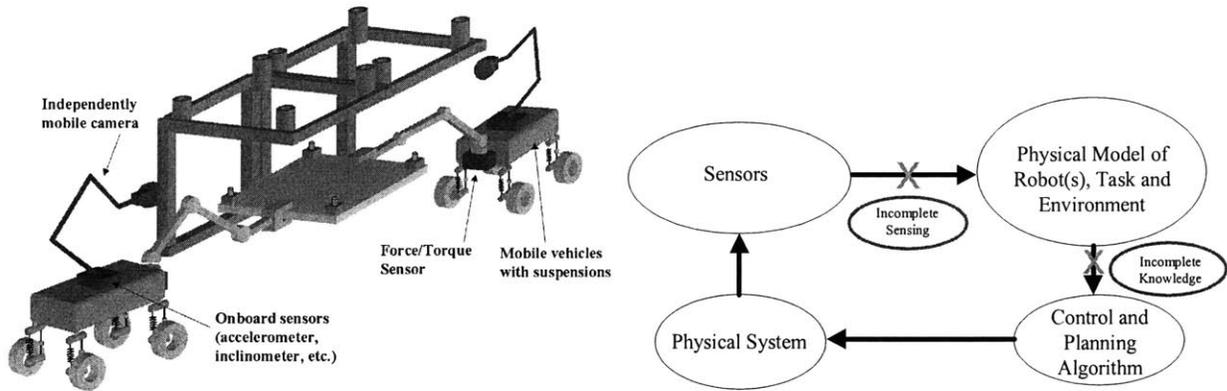


Figure 1-3: Representative physical system Figure 1-4: Conventional control architecture

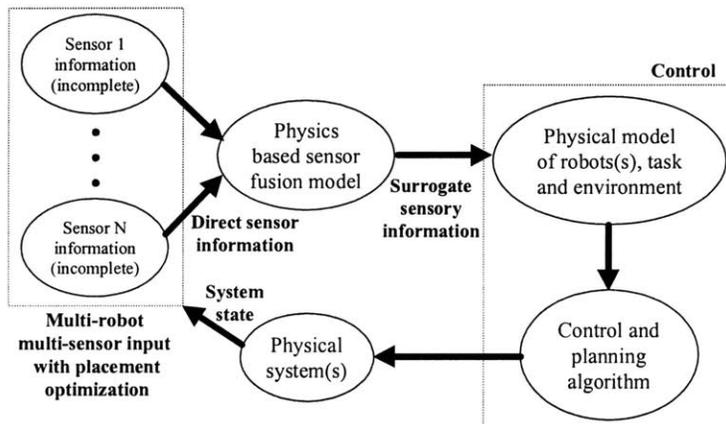
The key problem is that it is difficult or impossible to directly measure key information required for the control of interacting cooperative mobile field robots.

1.2. Purpose Of This Thesis

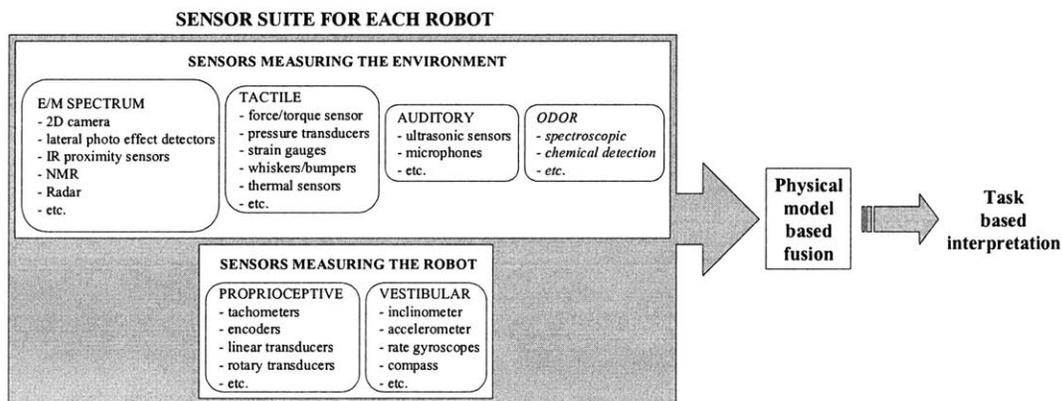
The objective of this research is to develop algorithms to compensate for sensor limitations and enable multiple mobile robots to perform cooperative tasks in unstructured field environments. The key theme is to develop optimal information gathering methods from distributed resources.

The proposed approach in this research is to use physics based models to fuse available sensor information with predictive models that can be used in lieu of missing sensory information. In other words, the physical models of the interacting systems are used as the “sensor fusion engines.” Observing these models will provide *virtual* or *surrogate* sensing. This virtual sensor information will be used to supplement the incomplete and insufficient direct

sensor data based on the information obtained from all the members of the robot workers (crew). Thus, experiences (measurements) of each individual robot become part of the collective experience of the group. Such a methodology fuses dynamic models, available sensor data, and prior sensed data from multiple robot team members. This inferred information can be applied to robot control and planning architectures. Figure 1-5(a) outlines this idea. Figure 1-5(b) shows a possible sensor suite of a robot or team of robots, fused using a physics based model, to yield *surrogate sensory information*.



(a) Control architecture with surrogate sensing



(b) Sensor suite example—used in model based fusion
Figure 1-5: Physical model based sensor fusion

The approach to this research is divided into three parts: “know one’s self”, “know one’s

environment” and “know one’s task”. Fundamentally this reduces to the following. First, field robots must have a good dynamic model of themselves to be used in model-based control algorithms. Second, the robots must have a good geometric environment model in which the task is being performed. Third, the robots must have a good view of the targets critical to performing the task. Algorithms in each successive step use the algorithms developed in the previous steps.

(a) **Know ones self:** Here, an algorithm is developed to allow a robot to measure its dynamic parameters in the presence of noisy sensor data. These parameters are required to successfully apply model-based control algorithms [Hootsman]. These parameters may be roughly known from design specifications or found off-line by simple laboratory tests. However, for field systems in hostile environments, they may not be well known, or may change when the robot interacts with the environment. For example, temperature fluctuations result in substantial changes in vehicle suspension stiffness and damping. Vehicle fuel consumption, rock sample collection, etc. cause changes in the location of the center of gravity, mass and inertia of the system. Hence, on-line identification of these parameters is critical for the system performance.

(b) **Know one’s environment:** Here, an algorithm is developed to allow multiple cooperating robots to efficiently model/map their environment. For robots working in unstructured environments, it is often not possible to have a-priori models of the environment. The robots will need to build these models using available sensory data. A number of problems can make this non-trivial. These include the uncertainty of the task in the environment, location and orientation uncertainty in the individual robots, and occlusions (due to obstacles, work piece, other robots).

(c) **Know one’s task:** Here an algorithm is developed to allow multiple cooperating robots to

observe their task effectively. Once the environment model is created, the robots need to position their sensors in a task-directed “optimal” way. That is, for a given task requiring visual guidance, there is an associated target to observe. For example, in assembly tasks, the target may be a single point or region in the environment, a distance between two objects, etc. The algorithm finds optimal view poses of the target for the individual robots as the task is carried out. These view poses provide the required visual guidance for task execution.

1.3. Background And Literature Review

There has been significant research in the area of cooperative robotics over the past decade [Choi, Clark, Khatib-1, Khatib-2, Luo, Marapane, Parker, Pfeffer, Schenker, Takanishi, Veloso-1, Veloso-2, Yeo, Donald, Mataric, Gerkey, Alur]. Relevant research is broken down into four areas: (a) Control and planning of cooperative mobile robots, (b) identification of system dynamic parameters, (c) environment modeling and (d) task modeling.

1.3.1. Control and planning cooperative mobile robots

Aspects of control and planning of cooperative mobile robots have been addressed by a number of researchers, including modeling the environment and task, modeling the physical interactions among robots and between robots and the environment, and assigning individual robot roles [Khatib-1, Marapane, Parker, Donald, Mataric, Gerkey, Alur]. A typical approach to the problem of modeling environment and task knowledge is to assume that both the environment and the task are well-defined or can be obtained with sufficient accuracy [Choi, Khatib-1, Khatib-2, Pfeffer, Donald, Mataric, Gerkey]. Further, many successful approaches have been developed to model dynamic interactions during cooperative manipulation of objects. These include: generating virtual linkages that account for internal forces; augmenting the object to provide a dynamic description at the operational point; dynamic hybrid force/position

compensation across a passive object; and including passive joints in the closed kinematic manipulator-object-manipulator chain [Choi, Khatib-2, Pfeffer, Yeo].

Some researchers have addressed the problem of tip-over stability due to the dynamic effects in a single mobile manipulator system [Dubowsky, Takanishi]. Researchers have also addressed the problem of role assignment in cooperative systems (hierarchical (master-slave) and equipotent team structures) [Khatib, Marapane, Veloso-1, Veloso-2]. Most work in the area of cooperative robots has focused on small laboratory systems that accomplish simple tasks (such as pushing small objects around a table) while avoiding collisions with other robots [Marapane, Parker]. The planning and control algorithms are primarily heuristic, probabilistic or based on fuzzy logic, do not exploit all the physical capabilities of the system, and testing in field environments has been limited.

1.3.2. Identification of dynamic system parameters

Dynamic system models are often used in robot control architectures, to enhance the system performance. Identification of system parameters is a well-studied problem [Alloum, Atkeson, Bard, Ljung, Nikravesh, Olsen, Serban, Schmidt, Soderstorm]. Various effective algebraic and numerical solution techniques have been developed to solve for unknown parameters using dynamic system models [Bard, Gelb, Nikravesh, Serban]. These include techniques based on pseudo-inverses, Kalman observers, Levenberg-Marquardt methods, and others. However, the accuracy/quality of the identified system parameters is a function of both the system excitation and the measurement noise (sensor noise). A number of researchers have developed metrics to evaluate the quality of identified system parameters [Armstrong, Gautier, Schmidt, Serban, Soderstorm]. Such metrics determine if a given set of parameters is identifiable, which is known as the “identifiability/observability” problem [Serban]. These include tests based on differential

algebra, where a set of differential polynomials describes the model under consideration [Bard, Ljung]. Other metrics monitor the condition number of an excitation matrix computed from the dynamic model. Examples of such excitation matrices include the Hessian of the model residual vector, the derivative of the system Hamiltonian, and the input correlation matrix [Serban, Gautier, Armstrong].

The metrics of parameter quality can be used to select the excitation imposed on the physical system and have been applied with limited success to industrial robotic systems [Armstrong, Atkeson, Gautier, Mayeda]. However, such approaches can be computationally complex, an important issue for space robots where computational power is very limited. For example, defining excitation trajectories for the identification of an industrial 3 DOF manipulator using an input correlation matrix requires 40 hours of VAX (40MHz) time [Armstrong, Gautier]. Additionally, these methods are unable to indicate which parameter estimates have low confidence values (low quality), since the quality metrics combines the performance into a single parameter. Thus it is not possible to assign higher weight to parameters of greater dynamic significance to system response.

1.3.3. Environment modeling

Environment modeling/mapping by mobile robots falls into the category of Simultaneous Localization and Mapping (SLAM). In such algorithms, the robot is constantly localizing itself as it maps the environment. Several researchers have addressed this problem for structured indoor-type environments [Asada, Burschka, Kruse, Thrun-1, Kuipers, Yamauchi, Castellanos, Leonard, Anousaki, Tomatis, Victorino, Choset]. Sensor movement/placement is usually done sequentially (raster scan type approach) or by following topological graphs [Choset, Victorino, Anousaki, Leonard, Kuipers, Rekleitis, Yamauchi]. Geometric descriptions of the environment

have been modeled in several ways, including generalized cones, graph models and Voronoi diagrams, occupancy grid models, segment models, vertex models, convex polygon models [Brooks, Choset, Crowley, Kuipers, Miller, Weisbin]. The focus of such work is accurate rather than efficient mapping process. Further, the environment is assumed to be effectively planar (e.g. the robot workspace is the floor of an office or a corridor) and readily traversable (i.e. terrain is flat and obstacles always have a route around them) [Anousaki, Thrun-1, Yamauchi, Choset, Kuipers, Lumelsky].

Localization is achieved by monitoring landmarks and their relative motions with respect to the vision systems. Several localization schemes have been implemented, including topological methods such as generalized voronoi graphs and global topological maps [Choset, Kuipers, Tomatis, Victorino], extended Kalman filters [Anousaki, Leonard, Park], and robust averages [Park]. Additionally, several different sensing methods have been employed, such as camera vision systems [Castellanos, Hager, Park], laser range sensors [Tomatis, Victorino], and ultrasonic sensors [Anousaki, Leonard, Choset]. Although some natural landmark selection methods have been proposed [Hager, Simhon, Yeh], most SLAM architectures rely on identifying landmarks as corners or edges in the environment [Anousaki, Kuipers, Castellanos, Victorino, Choset, Leonard]. This often limits the algorithms to structured indoor-type environments. Others have used human intervention to identify landmarks [Thrun-1].

Some studies have considered cooperative robot mapping of the environment [Jennings, Rekleitis, Thrun-2]. Novel methods of establishing/identifying landmarks and dealing with cyclic environments have been introduced for indoor environments [Jennings, Thrun-2]. In some cases, observing robot team members as references to develop accurate maps is required [Rekleitis].

1.3.4. Task modeling—visual sensing strategy

Previous work in visual sensing strategies can be divided into two areas [Luo, Tarabanis]. One area is concerned with sensor positioning i.e. placing a sensor to best observe some feature, and selecting a sensing operation that is most useful in object identification and localization. Researchers have used model-based approaches, requiring previously known environments [Burschka, Cowan, Hutchinson, Kececi, Laugier]. Target motions (if any) are assumed to be known [Laugier]. Brute force search methods divide the view volume (into grids, octrees, or constraint sets), and search algorithms for optimum sensor location are applied [Connolly, Cowan, Luo, Kececi, Nelson]. These methods require a priori knowledge of object/target models [Tarabanis, Chu]. Such methods can be effective, but are computationally expensive and impractical for many real field environments, where occlusions and measurement uncertainties are present.

The other direction of research in visual sensing strategies is sensor data fusion i.e. combining complementary data from either different sensors or different sensor poses to get an improved net measurement [Smith, Marapane, Nelson, Tarabanis, Veloso-1]. The main advantages of multi-sensor fusion are the exploitation of data redundancy and complementary information. Common methods for sensor data fusion are primarily heuristic (fuzzy logic) or statistical in nature (Kalman and Bayesian filters) [Betge-Brezetz, Luo, Repo, Clark, Marapane, Nelson, Tarabanis].

For target model building both sensor positions and sensor fusion play key roles. However, current methods do not effectively combine these methods to develop a sensing strategy for robot teams in unstructured environments.

In general, current research has not solved the problem of controlling multiple mobile

robots performing cooperative tasks in unstructured field environments, where limited sensing capabilities and incomplete physical models of the system(s)/environment dominate the problem.

1.4. Outline Of This Thesis

This thesis is composed of six chapters and five appendices. This chapter serves as an introduction and overview of the work, and summarizes related research.

Chapter 2 addresses the problem of “knowing ones self”. It presents a new information-based performance metric for on-line dynamic parameter identification of a multi-body system. The metric is used in an algorithm to optimally regulate the external excitation required by the dynamic system identification process. This algorithm is applied to identify the vehicle and suspension parameters of a mobile field manipulator. Simulations and experiments show the effectiveness of this algorithm.

Chapter 3 addresses the problem of “knowing ones environment”. An algorithm based on iterative sensor planning and sensor redundancy is proposed to enable field robots to efficiently build 3D models of the environment. The algorithm uses measured scene information to find new camera poses based on information content. Simulations and experiments show the effectiveness of this algorithm.

Chapter 4 addresses the problem of “knowing ones task”. Here, an algorithm is proposed to enable field robots to efficiently position their cameras with respect to the task/target. The algorithm uses the environment model, task/target model, measured scene information and camera models to find optimum camera poses for vision guided tasks. Simulations and experiments show the effectiveness of this algorithm.

Chapter 5 presents an experimental example of vision-guided cooperative assembly by mobile robots in unstructured field environments. Here, the algorithms developed in Chapters 2,

3 and 4 are combined.

Chapter 6 summarizes the contributions of this thesis and presents suggestions for future work.

The appendices to this thesis give detailed information on specific topics related to the work presented. Appendix A presents a cooperative mobile robots dynamic model used for model predictive control. Appendix B presents the derivation of the equations of motion of a mobile robot model used in Chapter 2. Appendix C presents a description of loss-less image compression schemes that is used for quantifying information content in a scene. Appendix D describes the Field and Space Robotics Laboratory cooperative rover test-bed, which is used to experimentally validate much of this work. Appendix E presents a concept for a lightweight hyper-redundant binary manipulator that may be used for camera/sensor placement tasks.

Dynamic Parameter Identification

2.1. Introduction

The first step in compensating for robot model uncertainty is to develop an algorithm that allows field robots to measure their dynamic parameters in the presence of noisy sensor data. These parameters are required to successfully apply model-based control algorithms

In this chapter a new performance metric, called a mutual information-based observability metric, is presented for on-line dynamic parameter identification of a multi-body system. This metric measures the uncertainty of each parameter's estimate. This measure is termed the "parameter observability." The metric is used to formulate a cost function that optimally controls the external system excitation during the identification process. The cost function weighs each parameter estimate according to its uncertainty. Hence, the excitation is controlled so that the identification favors parameters that have the greatest uncertainty at any given time. Parameters may also be given greater importance in the cost function based on its significance to the system's dynamic response. This method is more computationally efficient and yields faster convergence than single parameter methods [Armstrong, Gautier, Schmidt, Serban, Soderstorm].

Here the algorithm is applied to the on-line parameter identification of a mobile field robot system and is shown to be computationally efficient. A field robot may be equipped with a manipulator arm and onboard sensors such as inclinometers, accelerometers, vision systems, and force/torque sensors (see Figure 2-1). An onboard manipulator arm (with bandwidth constraints)

is moved to generate reaction forces, which excite vehicle base motions. The dynamic parameters include the mass, location of center of gravity, inertia, base compliance and damping.

The method assumes a robotic system composed of rigid elements, and there is no relative motion of the vehicle wheels with respect to the ground during the identification process. The algorithm also assumes that the robot is equipped with an inclinometer, accelerometer and arm base force/torque sensor mounted at the manipulator base. It is assumed that the onboard manipulator dynamic parameters are known, and the bandwidth of the arm actuators is sufficiently high to excite the vehicle dynamics. Finally, the motions of the base compliance are assumed to be small.

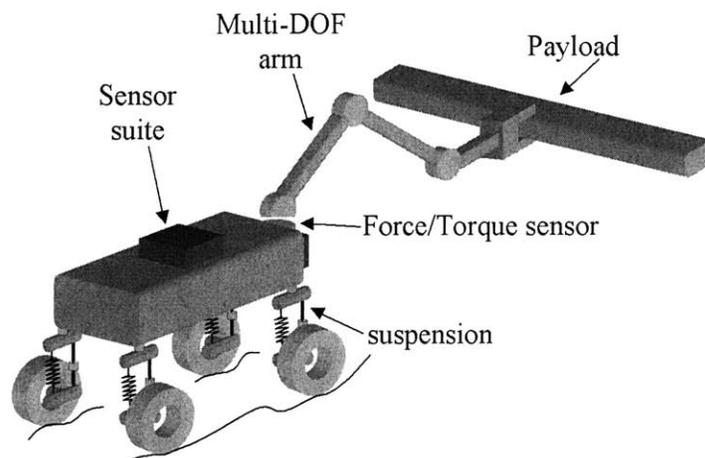


Figure 2-1: Representation of a general mobile field robot

The system is modeled using a Newton-Euler formulation (section 2-2). A Kalman filter is used to solve the dynamic parameters based on the physical model (section 2-3). The mutual information-based observability metric is used to determine the arm excitation trajectory (sections 2-4 and 2-5). Simulation and experimental results show the effectiveness of this algorithm (section 2-6).

2.2. System Dynamic Model

The algorithm to generate arm excitation trajectories for parameter identification requires a

dynamic model of the system. A number of models of vehicle suspension systems have been proposed [Alloum , Halfmann, Harris, Majjad, Nelles]. Many of these are quarter or half-vehicle models that consider stiffness and damping coefficients, but neglect vehicle mass and inertial properties. Here, a Newton-Euler formulation is used to model the full spatial dynamics of the system. The system represented in Figure 2-1 is reduced to three components: a rigid arm, a rigid vehicle body and a compliance module (see Figure 2-2). Rotational motions of the rigid arm result in reaction forces/moments felt by the vehicle base and in the suspension module. Motions of the base are measured through the onboard inclinometer, accelerometer and directional compass. Interaction forces/torques between the arm and vehicle base are measured by a base force/torque sensor (origin coincides with frame VI—Figure 2-2).

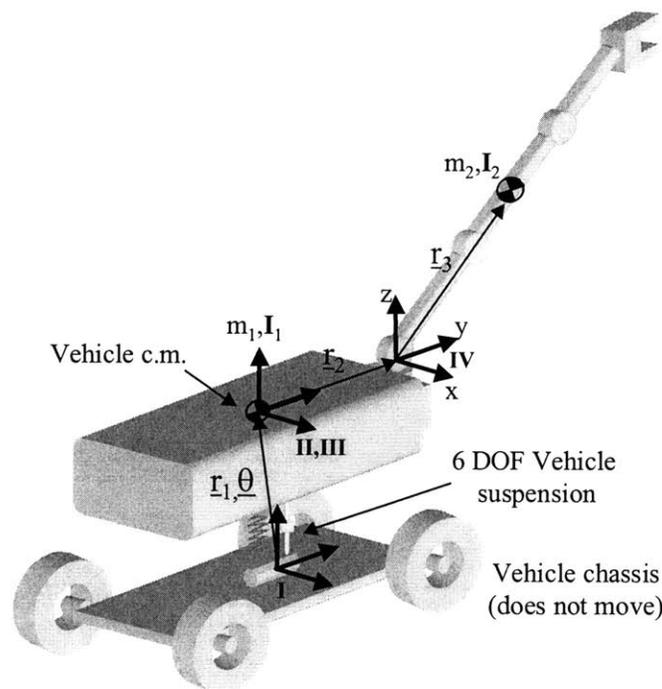


Figure 2-2: Representation of the simplified mobile robot

Although real vehicles have complex, multi-element suspension systems, only the net base compliance is modeled. This is modeled as a 6 DOF linear stiffness and damping system, located at the vehicle base center-of-gravity (see Appendix A). From the equations derived in Appendix

A, it can be seen that for small base motions, this simplified model can accurately model the vehicle dynamics. An advantage of the simplified model is that all coefficients can be identified by observing only the vehicle base motions, thus eliminating the need for more exotic sensors placed at each individual suspension. Additionally, a simplified suspension model accounts for all sources of compliance that would be difficult to model and measure individually.

Appendix B presents the dynamic model of the mobile robot presented in Figure 2-2. From Equation B-4, a set of 6 dynamic equations is obtained (forces and moments in 3D):

$$\begin{aligned} m_1(d(\mathbf{R}_0^{-1}\mathbf{g})-d(\ddot{\mathbf{r}}_1)_{II})-\mathbf{b}_{\bar{r}_1}^T \cdot d\dot{\mathbf{r}}_1-\mathbf{k}_{\bar{r}_1}^T \cdot d\mathbf{r}_1 &= d(\mathbf{F}_{1,2})_{II} \\ -\mathbf{I}_1 d(\ddot{\boldsymbol{\theta}})_{II}-d(\dot{\boldsymbol{\theta}} \times (\mathbf{I}_1 \dot{\boldsymbol{\theta}}))_{II}-d(\mathbf{r}_2 \times \mathbf{F}_{1,2})_{II}-\mathbf{b}_{\bar{\theta}}^T \cdot d\dot{\boldsymbol{\theta}}-\mathbf{k}_{\bar{\theta}}^T \cdot d\boldsymbol{\theta} &= d(\mathbf{N}_{1,2})_{II} \end{aligned} \quad (2-1)$$

where $\mathbf{F}_{1,2}$ and $\mathbf{N}_{1,2}$ are the arm base reaction forces and moments, m_2 and \mathbf{I}_2 are the arm mass and inertia tensors, \mathbf{a}_2 and $\boldsymbol{\omega}_2$ are the arm linear acceleration and angular velocity vectors, \mathbf{F}_{01} and \mathbf{N}_{01} are the suspension reaction forces and moments, m_1 and \mathbf{I}_1 are the base mass and inertia tensors, \mathbf{a}_1 and $\boldsymbol{\omega}_1$ are the base linear acceleration and angular velocity vectors, \mathbf{k}_r and \mathbf{k}_θ are the translational and rotational stiffness coefficients, \mathbf{b}_r and \mathbf{b}_θ are the translational and rotational damping coefficients. Using the onboard sensors described above, this set of equations present the following unknowns, knowns, and measureable quantities:

unknowns : $m_1, \mathbf{I}_1, \mathbf{r}_2, \mathbf{k}_{\bar{r}_1}, \mathbf{b}_{\bar{r}_1}, \mathbf{k}_{\bar{\theta}}, \mathbf{b}_{\bar{\theta}}$

knowns : m_2, \mathbf{I}_2

measured : $d\mathbf{r}_1, d\dot{\mathbf{r}}_1, d\ddot{\mathbf{r}}_1$ (III w.r.t. II), $d\boldsymbol{\theta}, d\dot{\boldsymbol{\theta}}, d\ddot{\boldsymbol{\theta}}$ (II w.r.t. I), $d\mathbf{F}_{1,2}, d\mathbf{N}_{1,2}$

By measuring the motions induced by three rotation modes or the arm (rotation about the x, y and z axes in Frame IV—see Figure 2-2) and applying the six dynamic equations of motion, results in a total of 18 independent equations. Note that the arm rotation motions about the x, y and z axes are done individually, and require rotation of only the arm base joints (Frame IV—see Figure 2-2). The remaining joints are held fixed. This configuration is sufficient to produce the

dynamic forces required to generate needed vehicle excitations. Additionally, this maintains the generality of the algorithm developed in this chapter, as no specific manipulator kinematics are assumed (other than two base rotational joints).

2.3. Estimating The Dynamic Parameters

To solve for the unknown parameters, Equation 2-1 is first recast into the form $\mathbf{A} \mathbf{x} = \mathbf{F}$ (where \mathbf{A} is a known matrix of measured position values, \mathbf{x} is the vector of unknowns, and \mathbf{F} is a known vector of measured forces/torques). Two common methods to solve equations in the form $\mathbf{A} \mathbf{x} = \mathbf{F}$ are pseudo-inverse and Kalman filters. Both result in a least-squares solution to the problem.

In a pseudo-inverse solution process, a discrete set of measurements combined with the 18 equations are used to formulate the matrix \mathbf{A} and the vector \mathbf{F} . A solution to $\mathbf{A} \mathbf{x} = \mathbf{F}$ is simply given by: $\mathbf{x}=(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{F}$.

A more efficient solution is to use a Kalman filter [Gelb]. A Kalman filter is a multiple-input, multiple-output digital filter that can optimally estimate the states of a system based on noisy measurements. The state estimates are statistically optimal in that they minimize the mean-square estimation error. Here, rather than estimating \mathbf{x} based on one large matrix \mathbf{A} containing all the position measurements, \mathbf{x} is estimated based on a single set of measurements and an associated covariance matrix. With each new measurement set, the estimate is improved and the covariance updated. Since there are numerous articles in the literature describing Kalman filters, only a flow-diagram of the process is presented here (see Figure 2-3) [Bard, Gelb, Nikravesh]. In Figure 2-3, \mathbf{Q}_k models the uncertainty which corrupts the system model, \mathbf{R}_k models the uncertainty associated with the measurement and \mathbf{C}_k gives the total uncertainty of the state estimate.

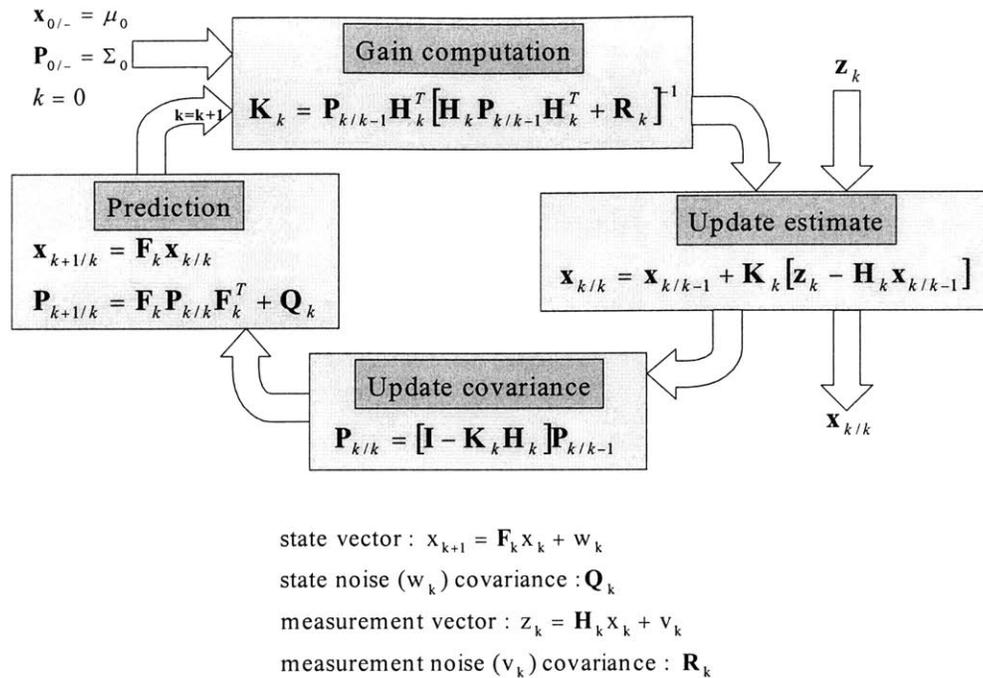


Figure 2-3: Flow-diagram of a Kalman Filter

2.4. A Metric For Observability

Although the above methods (pseudo-inverse and Kalman filter) produce solutions to the unknown dynamic parameters, a fundamental issue on the observability of unknown parameters is still to be addressed. Essentially, this provides a measure of accuracy of the current solution for the specific dynamic parameter. This is a difficult issue and a-priori tests are not available.

2.4.1. Classical observability metric

Classically, the concept of observability in the control literature is defined from a state model of the dynamic system. The idea is to determine if there are a sufficient number of independent equations relating the system states (from which these states may be inferred). Formally, a system is observable if the initial state can be determined by observing the output for some finite period of time. This metric is briefly outlined here and a discrete formulation is presented.

The linear system model (or state model) for a typical process in the absence of a forcing function is given by:

$$\begin{aligned}\dot{\bar{x}} &= F\bar{x} + G\bar{w} && \text{state model} \\ \bar{z} &= H\bar{x} + \bar{v} && \text{observation model}\end{aligned}\tag{2-2}$$

The discrete model is implemented by converting the continuous time model, given by:

$$\begin{aligned}\bar{x}_{k+1} &= \Phi_k \bar{x}_k + \Gamma_k \bar{w}_k && \text{state model} \\ \bar{z}_k &= H_k \bar{x}_k + \bar{v}_k && \text{observation model}\end{aligned}\tag{2-3}$$

where

\bar{x}_k is the (n x 1) system state vector at time t_k

Φ_k is the (n x n) transition matrix which relates \bar{x}_k to \bar{x}_{k+1}

Γ_k is the (n x n) process noise distribution matrix

\bar{w}_k is an (n x 1) white disturbance sequence with known covariance structure

\bar{z}_k is an (m x 1) measurement at time t_k

H_k is an (m x n) measurement matrix or observation matrix

\bar{v}_k is an (m x 1) white measurement noise sequence with known covariance

When the **F** matrix is constant in time and Equation 2-2 is linear, then the transition matrix is a function only of the time step Δt , and is given by the matrix exponential:

$$\Phi_k = e^{F\Delta t} = I + F\Delta t + \frac{(F\Delta t)^2}{2!} + \dots\tag{2-4}$$

It is assumed that process and measurement noise sequences are uncorrelated in time (white) and uncorrelated with each other. In practice, the transition matrix can often be written by inspection.

When Δt is much smaller than the dominant time constants in the system, a two term approximation is often sufficient [Kelly].

Consider the discrete n^{th} order constant coefficient linear system, $\bar{x}_{k+1} = \Phi_k \bar{x}_k$, for which there are m noise-free measurements, $z_k = Hx_k$ (where $k=0\dots m-1$), where each H is an (m x n) matrix. The sequence of the first i measurements can be written as:

$$\begin{aligned}
z_0 &= Hx_0 \\
z_1 &= Hx_1 = H\Phi x_0 \\
z_2 &= Hx_2 = H\Phi^2 x_0 \\
&\vdots \\
z_{i-1} &= Hx_{i-1} = H\Phi^{i-1} x_0
\end{aligned} \tag{2-5}$$

This can be written as the augmented set of equations $Z = \Xi^T x_0$. If the initial state is to be determined from this sequence of measurements, then $\Xi = \left[H^T \mid \Phi^T H^T \mid \dots \mid (\Phi^T)^{n-1} H^T \right]$ must have rank n . This definition is limited, in that it does not account for the effects of noisy data. Additionally, the unobservable state cannot be determined. To address both problems, a new mutual information based metric is proposed below.

2.4.2. Mutual information based metric

Consider a set of possible events with known probabilities of occurrence of p_1, p_2, \dots, p_n . If there is a measure of the amount of “choice” involved in selecting an event, $H(p_1, p_2, \dots, p_n)$, it is reasonable to require of it the following properties [Shannon]:

1. H should be continuous in the p_i .
2. $H(p_1, p_2, \dots, p_n)$ is a maximum for $p_k=1/n$ for $k=1 \dots n$. This implies that a uniform probability distribution possesses the maximum uncertainty
3. If a choice is broken down into two successive choices, the original H should be the weighted sum of the individual values of H .

It has been shown that the only H satisfying the three assumptions is of the form [Shannon]:

$$H = -K \sum_{i=1}^n p_i \log p_i \tag{2-6}$$

where K is a positive constant. Now consider the case where the signal is perturbed by noise during transmission i.e. the received signal is not necessarily the same as that sent out by the

transmitter. Two cases may be distinguished. If a particular transmitted signal always produces the same received signal, i.e. the received signal is a definite function of the transmitted signal, then the effect is called distortion. If this function has an inverse—no two transmitted signals produce the same received signal—distortion may be corrected. The case here is when the signal does not always undergo the same change in transmission. In this case the received signal, Y , is a function of the transmitted signal, X , and a second variable, the noise N : $Y=f(X,N)$. The noise is considered to be a chance variable. In general it may be represented by a suitable stochastic process [Shannon]. A finite number of states and a set of probabilities is assumed: $p_{\alpha,i}(\beta,j)$. This is the probability, if the channel is in state α and the symbol i is transmitted, that the symbol j will be received and the channel left in state β . Thus α and β range over the possible states, i over the possible transmitted signals and j over the possible received signals. In the case where successive symbols are independently perturbed by the noise there is only one state, and the channel is described by the set of transitional probabilities $p_i(j)$, the probability of transmitted symbol i being received as j [Shannon].

Thus, if a noisy channel is fed by a source, there are two statistical processes at work: the source and the noise. A number of important entropies can be calculated: the entropy of the source, $H(x)$; the entropy of the output of the channel, $H(y)$; the joint entropy of input and output, $H(x,y)$; the conditional entropies $H(y|x)$ and $H(x|y)$, the entropy of the output when the input is known and conversely. In the noiseless case $H(y)=H(x)$. These can be measured on a per-second or per-symbol basis. For a discrete channel transmitting a signal, an analogy with a sensor is made. The signal being read is the true value of the parameter being measured. The signal transmitted is the value that the sensor provides to a computer of the measured value (corrupted by noise).

The above definitions are used to understand the amount of information being transmitted by such a sensor i.e. the measure for observability. Consider the random variables x and y with joint probability distribution $p(x_i, y_j)$, $1 \leq i \leq N$, $1 \leq j \leq M$. The conditional entropy of x given y is defined as:

$$H(\mathbf{x} | \mathbf{y}) = - \sum_{i=1}^N \sum_{j=1}^M p(x_i, y_j) \log_2 p(x_i | y_j) \quad (2-7)$$

$H(x|y)$ can be interpreted as the average amount of uncertainty about x (the true value) after y (the measured value—sensor reading) has been revealed. Some important properties of the conditional entropy can be derived [Shannon]:

- (i) $H(x|y) \leq H(x)$ with equality if and only if x and y are independent
- (ii) $H(x, y) = H(y) + H(x|y) = H(x) + H(y|x)$

The average amount of information about x contained in y can now be defined in terms of the reduction in the uncertainty of x upon disclosure of y . Denoting this information by $\text{In}(x, y)$, define:

$$\text{In}(x, y) = H(x) - H(x|y) \quad (2-8)$$

With property (ii), it is easy to show that:

$$\text{In}(y, x) = H(y) - H(y|x) = \text{In}(x, y) \quad (2-9)$$

Thus, the information about x contained in y is equal to the information about y contained in x . For this reason, $\text{In}(x, y)$ is called the average mutual information between x and y . From property (i), $\text{In}(x, y) \geq 0$ with equality iff x and y are independent. As a direct consequence of the definition of $\text{In}(x, y)$:

$$\text{In}(x, y) = \sum_{i=1}^N \sum_{j=1}^M p(x_i, y_j) \log_2 \frac{p(x_i, y_j)}{p(x_i)p(y_j)} \quad (2-10)$$

To develop the relationship $p(x_i, y_j)$, sensor noise is now modeled. A single observation of a

point (\bar{x}) is modeled as a Gaussian probability distribution centered at \bar{y} . A Gaussian to model uncertainty in sensor data is based on two important observations. The use of the mean and the covariance of a probability distribution function is a reasonable form to model sensor data and is a second order linear approximation. This linear approximation corresponds to the use of a Gaussian (having all higher moments of zero). Additionally, based on the central limit theorem, the sum of a large number of independent variables has a Gaussian distribution regardless of their individual distributions [Kelly]. For example, the canonical form of the Gaussian distribution in 3 dimensions depends on the standard deviations of the measurement, $\sigma_{x,y,z}$, a covariance matrix (C) and the mean measurement (\bar{y}) [Ljung, Nikravesh, Smith]:

$$p(\bar{x}' | \bar{y}) = \frac{1}{(2\pi)^{n/2} \sqrt{|C|}} \exp\left(-\frac{1}{2}(\bar{y} - \bar{x}')^T C^{-1}(\bar{y} - \bar{x}')\right)$$

$$\text{where } C = \begin{bmatrix} \sigma_x^2 & \rho_{xy}\sigma_{xy}\sigma_{xy} & \rho_{zx}\sigma_{zx}\sigma_{zx} \\ \rho_{xy}\sigma_{xy}\sigma_{xy} & \sigma_y^2 & \rho_{yz}\sigma_{yz}\sigma_{yz} \\ \rho_{zx}\sigma_{zx}\sigma_{zx} & \rho_{yz}\sigma_{yz}\sigma_{yz} & \sigma_z^2 \end{bmatrix} \quad (2-11)$$

where the exponent is called the Mahalanobis distance. For uncorrelated measured data $\rho=0$. This can be generalized for an n dimensional sensor. $H(x)$ and $H(x|y)$ can be explicitly defined in terms of a given sensor:

$$H(x) = \sum_{i=1}^n p_i \log p_i \quad (2-12a)$$

where, for example in a special case of a discrete sensor $p_i = \frac{1}{n}$ and $n =$ number of sensor discrete states.

$$H(x | y) = \sum_{i=0}^{n-1} q_i \log q_i \quad \text{where } q_i = \frac{\int_a^b p(\bar{x}' | \bar{y}) d\bar{x}'}{\int_{\min}^{\max} p(\bar{x}' | \bar{y}) d\bar{x}'}$$

$$\text{and } \begin{cases} a = \min + i \left(\frac{\max - \min}{n} \right); b = \min + (i + 1) \left(\frac{\max - \min}{n} \right) \\ \max = \text{maximum sensor reading;} \\ \min = \text{minimum sensor reading} \\ p(\bar{x}' | \bar{y}) \text{ is obtained from Equation 2 - 11} \end{cases} \quad (2-12b)$$

$\ln(y,x)$, reflects the information content of the current estimate of the dynamic parameter being estimated. In other words, increasing certainty of a parameter estimate is reflected in the increasing value of $\ln(y,x)$ associated with that parameter. This metric makes no assumption on the noise statistics (Gaussian, etc.). It is convenient to establish the details using Gaussian noise.

2.5. Formulation Of Exciting Trajectories

Using the observability metric defined in Section 2.4, a method to formulate the appropriate arm-exciting trajectory is now developed. The idea is to use the observability metric to modify the arm motion, thus increasing the information associated with the dynamic parameter estimates.

From Section 2 a set of differential algebraic equations of motion of the form: $\mathbf{Ax} = \mathbf{F}$ is obtained. For the robot system in the situation considered here, the arm excitation function is sinusoidal, namely $f(t)=a_0+a.\sin(\omega t)$ (Equation B-1). The only parameters that can be varied are the amplitude, frequency and offset of the sinusoidal excitation function i.e. amplitude (a), frequency (ω) and offset (a_0) of motion of the robotic arm.

Based on the associated information (Section 2-4) for the parameter estimate vector, \tilde{x}_k , a cost function is defined as follows:

$$\begin{aligned}
V(\mathbf{d}) &= \frac{1}{2} \sum_i \left(1 - \left(1 - \frac{\text{In}_{x_i}}{\text{In}_{x_i}^{\max}} \right) \frac{\text{In}_{x_i}}{\text{In}_{x_i}^{\max}} \right) \\
&= \frac{1}{2} \sum_i \left(1 - \frac{\text{In}_{x_i}}{\text{In}_{x_i}^{\max}} + \left(\frac{\text{In}_{x_i}}{\text{In}_{x_i}^{\max}} \right)^2 \right) = \frac{1}{2} \sum_i r_i^2(\mathbf{d})
\end{aligned} \tag{2-13}$$

where i is summed over the number of dynamic parameters to identify. In_{x_i} is the current information associated with parameter estimate x_i . $\text{In}_{x_i}^{\max}$ is the current maximum information (observability) associated with any of the parameter estimates. The control parameters vector $\mathbf{d} \in R^3$ consists of the amplitude, frequency and offset of the arm excitation function. Note, in this cost function the information associated with each parameter is weighted such that parameters with a higher uncertainty receive a higher weight. Further, this cost function may be easily amended to include weightings that reflect the relative importance of the individual dynamic parameters.

A numerical minimization routine is applied to this cost function, by changing the excitation function in amplitude, frequency and offset (the current estimates for the unknowns are used here). By assembling the terms $r_i(\mathbf{d})$ into a vector $\mathbf{R}(\mathbf{d})$, given as:

$$\mathbf{R}(\mathbf{d}) = \{r_1(\mathbf{d}), \dots, r_{n_m}(\mathbf{d})\}^T \tag{2-14}$$

the control parameters \mathbf{d} must be chosen so that the residual vector \mathbf{R} is as small as possible. The quadratic cost function $V(\mathbf{d})$ of Equation 2-13 becomes:

$$V(\mathbf{d}) = \frac{1}{2} \mathbf{R}^T(\mathbf{d})\mathbf{R}(\mathbf{d}) \tag{2-15}$$

The problem of finding \mathbf{d} from $V(\mathbf{d})$ is a nonlinear least-squares problem [Nikravesh]. If the vector $\mathbf{R}(\mathbf{d})$, is continuous, and if both first and second-order derivatives are available, then the nonlinear least-squares problem can be solved by standard unconstrained optimization methods.

Otherwise, a method that requires only the first derivatives of $\mathbf{R}(\mathbf{d})$ must be used [Serban]. The first derivative of Equation 2-15 with respect to the design parameters, \mathbf{d} , is defined as:

$$G(\mathbf{d}) = \sum_{i=1}^{n_m} \nabla r_i(\mathbf{d}) r_i(\mathbf{d}) = \mathbf{J}^T(\mathbf{d}) \mathbf{R}(\mathbf{d}) \quad (2-16)$$

where $\mathbf{J}(\mathbf{d}) \in R^{n_m \times 3}$ is the Jacobian matrix of $\mathbf{R}(\mathbf{d})$ with respect to the design parameters. The second derivative of Equation 2-15 with respect to the design parameters, \mathbf{d} , is defined as:

$$H(\mathbf{d}) = \sum_{i=1}^{n_m} [\nabla r_i(\mathbf{d}) r_i(\mathbf{d})^T + \nabla^2 r_i(\mathbf{d}) r_i(\mathbf{d})] = \mathbf{J}^T(\mathbf{d}) \mathbf{J}(\mathbf{d}) + \mathbf{S}(\mathbf{d}) \quad (2-17)$$

where $\mathbf{S}(\mathbf{d}) \in R^{3 \times 3}$ is part of $H(\mathbf{d})$ that is a function of second derivatives of $\mathbf{R}(\mathbf{d})$. Thus the knowledge of $\mathbf{J}(\mathbf{d})$ supplies $G(\mathbf{d})$ and the part of $H(\mathbf{d})$ dependent on first-order derivative information, but not on the second order part $\mathbf{S}(\mathbf{d})$. Levenberg-Marquardt methods simply omit $\mathbf{S}(\mathbf{d})$, and base the step selection (\mathbf{d}') on the approximation given by [Serban]:

$$V(\mathbf{d} + \mathbf{d}') = V(\mathbf{d}) + G^T(\mathbf{d}) \mathbf{d}' + \frac{1}{2} \mathbf{d}' \mathbf{J}^T(\mathbf{d}) \mathbf{J}(\mathbf{d}) \mathbf{d}' \quad (2-18)$$

Equation 2-18 leads to the following optimization procedure:

$$\mathbf{d}^{(k+1)} = \mathbf{d}^{(k)} + \alpha_k \mathbf{d}'^{(k)} \quad (2-19)$$

with

$$\mathbf{d}'^{(k)} = \frac{-G(\mathbf{d}^{(k)})}{\mathbf{J}^T(\mathbf{d}^{(k)}) \mathbf{J}(\mathbf{d}^{(k)})} \quad (2-20)$$

$$\alpha_k = \arg \min_{\alpha} [V(\mathbf{d}^{(k)} + \alpha \mathbf{d}'^{(k)})]$$

$\mathbf{d}'^{(k)}$ given by Equation 2-20 represents a descent direction. Thus, using equations 2-19 and 2-20, the amplitude, frequency and offset of the robot arm are refined during the identification process, leading to an optimal excitation trajectory. Note, in physical systems, evaluation of the information metric and optimization of arm motion should be carried out at time intervals larger

than the sampling time. This permits the physical system to respond to the changes in arm motion.

2.6. Results

2.6.1. Simulation studies

Two tests have been conducted using a 3D simulation of a mobile robot system with a manipulator and suspension compliance. The first uses a constant parameter excitation function to drive the arm motion. The second uses a variable parameter excitation function (based on the formulation presented above) to drive the arm. The parameter identification results are compared. The system was simulated for 10 seconds. The manipulator arm mass is assumed to be 1Kg and inertias $I_x=0.02\text{kg}\cdot\text{m}^2$ $I_y=0.001\text{kg}\cdot\text{m}^2$ $I_z=0.02\text{kg}\cdot\text{m}^2$ $I_{xz}=0 \text{ kg}\cdot\text{m}^2$ $I_{yz}=0 \text{ kg}\cdot\text{m}^2$ $I_{xy}=0 \text{ kg}\cdot\text{m}^2$. In the simulation, sensor data is corrupted by adding white noise of up to 10% of the maximum sensed value. Evaluation of the information metric and refining arm motion occur every 0.4 secs, with a sampling time of 0.005 secs.

For the first case, the constant parameter excitation function is given by the form:

$$f(t) = a_0 + a \cdot \sin(\omega t) = \pi/4 + 2\pi/9 \cdot \sin(\pi/2 t) \quad (2-21)$$

This was chosen based on the arm kinematic and dynamic limitations (i.e. to be well within the manipulator capabilities). Figures 2-4 and 2-5 show the arm excitation functions for the two test cases. For a sensor with n-bit precision (i.e. 2^n possible values), the maximum mutual information associated with the reading is n bits (i.e. no uncertainty, see Equation 2-9). In both test cases, a 10-bit accuracy sensor is assumed i.e. 2^{10} possible values. Figure 2-6 shows the value for the mutual information metric in identifying the stiffness in θ_p for the two test cases. It is seen that by using the variable parameter excitation function (as opposed to constant parameter excitation function), the amount of information associated with the unknown parameter, $I_{n\theta}$, has

low uncertainty (i.e. identified parameter value has high quality). Figure 2-7 shows the convergence in identification of several parameters for the two test cases. In general, using the variable parameter excitation function results in faster and more accurate convergence of the estimates to the true values. Table 2-1 presents the identification results of the 22 unknowns (see Section 2) using both arm excitation tests. The average percentage error shows an improvement of almost a factor of six for the variable parameter over the constant parameter excitation function. The average computational time per evaluation step for 22 unknown parameters with $\ln_{x_i}^{\max} = 10$ bits on a PIII 750MHz platform is 75 ms.

For comparison, the simulation is also run using a parameter quality metric based on the condition number of a matrix formed by the sensed values (matrix **A** in section 3). The arm motion is refined to generate lower condition numbers (i.e. matrix **A** is better behaved w.r.t. inversion). On average, parameter estimates converge an order of magnitude faster (in simulation time) using the information-based quality metric.

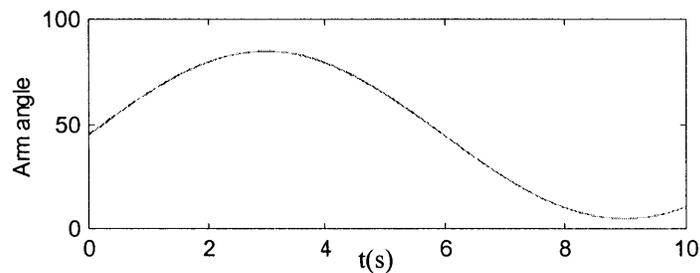


Figure 2-4: Constant parameter arm motion

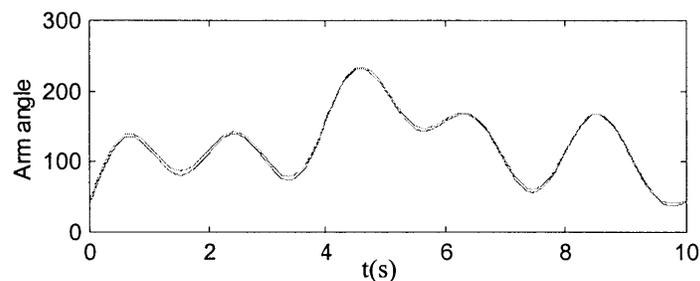
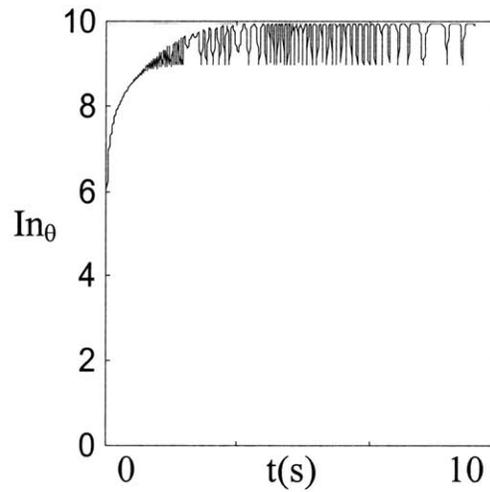
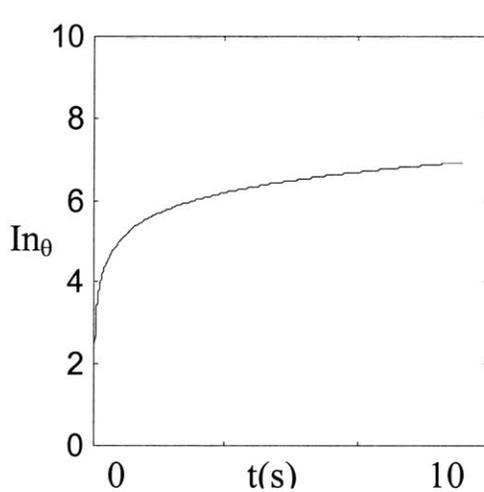


Figure 2-5: Variable parameter arm motion

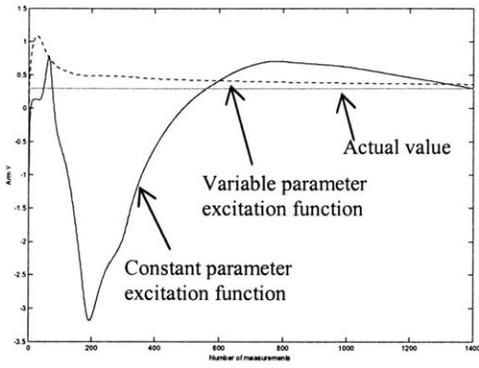
Table 2-1: Simulation parameter identification

Parameter	True value	Constant parameter excitation function	Variable parameter excitation function
Mass (kg)	3.0	4.021	3.655
Inertia I_x ($\text{kg}\cdot\text{m}^2$)	0.15	0.123	0.144
Inertia I_y ($\text{kg}\cdot\text{m}^2$)	0.10	0.054	0.089
Inertia I_z ($\text{kg}\cdot\text{m}^2$)	0.20	0.119	0.177
Inertia I_{xz} ($\text{kg}\cdot\text{m}^2$)	0.0	0.045	0.011
Inertia I_{yz} ($\text{kg}\cdot\text{m}^2$)	0.03	0.071	0.047
Inertia I_{xy} ($\text{kg}\cdot\text{m}^2$)	0.0	0.037	0.014
c.g. x (m)	0.01	0.030	0.012
c.g. y (m)	0.3	0.512	0.360
c.g. z (m)	0.1	-1.053	0.042
Damping x (kg/s)	100	142.554	78.362
Damping y (kg/s)	100	139.956	75.559
Damping z (kg/s)	300	22.368	16.757
Damping θ_p (kg/s)	200	78.665	195.264
Damping θ_r (kg/s)	300	105.061	266.254
Damping θ_v (kg/s)	400	133.714	333.860
Stiffness x (kg/s^2)	1000	1103.281	979.494
Stiffness y (kg/s^2)	1000	1094.529	1244.836
Stiffness z (kg/s^2)	500	390.469	375.479
Stiffness θ_p (kg/s^2)	2000	945.103	1934.569
Stiffness θ_r (kg/s^2)	2500	1950.426	2394.257
Stiffness θ_v (kg/s^2)	3000	2402.537	2891.637
% Avg. Error		108.8	18.9

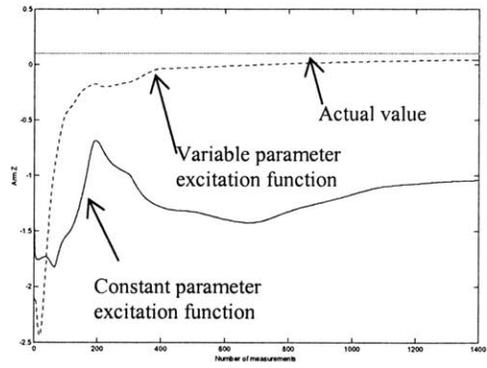


(a) constant parameter excitation function (b) variable parameter excitation function

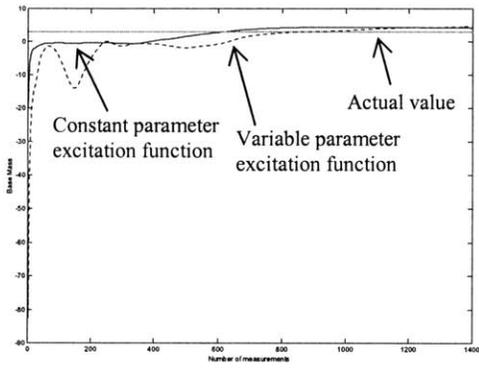
Figure 2-6: Mutual information metric for stiffness θ_p



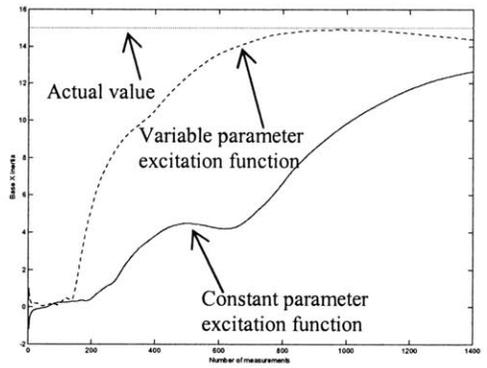
(a) c.g. Y



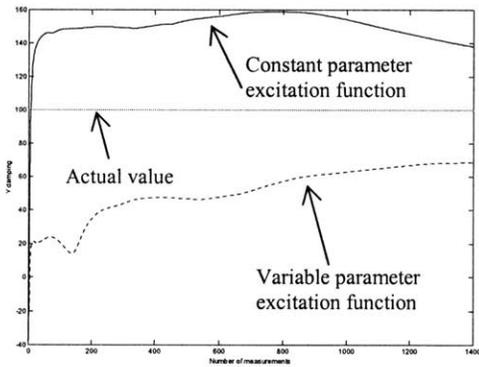
(b) c.g. Z



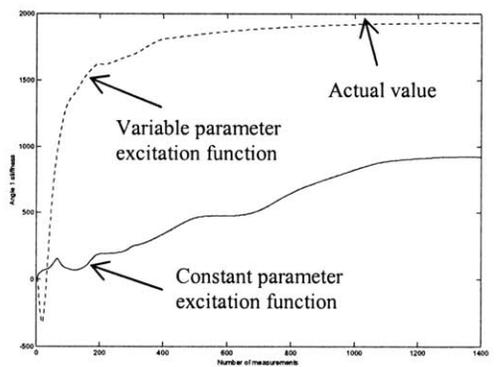
(c) base mass



(d) Inertia X

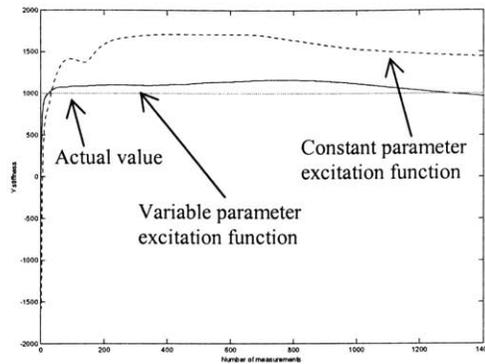


(e) damping Y



(f) Stiffness θ_p

Figure 2-7: Example of identification converge curves



(g) stiffness Y

Figure 2-7 (cont): Example of identification converge curves

2.6.2. Experimental studies

Experiments were performed on a four-wheeled robot with a four DOF manipulator arm mounted on a 6-axis force/torque sensor (see Figure 2-8). On-board sensors also include a two-axis inclinometer. An off-board computer system (Pentium 166 MHz) is used for real-time control, data acquisition, and data processing. All programs are written in C++ operating on Windows NT. Due to the absence of an accelerometer in the experimental platform, tests only demonstrate the identification of rotational dynamic components (inertia, stiffness, damping of roll/pitch axes and the location of the center of gravity). A more detailed description of the hardware is presented in Appendix D.

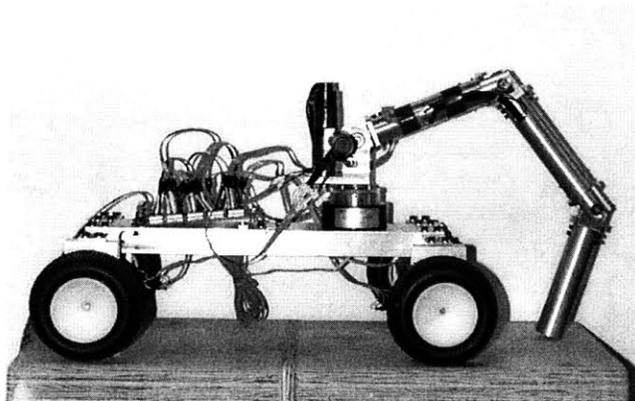


Figure 2-8: Experimental mobile manipulator

Before any parameters identification tests are run, the real dynamic parameters must be identified. Conventional laboratory approaches are used here [Bard, Nikravesh, Serban]. Mass is measured using a precision weighing scale. The location of the center of mass is found by titling the vehicle on one axis supported at two points, and measuring the reaction forces at the support points. Inertia is measured using pendulum oscillatory tests. Stiffness is measured by measuring deflection as a function of added load. Damping is measured by fitting the impulse response of the system to a second order equation. Linear models are used for stiffness and damping tests. Use of the force/torque sensor in the experimental system eliminates the need to measure the actual arm inertia tensor (see Equation 2-1).

As in the simulation studies, two tests have been conducted using the constant parameter excitation function and the variable parameter excitation function to drive the arm. The parameter identification results are compared. The experiments were run for approximately 10 seconds. Figures 2-9 and 2-10 show the arm excitation functions for the two test cases. Figure 2-11 shows the inclinometer pitch reading for the 2 test cases.

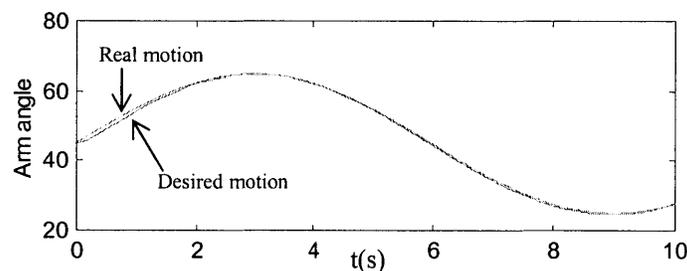


Figure 2-9: Constant parameter arm motion

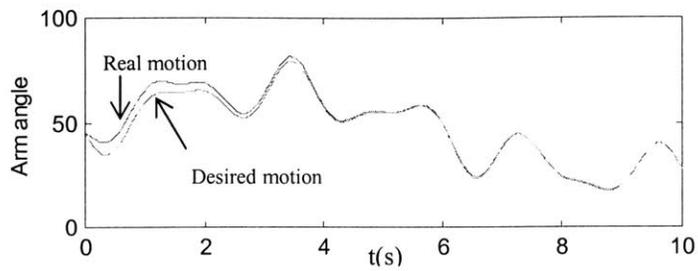
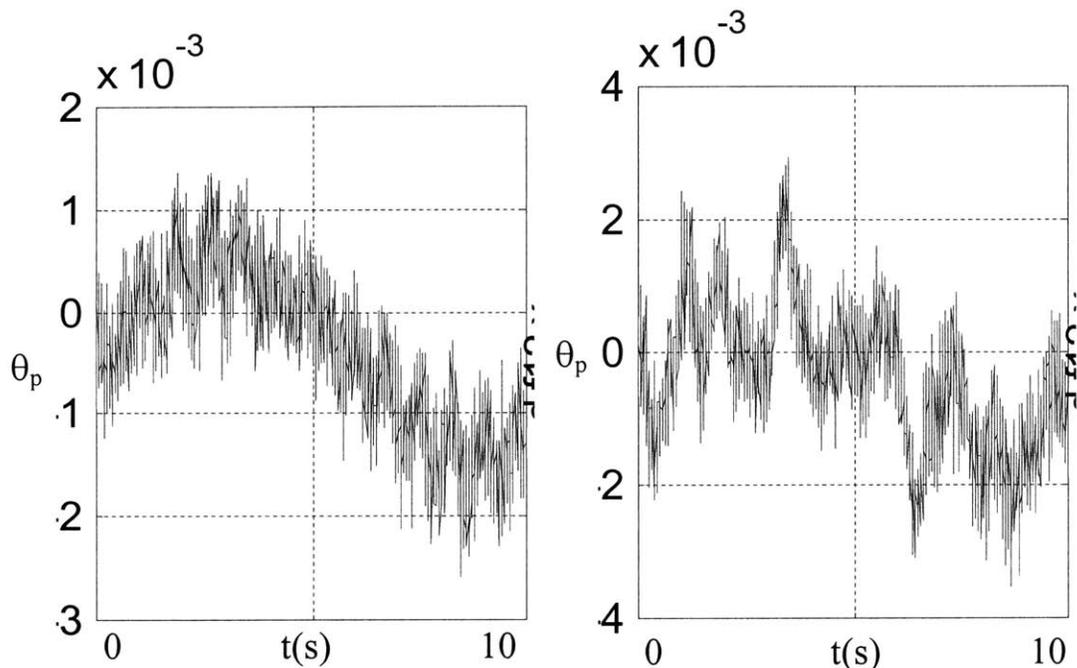


Figure 2-10: Variable parameter arm motion



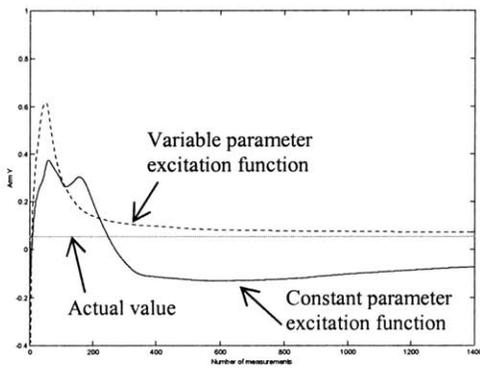
(a) constant parameter excitation function (b) variable parameter excitation function

Figure 2-11: Inclinometer pitch reading (radians)

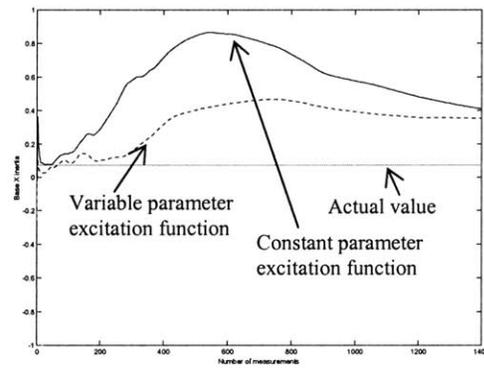
Table 2-2: Experimental parameter identification

Parameter	True value	Constant parameter excitation function	Variable parameter excitation function
Inertia x	0.072	0.041	0.035
c.g. y	0.054356	-0.072	0.072
c.g. z	0.036195	-0.091	0.052
Damping θ_p	1.5	7.543	1.459
Stiffness θ_p	317.0	354.711	327.885
% Avg. Error		208.4	26.8

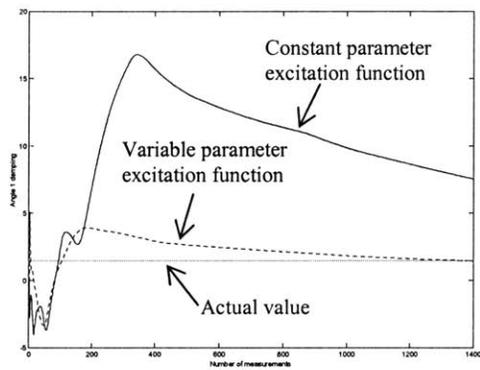
Table 2-2 presents the identification results of the five unknowns using both arm excitation tests. The average percentage error shows an improvement of almost a factor of 8 for the variable parameter over the constant parameter excitation function. Figure 2-12 shows the convergence in identification of several parameters for the two test cases. In addition to the significant corruption of data due to sensor noise, inaccuracies in laboratory measurements of “true” vehicle dynamic parameters contribute to the errors seen.



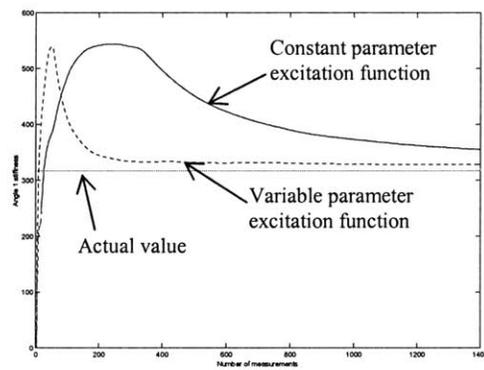
(a) c.g. Y



(b) Base X inertia



(c) Damping θ_p



(d) Stiffness θ_p

Figure 2-12: Example of identification converge curves

2.7. Conclusions

This chapter presented an algorithm for dynamic parameter estimation based on iterative excitation of vehicle dynamics. This algorithm enabled mobile robots in field environments to efficiently estimate their dynamic parameters, including the mass, location of center of gravity, inertia, base compliance and damping. The algorithm uses an onboard robotic arm to generate base motions, which are measured with simple onboard sensors, and fit to a physical model. A mutual information theoretic basis for a metric on parameter identification is developed. This metric provides a measure on how well a given parameter's value is known. Using this metric, the arm trajectory is defined. Simulations and experimental results show the effectiveness of this algorithm.

Environment Modeling

3.1. Introduction

The next step in compensating for field robot model uncertainties is to develop an algorithm for effective modeling of the robots' environment.

This chapter presents an algorithm for geometric environment model building using vision, for robot teams working cooperatively in unstructured field environments. In such scenarios, it is often not possible to have a priori environment models. Robots need to build these models using available sensory data. A number of problems make this non-trivial. These include the task uncertainty, location and orientation uncertainty of the individual robots, and visual occlusions (due to obstacles, work piece, other robots). If the systems are equipped with cameras mounted on articulated mounts, intelligent planning of the camera motion can often alleviate occlusion problems, providing an accurate geometrical model of the task and environment. If the system consists of cooperating robots, planning the behavior of these information-sharing systems can further improve the system performance.

It is assumed that dimensional geometric information is relevant and required for robots to perform their operations, such as the construction of field facilities. It is also assumed that the system consists of two or more mobile robots working in an unknown environment (such as constructing a planetary structure—see Figure 3-1). It is also assumed that there are no physical interactions between the robots and that the environment is static. Each has a 3D vision system

mounted on an articulated arm. Sensing and sensor placement is limited, resulting in occlusions and uncertainties. Again, the objective is to efficiently build a geometrically consistent dimensional model of the environment and target, available to all robots. The key idea is that the algorithm builds an environment and task model by fusing data from all robots, providing both improved accuracy and knowledge of regions not visible by all robots. Using this algorithm, the individual robots can also be positioned “optimally” with respect to the target (see Chapter 4).

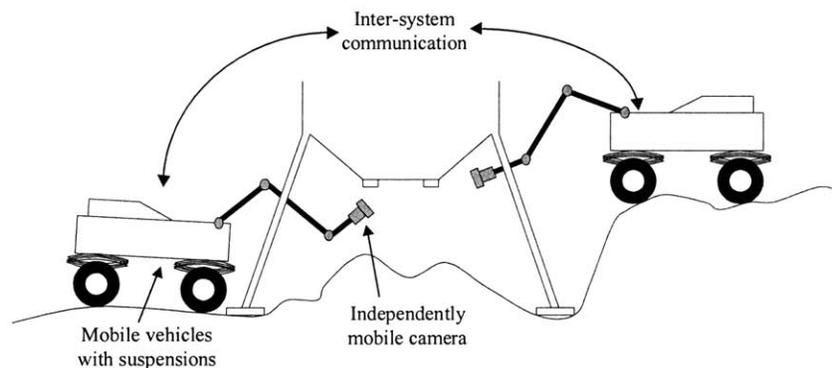


Figure 3-1: Cooperative mapping by robots

3.2. Algorithm Description

3.2.1. Overview

The first step in cooperative model building is to visually construct a model of the local environment, including the locations of the task elements and the robots themselves. We assume that only the geometry of the task elements (such as the parts of a solar panel that needs to be assembled [Huntsberger]) are well known. Obstacles and robot positions are unknown.

Figure 3-2 outlines the map building algorithm. The algorithm consists of two parts. In the first part, the articulated cameras cooperatively scan the region around a target, generating a 3D geometric model. This allows the robots to locate themselves and the obstacles in the target reference frame. In the second part, these models are used to find an optimum target viewing pose for the multiple camera systems (see Chapter 4). The 3D map is modeled as a probabilistic discretized occupancy grid. Every voxel in the map has a value for probability-of-occupancy that

ranges from 0 (empty) to 1 (occupied). A value of 0.5 indicates maximum uncertainty in occupancy of the voxel. The process is initialized by visually finding the target and robots in a common reference frame. This is done by “looking around” and matching the known target element geometric CAD model with visual data [Lara]. Next, a new camera pose is found for each of the cameras by defining and evaluating a rating function (RF) over the known environment map, subject to kinematic constraints of the sensor placement mechanisms for the individual robots. Then, the cameras move to their new poses and acquire 3D data. Based on the camera mount kinematics, the motions of the cameras are known. Small motions of the robot base (due to suspension compliance) and errors in camera mounts lead to additional uncertainties. These are accounted for by measuring common features during the camera motion (section 3.2.5). Finally, the new data and its associated uncertainty are fused with the current environment map, resulting in an updated probabilistic environment map.

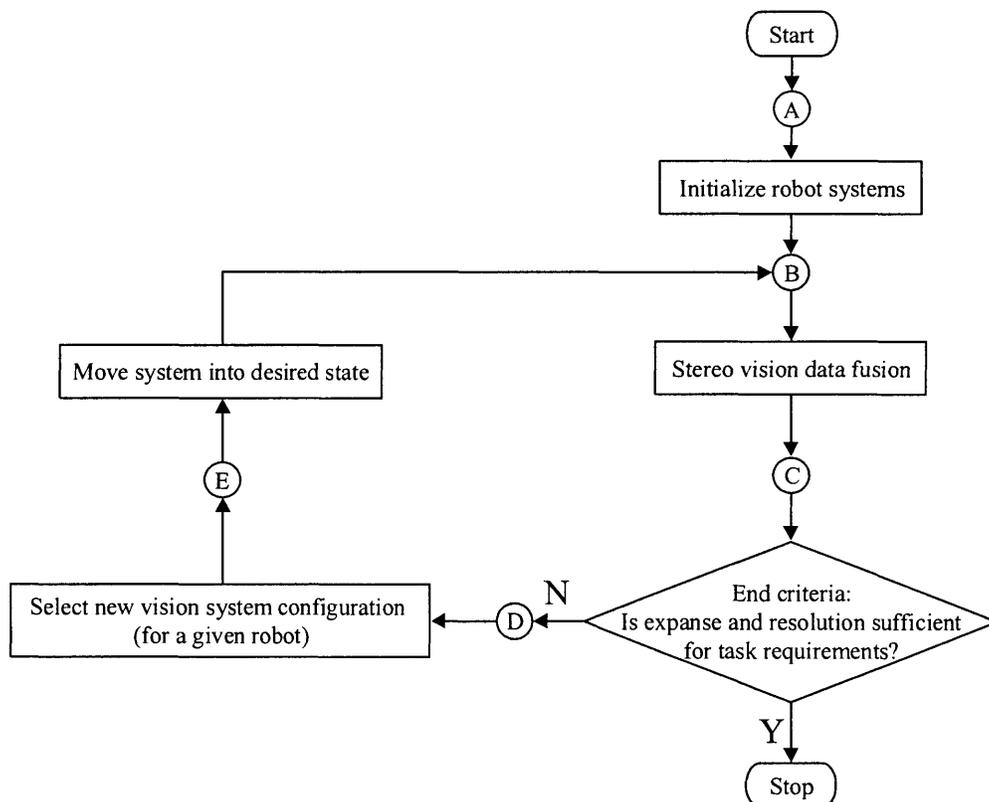


Figure 3-2: Outline of model building and placement algorithm

3.2.2. Algorithm initialization

As described above, a common target must be located to establish a common inertial reference frame between the robots and the environment. Searching for the target by moving the robot cameras can be done in many ways, such as exhaustive raster scanning, random walking, tracking “space filling curves”, and model-based image understanding methods [Luo, Tarabanis]. In this study, camera positioning for target searching is done in the same way as camera positioning for environment model building (described in Section 3.2.4.). The initialization process is outlined in Figure 3-3. At this stage, the environment model (occupancy grid with associated measurement uncertainties) is considered empty i.e. no points are known. The first stereo range map is taken and fused to the model. This is described next.

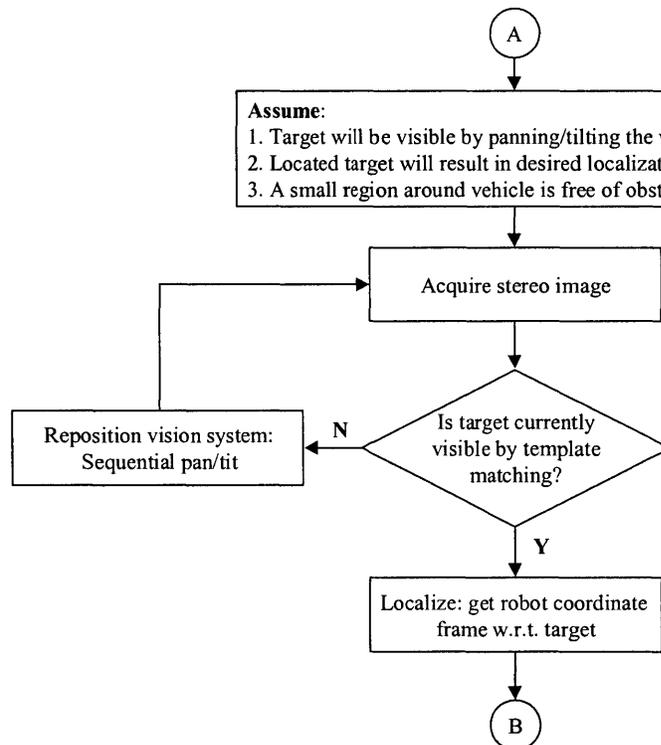


Figure 3-3: Flowchart of the initialization of environment mapping algorithm

3.2.3. Data modeling and fusion

At any time, the cameras on each mobile robot can only observe a small part of their environment.

However, measurements obtained from multiple view-points can provide reduced uncertainty, improved accuracy, and increased tolerance in estimating the location of observed objects [Smith]. To fuse multiple range measurements of a feature by sensors, a statistical model of sensor uncertainty is employed (see Figure 3-4). Current and previous range sensor measurements and their uncertainty models can be integrated to give an updated probabilistic geometric model of the environment.

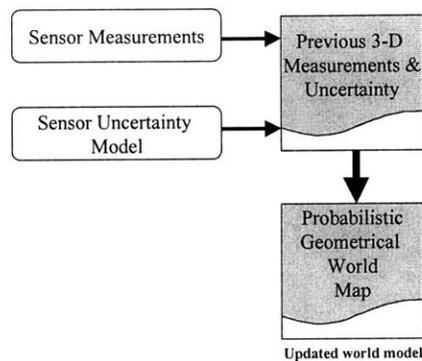


Figure 3-4: 3-D range measurement fusion with sensor uncertainty

A single range observation of a point (\bar{x}) is modeled as a 3-D Gaussian probability distribution centered at \bar{x} , based on two important observations. First, the use of the mean and covariance of a probability distribution function is a reasonable form to model sensor data and is a second order linear approximation [Smith]. This linear approximation corresponds to the use of a Gaussian (having all higher moments of zero). Second, from the central limit theorem, the sum of a number of independent variables has a Gaussian distribution regardless of their individual distributions.

The standard deviations along the three axes of the distribution correspond to estimates of the uncertainty in the range observation along these axes. These standard deviations are a function of intrinsic sensor parameters (such as camera lens shape accuracy) as well as extrinsic sensor parameters (such as the distance to the observed point or feature).

The basic stereo triangulation formula for perfectly aligned cameras of epipolar geometry is shown below [Kelly]:

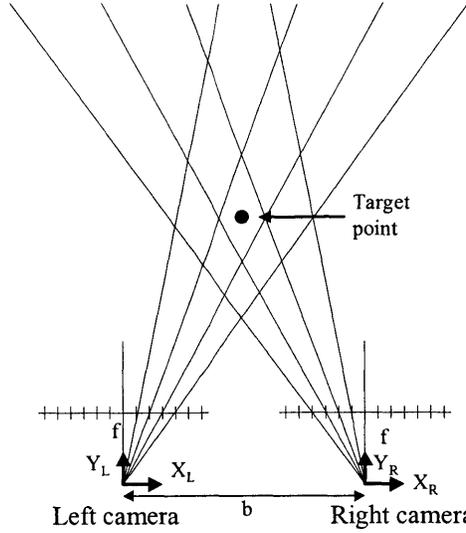


Figure 3-5: Stereo imaging error evaluation

From similar triangles we get:

$$\begin{aligned} \frac{X_L}{Y_L} &= \frac{x_L}{f} & \frac{X_R}{Y_R} &= \frac{x_R}{f} \\ X_L - X_R &= \frac{Y(x_L - x_R)}{f} & & (3-1) \\ b &= \frac{Yd}{f} \end{aligned}$$

Uppercase letters signify scene (3D) quantities and lowercase signify image plane coordinates. Once stereo correspondence is performed, the Y coordinate can be determined from the disparity, d, of each pixel. Then the X and Z coordinates come from the known unit vector through each pixel which is given by the camera kinematic model. A quantity of disparity is defined as:

$$\delta = \frac{d}{f} = \frac{b}{Y} \quad (3-2)$$

which then gives:

$$Y = \frac{b}{\delta} \quad (3-3)$$

Thus if $s_{\delta\delta}$ is the uncertainty in disparity, then the uncertainty in range is:

$$s_{yy} = J s_{\delta\delta} J^T \quad (3-4)$$

where the jacobian, J, is given by:

$$J = \frac{\partial Y}{\partial \delta} = \frac{-b}{\delta^2} \quad (3-5)$$

If the disparity uncertainty is equal to the constant angle subtended by one pixel, then the range variance goes as the fourth power of the range itself:

$$s_{yy} = \left(\frac{b^2}{\delta^4} \right) s_{\delta\delta} = \left(\frac{Y^4}{b^2} \right) s_{\delta\delta} \quad (3-6)$$

and hence, the standard deviation of the range goes with range squared.

For most range sensing systems, this model can be approximated as:

$$\begin{aligned} \sigma_{x,y,z} &= f(\text{extrinsic parameters, intrinsic parameters}) \\ &\approx S \cdot T_{x,y,z} \cdot L^n \end{aligned} \quad (3-7)$$

where S is an intrinsic parameter uncertainty constant, $T_{x,y,z}$ is an extrinsic parameter uncertainty constant, L is the distance to the feature/environment point, and n is a constant (typically 2). Provided two observations are drawn from a normal distribution, the observations can be merged into an improved estimate by multiplying the distributions. Since the result of multiplying two Gaussian distributions is another Gaussian distribution, the operation is symmetric, associative, and can be used to combine any number of distributions in any order [Stroupe]. The canonical form of the Gaussian distribution in n dimensions depends on the standard distributions, $\sigma_{x,y,z}$, a covariance matrix (C) and the mean (\bar{x}) [Stroupe, Smith]:

$$p(\bar{x}' | \bar{y}) = \frac{1}{(2\pi)^{n/2} \sqrt{|C|}} \exp\left(-\frac{1}{2}(\bar{y} - \bar{x}')^T C^{-1}(\bar{y} - \bar{x}')\right)$$

$$\text{where } C = \begin{bmatrix} \sigma_x^2 & \rho_{xy} \sigma_{xy} \sigma_{xy} & \rho_{zx} \sigma_{zx} \sigma_{zx} \\ \rho_{xy} \sigma_{xy} \sigma_{xy} & \sigma_y^2 & \rho_{yz} \sigma_{yz} \sigma_{yz} \\ \rho_{zx} \sigma_{zx} \sigma_{zx} & \rho_{yz} \sigma_{yz} \sigma_{yz} & \sigma_z^2 \end{bmatrix} \quad (3-8)$$

For un-correlated measured data $\rho=0$. The formulation in Equation 3-8 is in the spatial coordinate

frame. However, all measurements are made in the camera (or sensor) coordinate frame. This problem is addressed by a transformation of parameters from the observation frame to the spatial reference frame as follows [Stroupe]:

$$C_{\text{transformed}} = R(-\bar{\theta})^T \cdot C \cdot R(-\bar{\theta}) \quad (3-9)$$

where $R(\theta)$ is the rotation matrix between the two coordinate frames. The angle of the resulting principal axis can be obtained from the merged covariance matrix:

$$C_{\text{merged}} = C_1 \left(I - C_1 (C_1 + C_2)^{-1} \right) \quad (3-10)$$

where C_i is the covariance matrix associated with the i^{th} measurement (see Figure 3-6). Additionally, a translation operation is applied to the result from Equation 3-8, to bring the result into the spatial reference frame. Determining this transformation matrix is described in section 3.2.5.

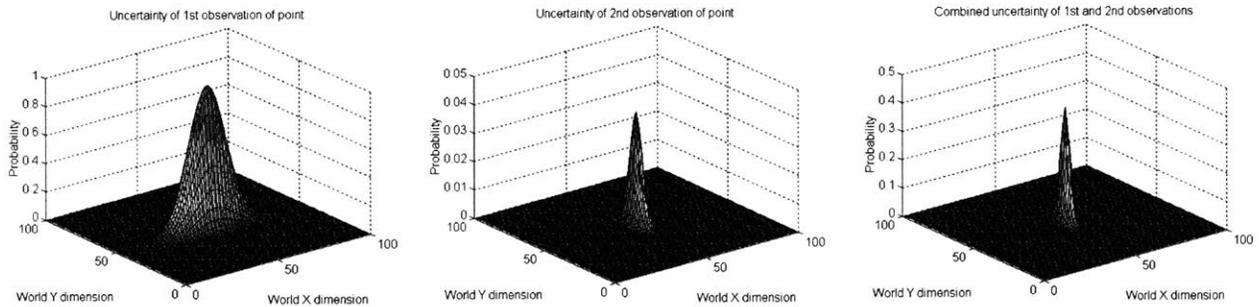


Figure 3-6: Sample of merging two Gaussian probability distributions (2-D case)

To contribute to the probabilistic occupancy environment model, all measured points corresponding to obstacles are merged. That is, all measured points falling in a particular grid voxel contribute to the error analysis associated with that voxel. Note that grid voxels falling within the field of view of the vision system that correspond to empty space result in no contribution to the uncertainty model (since these are not measured). However, these points are tabulated as measured (or known) points. This will be used to select the pose for the vision system. Obtaining the covariance matrix (C) and

the mean (\bar{x}) while accounting for uncertainties in camera motion, is described in Section 3.25.

The data fusion process is summarized in Figure 3-7.

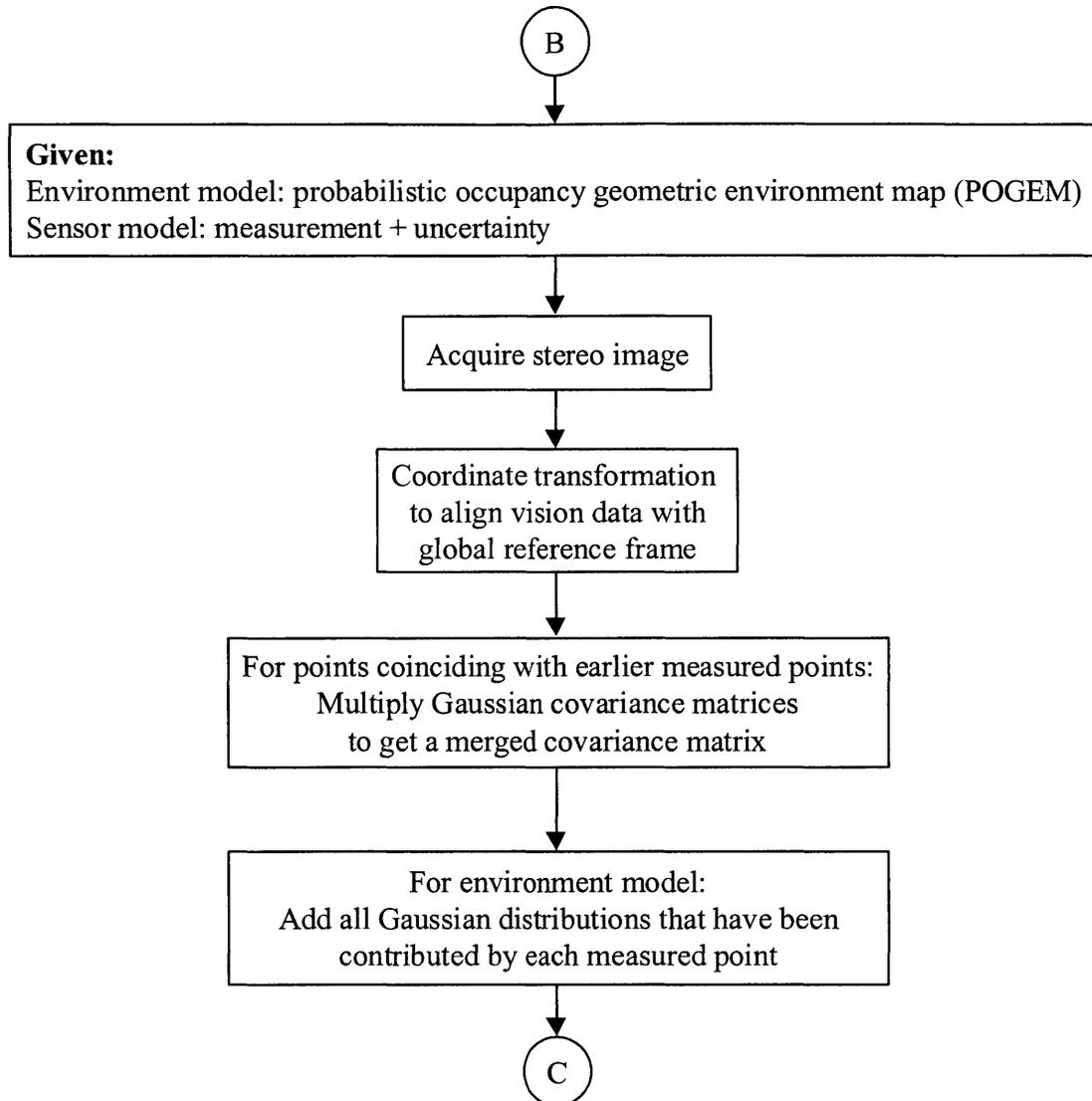


Figure 3-7: Flowchart for data fusion using known vision system motion

3.2.4. Vision system pose selection

A rating function is used to determine the next pose of the camera from which to look at the unknown environment. The aim is to acquire new information of the environment that would lead to a more complete environment map. In selecting this new camera pose the following four constraints are considered:

- (i) *Goal configuration is collision free*—from the probabilistic geometric environment model, (x,y,z) locations with $P_{x,y,z} < P_{\text{empty}} = 0.05 (2\sigma)$ are considered as unoccupied. Such points form candidate configuration space camera pose coordinates.
- (ii) *Goal reached by a collision free path*—this is a function of the camera manipulator kinematics and the known environment model.
- (iii) *Goal configuration should not be far from the current one*—a Euclidean metric in configuration space, with individual weights α_i on each degree of freedom of the camera pose \bar{c} , is used to define the distance moved by the camera ($\alpha_i = 1$ for this thesis):

$$d = \left(\sum_{i=1}^n \alpha_i (c_i - c'_i)^2 \right)^{1/2} \quad (3-11)$$

where \bar{c} and \bar{c}' are vectors of the new and current camera poses respectively.

- (iv) *Measurement at the goal configuration should maximize information throughput*—Specifically, the new information (NI) is equal to the expected information of the unknown/partially known region viewed from the camera pose under consideration. This is based on the known obstacles from the current environment model, the field of view of the camera (see Figure 3-8) and a framework for entropic thresholding of information. Shannon showed that a definition of entropy, similar in form to a corresponding definition in statistical mechanics, can be used to measure the information gained from the selection of a specific event among an ensemble of possible events [Shannon] (see Chapter 2). This entropy function, H , can be represented as:

$$H(q_1, q_2, \dots, q_n) = - \sum_{k=1}^n q_k \log_2 q_k \quad (3-12)$$

where q_k represents the probability of occurrence for the k^{th} event.

Shannon's emphasis was in describing the information content of 1-D signals. In 2-D the gray level histogram of an image can be used to define a probability distribution:

$$q_i = f_i / N \text{ for } i = 1 \dots N_{\text{gray}} \quad (3-13)$$

where f_i is the number of pixels in the image with gray level i , N is the total number of pixels in the image, and N_{gray} is the number of possible gray levels. With this definition, the entropy of an image for which all the q_i are the same—corresponding to a uniform gray level distribution or maximum contrast—is a maximum. The less uniform the histogram, the lower the entropy.

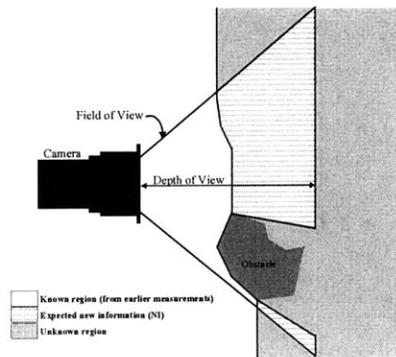


Figure 3-8: Evaluation of expected new information

It is possible to extend this idea of entropy to a 3-D signal—the environment model. In such an instance the scene probability distribution for entropy (information) analysis is still given by Equation 3-13. However, N is the maximum number of voxels visible by the vision system (limited by the depth of field and the field of view), and f_i is the number of voxels in the scene with gray level i . The equation is evaluated separately for mapped versus unmapped regions.

$$H(q) = - \left(\left(\sum_{k=1}^n q_k \log_2 q_k \right)_{\text{known}} + \left(\sum_{k=1}^n q_k \log_2 q_k \right)_{\text{unknown}} \right) \quad (3-12b)$$

The possible gray level values are defined as follows. For all unknown/unsampled voxels, a gray value may be defined as: $p(\bar{x}) = \frac{d_{\text{voxel}}}{d_{\text{max}}}$ where d_{max} is the maximum distance from the camera to any voxel in the camera field of view (equal to the depth of field). This is a subjective choice and other similar choices may be conceived. However, a more complex but better form for $p(\bar{x})_{\text{unknown}}$ would be a Markovian chain i.e. $p(\bar{x})$ of a particular voxel is the average value of $p(\bar{x})$ of the

neighboring voxels. Intuitively this results in unknown regions that are mapped as averages of closest known regions. Thus, for all spatial voxels, a gray (probabilistic) occupancy value between 0 and 1 is found. Next the values for $p(\bar{x})$ are modified as follows:

$$\text{stretching: } p'(\bar{x}) = \begin{cases} \frac{1}{1-p(\bar{x})} \cdot \frac{1}{d_{\text{voxel}}} & \forall p(\bar{x}) < 0.5 \\ \frac{1}{p(\bar{x})} \cdot \frac{1}{d_{\text{voxel}}} & \forall p(\bar{x}) \geq 0.5 \end{cases} \quad + \text{ scaling: } p''(\bar{x}) = \begin{cases} \frac{p'(\bar{x})-1}{2} & \forall p(\bar{x}) < 0.5 \\ 1 - \frac{p'(\bar{x})-1}{2} & \forall p(\bar{x}) \geq 0.5 \end{cases} \quad (3-14)$$

where d_{voxel} is the Euclidean distance of the voxel from the camera coordinate frame. This process causes regions with probability densities closer to 0 or 1 (regions of most certainty) to have a reduced effect on the new information expected. Regions that have a probability density closer to 0.5 (regions of least certainty of occupancy) are “stretched out” in the scene probability distribution, thus increasing the new expected information associated with those regions. A uniform discretization of this range of $p''(\bar{x})$ values may be performed to define f_i , N_{gray} and N (equation 3-13). With these definitions, q_k (Equation 3-13) is evaluated and the results applied to Equation 3-12b resulting in a metric for new information (NI). Alternatively (better) a uniform discretization of $p(\bar{x})$ may be used to define f_i , N_{gray} and N . To increase the contribution of regions with higher occupancy uncertainty to the information metric, the term $q_k \log_2 q_k$ of equation 3-12b is premultiplied by $-p_k \log_2 p_k$.

This definition for NI does behave in an intuitively correct form. For example, for a given camera pose, if the field of view is occluded, then NI decreases. If every point in the field of view is known and is empty then NI=0. NI increases as the number of unknowns in the field of view increases. Further, Equation 3-14a increases the new information expected with regions that are known with median probabilistic values i.e. values that indicate with least amount of certainty whether a voxel is occupied or not. On the other hand, regions with high probabilistic values for

occupancy result in reduced associated information.

Constraints (iii) and (iv) are merged into a rating function (RF):

$$RF = (NI - K \cdot d^n) \cdot (1 - P_{x,y,z}) \quad (3-15)$$

where K and n are scaling constants. Shorter distances exhibit a higher rating. This rating function can be evaluated and optimized to find the next camera configuration from which to make future measurements of the environment. Although this choice of rating function is somewhat arbitrary, good results were obtained. Additional constraints can also be accommodated.

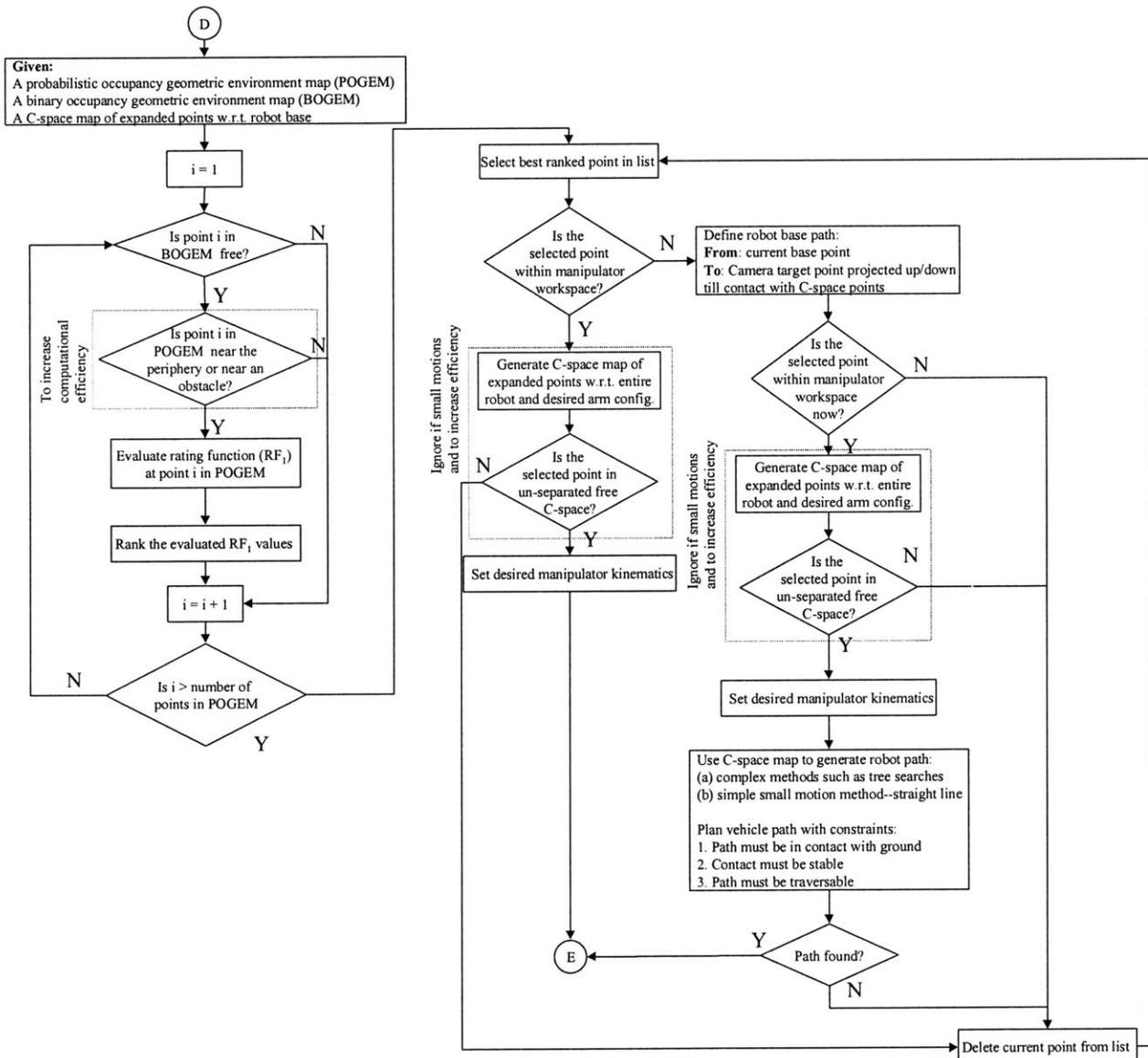


Figure 3-9: Flowchart for vision system pose selection of environment mapping algorithm

The vision system pose selection algorithm is outlined in Figure 3-9. Note that the movement of the vision system may require motions by the mobile robot (in addition to manipulator motions). The flowchart in figure 3-9 includes a simple path planning approach based on the principle of convex hulls (see Chapter 4).

3.2.5. Camera motion correction

A final step in environment map building is to identify the motion of the camera. This process eliminates manipulator positioning errors and vehicle suspension motions, and allows for accurate data fusion. A single spatial point in the base frame, \bar{r}_i , is related to the image point (u_i, v_i) by the 4x4 transformation matrix \mathbf{g}_{01} (see Figure 3-10).

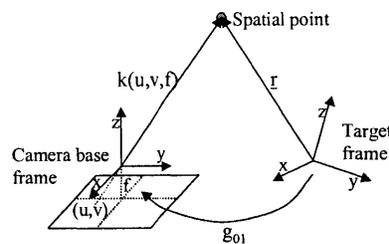


Figure 3-10: Relationship of camera and target frames

For motion calibration we need to identify \mathbf{g}_{01} :

$$\begin{bmatrix} k_i u_i \\ k_i v_i \\ k_i f \\ 1 \end{bmatrix} = \mathbf{g}_{01} \cdot \bar{r}_i = \begin{bmatrix} [\mathbf{R}_{01}]_{3 \times 3} & \bar{\mathbf{X}}_{3 \times 1} \\ \bar{\mathbf{0}} & 1 \end{bmatrix} \cdot \begin{bmatrix} r_i^x \\ r_i^y \\ r_i^z \\ 1 \end{bmatrix} \quad (3-16)$$

where \mathbf{R}_{01} is the rotational matrix, $\bar{\mathbf{X}}$ is the translation vector, f is the camera focal length, and k_i is a scaling constant. For computational reasons it is more convenient to treat the 9 rotational components of \mathbf{R}_{01} as independent, rather than a transcendental relation of 3 independent parameters. Each spatial point gives 3 algebraic equations, but also introduces a new variable, k_i —multiplicative constant to extend the i^{th} image point vector $(u, v, f)_i$ to the i^{th} spatial point in the camera coordinate frame. k_i may be found from the disparity pair of the stereo images. For n

points we have:

$$\mathbf{u} = \mathbf{g}_{01} \mathbf{r} \Rightarrow \begin{bmatrix} k_1 u_1 & k_2 u_2 & & k_n u_n \\ k_1 v_1 & k_2 v_2 & \dots & k_n v_n \\ k_1 f & k_2 f & \dots & k_n f \\ 1 & 1 & & 1 \end{bmatrix} = \mathbf{g}_{01} \begin{bmatrix} r_1^x & r_2^x & & r_n^x \\ r_1^y & r_2^y & \dots & r_n^y \\ r_1^z & r_2^z & \dots & r_n^z \\ 1 & 1 & & 1 \end{bmatrix} \quad (3-17)$$

This set of linear equations can be readily solved using conventional techniques. A least mean square error solution is given by:

$$\mathbf{g}_{01} = \mathbf{u}(\mathbf{r}^T \mathbf{r})^{-1} \mathbf{r}^T \quad (3-18)$$

The rotation matrix, \mathbf{R}_{01} , and the translation vector, \bar{X} , of the camera frame with respect to the base frame are extracted directly from this solution of \mathbf{g}_{01} . However, for real measured data and associated uncertainty, a larger number of spatial points are required to more correctly identify the geometric transformation matrix, \mathbf{g}_{01} . Given the $(i+1)^{\text{st}}$ spatial and image point, from Equation 3-18 \mathbf{R}_{i+1} and \bar{X}_{i+1} can be obtained. A recursive method can be used to determine the mean and covariance of \bar{X} and \mathbf{R}_{01} based on the previous i measurements as follows:

$$\begin{aligned} \hat{\bar{X}}_{i+1} &= \frac{(i\hat{\bar{X}}_i + \bar{X}_{i+1})}{i+1} \\ C_{i+1}^{\bar{X}} &= \frac{iC_i^{\bar{X}} + [\bar{X}_{i+1} - \hat{\bar{X}}_{i+1}] [\bar{X}_{i+1} - \hat{\bar{X}}_{i+1}]^T}{i+1} \\ \hat{\mathbf{R}}_{i+1}^{(l,m)} &= \frac{(i\hat{\mathbf{R}}_i^{(l,m)} + \mathbf{R}_{i+1}^{(l,m)})}{i+1} \\ C_{i+1}^{\mathbf{R}^{(l,m)}} &= \frac{iC_i^{\mathbf{R}^{(l,m)}} + [\mathbf{R}_{i+1}^{(l,m)} - \hat{\mathbf{R}}_{i+1}^{(l,m)}] [\mathbf{R}_{i+1}^{(l,m)} - \hat{\mathbf{R}}_{i+1}^{(l,m)}]^T}{i+1} \end{aligned} \quad (3-19)$$

This essentially maintains a measure on how certain the camera motion is w.r.t. its original configuration (assuming the original configuration is known very precisely w.r.t. the common reference frame). To obtain an estimate on the position uncertainty of a measured point in the environment, this camera pose uncertainty must be accounted for. Let the measurement \bar{z} be related

to the state vector (actual point position) \bar{x} by a non-linear function, $h(\bar{x})$. The measurement vector is corrupted by a random noise vector \bar{v} of known covariance matrix, \mathbf{R} .

$$\bar{z} = h(\bar{x}) + \bar{v} \quad (3-20a)$$

Assume that the measurement of the state vector \bar{x} is done multiple times. In terms of the current measurement, a Jacobian matrix of the measurement relationship evaluated at the current state estimate is defined as:

$$H_k = \left. \frac{\partial h(\bar{x})}{\partial \bar{x}} \right|_{\bar{x}=\bar{x}_k} \quad (3-20b)$$

The state (or position) may then be estimated as follows:

$$\begin{aligned} K_k &= P_k H_k^T [H_k P_k H_k^T + R_k]^{-1} \\ \hat{\bar{x}}_{k+1} &= \hat{\bar{x}}_k + K_k [\bar{z}_k - h(\bar{x}_k)] \\ P_{k+1} &= [1 - K_k H_k] P_k \end{aligned} \quad (3-20c)$$

This estimate is also known as the Extended Kalman Filter which reduces to the standard Kalman filter described in Section 2.3. for linear $h(\bar{x})$ [Gelb, Kelly]. Using this updated value for both the measured point \bar{x} and the absolute uncertainty \mathbf{P} , the measured point may then be merged with the current environment model using equations 3-8 and 3-10. Note that combining noisy measurements leads to a noisier result. For example, the camera pose uncertainty increases as the number of camera steps increase. With every new step, the current uncertainty is merged with the previous uncertainty to get an absolute uncertainty in camera pose. However, by merging redundant measurements (filtering) leads to a less noisier result (e.g. the environment point measurements).

The issue of obtaining appropriate spatial points is now addressed. Spatial points are obtained by maintaining a visible set of fiducials that are tracked during map building. As the camera moves, the fiducials move relative to the camera, eventually moving out of the camera view. This requires methods to identify and track new fiducials. Fiducials are selected from the probabilistic

environment model based on three criteria: the degree of certainty with which a sampled point is known, the visual contrast of the sampled point with its surroundings, and depth contrast of the sampled point with its surroundings. These are combined into a single fiducial evaluation function:

$$F.E.F. = f(P(\mathbf{x})) + g(C(u, v)) + h(H(\mathbf{x})) \quad (3-21)$$

- Fiducial certainty: $f(P(\mathbf{x})) \sim P(\mathbf{x})/r$, where r is the radius of a sphere centered at the potential fiducial within which neighboring voxels have decending certainty levels. Outside this sphere voxel certainty levels increase. Lower values for r suggest that the region surrounding a potential fiducial is well known—a desirable property.
- Fiducial visual contrast: $g(C(u, v)) \sim \text{contrast } (C) * \text{window size } (w)$. Contrast is defined as:

$$C(u, v) = \frac{I(\mathbf{x}) - \bar{I}_w}{\bar{I}_w} \quad (3-22)$$

where $I(\mathbf{x})$ is the 2D image intensity value of the potential fiducial at \mathbf{x} , \bar{I}_w is the average intensity of a window centered at the potential fiducial in the 2D image, and w is the maximum window size after which the contrast starts to decrease.

- Fiducial depth contrast: $h(H(\mathbf{x})) \sim H(\mathbf{x}) * \text{window size } (w)$, where $H(\mathbf{x})$ is the maximum spatial frequency (from a 3D Fourier transform) at the potential fiducial within a window, and w is the maximum window size after which the power spectrum (of the 3D Fourier transform) starts shifting to higher frequencies. To simplify computation, this may be approximated with some heuristics.

Additionally, a penalty is added if a potential fiducial is too close to other identified fiducials. Using the identified fiducials, camera motion can be identified. Fiducials can be tracked with simple methods such as region growing or image disparity correspondence. This algorithm is summarized in Figure 3-11.

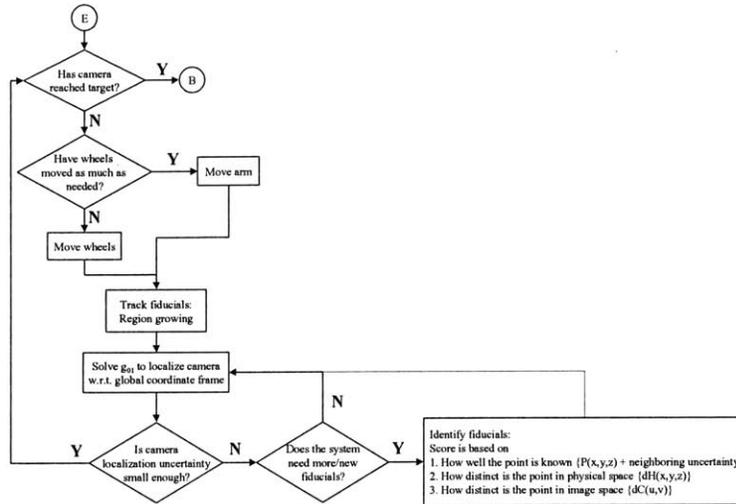


Figure 3-11: Flowchart for vision system motion identification using scene fiducials

3.3. Results

3.3.1. Simulation studies

Results using the rating function for vision system pose selection to develop a probabilistic model of a planar environment, are given here. Three simulation tests have been conducted: single camera/robot modeling of an unstructured environment, two cooperative cameras/robots modeling of an unstructured environment, and single camera/robot modeling of an indoor environment. Five camera pose selection methods are compared:

- (i) *random pose selection*—the next camera pose is selected randomly within the known environment
- (ii) *sequential/raster pose selection*—the next camera pose is selected as the next free location in the known environment from which measurements have not yet been made
- (iii) *pose with maximum expected unmapped (new) region*—the next camera pose is selected as the location with the largest expected new region while accounting for known occlusions
- (iv) *pose with minimum mapped (old) region*—the next camera pose is selected as the location that will yield the smallest known region
- (v) *pose with maximum expected information*

The rating function (RF) cannot be optimized analytically. In practice, finding an optimum value for RF requires exhaustive searching through the entire known configuration space—a process that takes $O(n)$ time, where n is the number of discrete points in the configuration space. One way to reduce the search time is to employ a finite set of goal configurations. This set of goal configurations may be selected in several ways (random, closest to current pose, etc.). For m possible configurations, this process takes $O(m)$ time— m is a constant. Thus, while the best goal configuration would be the one maximizing RF, any configuration with a high value for RF should suffice. Such a configuration can be found with reasonable effort.

Figure 3-12 shows an unstructured environment (100m x 100m) with occlusions (black) to be mapped/modeled. Figure 3-13 shows the fraction of the environment mapped and the net distance moved by the vision system for the five mapping methods, using a single mobile vision system (with 90° field of view, 15m depth of field). The energy consumption by the system is proportional to the net distance moved by the vision system. Hence it is desirable to have large fraction of the environment mapped with small net displacements. These results show the effectiveness of the information theoretic approach to vision system pose selection in environment modeling. Figure 3-14 shows the path of the vision system for mapping/modeling methods 1 and 5.

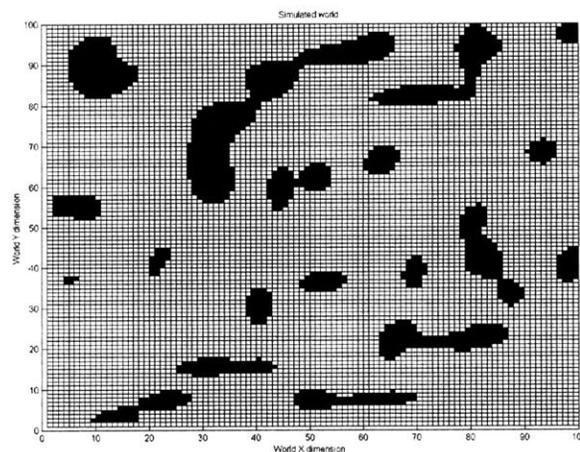
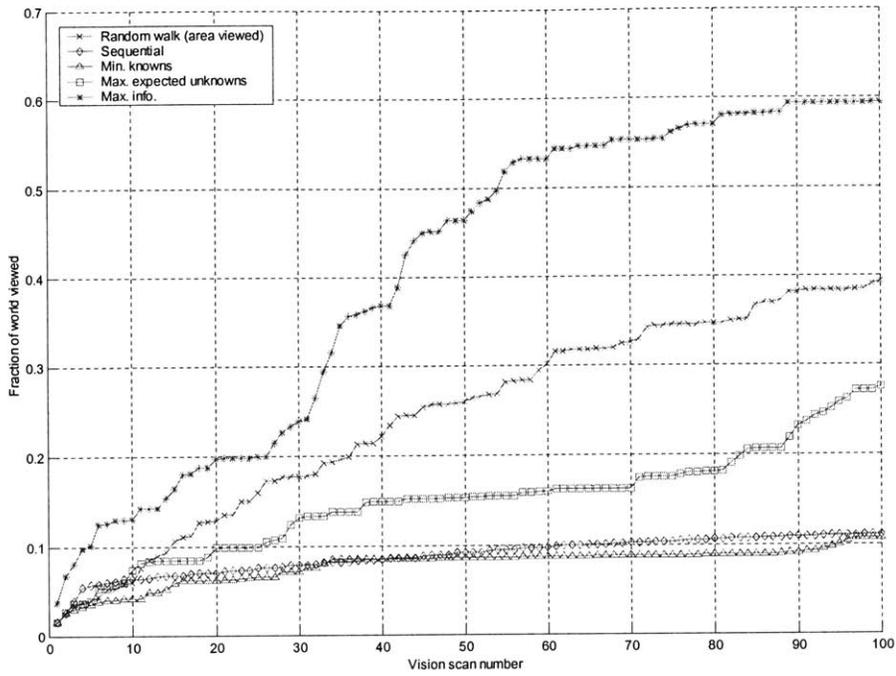
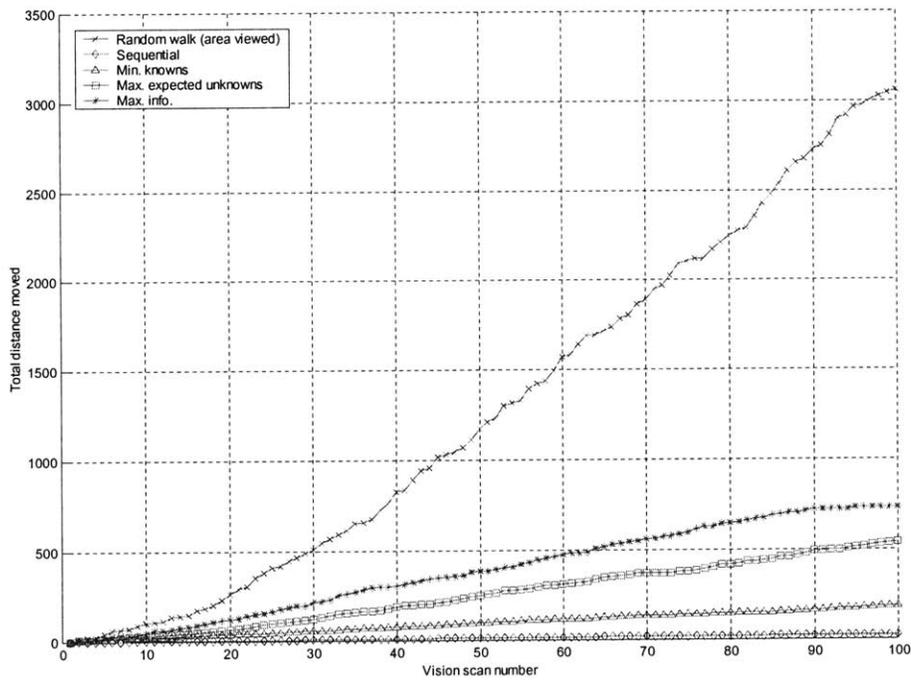


Figure 3-12: Unstructured planar environment

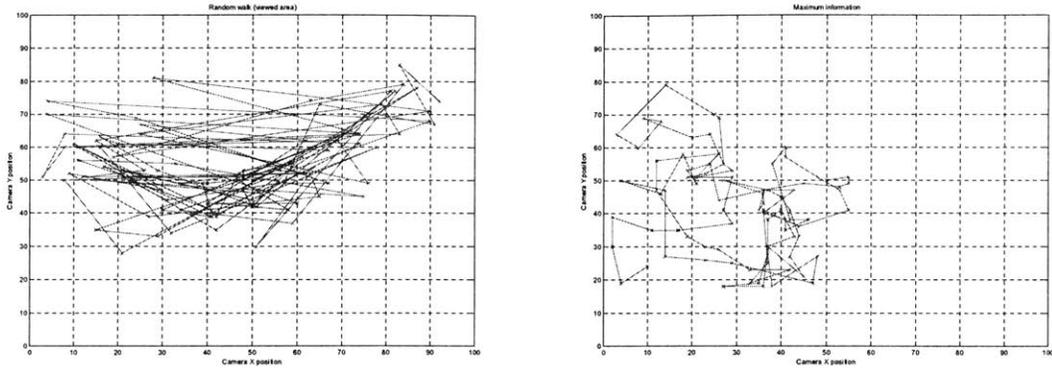


(a) Fraction of environment modeled as a function of scan number



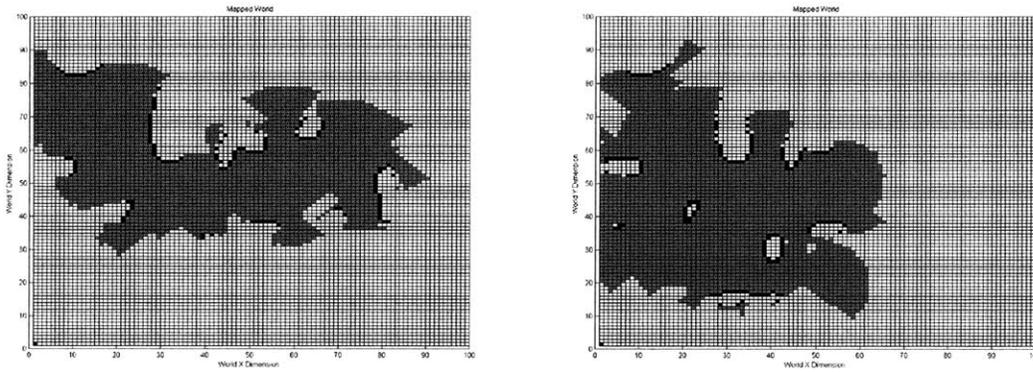
(b) Distance moved by vision system as a function of scan number

Figure 3-13: Results of single vision system modeling an unknown environment



(a) Random walk pose selection (b) Maximum information pose selection

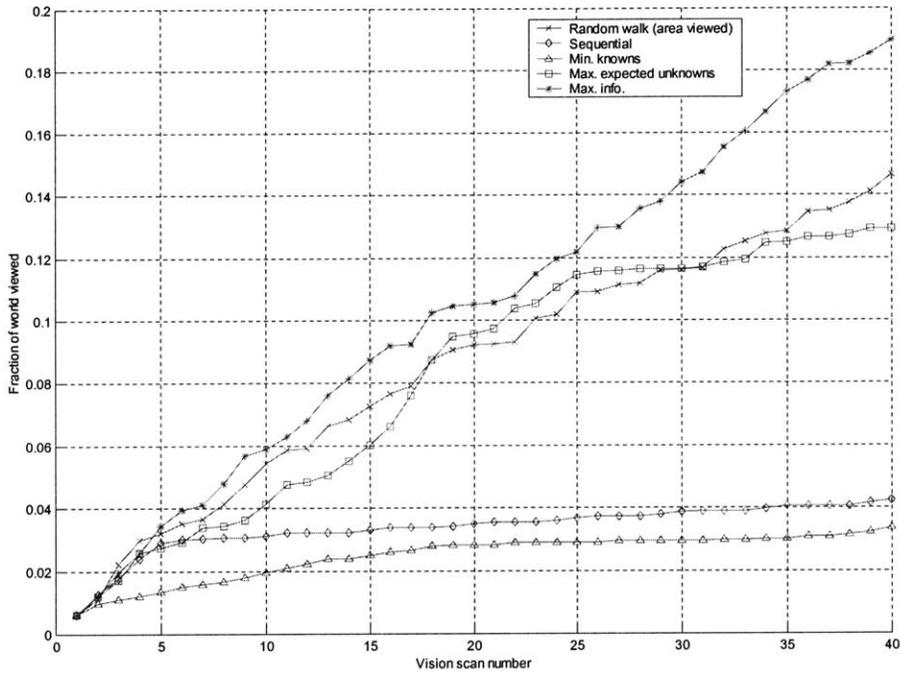
Figure 3-14: Single vision system path



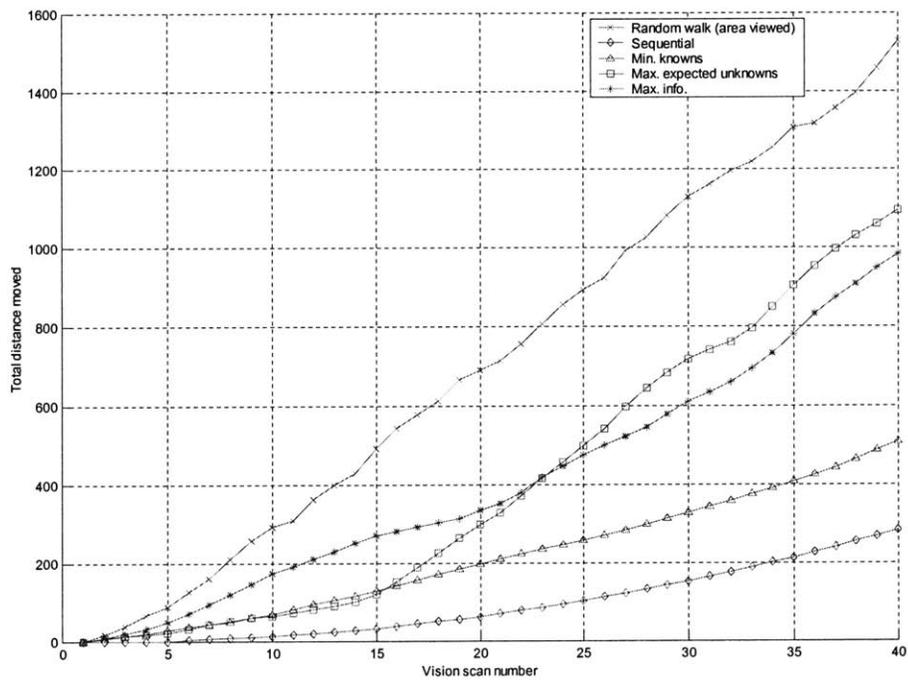
(a) Random walk pose selection (b) Maximum information pose selection

Figure 3-15: Single vision system area mapped (gray=empty space, black=obstacle, white=unknown)

Figure 3-16 shows the fraction of the environment mapped and the net distance moved by the vision system for the five mapping methods, using two cooperating mobile vision systems (with 75° field of view, 10m depth of field). These results again show the effectiveness of the information theoretic approach to vision system pose selection in environment modeling. Figure 3-17 shows the path of the vision system for mapping/modeling methods 1 and 5.

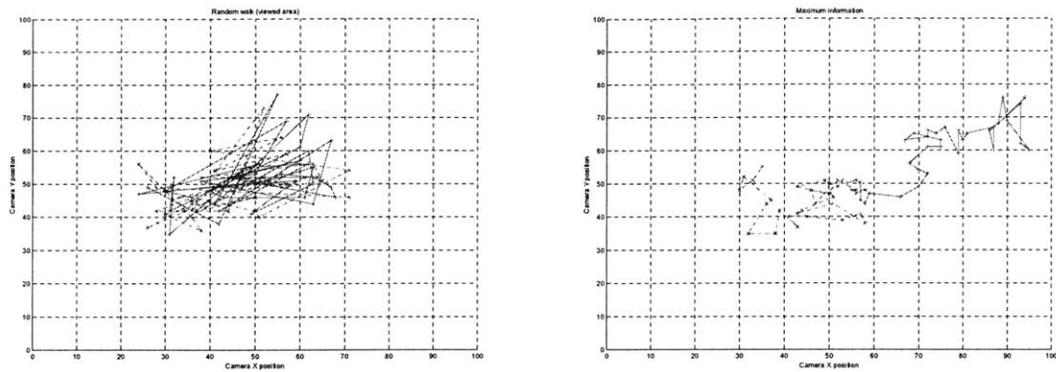


(a) Fraction of environment modeled as a function of scan number

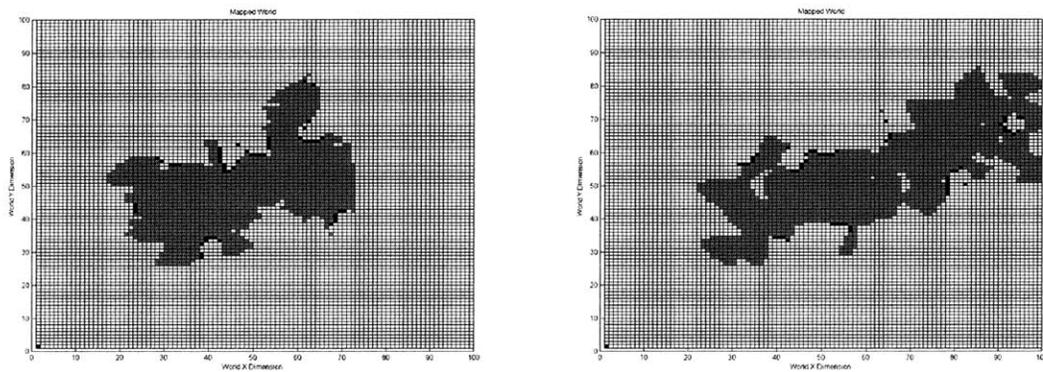


(b) Distance moved by vision system as a function of scan number

Figure 3-16: Results of two vision systems modeling an unknown environment



(a) Random walk pose selection (b) Maximum information pose selection
Figure 3-17: Two cooperating vision systems path



(a) Random walk pose selection (b) Maximum information pose selection
Figure 3-18: Two cooperating vision systems area mapped (gray=empty space, black=obstacle, white=unknown)

Figure 3-19 shows an unstructured indoor-type environment (100m x 100m) with rooms to be mapped/modeled. Figure 3-20 shows the fraction of the environment mapped and the net distance moved by the vision system for the five mapping methods, using a single mobile vision system (with 75° field of view, 10m depth of field). These results again show the effectiveness of the information theoretic approach to vision system pose selection in environment modeling. It is important to note that the information theoretic vision algorithm is the first to map the room and find its way out into the corridor. Figure 3-21 shows the path of the vision system for mapping/modeling methods 1 and 5.

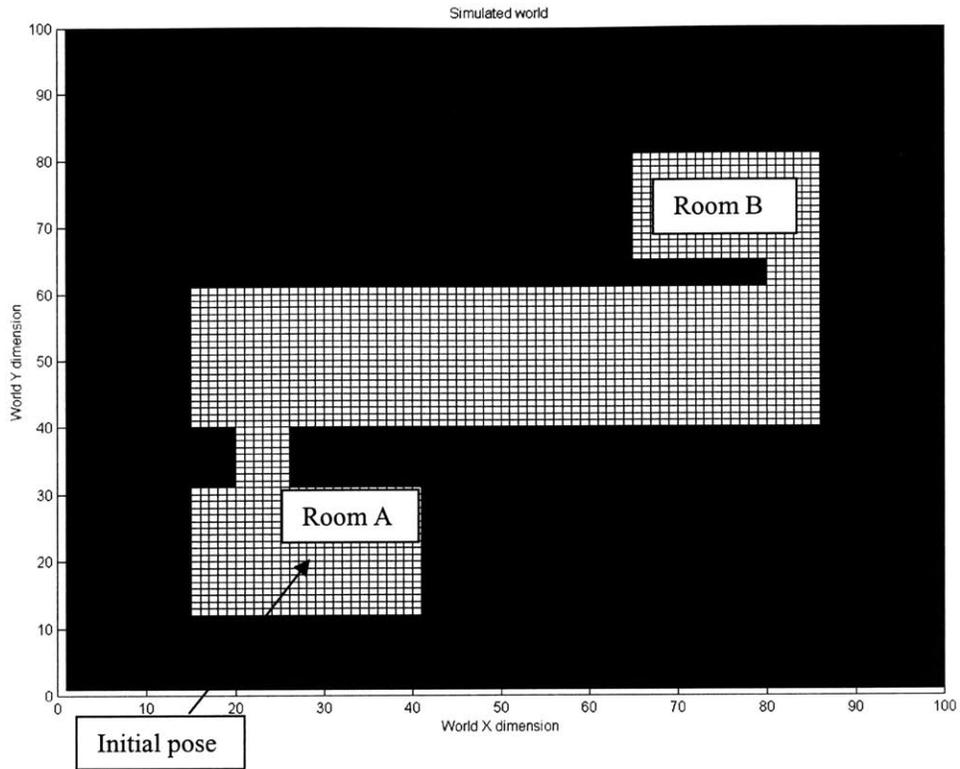
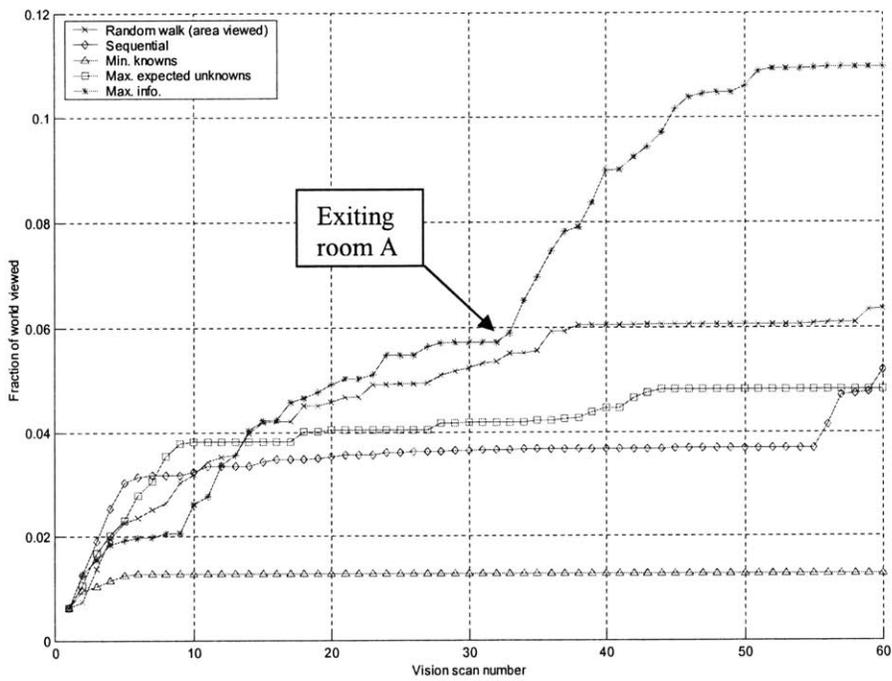
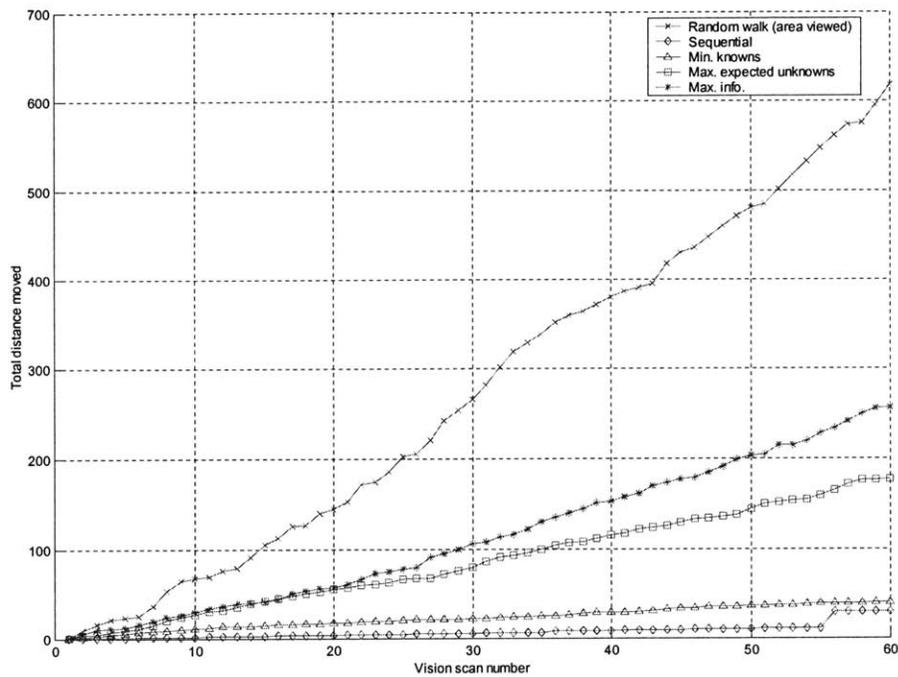


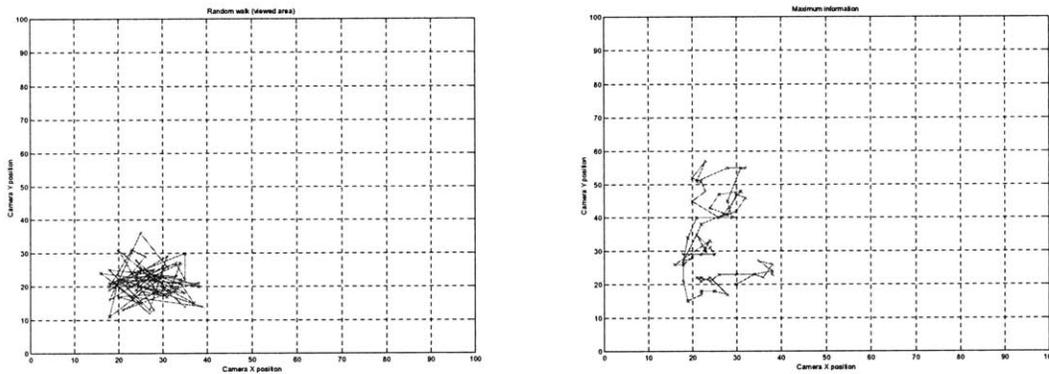
Figure 3-19: Unstructured indoor-type planar environment



(a) Fraction of environment modeled as a function of scan number



(b) Distance moved by vision system as a function of scan number
Figure 3-20: Results of single vision system modeling an unknown environment



(a) Random walk pose selection (b) Maximum information pose selection
Figure 3-21: Single vision system path

3.3.2. Experimental studies

The experimental platform has been briefly described in Chapter 2 and in more detail in Appendix D. Figure 3-22 shows a single vision system (stereo pair) mounted on a mobile manipulator. Mapping is done by breaking up the world into a grid of voxels of specified resolution. All measured points falling in a particular voxel, contribute to the error analysis associated with that

voxel. Voxels corresponding to empty space falling in the field of view of the vision system are tabulated as known, but with no contribution to the uncertainty model. The desired voxel resolution is a function of the task. For this test the resolution is set at 1mm. Figure 3-23 shows the identification and tracking of scene fiducials using the method described in section 3.2.5. Figure 3-24 shows the accumulated r.m.s. translation error as a function of scan step while tracking fiducials at an average distance from the cameras of 1100mm and 350mm. Note that redundancy in scene fiducials help reduce this error dramatically. This error is directly mapped into the uncertainty of measurements of the environment. Figure 3-25 shows the number of points mapped in the environment for two pose selection methods: sequential/raster scan and maximum information based pose selection. Once again, the effectiveness of the information based pose selection process is seen. Figures 3-26 and 3-27 show the regions mapped using sequential camera pose selection and maximum information based camera pose selection.

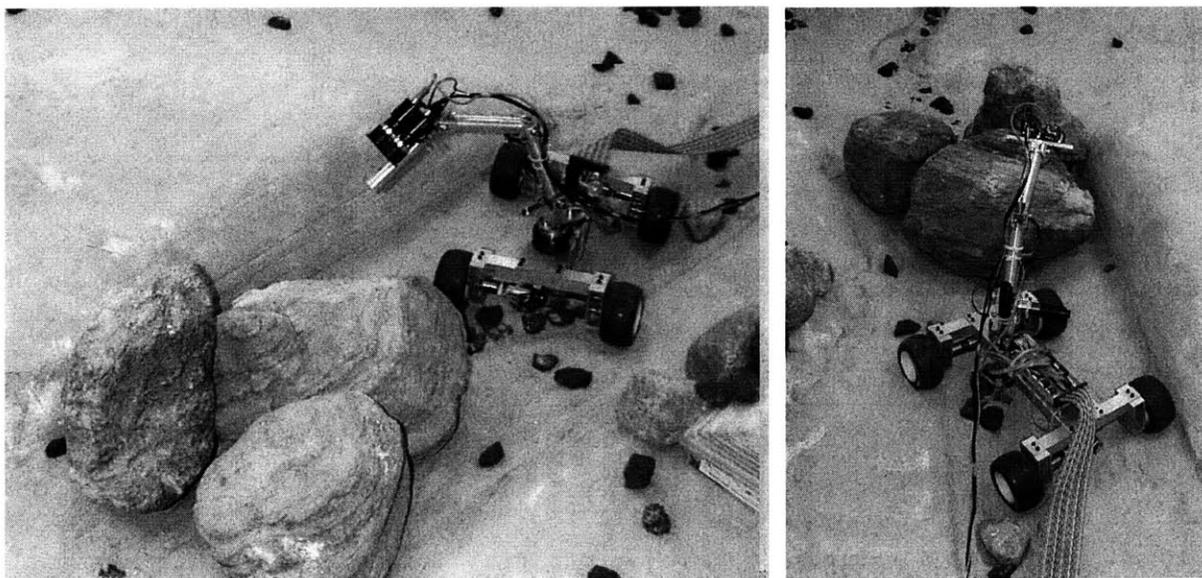


Figure 3-22: Experimental mobile vision system modeling an unstructured environment

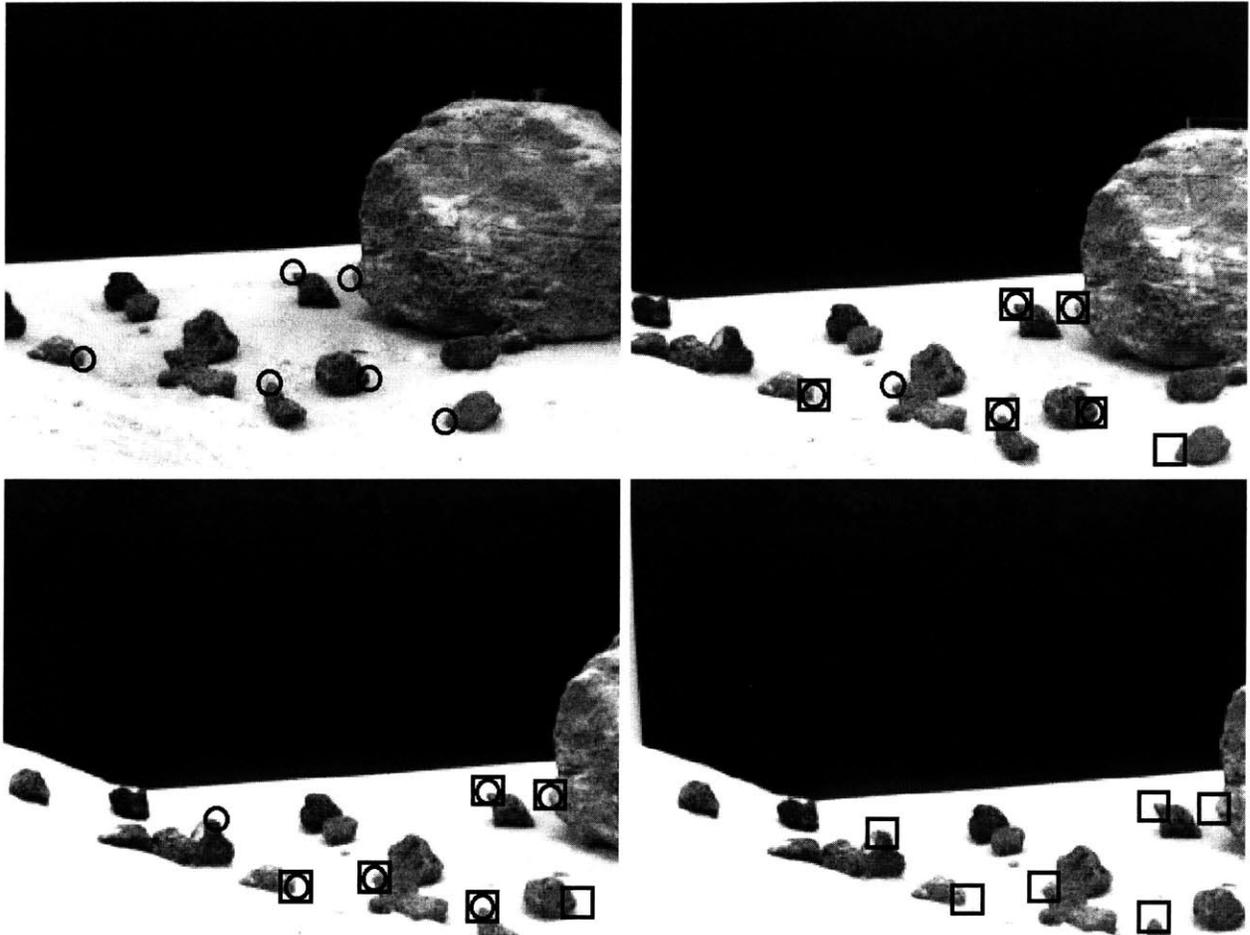
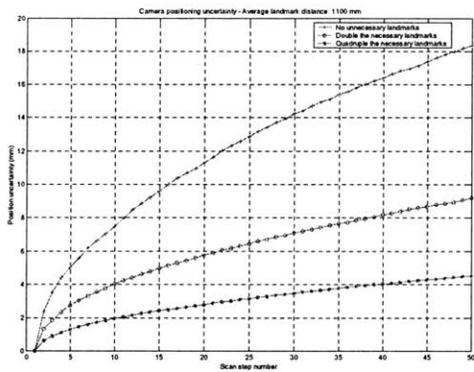
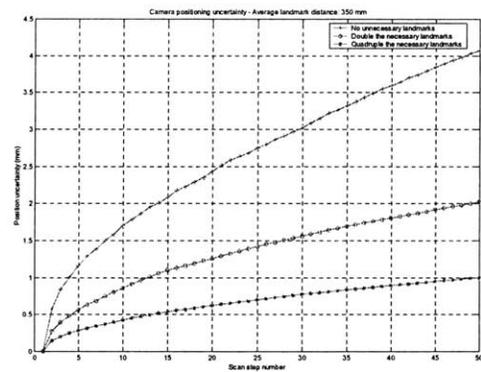


Figure 3-23: Identification and tracking of 6 fiducials (□--tracked with previous image, ○--tracked with next image)



(a) Avg. fiducial distance 1100mm



(b) Avg. fiducial distance 350mm

Figure 3-24: Accumulated r.m.s. translation error of vision system

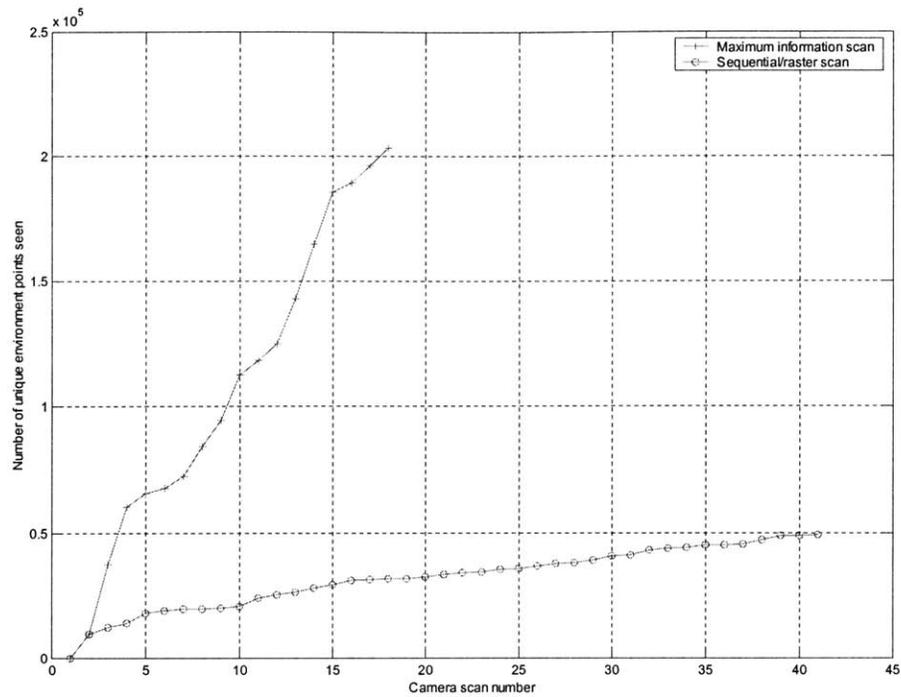
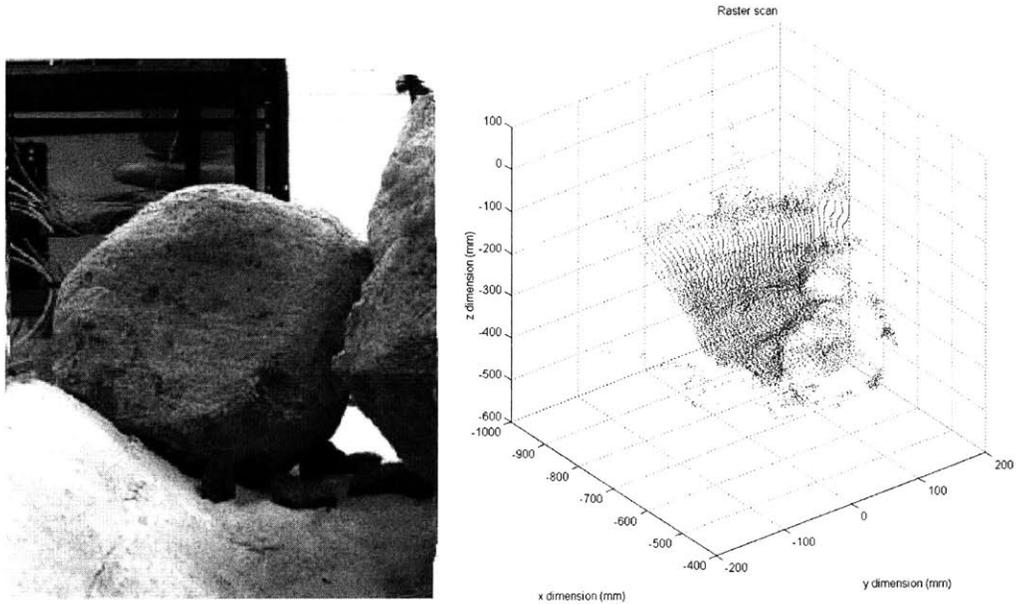


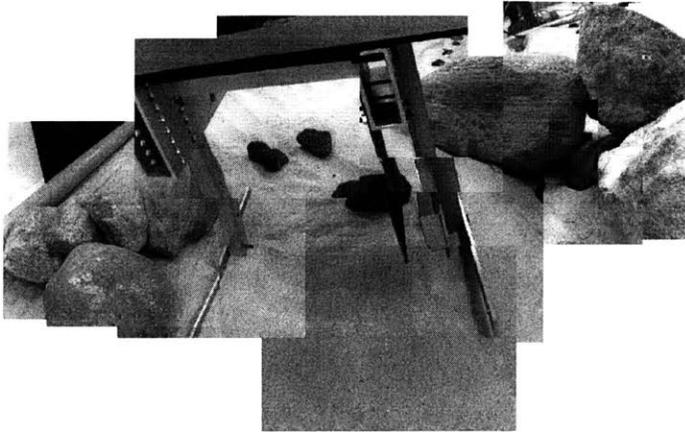
Figure 3-25: Number of mapped environment points as a function of scan number



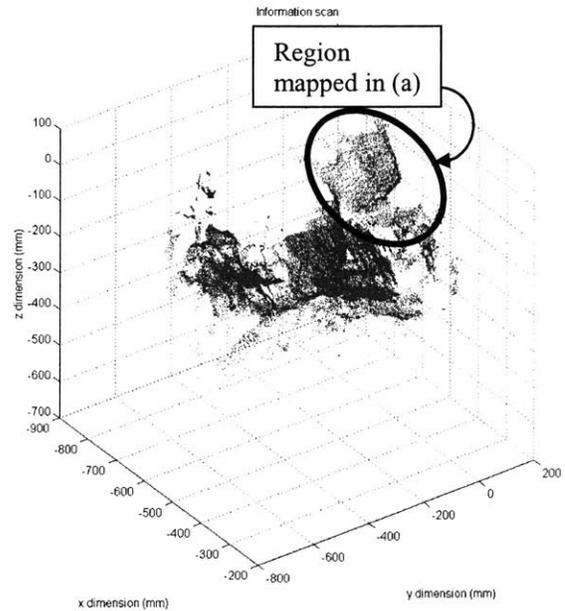
(a) patched left-eye images

(b) environment point cloud

Figure 3-26: Environment mapped/modeled-Sequential camera pose selection



(a) patched left-eye images



(b) environment point cloud

Figure 3-27: Environment mapped/ modeled-Maximum information based camera pose selection

3.4. Conclusions

In field environments it is often not possible to provide robotic teams with detailed a priori environment and task models. In such unstructured environments, cooperating robots will need to create an accurate 3-D geometric model. However, uncertainties in robot locations and sensing limitations/occlusions make this difficult. A new algorithm based on iterative sensor planning and sensor redundancy is proposed to build a map of the environment for mobile robots that have articulated sensors. This algorithm is unique in that it uses a metric of the quality of information previously obtained by the sensors to find new viewing positions for the cameras. Simulations and experiments show promising results.

Task Modeling

4.1. Introduction

Once the environment model is created, the robots need to position their sensors in a task directed optimal way. That is, for a given task requiring visual servo control, there is an associated target to observe. For example, in assembly tasks, the target may be a single point/region in the environment, a distance between two objects, etc. As before, a number of problems can make this non-trivial. These include the uncertainty of the task in the environment, location and orientation uncertainty in the individual robots, and occlusions (due to obstacles, work piece, other robots). If the systems are equipped with cameras mounted on articulated mounts, intelligent planning of the camera motion can alleviate some of these problems.

This chapter describes an algorithm for task directed optimal camera placement for cooperative robots in field environments. Using the environment model created by the algorithm in Chapter 3, the individual robots are positioned “optimally” with respect to the target. This process is described in this chapter. It is assumed that the system consists of two (or more) mobile robots working in an unknown environment (such as constructing a planetary structure—see Figure 4-1). Each has a 3D vision system mounted on an articulated arm. Sensing and sensor placement is limited, resulting in occlusions and uncertainties. The objective is to efficiently find poses for each camera system to optimally view the target. The algorithm iterates on the known environment model, accounting for object motions, occlusions, and camera characteristics.

Fundamentally, this is analogous to the problem of a human hanging a clock in a room. First, a

map of the room (including the target—the clock hook) must be generated. Next, appropriate eye/head/body motions must be generated to visually guide the clock into place. The first part of this problem—environment modeling—is addressed in Chapter 3. This chapter addresses the second part of the problem—task directed appropriate eye positioning.

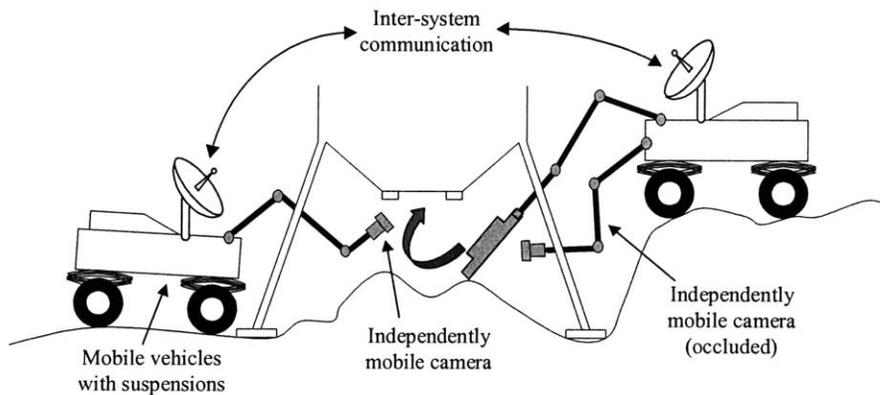


Figure 4-1: Cooperative assembly by robots

4.2. Algorithm Description

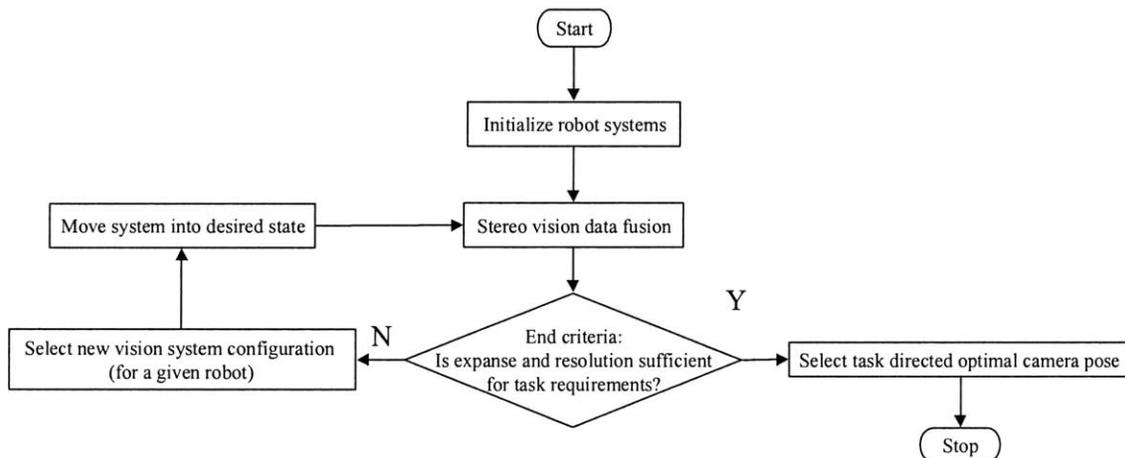
4.2.1. Overview

Figure 4-2 outlines the map building and camera placement algorithm. The algorithm consists of two parts. In the first part, the articulated cameras cooperatively scan the region around a target generating a 3D geometric model. This allows the robots to locate themselves and the obstacles in the target reference frame (see Chapter 3). The second part uses this model to find an optimum pose for the multiple camera systems to view the target(s). As described in Chapter 3, the 3D environment map is modeled as a discretized probabilistic occupancy grid. Every voxel in the map has a value for probability-of-occupancy that ranges from 0 (empty) to 1 (occupied). A value of 0.5 indicates maximum uncertainty in occupancy of the voxel. Using this environment map, and a task description, the algorithm finds an optimum vision system pose with which to view the targets. This is repeated for every vision system available for the task.

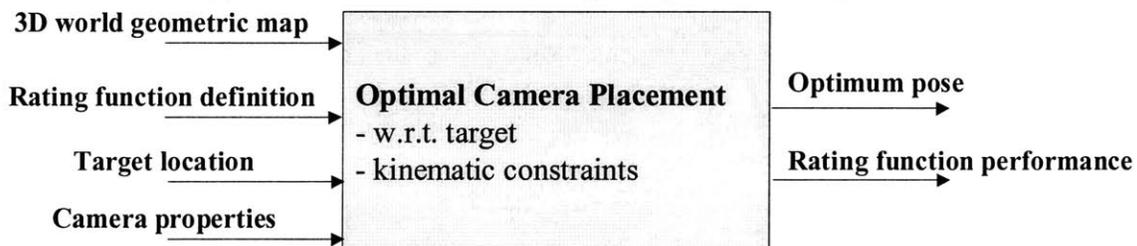
For task directed optimal camera placement, there are four primary constraints. These are depth

of field (DOF), target resolution, target field visibility (TFV) and target angular visibility (TAV). Also physical mobility constraints of the sensor placement mechanism must not be violated. These four constraints will be discussed in detail in the following sections. A rating function weighs the relative importance of these constraints. Optimum sensor placement is accomplished by optimizing this rating function over the environment model.

Examples of tasks include: monitoring a specific target (e.g. an insertion site), guiding the insertion or movement of an object, and other visual servo control operations. For such tasks the visual targets may include: a point/area in the environment (e.g. a hole in a wall), a distance between two points (e.g. an object corner and the insertion site corner), etc. In this chapter, the task is to monitor a specific target, and is demonstrated using simulations and experiments. In the following chapter, a cooperative insertion task is demonstrated experimentally, where the target is the distance between the object and the insertion site.



(a) Outline of model building and placement algorithm



(b) Block diagram for optimal camera placement
Figure 4-2: Task directed optimal camera placement

4.2.2. Algorithm initialization and target identification

As in the environment modeling algorithm, the first step of the task modeling and camera placement algorithm is to identify the targets. This establishes a common reference frame for all further mapping and camera placement operations. For the purposes of this chapter, simple circular holes in a work-piece were used as targets. Although the choice of targets is arbitrary, it is important to distinguish them clearly and consistently. A modified Hough transform is used to identify the circular targets. This process is described in detail in Chapter 5.

4.2.3. Optimum camera pose identification

The approach to finding the optimal camera position is described here. Given the geometric environment model with its uncertainties, an optimum pose for a camera to view a given target can be developed. Based on the probabilistic geometric world map, (x,y,z) locations with $P_{x,y,z} < 0.05$ are considered as unoccupied. Such points form candidate configuration space camera pose coordinates. A rating function (RF) is defined, and optimized over the known configuration space for the new camera pose. This rating function is defined as:

$$RF(x, y, z) = \text{DOF}_{\text{RF}}^{\alpha} \cdot \text{Res}_{\text{RF}}^{\beta} \cdot \text{TFV}_{\text{RF}}^{\gamma} \cdot \text{TAV}_{\text{RF}}^{\delta} \cdot (1 - P_{x,y,z}) \quad (4-1)$$

where DOF_{RF} , Res_{RF} , TFV_{RF} and TAV_{RF} are the contributions to the rating function due to the depth of field, resolution of the target, target field visibility and target angular visibility respectively, from the camera position (x,y,z) . These contributions are defined in the following sections. α , β , γ , δ are constants (set to unity for the experiments conducted in this chapter). It is assumed that the camera normal vector (i.e. principal axis of the lens system) points directly at the target. Camera position improves as the rating function value increases.

4.2.3.1. Depth of field (DOF)

The DOF constraint of a camera system is defined as the maximum and minimum distance from the

camera-lens system between which all feature points will be sufficiently in focus. This tolerance is based on the lens aperture effects as well as the flexibility allowed by the image processing algorithms (such as range finding, feature identification, etc.). The value of $DOF_{RF}=0$ if the feature point is outside the depth of field, and $DOF_{RF}=1$ if the feature point is inside the depth of field. More complex non-linear functions may be used here that quantify how much a given feature point is in focus. However, this simple binary function gives good solutions.

4.2.3.2. Resolution of target

The resolution of the target along an axis from the given position of the camera is simply:

$$R = \frac{2d \tan(\alpha/2)}{n} \quad \text{and} \quad Res_{RF} = \frac{1}{R} \quad (4-2)$$

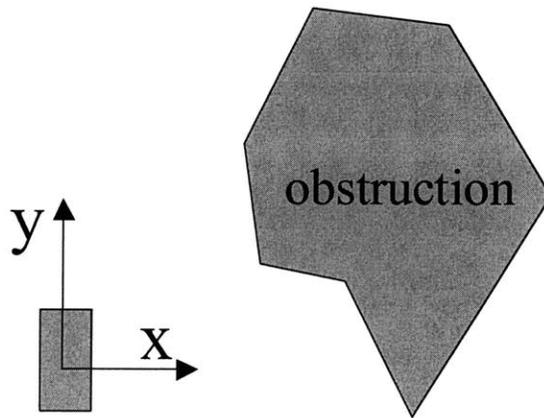
where d is the distance of the camera from the target, α is the camera field of view and n is the number of pixels along the detector axis. The contribution to the rating function from the resolution is given by Res_{RF} .

4.2.3.3. Target field visibility

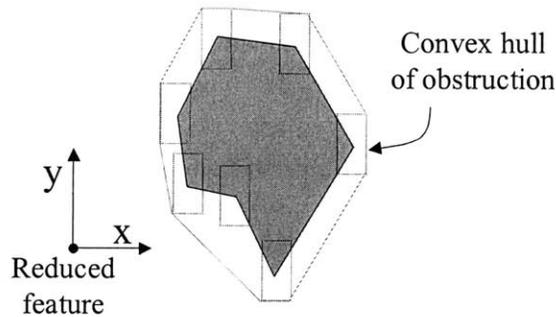
The target field visibility (TFV_{RF}) at any point in space, is defined as the largest angle of excursion the camera can traverse around a circle centered about the target, before the target is occluded. The target field visibility must account for occlusions in the workspace and the finite size of the target. Finding the target field visibility (for finite sized targets) in a 3-D space while accounting for occlusions can be very difficult and time consuming, growing exponentially with the number of occlusions. To reduce this difficulty, a method of occlusion expansion using convex hulls is proposed. In this process the target is reduced to a single mathematical point, while all potential occlusions are appropriately expanded in size. As such, if the target point is now visible from a given location, then it is guaranteed that the entire physical target will also be visible from that location. This process consists of three steps:

(i) Occluding object expansion and target reduction

The finite sized target is reduced to a single mathematical point while the occlusions are expanded. Determining the field visibility of the target from any given point is thus simplified. Figure 4-3 outlines the idea of object expansion in 2-D. A coordinate frame is attached to the target at a reference point. This point is now placed at every vertex of the occluding object and the target projected accordingly. The new vertices of the target/feature are computed. This new set of vertices forms the expanded object. This is known as the Minkowski sum [O'Rourke].



(a) original target (feature) and obstruction



(b) reduced target (feature) and expanded obstruction

Figure 4-3: Target reduction and occlusion expansion

(ii) Convex hulls of the expanded objects

The expanded object is simplified to occlusion region defined by the convex hull of the expanded

object. The convex hull is the minimum set of points from the Minkowski sum of the expanded object, such that all other points of the expanded object fall “inside” this set. See Figure 4-3(b).

(iii) Projection of expanded object

The convex hull of the expanded occlusion is now projected to a sphere centered on the reduced target. The radius of the sphere is defined as the distance from the camera to the target. This projection is seen in Figure 4-4(a) for a 2-D case and Figure 4-4(b) for a 3-D case. The TFV_{RF} can be directly computed from these projections, as demonstrated for a 2-D case in Figure 4-5.

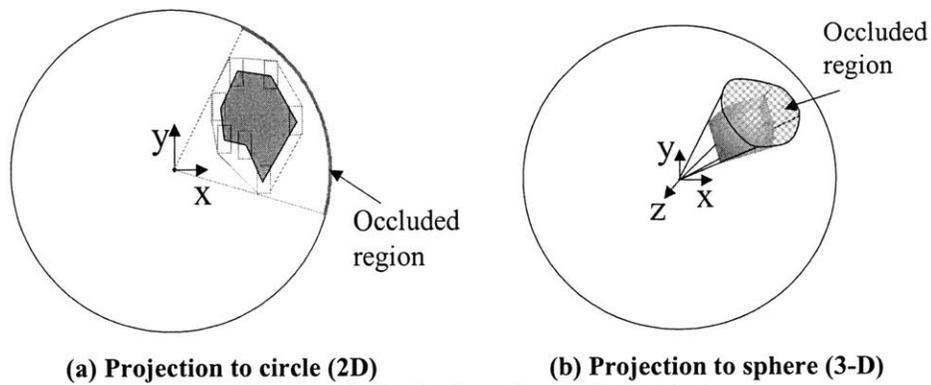


Figure 4-4: Projection of expanded object

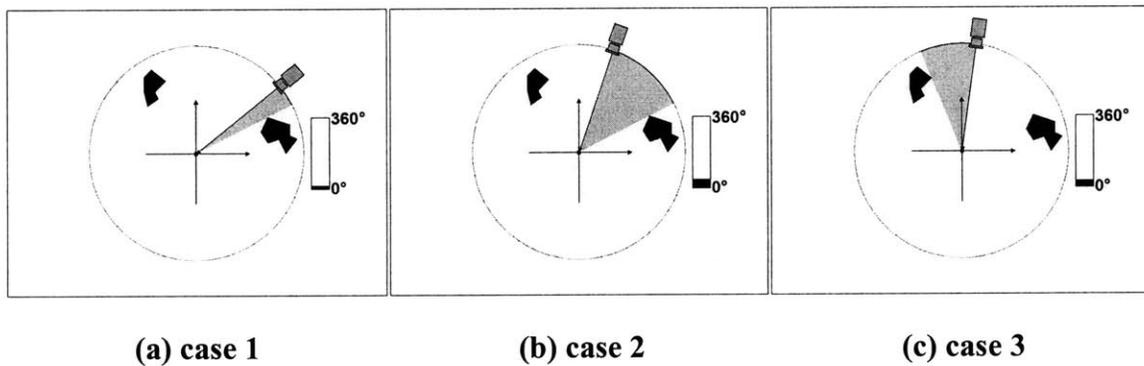


Figure 4-5: Computing the TFV_{RF} (shaded regions)

A difference exists between the actual occluded region and the one found using the expanded object. The latter is larger than or equal to the former. This difference is seen more clearly in Figure 4-6. The occluded region in Figure 4-6(a) is given by $\alpha_2 - \alpha_1$ and in Figure 4-6(b) is given by $\beta_2 - \beta_1$. As the distance between the occlusion and the target increase then $\alpha_2 \rightarrow \beta_2$ and $\alpha_1 \rightarrow \beta_1$.

However, visibility of the target is still guaranteed using the method of convex hulls.

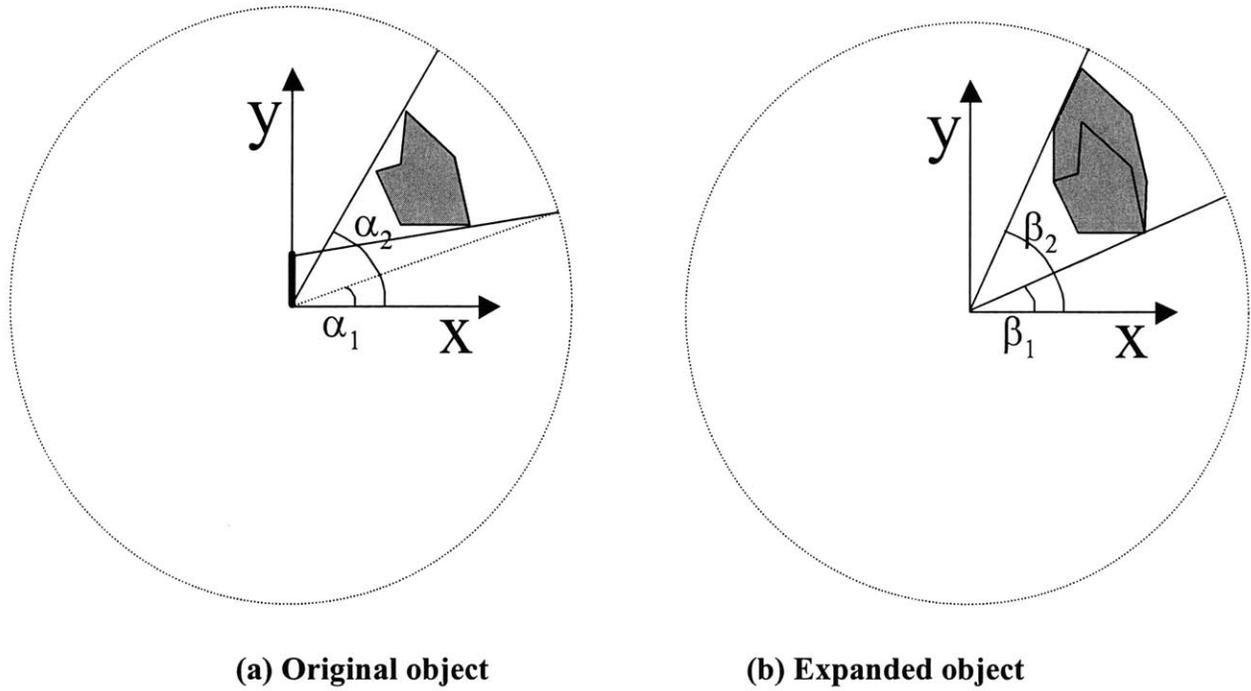


Figure 4-6: Difference between true and expanded object occluded regions

4.2.3.4. Target angular visibility

Target angular visibility is defined as the dot product of the camera image normal vector and the target normal vector, or the cosine of the inter-normal angle, β . Essentially, this is the angle at which the target is viewed. Additionally, for most practical cases, the target may only be visible from one side. In such a situation, the contribution to the rating function due to target angular visibility is given by:

$$TAV_{RF} = \begin{cases} \cos \beta & \text{for } \frac{\pi}{2} \geq \beta \geq \frac{-\pi}{2} \\ 0 & \text{for } \beta \geq \frac{\pi}{2} \text{ or } \beta \leq \frac{-\pi}{2} \end{cases} \quad (4-3)$$

Figure 4-7 shows the TAV_{RF} for a hole viewed at three different angles. It is clear from this that viewing the target at certain angles may be preferred over others.

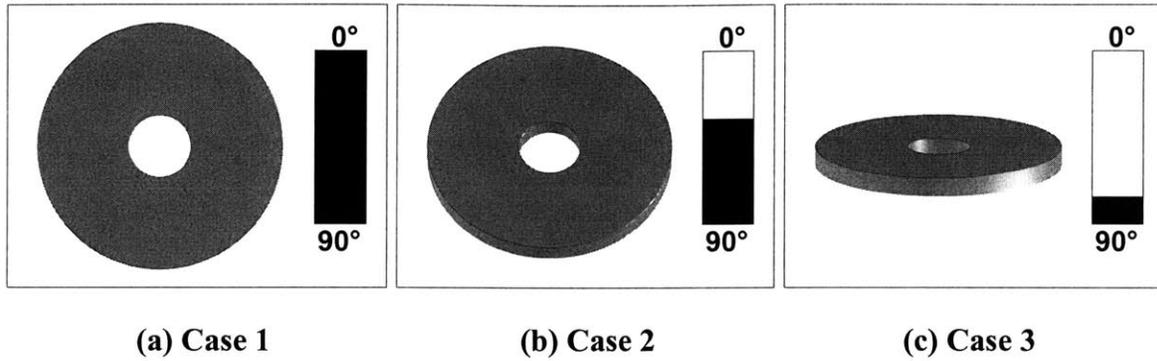


Figure 4-7: Target angular visibility

4.2.3.5. Alternate/secondary targets

In the representative scenarios presented in section 4.2.1, where the target may be a point/area in the environment or a distance between two points, motion of the cooperating robot systems could lead to occlusions of the target. Even with placement optimization, it is possible that the target would not be sufficiently visible to perform the task (e.g. the target angular visibility may be too low). One way to resolve this problem is to identify additional targets, and evaluate the rating function for these *secondary* targets. A secondary target is a feature whose geometric relationship to the original or primary target is known (within defined tolerances).

The above rating function is readily modified to evaluate all known secondary targets. This rating function reflects uncertainty in the geometric relationship between the secondary and primary targets (ST_{error}) with modification to RES_{RF} :

$$R = \frac{2d \tan(\alpha/2)}{n} + ST_{error} \quad \text{and} \quad Res_{RF} = \frac{1}{R} \quad (4-4)$$

In such a situation, a secondary target may be selected even when the primary target is not completely occluded. For example, a hole viewed at an oblique angle (primary target) may be less desirable to view than a hole viewed normally (secondary target).

4.2.3.6 Camera motion identification

Once an optimum pose for the vision system is obtained the physical motion of the cameras to the

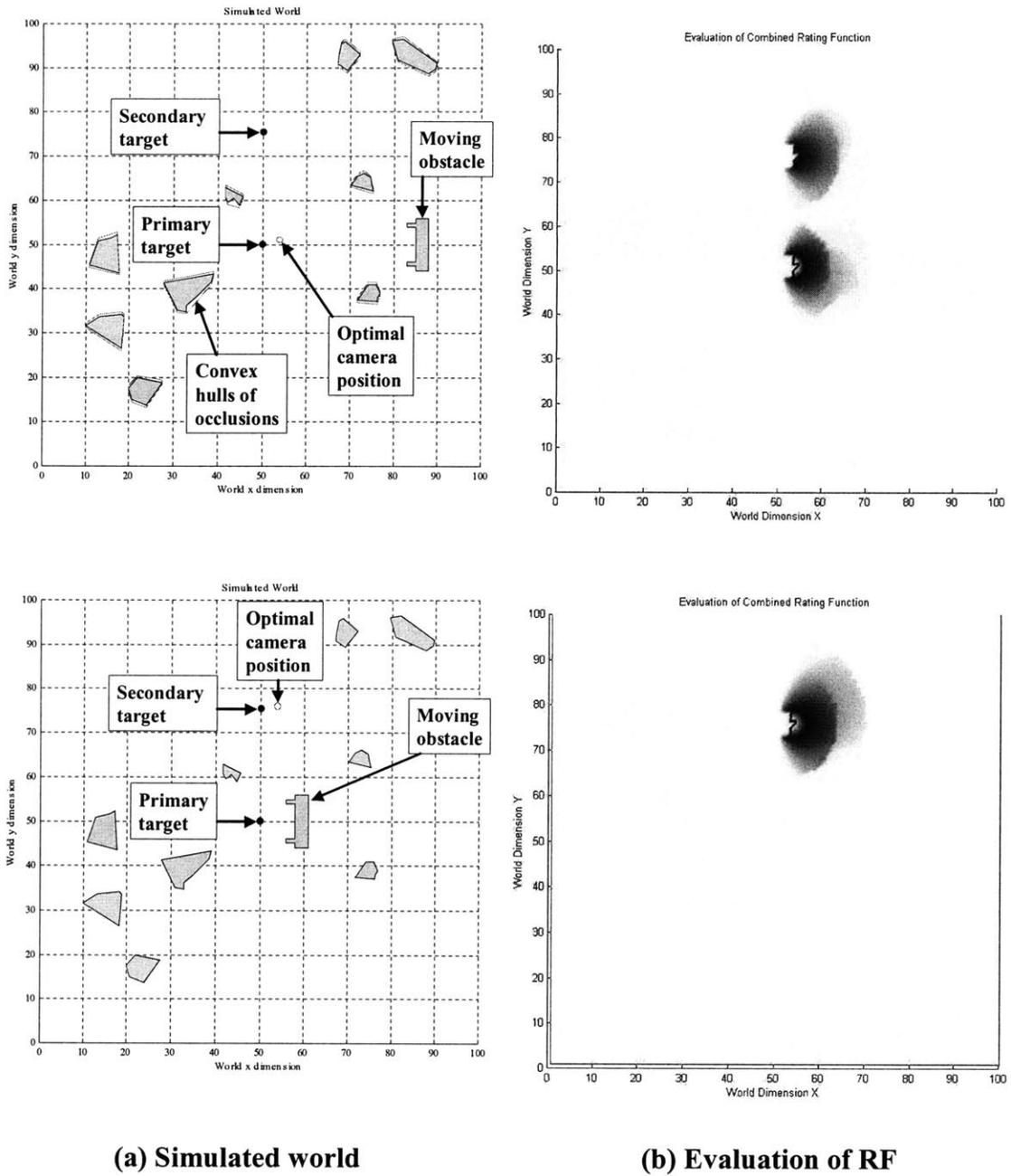
desired pose is achieved using the method described in Section 3.2.5.

4.3. Results

4.3.1. Simulation

In this chapter the task is to monitor a specific target in the environment. The algorithm is first tested on a 2-D simulated environment. Results using the RF to define an optimal camera pose given the probabilistic geometric world map are shown here. Regions where the probability of occlusion < 0.05 are considered empty, and form candidates for optimal camera placement locations. The rating function (RF) cannot be optimized analytically. In practice, finding an optimum value for RF requires exhaustive searching through the known configuration space—a process that takes $O(n)$ time, where n is the number of discrete points in the configuration space. Methods to reduce the search time include: (i) increasing the environment grid “coarseness”, (ii) bounding the evaluation of RF by distance to the target, (iii) employ a finite random selection of goal configurations to evaluate. Thus, while the best goal configuration would be the one maximizing RF, any configuration with a high value for RF will suffice. Such a configuration can be found with reasonable effort.

In the first simulation study, a planar environment is set up (see Figure 4-8a). The primary target center is located at world coordinate (50,50). A secondary target is located at world coordinate (50,75). Figure 4-8b shows the evaluation of the RF (Equation 4-1) over the entire environment, for two positions of a potential occlusion. It is assumed that the environment is known in both instances. Accounting for actual motions of objects will be demonstrated later. Note that the RF value increases as the pixel intensity increases i.e. the darker the intensity the better the camera location. In the first instance, the optimum location is found by viewing the primary target. However, in the second instance, the optimum location is found by viewing the secondary target.



(a) Simulated world

(b) Evaluation of RF

Figure 4-8: Optimal camera placement

For most practical situations, it is expected that there would be movement of some objects in the environment, since often the task would involve motions by one or more of the cooperating robots. For example, Figure 4-9 shows the cooperative assembly of a panel into a mating slot. Here, there is movement by the panel and the robot(s) carrying the panel. Clearly, the optimum camera location

would change as a function of the panel position. This is analogous to a human mounting a clock or picture frame on a wall. As the object is brought toward the target (e.g. a hook), the human repositions his or her head to continue monitoring the approach of the object to the target.

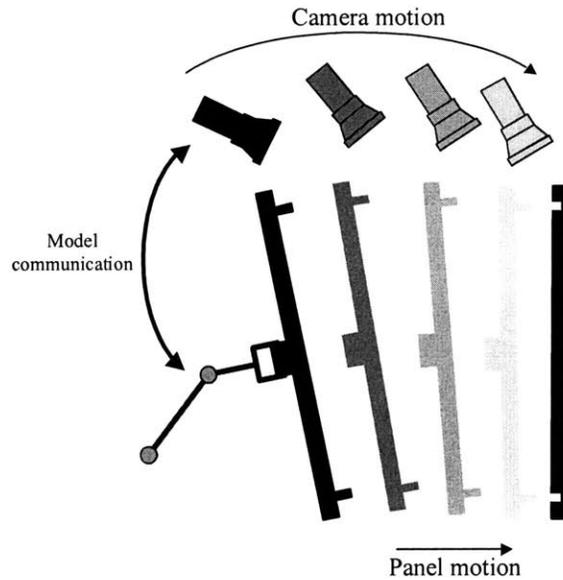


Figure 4-9: Cooperative assembly concept

For cooperative robots, it is assumed that moving objects are well known (e.g. CAD models are given). However, the measured uncertainty associated with their position must be accounted for. As described in Chapter 3, for every mapped point in the environment, there is an associated uncertainty. Rather than remap the environment every time an object (with a known model) moves, the algorithm simply updates the environment model using the object CAD model. This is achieved as follows.

1. Grid points/voxels in the environment model belonging to the moving object are identified. This is achieved using conventional image processing approaches such as template matching, Hough transforms, etc [Lara]. The CAD model of the object is fit to the mapped points. This is similar (in principle) to the method described in section 3.2.5 using Equations 3-16, 3-17 and 3-18. These grid points are removed from the environment model and assumed to be empty measured

space (see Section 3.2.3).

2. Points in the current field of view of the vision system corresponding to the moving object are identified. The identified points are fit to the object CAD model (as in step 1).
3. The current position of the moving object is identified. This is achieved using the known vision system pose and the identified object pose relative to the vision system.
4. Measurements from step 2 and the object CAD model are used to update the environment model (see section 3.2.3.)

The optimal positioning of a vision system in the presence of moving objects is addressed in the second simulation study. Figure 4-10(a) shows a planar simulated environment with obstacles. Primary and secondary targets are set up. Figure 4-10(b) shows the probabilistic occupancy environment model for this scenario. Higher intensity values correspond to greater certainty in occupancy. Note that if the number of scan steps done in modeling the environment are increased, there would be less uncertainty in occupancy values.

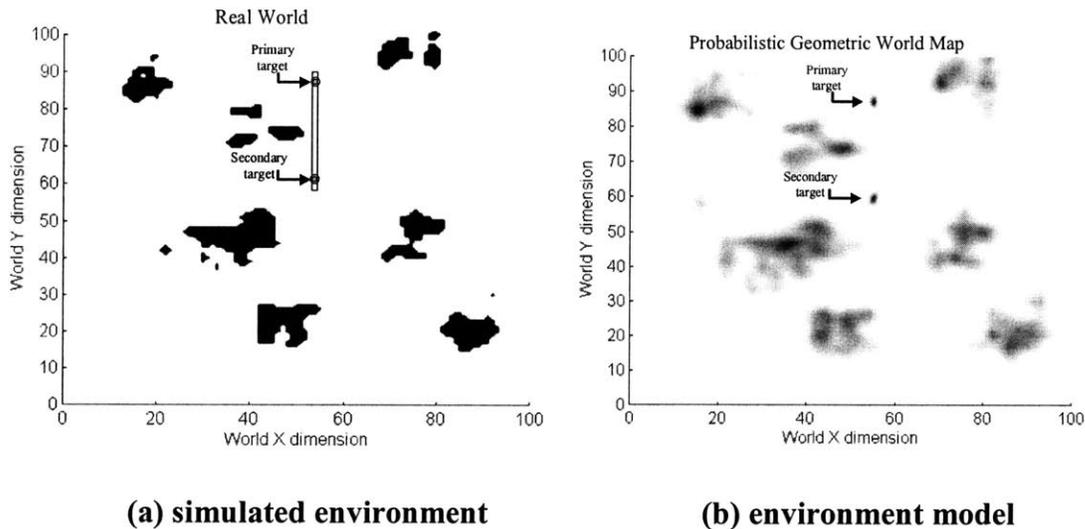


Figure 4-10: Simulated planar environment

Figure 4-11 shows a few steps in the selection of optimum placement of a vision system for fixed target monitoring in the presence of a moving object (with known CAD model). Initially (Figure 4-

11a) the moving object is far enough away so that the vision system can look directly at the primary target. As the moving object approaches the primary target (Figure 4-11b), the algorithm repositions the vision system to avoid making contact with the target and other occlusions. Finally, the object is too close to the primary target making it more effective for the vision system to monitor the secondary target, rather than the primary target (Figure 4-11c). Note, the vision system is actually modeled as a single point located at the front principal node of the lens system of Figure 4-11.

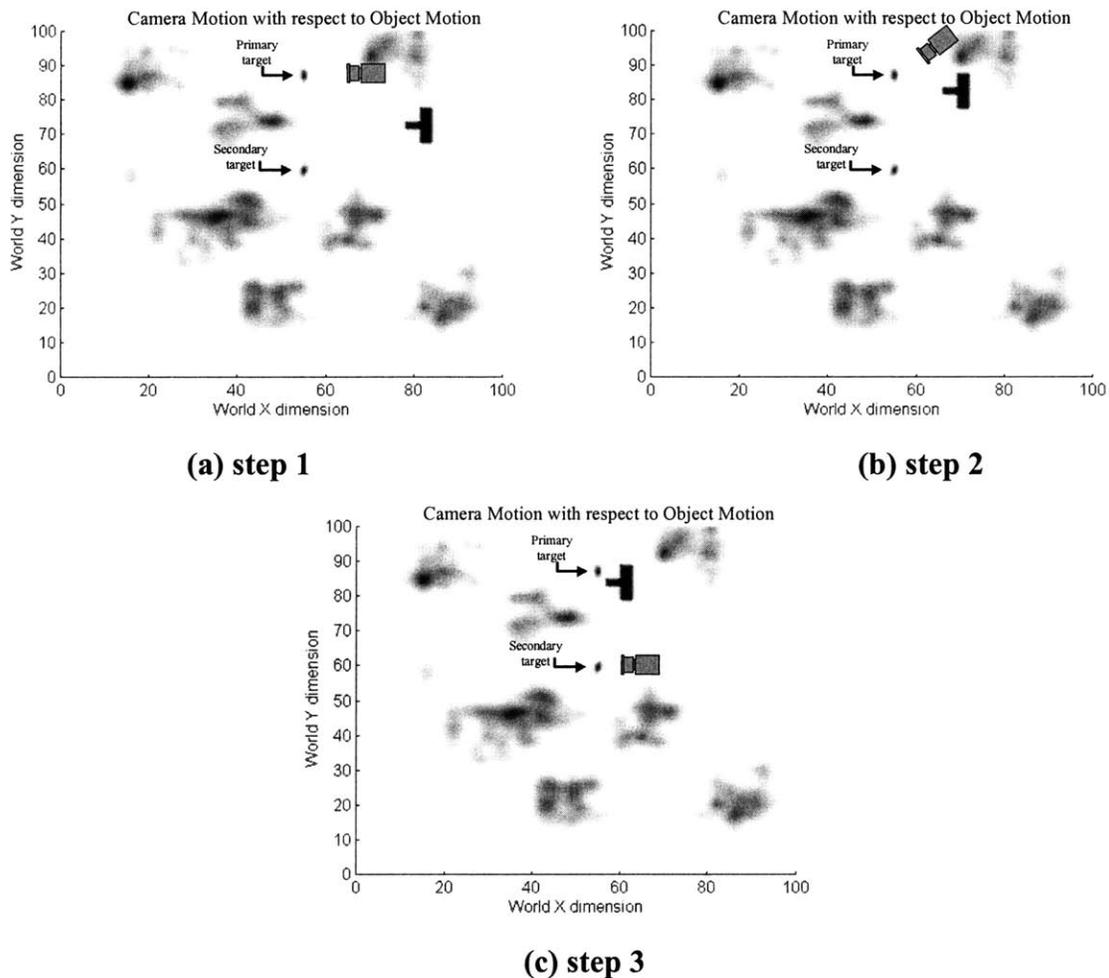


Figure 4-11: Optimal camera placement with moving object (known CAD model)

Table 4-1 presents the results of simulating a similar scenario. However, here the task is modified. The vision system is used to guide the object to the goal (insertion site). Hence, the target is the

distance between the object and the goal. The task is successful if the object can be visually guided to the goal within the defined tolerances. The simulation is carried out 300 times and compares the approach developed in this chapter with two other methods: biased random placement (allowing for re-placement during task execution with some heuristics) and biased random fixed placement (not allowing for re-placement during task execution). This is experimentally demonstrated in Chapter 5.

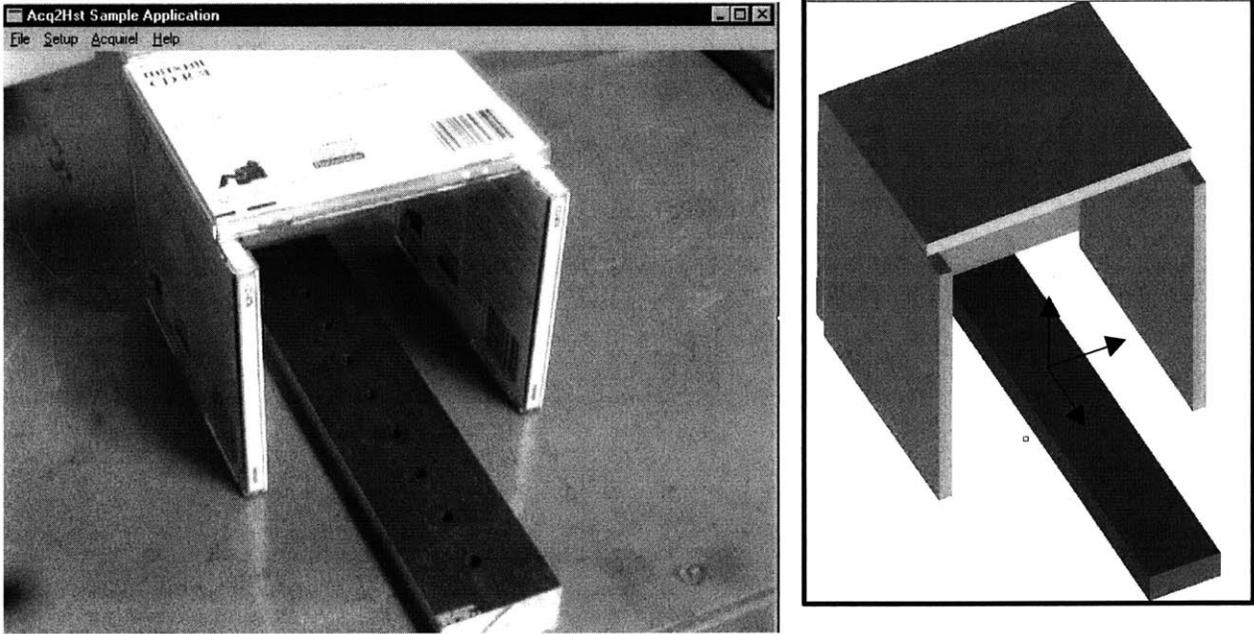
Table 4-1: Results of changing task difficulty, occlusion density and task execution success

300 tests per scenario		Occlusion Density 1 (5%)		Occlusion Density 2 (20%)		Occlusion density (35%)	
		Without secondary target	With secondary target	Without secondary target	With secondary target	Without secondary target	With secondary target
Task difficulty: easy→20% tolerance	Optimal camera re-placement Success (%)	100	100	76	95	13	25
	Random camera re-placement Success (%)	51	63	18	31	5	10
	Random camera placement Success (%)	45	58	16	28	5	9
Task difficulty: medium→10% tolerance	Optimal camera re-placement Success (%)	99	100	63	86	10	18
	Random camera re-placement Success (%)	30	37	11	18	3	6
	Random camera placement Success (%)	23	30	8	15	3	4
Task difficulty: hard→1% tolerance	Optimal camera re-placement Success (%)	97	99	30	52	3	7
	Random camera re-placement Success (%)	3	4	1	2	<<1	1
	Random camera placement Success (%)	1	1	<<1	<<1	<<1	<<1

Note that as task difficulty increases, the optimal camera placement algorithm continues to perform well. The influence of secondary targets is seen as task difficulty or occlusion density increases.

4.3.2. Experiments

Figure 4-12(a) shows an arrangement of simple block components. The target consists of a workpiece with holes. For this demonstration, an accurate 3-D CAD model with added uncertainty of the arrangement is provided to the planning algorithm (see Figure 4-12(b)). The optimal camera placement is found by optimizing RF in 3-D. For the arrangement shown, this position is $(r, \theta, \phi) = (5", 0, \pi/3)$, where r is the radial distance, θ is azimuth angle, ϕ is the elevation angle—all measured with respect to the primary target coordinate frame. Note that the camera principal axis always points directly to the origin of the target (center of the hole). Figure 4-13 shows the camera view of the target from this point.



(a) Environment setup (b) CAD model

Figure 4-12: Experimental test

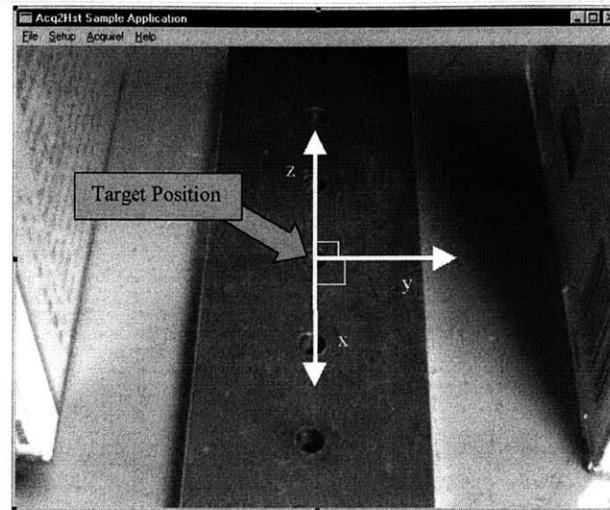


Figure 4-13: Experimental test- Camera view of target from optimum pose

4.4. Conclusions

In field environments, it is often not possible to provide robotic teams with detailed a priori environment and task models. In such unstructured environments, cooperating robots will need to create a 3-D geometric model positioning their sensors in a task directed optimal way. A new algorithm based on iterative sensor planning and sensor redundancy is proposed. This algorithm overcomes problems due to camera occlusions from fixed poses, to build the 3-D environment model, and to position sensors. The environment modeling stage of the algorithm was developed in detail in the previous chapter. This chapter addresses the process of optimal camera placement, given the developed environment model. This algorithm is based on iterative sensor planning and exploiting the sensor redundancy of cooperative robotic systems. A rating function is developed and optimized to find the most suitable pose to view the target. Simulations and experiments show promising results.

Cooperative Task Execution

5.1. Problem Overview

In chapter 4 a method to position a vision system in a task directed optimal way, was presented. Simulation and experimental results show the effectiveness of this algorithm for single point monitoring. Simulation results were also presented for the task of guiding an object to a goal. An example of such a task would be the insertion of a computer card into a mating slot in a field environment—a typical maintenance task. In this chapter, an experimental demonstration of this task using two mobile field robots is presented (see Figure 5-1).

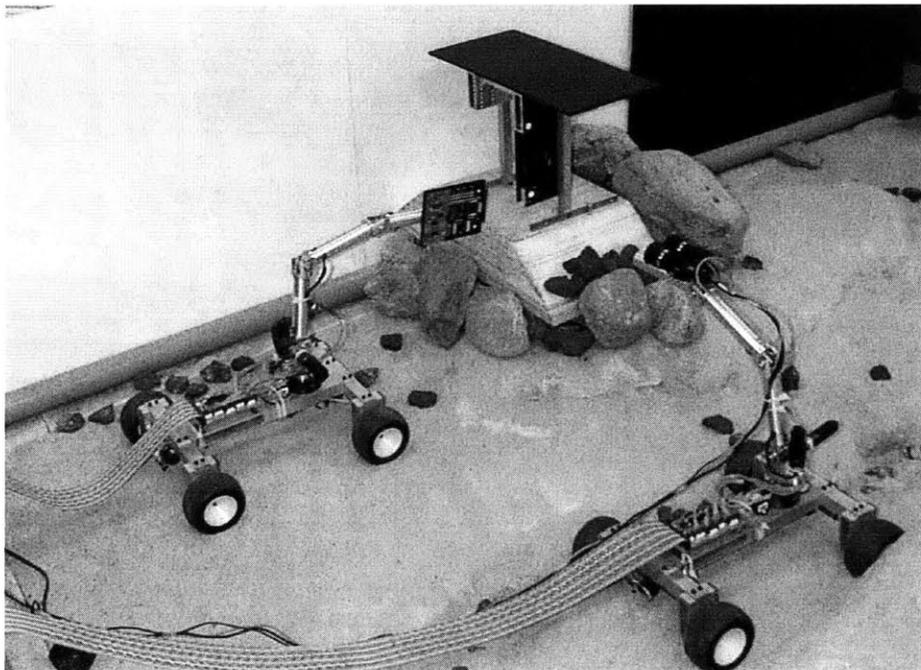


Figure 5-1: Cooperative insertion task layout

The target for this task is the distance between the object and insertion site. This can be viewed in Figures 5-2 and 5-3.

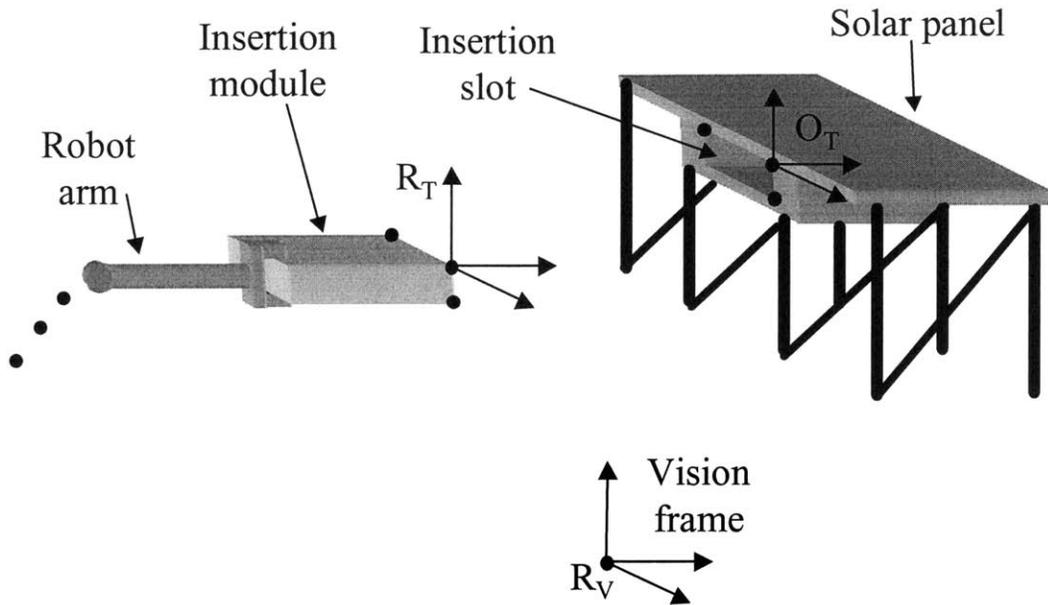


Figure 5-2: Cooperative task target model - representative problem

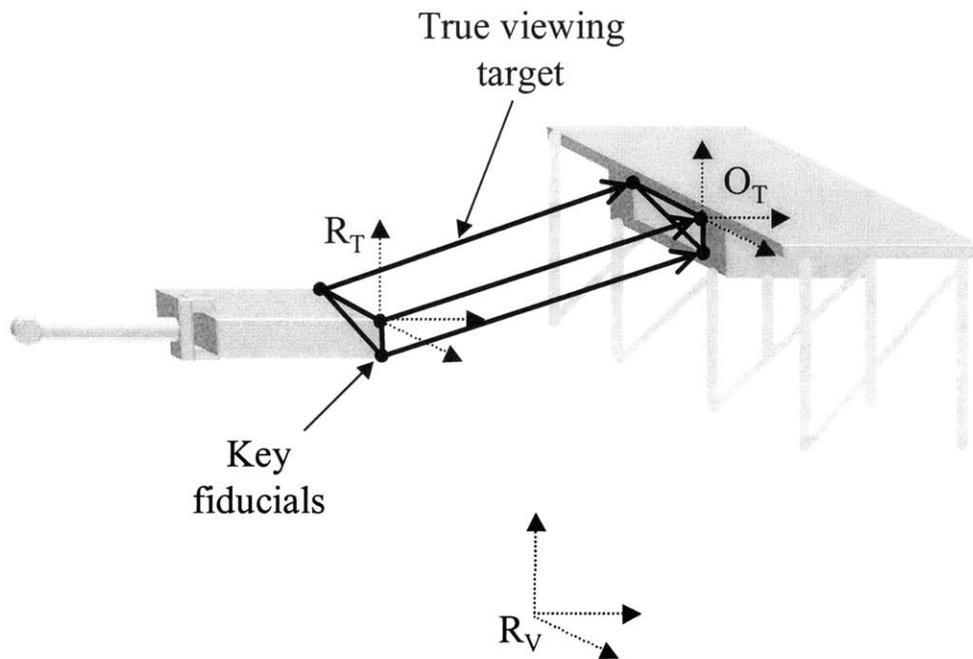


Figure 5-3: Cooperative task target model - extrapolation of true viewing target

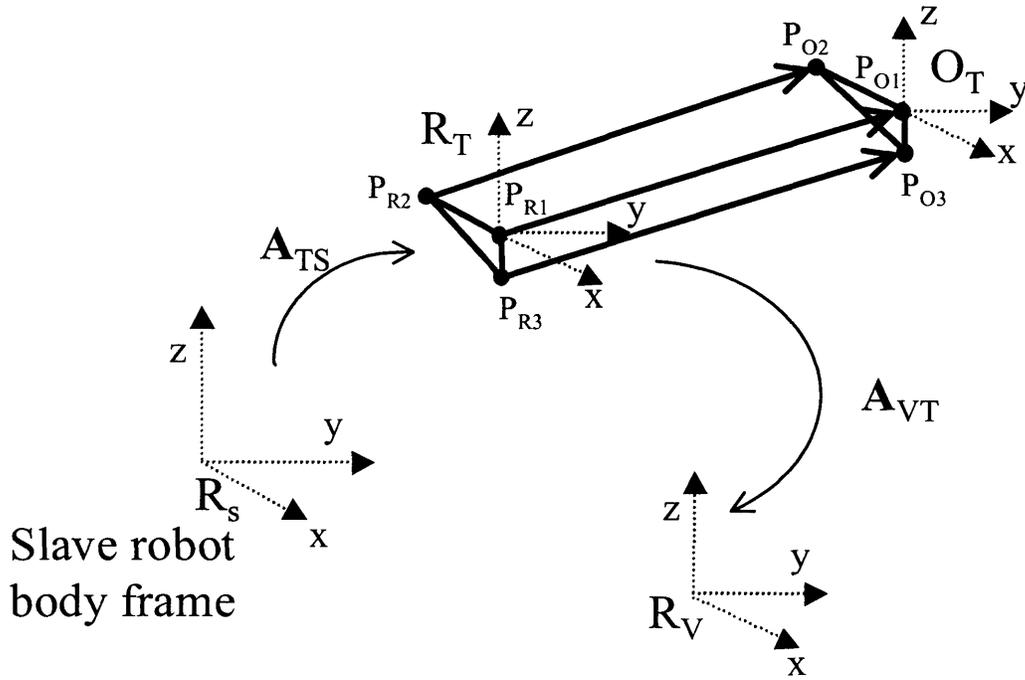


Figure 5-4: Cooperative task target model - relating the coordinate frames of the cooperating robots

The slave robot manipulates the insertion object and the master robot visually guides the slave robot. To determine the required motion by the slave robot, the two robot coordinate frames must be related. In the camera frame the rotational angles (Euler angles) for any set of cartesian axes (formed by 3 points set at the origin and one along each of two principle directions) with respect to the camera frame (as shown in figure 5-4) is given by:

$$\begin{aligned}
 R_z &= \tan^{-1} \left(\frac{P_{1y} - P_{2y}}{P_{1x} - P_{2x}} \right) \\
 R_y &= \tan^{-1} \left(\frac{P_{1z} - P_{2z}}{\sqrt{P_{1x}^2 + P_{1y}^2} - \sqrt{P_{2x}^2 + P_{2y}^2}} \right) \\
 R_x &= \tan^{-1} \left(\frac{P_{3y} - P_{1y}}{P_{1z} - P_{3z}} \right)
 \end{aligned} \tag{5-1}$$

The transformation matrix from the vision system (\mathbf{R}_v) to the slave robot end-effector frame

(\mathbf{R}_T) is given by:

$$\mathbf{A}_{VT} = \begin{bmatrix} \mathbf{R}_{R_T}^x \cdot \mathbf{R}_{R_T}^y \cdot \mathbf{R}_{R_T}^z & \bar{\mathbf{P}}_{R_1} \\ \bar{\mathbf{0}} & 1 \end{bmatrix} \quad (5-2)$$

From the slave robot arm kinematics, a transformation matrix from the arm base (\mathbf{R}_S) to the arm end-effector (\mathbf{R}_T) is given as:

$$\mathbf{A}_{TS} = \begin{bmatrix} \mathbf{R}_{TS} & \bar{\mathbf{P}}_{TS} \\ \bar{\mathbf{0}} & 1 \end{bmatrix} \quad (5-3)$$

Thus, in the camera frame (\mathbf{R}_V) , the required object motion to reach the goal is given by:

$$\begin{aligned} \text{Required Translation} &= \bar{\mathbf{P}}_{O_1} - \bar{\mathbf{P}}_{R_1} \\ \text{Required Rotation} = \mathbf{R}_O - \mathbf{R}_{R_T} &= \begin{bmatrix} \mathbf{R}_O^x - \mathbf{R}_{R_T}^x \\ \mathbf{R}_O^y - \mathbf{R}_{R_T}^y \\ \mathbf{R}_O^z - \mathbf{R}_{R_T}^z \end{bmatrix} \end{aligned} \quad (5-4)$$

Since the transformation of the camera frame (\mathbf{R}_V) and the slave robot base frame (\mathbf{R}_S) can be inferred from \mathbf{A}_{VT} and \mathbf{A}_{TS} , Equation 5-4 can be transformed in the robot base frame (\mathbf{R}_S) , giving the required motions to get the object to the goal. Note that to achieve the required object motions, both vehicle and arm motions are required. Since vehicle motions are not known exactly, the vision guide acts in a classical visual servo control mode.

5.2. Experimental Setup

For the experimental demonstration of the computer card-to-insertion slot task, certain assumptions/simplifications have been made:

1. The insertion problem is assumed to be planar
2. Since motions are relatively slow, dynamic effects are assumed to be negligible. However, inclinometer measurements of vehicle pitch and roll are not negated. These dynamic effects could not be negated if (a) the vision system is looping at a much slower rate than servo loop,

- (b) task requirements are high (tolerances are tight), and (c) task needs to be done fast.
3. Optimal re-positing of the vision system is assumed to be within the kinematic limitations of the manipulator with linear (straight line) vehicle base motions
 4. An environment model consisting primarily of the computer card and insertion site is generated using the method proposed in Chapter 3, and assumed to be sufficient (optimal camera poses and a neighborhood of these poses are visible and free of natural occlusions)

These assumptions do not violate the principles and algorithms presented for optimal camera placement and cooperative task execution. The assumptions have been included here due to hardware limitations (kinematic limitations of the rovers and lack of all necessary sensors).

The experimental system was introduced in Chapter 2 with more details in Appendix D. The computer card and the insertion site each have two clear circular markers (see Figure 5-5). Identification of these markers using grayscale cameras is presented in Section 5.3. As described in Chapter 4, this choice of marker type is arbitrary, but is a convenient choice based on the vision hardware.

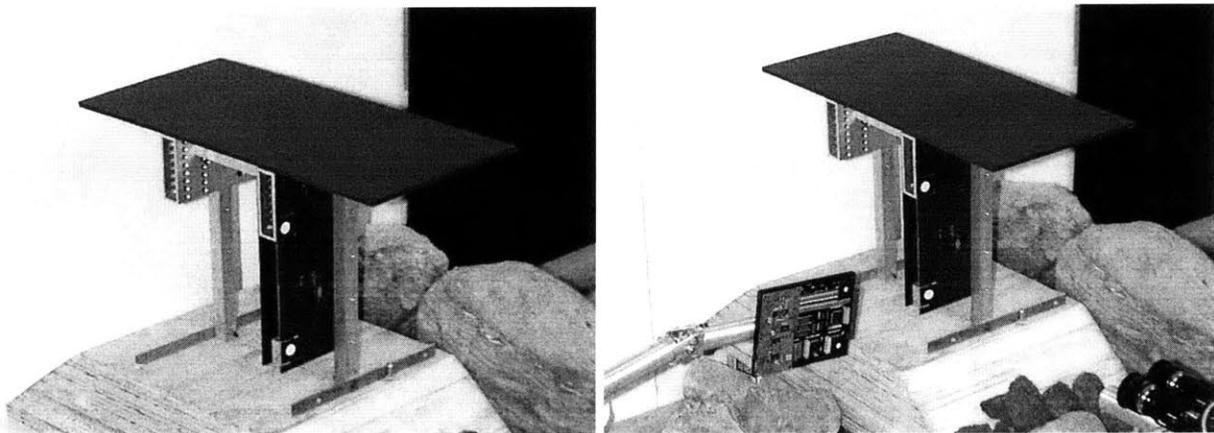


Figure 5-5: White circular markers to locate object and insertion site

Task execution consists of two stages: goal approach and object insertion. In the first stage, goal approach is accomplished using visual guidance, as described before. In the second

stage, object insertion is accomplished using force feedback models developed in Appendix A. The model-based sensor fusion for these two stages are shown schematically in Figure 5-6.

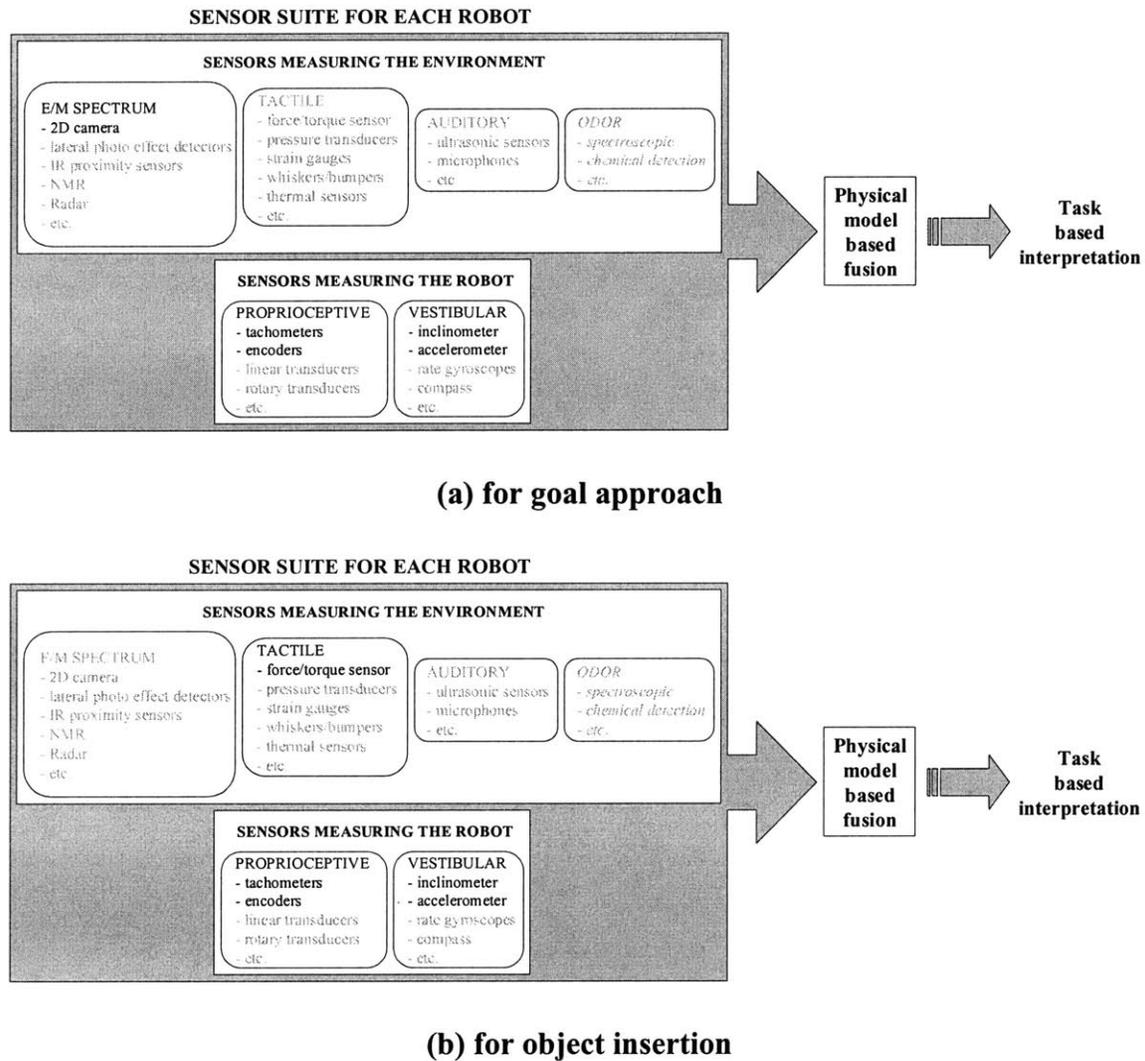


Figure 5-6: Model based sensor fusion from a sensor suite

5.3. Marker Identification

To detect the circular markers (or holes), an algorithm capable of recognizing circles and computing the coordinates of the circle centers in grayscale 2-D images is needed. This is described here. Two steps are required to obtain circle center coordinates: image reduction to black and white, and a Hough transform. In the first step, the image is reduced to a black and

white image of edge points. The black points represent candidate points that may lie along the circumference of a circle. This is accomplished in a sequence of simple image processing steps with the only assumption that points lying in the circular markers (or holes) tend to differ significantly in intensity than points not lying in markers. This sequence of steps is generated by experimentation and proves to be robust to significant lighting variation and image noise. These steps are outlined as follows:

- Process the image through a histogram equalization filter: this helps reduce variations due to illumination conditions (for simplicity, in this work illumination was not varied).
- Threshold the image to obtain a black and white image: the threshold value is selected based on the expected value of the pixels located in the circular holes. Clearly, this value will change depending on illumination conditions. However, the histogram equalization step reduces the variation in images due to this variable.
- Image is processed with a Laplacian filter: this edge detection filter is defined as the sum of second derivatives, or in 2-D this is given as:

$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (5-5)$$

This can be discretized based on second central differences. Considering only first order effects, we get the following 3x3 mask for the 2-D discrete Laplace filter:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (5-6)$$

This steps identifies pixels that tend to belong to an edge (the rim of the marker).

- Image filtered with a 3x3 constant low pass filter: this effectively blurs the image, reducing image noise. This tends to give “well behaved” image histograms, with a single clean peak (see Figure 5-7).

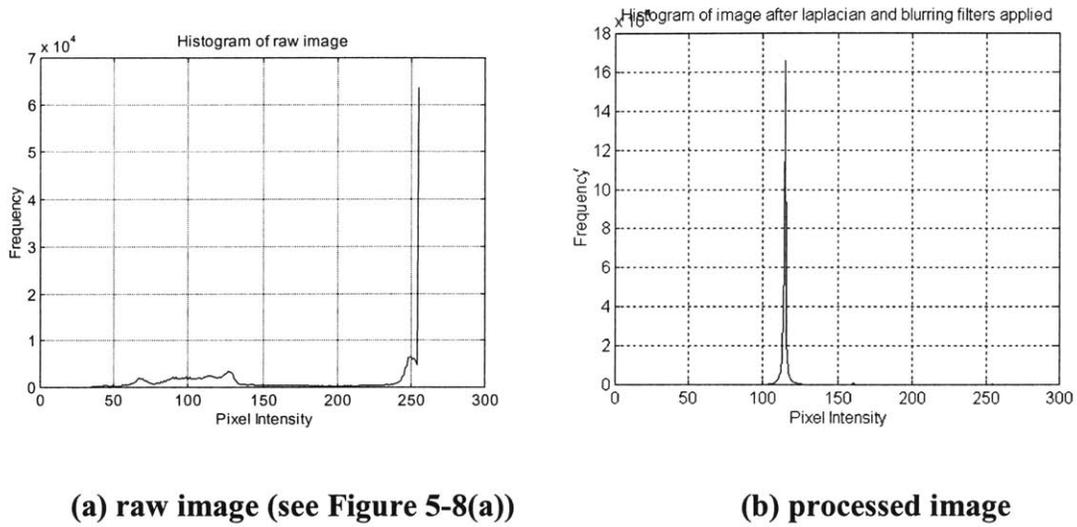
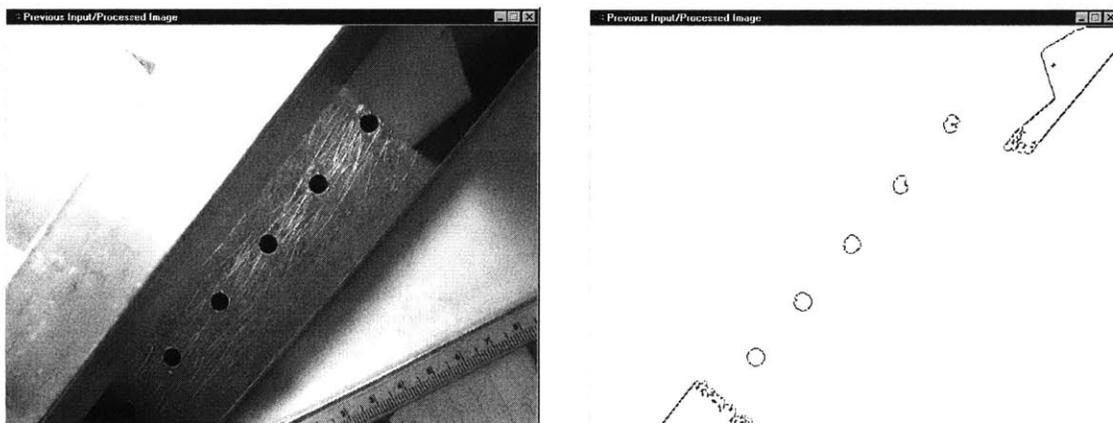


Figure 5-7: Histogram of images

- Threshold the image to obtain a black and white image: the threshold value is empirically selected as `histogram_peak_value - 5`. As an example, Figure 5-8(b) shows the reduction of the image in Figure 5-8(a).



(a) Original image **(b) Reduced (black and white) image**

Figure 5-8: Example of image reduction process

In the second step, the coordinates of the holes are found using a Tracking Hough transform (THT), which is an extension the general Hough transform [Lara]. Only black pixels are considered here and are referred to as feature points . Three feature points are selected at random. The circle radius and center are defined by 3 points.

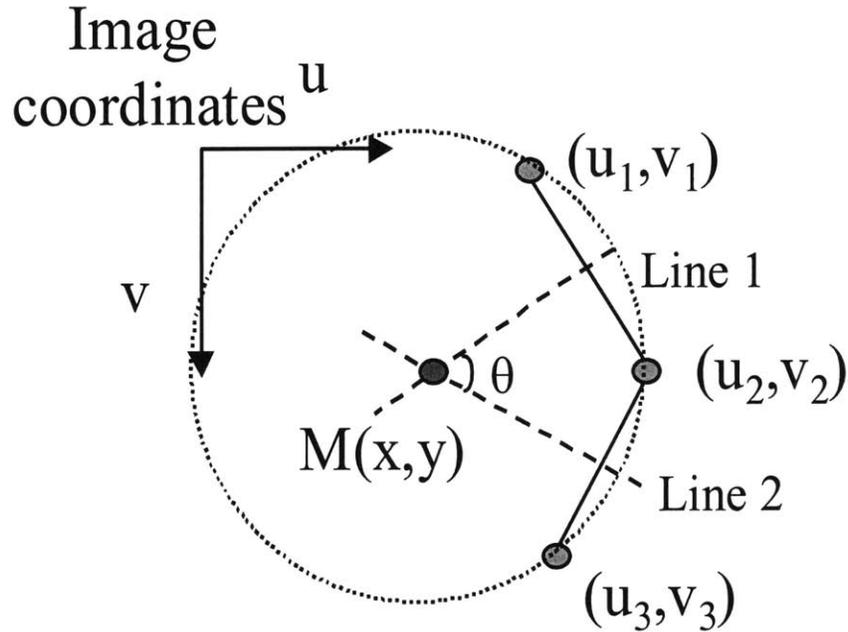


Figure 5-9: Circle defined by three points

From Figure 5-9, we can solve for (x,y), the circle center, by solving:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -m_1 & 1 \\ -m_2 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} y_1 - m_1 x_1 \\ y_2 - m_2 x_2 \end{bmatrix} \quad (5-7)$$

where:

$$\begin{aligned} m_1 &= \frac{u_2 - u_1}{-v_2 + v_1} & m_2 &= \frac{u_3 - u_2}{-v_3 + v_2} \\ x_1 &= \frac{u_2 + u_1}{2} & x_2 &= \frac{u_3 + u_2}{2} \\ y_1 &= \frac{v_2 + v_1}{2} & y_2 &= \frac{v_3 + v_2}{2} \end{aligned} \quad (5-8)$$

Additionally, we can solve for r (circle radius) and θ (angle of intersecting segment bisectors) as:

$$r_{radius} = \frac{\sum_{i=1}^3 \sqrt{(x-u_i)^2 + (y-v_i)^2}}{3} \quad (5-9)$$

$$\theta = \arccos\left(\frac{\overline{a_1} \cdot \overline{a_2}}{|\overline{a_1}| |\overline{a_2}|}\right) \quad \text{where} \quad \overline{a_1} = \begin{bmatrix} -v_2 + v_1 \\ +u_2 - u_1 \end{bmatrix} \quad \overline{a_2} = \begin{bmatrix} -v_3 + v_2 \\ +u_3 - u_2 \end{bmatrix}$$

To increase the likelihood that the three points lie on a circle of interest, certain restrictions are placed on feature point selection:

- The first point is chosen at random
- The second point is selected at a distance $< D$ from the first point; where D is the projected diameter of the hole onto the image
- The third point is selected at a distance $< D$ from the first and second point. Additionally, the angle generated by the intersecting segment bisectors (of segments formed by points 1 & 2 and 2 & 3) $> d\theta$. This ensures that the three points do not lie along or almost along a straight line. $d\theta$ is chosen empirically to be $\pi/6$.

Once a set of three points is identified to lie along a circle, they are removed from the list of feature points. The process is continued until all feature points have been processed. Finally, a circle is identified if a substantial percentage of its circumference has been identified (empirically chosen as $>35\%$). To improve the accuracy of the center coordinates, a standard region growing procedure is applied at the identified center coordinates. The new pair of center coordinates is defined as the centroid of the “grown” region. A final result is shown in Figure 5-10. This entire process takes approximately 0.3 seconds on a Pentium 166 MHz system. Hence the visual loop runs at about 3-4 Hz, whereas the servo control loop runs at about 100 Hz. Commanded motions must account for this for effective visual servo control. Although faster computers may easily be employed, markers may be identified more quickly using color cameras

and palette matching methods.

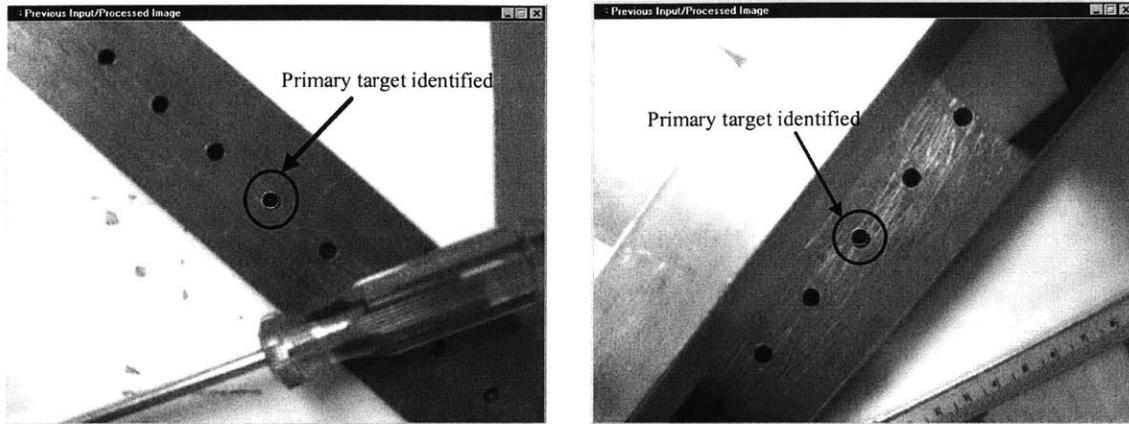


Figure 5-10: Identification of marker (centermost hole with 0.25” diameter)

5.4.Task Execution Results

Goal approach is performed using the optimal camera placement algorithm and compared with random camera placement. Note that some heuristics are applied to the random camera placement method to keep camera poses bounded to reasonable locations. The insertion object (computer card) is 4.25” wide. The experiments are carried out for three different task difficulty levels: low difficulty (~18% object width tolerance in insertion site), medium difficulty (~6% object width tolerance in insertion site) and high tolerance (~3% object width tolerance in insertion site). Table 5-1 presents the results of the two methods for 10 attempts at the three different task difficulty levels. Figure 5-11 shows four steps in the task execution process using optimal camera placement. Figure 5-12 shows the stereo camera view during an approach and object-insertion site contact step. Note that the vision system repositions during task execution. However, these motions are relatively small for this scenario. Figures 5-13 and 5-14 show the object pose convergence using optimal placement (successful) verses random placement (unsuccessful) for a high difficulty task example. Also shown are the base sensed forces in the plane of motion after object-insertion site contact is made. At this point, the force-feedback

insertion method developed in Appendix A attempts to guide the object into the insertion slot. It is seen that with good initial object-insertion site alignment achieved using optimal camera placement, this method works well (i.e. it results in low interaction forces).

Table 5-1: Task execution success for varying task difficulty levels

	Optimal camera placement	Random placement
Easy (insertion site width: 5.0")	10/10	7/10
Medium (insertion site width: 4.5")	10/10	4/10
Hard (insertion site width: 4.375")	9/10	2/10

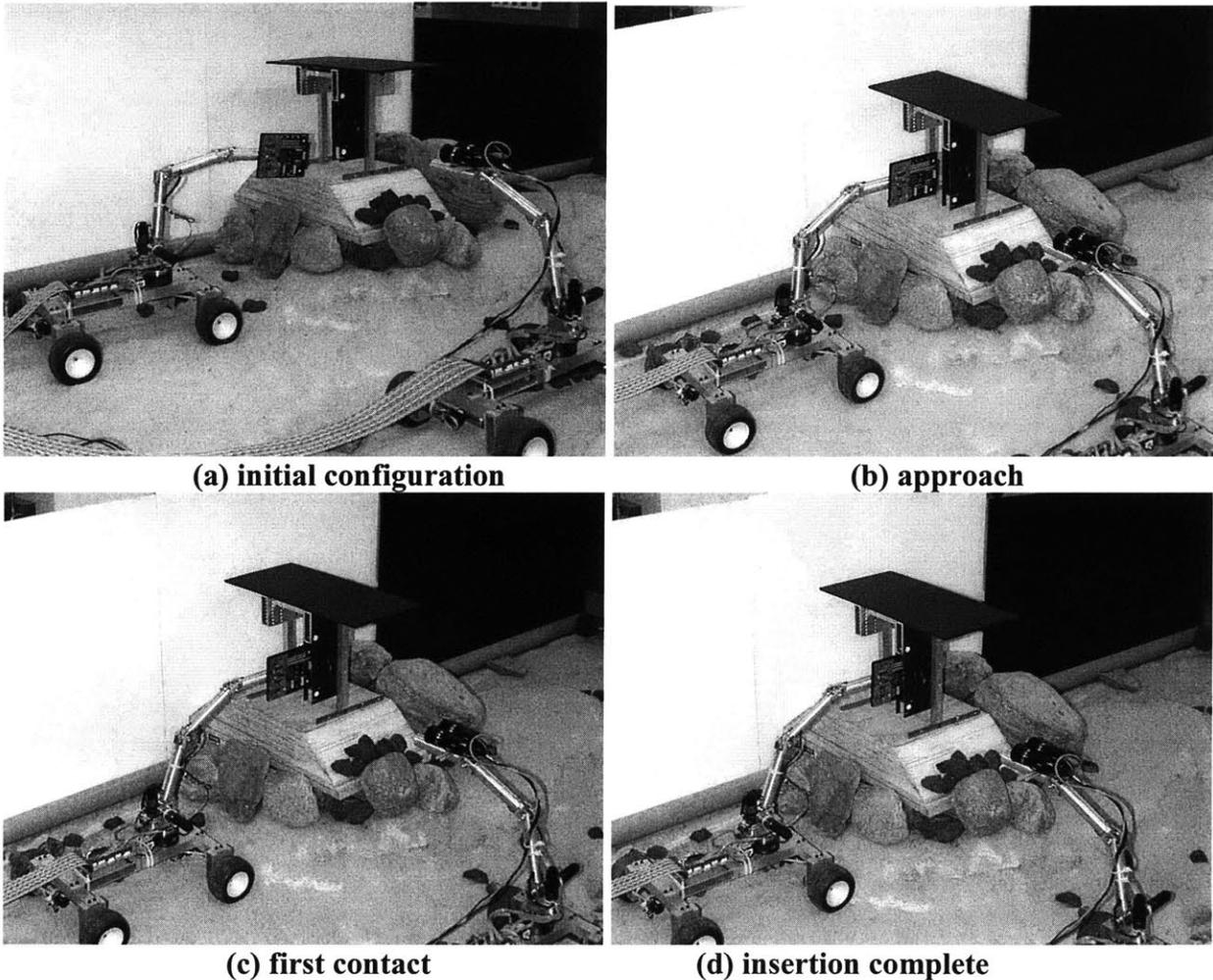
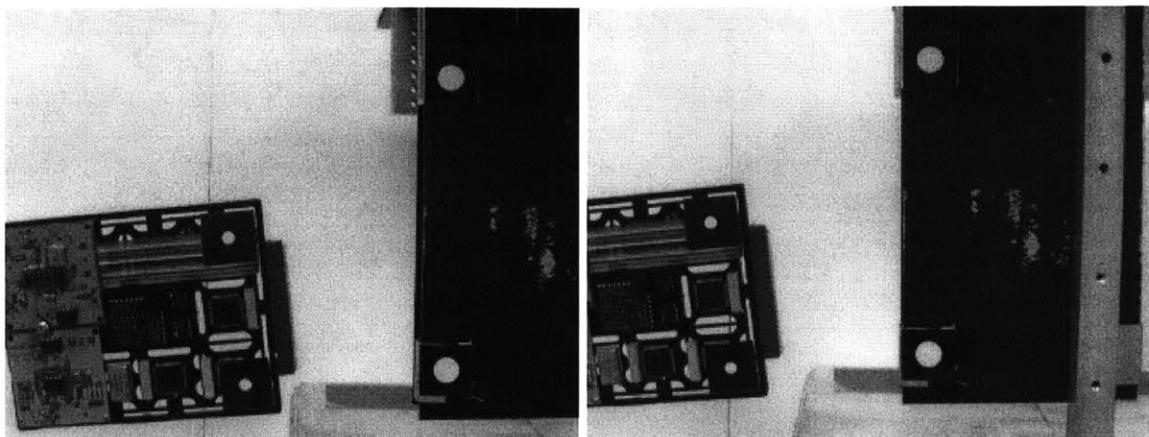
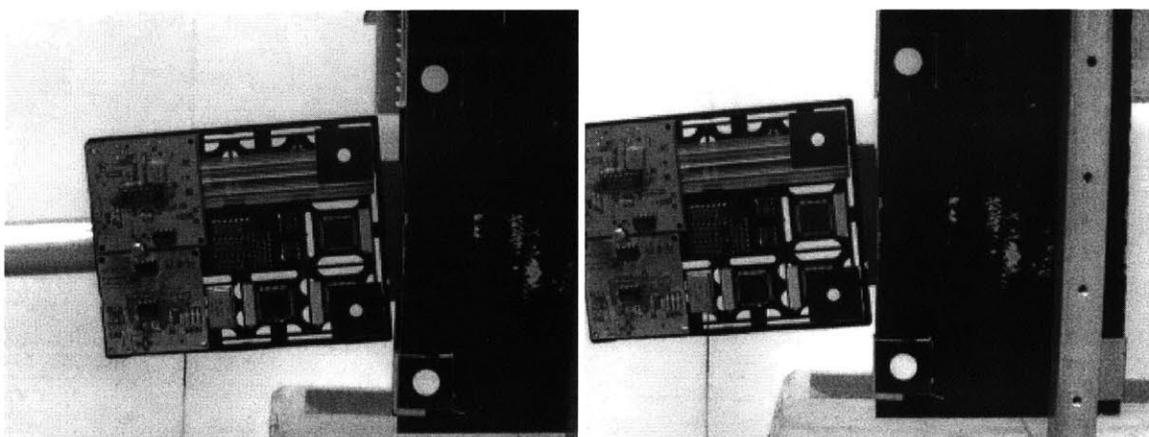


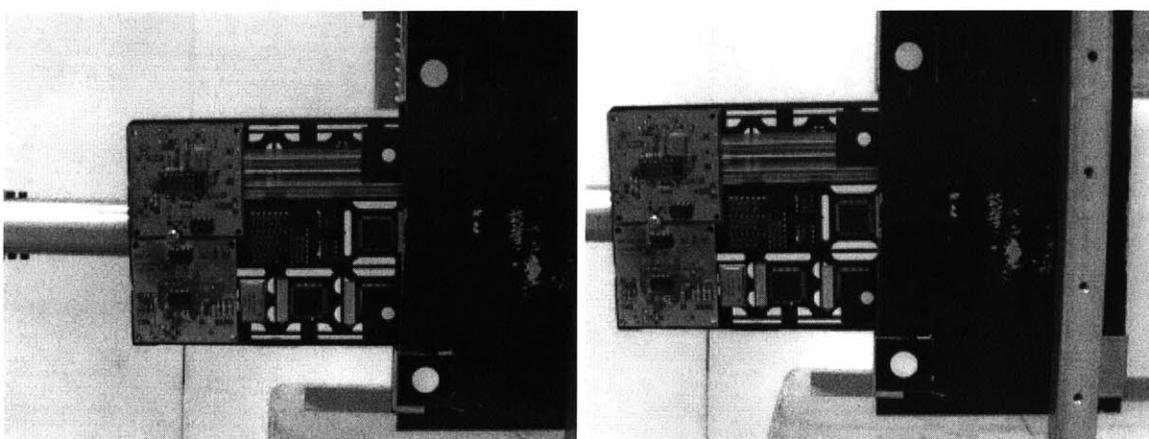
Figure 5-11: Intermediate steps during task execution (success)



(a) approach (left and right images)

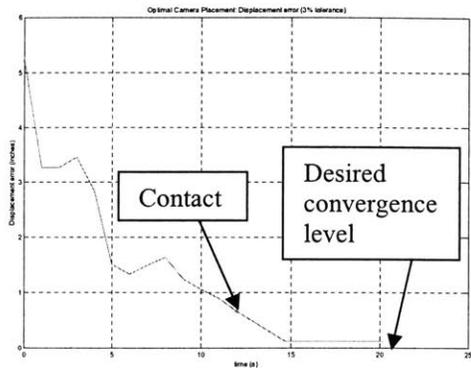


(b) almost contact (left and right images)

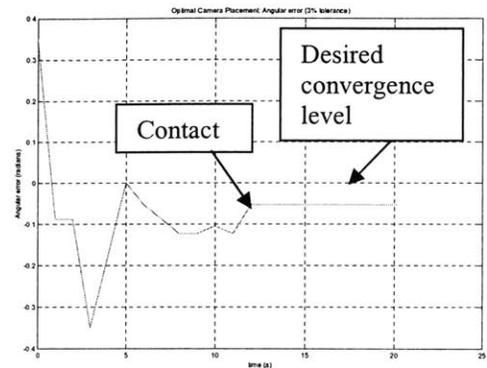


(c) first contact (left and right images)

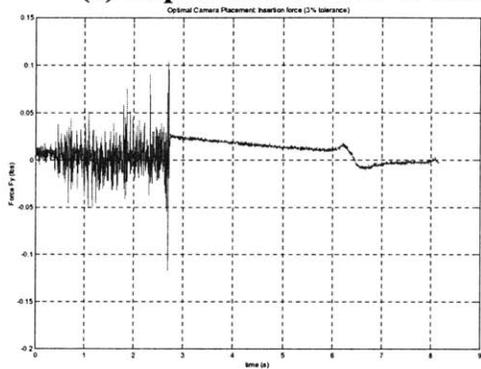
Figure 5-12: Camera view (stereo pairs)



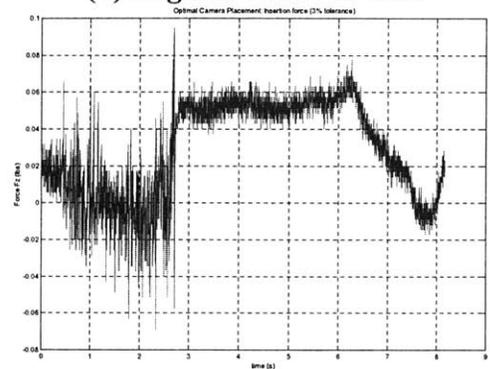
(a) displacement error vs time



(b) angular error vs time

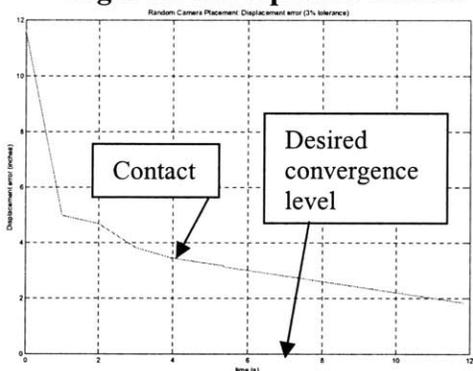


(c) Force F_y vs time (after contact)

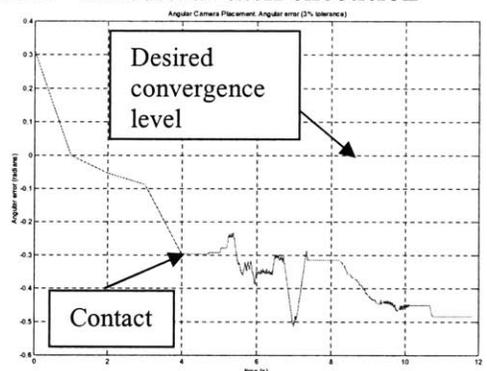


(d) Force F_z vs time (after contact)

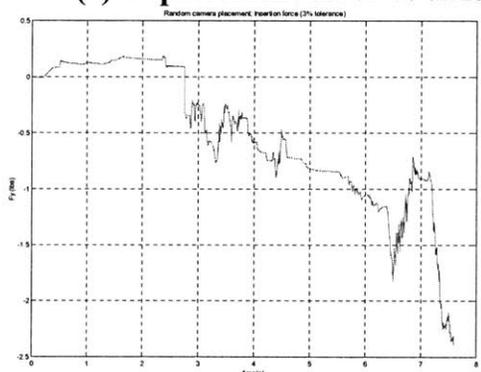
Figure 5-13: Optimal camera placement—successful task execution



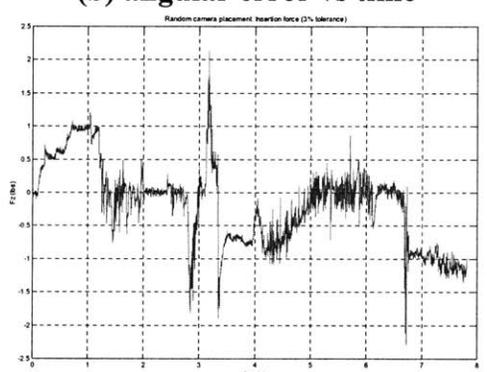
(a) displacement error vs time



(b) angular error vs time



(c) Force F_y vs time (after contact)



(d) Force F_z vs time (after contact)

Figure 5-14: Random camera placement—unsuccessful task execution

5.5. Conclusions

This chapter addressed the process of task modeling with optimal camera placement on an experimental platform, given the developed environment model. The task here is to cooperatively insert an object (a computer card) in an insertion site (representing a section of a solar panel array) in an unstructured environment—known as a cooperative guiding task. The algorithm is developed in Chapter 4 where simulations for the cooperative guiding task show promising results. Here, experiments compare task execution using the task directed optimal camera placement algorithm with random camera placement. These show high improvements in performance, especially as task difficulty increases.

Conclusions and Future Work

6.1. Contributions of This Thesis

The performance of cooperative robots in field environments is limited by model uncertainties and on-board sensor limitations. This thesis has developed sensing and estimation algorithms to enable multiple mobile robots to compensate for model uncertainties and successfully perform interacting cooperative tasks in highly unstructured/unknown field environments using optimal information gathering methods.

In Chapter 2, an algorithm based on adaptive excitation of vehicle dynamics to enable mobile robots in field environments to efficiently estimate their dynamic parameters, including the mass, location of center of gravity, inertia, base compliance and damping, was presented. The algorithm used an onboard robotic arm to generate base motions, which were measured with simple onboard sensors, and fit to a physical model. A mutual information theoretic basis for a metric on parameter identification was developed. This metric provided a measure of how well the value of a given parameter was known. Using this metric, the arm trajectory was refined. Simulations and experimental results show the effectiveness of this algorithm.

Chapter 3 addressed the problem of incomplete or unknown environment models. In field environments it is often not possible to provide robotic teams with detailed a priori environment and task models. In such unstructured environments, cooperating robots need to create a

dimensionally accurate 3-D geometric model by performing appropriate sensor actions. However, uncertainties in robot locations and sensing limitations/occlusions make this difficult. A new algorithm based on iterative sensor planning and sensor redundancy was presented to build a geometrically consistent dimensional map of the environment for mobile robots that have articulated sensors. This algorithm is unique in that it used a metric of the quality of information previously obtained by the sensors to find new viewing positions for the cameras. Simulations and experiments showed very promising results.

Chapter 4 addressed the problem of task modeling using the process of camera placement in a task directed optimal way, given the developed environment model. The algorithm was again based on iterative sensor planning and exploiting sensor redundancy of cooperative robotic systems. The algorithm used a priori task and target definitions to formulate the placement plan. A rating function was developed and optimized to find the most suitable pose to view the target. Although, the algorithm was applied to the optimal placement of a single camera pair, it is directly applicable to multiple systems. Simulations and experiments showed promising results. Chapter 5 combines the algorithms developed in this thesis to execute a cooperative insertion task in an unstructured environment. This was done on an experimental platform consisting of two mobile field robots (also developed for this work) using a model predictive control architecture (developed in Appendix A).

6.2. Suggestions for Future Work

This thesis has shown that mobile field robots, working in highly unstructured and unknown environments, can compensate for sensing limitations and for uncertainties in the environment, task, and robot by cooperatively developing appropriate robot dynamic models, environment models and task models. These models are then used for successful task execution.

Although substantial work has been completed in this area, some improvements can be achieved in the presented methodologies.

- (a) Cooperative stereo vision—In the environment modeling algorithm, presented in Chapter 3, each individual robot directly contributes range data to the environment model using onboard stereo vision systems. Extrinsic parameters, such as camera baseline distance, limit the resolution. Often higher resolutions may be needed without allowing the vision system to compensate for short baseline distances by getting closer to the target (terrain may be untraversable, kinematic limitations, target too far away, etc). As such a promising solution is to use cooperative stereo vision, where two or more robots combine their views and locations to form a more accurate range measurement. Uncertainties in robot/camera locations and orientations would have to be accounted for. The environment modeling algorithm may be easily modified to accommodate cooperative stereo imaging.
- (b) Vehicle/ground friction coefficient estimation and slip prediction—Model predictive control compensation is shown to be highly effective in disturbance rejection and improving robot performance (Appendix A). Chapter 2 presents a method to estimate the dynamic parameters needed for such models. However, it is assumed that the individual robots are stationary during the estimation process. This constraint may be mathematically relaxed by incorporating friction models and estimating the friction coefficients along with the dynamic system parameters. Once estimated, the friction coefficients could be used in model predictive control compensation methods to predict the incipience of vehicle slip during task execution.
- (c) Payload identification—In addition to dynamic parameter estimation, the algorithm presented in Chapter 2, may be readily extended to estimating the inertial parameters of a payload. This

problem gets more intricate when the unknown payload is grasped by multiple robots/manipulators. This scenario would require cooperative excitation of the payload dynamics.

- (d) Swarm robots—Although, the algorithms developed in this thesis have not assumed single or a few cooperating systems, they have only been tested as such. Emergent behaviors and computation/communication limitations may not be entirely predictable until the algorithms have been tested on robotic swarm systems. Such behaviors would be used to develop local rules that would be used by the robotic swarm to achieve a given task. For example, the robotic swarm may be used for information (intelligence) gathering in such environments as battlefields, large scale extraterrestrial exploration, etc. This is a similar scenario to environment modeling (Chapter 3). However, modifications may be needed to define individual roles/behaviors. Additionally, in this problem, individual robots (resources) can be allocated to regions of interest. As the swarm size and functionality increase, there are increasing concerns on the handling the quantity of data, the complex bi-directional communications, and the limited computational resources. This may be addressed by the psychological phenomenon of attention (nature’s solution to the complexity problem in the context of sensory processes interacting with cognitive processes) [Barbastathis].

References

- Alloum, A.; Charara, A.; Machkour, H. *Parameters nonlinear identification for vehicle's model*. Proceedings of the 1997 IEEE International Conference on Control Applications. 5-7 Oct. 1997. Hartford, CT, USA.
- Alur, R., Das, A., Esposito, J., Fierro, R., Grudic, G., Hur, Y., Kumar, V., Lee, I., Ostrowski, J., Pappas, G., Southall, B., Spletzer, J. and Taylor, C.J. *A Framework and Architecture for Multirobot Coordination*. Proceedings of the International Symposium on Experimental Robotics 2000, December 10 – 13, 2000, Waikiki, Hawaii.
- Anousaki, G.C., Kyriakopoulos, K.J. *Simultaneous localization and map building for mobile robot navigation*. IEEE Robotics & Automation Magazine , Volume: 6 Issue: 3 , Sept. 1999 Page(s): 42 -53
- Armstrong, B. *On finding exciting trajectories for identification experiments involving systems with nonlinear dynamics*. International Journal of Robotics Research, Vol. 8, No. 6, December 1989.
- Asada, M. *Map building for a mobile robot from sensory data*. IEEE Transactions on Systems, Man, and Cybernetics. Vol. 37, no. 6, November/December 1990.
- Ashby, M. *Material selection in mechanical design*. Butterworth-Heinemann, Oxford, 1992.
- Atkeson, C.G., An, C.H., and Hollerbach, J.M. *Estimation of inertial parameters of manipulator loads and links*. Int. Jour. of Robotics Research. Vol. V, No. 3, 1986.
- Barbastathis, G. *Intelligent holographic databases*. Ph.D. Thesis, Department of Electrical Engineering, California Institute of Technology, 1998.
- Bard, Y. *Nonlinear parameter estimation*. Academic Press. London, 1974.
- Baumgartner, E.T., P. S. Schenker, C. Leger, and T. L. Huntsberger. *Sensor-fused navigation and manipulation from a planetary rover*. Proceedings SPIE Symposium on Sensor Fusion and Decentralized Control in Robotic Systems, Vol. 3523, Boston, MA, Nov 1998.
- Betge-Brezetz, S., Hebert, P., Chatila, R., and Devy, M. *Uncertain map making in natural environments*. Proceedings of the 1996 IEEE International Conference on Robotics and Automation, Minneapolis, Minnesota, April, 1996.
- Burschka, D. Eberst, C. and Robl, C. *Vision based model generation for indoor environments*.

Proceedings of the 1997 IEEE International Conference on Robotics and Automation, Albuquerque, New Mexico, April, 1997

Castellanos, J.A., Martinez, J.M., Neira, J., Tardos, J.D. *Simultaneous map building and localization for mobile robots: a multisensor fusion approach*. Proceedings of the 1998 IEEE International Conference on Robotics and Automation. Volume: 2, 1998 Page(s): 1244 -1249

Chirikjian, G.S.; Burdick, J.W. *The kinematics of hyper-redundant robot locomotion*. IEEE Transactions on Robotics and Automation. Volume: 11 Issue: 6 , Dec. 1995 Page(s): 781 -793

Choi, J-D., Kang, S., Munsang, K., Lee, C.W., and Song, J-B. *Two-arm cooperative assembly using force-guided control with adaptive accommodation*. Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, vol. 2, pp. 1253-1258. 1999.

Choset, H., Nagatani, K. *Topological simultaneous localization and mapping (SLAM): toward exact localization without explicit localization*. IEEE Transactions on Robotics and Automation, , Volume: 17 Issue: 2, April 2001 Page(s): 125 -137

Chu, G-W., Yu, W.P., and Chung, M.J. *A simple method of finding a visible region for the optimal camera position*. Proceedings of the 1997 IEEE ICAR. Monterey, CA. July 1997.

Clark, J.J. and A.L. Yuille. *Data fusion for sensory information processing systems*. Kluwer international series in engineering and computer science. Kluwer Academic Publishers, 1990.

Connolly, C.I. *The determination of the next best views*. Proceedings of the IEEE International Conference on Robotics and Automation, pp. 432-435, 1985.

Cowan, G.K. and Kovesi, P.D. *Automatic sensor placement from vision task requirements*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 10, no. 3, pp. 407-416, May 1988.

Darby, A. P. and Pellegrino, S. *Modeling and control of a flexible structure incorporating inertial stick-slip actuators*. Journal of Guidance, Control, and Dynamics, 22, 36-43, 1999.

Donald, B., J. Jennings, and D. Rus. *Information invariants for cooperating autonomous mobile*

- robots*. International Journal of Robotics Research, 16(5) 673-702, 1997.
- Dotson RD. *Spacecraft deployable structure testing*. Space Systems Design and Development Testing (AGARD-CP-561). AGARD. 1995, pp.6/1-12. Neuilly sur Seine, France.
- Dubowsky, S. and Vance, E. E. *Planning Mobile Manipulator Motions Considering Vehicle Dynamic Stability Constraints*. IEEE Int. conf. on Robotics and Automation, Scottsdale, AZ, May 1989, pp. 1271 -1276 vol.3
- Ebert-Uphoff, I.; Chirikjian, G.S. *Inverse kinematics of discretely actuated hyper-redundant manipulators using workspace densities*. Proceedings of the IEEE International Conference on Robotics and Automation, 1996, Volume: 1, Page(s): 139 -145 vol.1.
- Erdmann, M.A. and Mason, M.T. *Exploration of sensor-less manipulation*. IEEE Journal of Robotics and Automation, Vol. 4, pp 369-379, August 1988.
- Gantes, C., Connor, J., and Logcher, R.D. *Structural analysis and design of deployable structures*. Computers and Structures. Vol. 32, No. ¾, pp. 661-669, 1989.
- Gautier, M. and Khalil, W. *Exciting trajectories for the identification of base inertial parameters of robots*. International Journal of Robotics Research, Vol. II, No. 4, August 1992.
- Gelb, A. *Applied optimal estimation*. 1974, MIT press, Cambridge, MA, U.S.A.
- Gerkey, B. and M. J. Mataric. *Principled Communication for Dynamic Multi-Robot Task Allocation*. To Proceedings of the International Symposium on Experimental Robotics 2000, Waikiki, Hawaii, December 10-13, 2000.
- Gilbertson, R. *Muscle wires*. San Alselmo, CA 1994.
- Giraud, A. and Sidobre, D. *A heuristic motion planner using contact for assembly*. Proceedings of the 1992 IEEE International Conference on Robotics and Automation, Vol. 3, pp. 2165-2170. Nice, France, May 1992.
- Goldberg, D. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA 1989
- Goldberg, K. *Orienting polygonal parts without sensors*. Algorithmica, 1992, Special Robotics Issue.
- Hager, G.D., Kriegman, D., Teh, E., Rasmussen, C. *Image-based prediction of landmark features for mobile robot navigation*. Proceedings of the 1997 IEEE International Conference on Robotics and Automation. Volume: 2, Page(s): 1040 -1046
- Halfmann, C., Nelles, O. and Holzmann, H. *Modeling and identification of the vehicle*

- suspension characteristics using local linear model trees*. Proc. of 1999 IEEE Int. Conf. on Control Applications, Hawaii, USA, August 22-27, 1999.
- Harris, C.J.; Wu, Z.Q.; Gan, Q. *Neurofuzzy state estimators and their applications*. 1999 IEE Colloquium on Advances in Control Technology. 25 May 1999, London, UK.
- Hirai, S., and Iwata, K. *Recognition of Contact State Based on Geometric Model*. Proceedings of the 1992 IEEE International Conference on Robotics and Automation. Vol. 2, pp. 1507-1512. Nice, France, May 1992.
- Hootsman, N.A.M., Dubowsky, S. and Mo, P.Z. *The experimental performance of a mobile manipulator control algorithm*. Proc. of the 1992 IEEE ICRA, Nice, France, May 10-15, 1992. Vol. 3, pp. 1948-1954.
- Huang, M.Z.; Shou-Hung Ling *Kinematics of a class of hybrid robotic mechanisms with parallel and series modules* Proceedings of the 1994 IEEE International Conference on Robotics and Automation, Page(s): 2180 -2185 vol.3
- Hughes, P.C. *Trussarm – a variable geometry truss manipulator*. Journal of intelligent materials, systems and structures, vol. 2, pp. 148-160, April '91.
- Huntsberger, T.L. *Autonomous multi-rover system for complex planetary retrieval operations*. Proceedings. SPIE Symposium on Sensor Fusion and Decentralized Control in Autonomous Robotic Agents, Pittsburgh, PA, Oct 1997, pp. 220-229.
- Huntsberger, T., H. Aghazarian, E. Baumgartner, and P. S. Schenker. *Behavior-based control systems for planetary autonomous robot outposts*. Proceedings of IEEE Aerospace 2000, Big Sky, Montana, Mar 2000.
- Huntsberger, T.L., G. Rodriguez, and P. S. Schenker. *Robotics: challenges for robotic and human Mars exploration*. Proceedings of ROBOTICS2000, Albuquerque, NM, Mar 2000.
- Hutchinson, S.A. and Kak, A. *Planning sensing strategies in a robot work cell with multi-sensor capabilities*. IEEE Transactions on Robotics and Automation, vol. 5, no. 6, December, 1989.
- Jennings, C., Murray, D., and Little, J. *Cooperative robot localization with vision-based mapping*. Proceedings of the 1999 IEEE International Conference on Robotics and Automation, Detroit, Michigan, May 1999.
- Kang, S.C., Hwang, Y.K., Kim, M.S., Lee, C.W., and Lee, K. *A compliant motion control for*

- insertion of complex shaped objects using contact.* Proceedings of the 1997 IEEE International Conference on Robotics and Automation, Albuquerque, New Mexico. April 1997
- Kang, S., Kim, M., Lee, W.L., and Lee, K-I. *A target approachable force-guided control for complex assembly.* Proceedings of the 1998 IEEE International Conference on Robotics and Automation. Leuven, Belgium. May 1998.
- Kececi, F., Tonko, M., Nagel, H.H., and Gengenbach, V. *Improving visually servoed disassembly operations by automatic camera placement.* Proceedings of the 1998 IEEE International Conference on Robotics and Automation, Leuven Belgium. May 1998.
- Kelly, A. *Introduction to Mobile Robots.* Lecture notes, Robotics Institute, Carnegie-Mellon University, 1999.
- Khatib, O., Yokoi, K., Chang, K., Ruspini, D., Holmberg, R., Casal, A., Baader, A. *Force strategies for cooperative tasks in multiple mobile manipulation systems.* International Symposium of Robotics Research, Munich, October, 1995.
- Khatib, O., Yokoi, K., Brock, O., Chang, K., and Casal, A. *Robots in Human Environments: Basic Autonomous Capabilities.* The International Journal of Robotics Research, Vol. 18, No. 7, pp. 684-696 July 1999.
- Kitagaki, K., Ogasawara, T. and Suehiro, T. *Methods to detect contact state by force sensing in an edge mating task.* Proceedings of the 1993 IEEE International Conference on Robotics and Automation, pp. 701-706, vol. 2. Atlanta, GA., 1993.
- Kittipongpattana, T. and Laowattana, D. *Robotic Insertion of Dual Pegs based on Force Feedback Signals.* IEEE Proceedings of the Asia Pacific Conference on Circuits and Systems. Pp. 527-530. November, 1998.
- Kritofek, G. Bachelors thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, June 2002.
- Kruse, E., Gutsche, R., and Wahl, F.M. *Efficient, iterative, sensor based 3-D map building using rating functions in configuration space.* Proceedings of the 1996 IEEE International Conference on Robotics and Automation, Minneapolis, Minnesota, April, 1996.
- Kuipers, B., Byun, Y. *A robot exploration and mapping strategy based on semantic hierarchy of spatial representations.* Journal of Robotics and Autonomous Systems **8**, 47-63, 1991
- Kwon, S; Youngil Youm *General algorithm for automatic generation of the workspace for n-*

- link redundant manipulators*. Proceedings of the International Conference Advanced Robotics, 1991. 'Robots in Unstructured Environments', Page(s): 1722 -1725 vol.2
- Lara, B, Althoefer, K., and Seneviratne, L.D. *Automated robot-based screw insertion system*. Proceedings Conference of the IEEE of the 24th Annual Industrial Electronics Society, 1998. IECON '98. Volume: 4 , Page(s): 2440 -2445 vol.4
- Laugier, C. and Triggs, B. *Automatic camera placement for robot vision tasks*. IEEE International conference on robotics and automation, Nagoya, Japan, 1995.
- Lee, S. and Asada, H. *A perturbation/correlation method for force guided robot assembly*. IEEE Transactions on Robotics and Automation, Vol. 15, No. 4, pp. 764-773. August, 1999.
- Lees, D.S. and Chirikjian, G.S. *A combinatorial approach to trajectory planning for binary manipulators*. Proceedings of the IEEE International Conference on Robotics and Automation, Minneapolis, Minnesota, April 1996.
- Leonard, J.J., Durrant-Whyte, H.F. *Simultaneous map building and localization for an autonomous mobile robot*. IEEE/RSJ 1991 International Workshop on Intelligent Robots and Systems. Page(s): 1442 -1447 vol.3. Nov 3-5, 1991, Osaka, Japan.
- Lichter, M.D., Sujan, V.A., Dubowsky, S. *Experimental Demonstration of a New Design Paradigm in Space Robotics*. Proceedings of the Seventh International Symposium on Experimental Robotics, ISER 00. Dec 10-13, 2000, Honolulu, Hawaii.
- Lichter, M.D., Sujan, V.A. and Dubowsky, S. *Computational Evaluation of Binary Robots for Exploratory Space Systems*. Proceedings of the 2002 IEEE International Conference on Robotics and Automation, May 11-15, 2002, Washington, D.C., U.S.A.
- Ljung, L and Glad, T. *Testing global identifiability for arbitrary model parameterizations*. Identification and System Parameter Estimation, Vols 1,2. Pergamon Press, New York 1991, pp. 10877-1082.
- Lumelsky, V., Mukhopadhyay, S. and Sun, K. *Sensor-based terrain acquisition: the "sightseer" strategy*. Proceedings of the 28th Conference on Decision and Control. Tampa, Florida. December 1989.
- Luo, R.C. and M.G. Kay. *Multisensor integration and fusion in intelligent systems*. IEEE Transactions on Systems, Man, and Cybernetics, Vol. 19, No. 5, September, 1989.
- Madden J D, Cush R A, Kanigan T S, et al. *Fast-contracting Polypyrrole Actuators*. *Synthetic*

Metals, 113: 185-193, 2000

- Majjad, R. *Estimation of suspension parameters*. Proceedings the 1997 IEEE International Conference on of Control Applications. 5-7 Oct. 1997, Hartford, CT, USA.
- Marapane, S.B., Trivedi, M.M., Lassiter, N., Holder, M.B. *Motion control of cooperative robotic teams through visual observation and fuzzy logic control*. Proceedings of International Conference on Robotics and Automation. pp. 1738-1743 vol. 2, Minnesota, MN, 1996.
- Mataric, M.J. *Coordination and Learning in Multi-Robot Systems*, IEEE Intelligent Systems, Mar/Apr 1998, 6-8
- Mayeda, H., Yoshida, K. and Osuka, K. *Base parameters of manipulator dynamics*. Proc. of IEEE Int. Conf. on Robotics and Automation. April 24-29, 1988. Philadelphia, U.S.A.
- McCarragher, B.J. and Asada, H. *A Discrete Event Approach to the Control of Robotic Assembly Tasks*. Proceedings of the 1993 IEEE International Conference on Robotics and Automation. Atlanta, GA, 1993.
- Meguro A, Mitsugi J, Ando K. *A modular cable-mesh deployable structure for large scale satellite communication antennas*. Transactions of the Institute of Electronics, Information & Communication Engineers B-II, vol.J76B-II, no.5, May 1993, pp.476-84. Japan.
- Nelles, O. and Isermann, R. *Basis function networks for interpolation of local linear models*. Proc. of IEEE Conference on Decision and Control. Kobe, Japan, 11-13 December 1996.
- Nelson, B.J., Morrow, J.D. and Khosla, P. K. *Fast stable contact transitions with stiff manipulator using force and vision feedback*. IEEE Proceedings of the International Conference on Intelligent Robots and Systems—Human Robot Interactions and Cooperative Robots. Pittsburgh, PA. Vol. 2, pp. 90-95. 1995.
- Nikravesh, P.E. *Computer-aided analysis of mechanical systems*. Prentice Hall, Englewood Cliffs, NJ, 1988
- Olsen, H.B. and Beckey, G.A. *Identification of parameters in models of robots with rotary joints*. Proc. of IEEE Int. Conf. on Robotics and Automation, St. Louis, March 25-28, 1985.
- Oropeza, G. *The Design of Lightweight Deployable Structures for Space Applications*. Thesis for the Bachelors of Science in Mechanical Engineering, Massachusetts Institute of Technology, May 1999.

- O'Rourke, J. *Computational geometry in C*. Cambridge University Press, 1993.
- Osborn, J.F. *Applications of Robotics in Hazardous Waste Management* Proceedings of the SME 1989 World Conference on Robotics Research, Gaithersburg, MD.
- Park J., Bolan Jiang, Neumann, U. *Vision-based pose computation: robust and accurate augmented reality tracking*. Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality, 1999. (IWAR '99). Page(s): 3 -12
- Parker. L.E. *The Effect of Action Recognition and Robot Awareness in Cooperative Robotic Teams*. Proceedings of IEEE International Conference on Robotics and Automation, Nagoya, Japan, 1995. Vol. 1, pp. 212-219.
- Pellegrino, S. and Guest, S. D. *Deployable Structures: Theory and Applications*. Proceedings of IUTAM-IASS Symposium held in Cambridge, September 1998, Kluwer Academic Publishers, Dordrecht
- Pelrine R, Kornbluh R, Pei Q, et al. High-speed Electrically Actuated Elastomers with Over 100% Strain. *Science*, Vol. 287, No. 5454, 836-839, 2000.
- Pfeffer, L.E. and Cannon, R.H. Experiments with a dual-armed cooperative, flexible-drive train robot system. Proceedings of the 1993 IEEE International Conference on Robotics and Automation. Atlanta, GA, 1993. pp. 601-608, vol. 3.
- Rekleitis, I., Dudek, G., and Milios, E. *Multi-robot collaboration for robust exploration*. Proceedings of the 2000 IEEE International Conference on Robotics and Automation, San Francisco, CA. April, 2000.
- Repo, T. and Roning, J. *Modeling structured environments by a single moving camera*. Proceedings. Second International Conference on 3-D Digital Imaging and Modeling, 1999. On page(s): 340 - 347, 4-8 Oct. 1999
- Schenker, P.S., E. T. Baumgartner, R. A. Lindemann, H. Aghazarian, A. J. Ganino, G. S. Hickey, D. Q. Zhu, L. H. Matthies, B. H. Hoffman, T. L. Huntsberger, *New planetary rovers for long-range Mars science and sample return*. Proc. SPIE Symposium on Intelligent Robots and Computer Vision XVII: Algorithms, Techniques, and Active Vision, Vol. 3522, Boston, MA, Nov 1998.
- Schmidt, T. and Muller, P.C. *A parameter estimation method for multibody systems with constraints*. Advanced Multibody System Dynamics: Simulation and Software Tools. Kluwer Academic Publishers, 1993, pp 427-432.

- Serban, R. and Freeman, J.S. *Identification and identifiability of unknown parameters in multibody dynamic systems*. Multibody Systems DynaMics. Vol. 5, No. 4, pp. 335-350, May 2001. Kluwer Academic Publishers.
- Shaffer, G.; Stentz, A. *A robotic system for underground coal mining*. Proc. of 1992 IEEE ICRA. Page: 633 -638 vol.1
- Shannon, C. E. *A mathematical theory of communication*. The Bell System Technical Journal, vol. 27, pp. 379-423 and 623-656, July and October, 1948.
- Shimokura, K. And Muto, S. *A high speed method of detecting contact-state transitions and its implementation in a task-coordinate manipulation system*. Proceedings of the 1996 IEEE International Conference on Robotics and Automation, Minneapolis, Minnesota. April, 1996.
- Simhon, S., Dudek, G. *Selecting targets for local reference frames*. Proceedings of the 1998 IEEE International Conference on Robotics and Automation. Volume: 4, Page(s): 2840 - 2845 vol.4
- Smith, R.C. and Cheeseman, P. *On the representation and estimation of spatial uncertainty*. International Journal of Robotics Research. 5(4): 56-68
- Smith, S. W. *The Scientist and Engineer's Guide to Digital Signal Processing*. 2nd Edition, California Technical Publishing, San Diego, CA, 1999.
- Soderstorm, T and Stocia, P. *System identification*. Prentice Hall International. London, 1989.
- Stroupe, A. W., Martin, M. and Balch, T. *Merging gaussian distributions for object localization in multi-robot systems*. Proceedings of the Seventh International Symposium on Experimental Robotics, ISER '00. Hawaii, December 10 - 13, 2000.
- Sujan, V. A., Lichter, M. D. and Dubowsky, S. *Lightweight Hyper-redundant Binary Elements for Planetary Exploration Robots*. Proceedings of the IEEE/ASME Conference on Advanced Intelligent Mechatronics (AIM '01), July, 2001. Como, Italy.
- Sujan, V.A. and Dubowsky, S. *Visually Built Task Models for Robot Teams in Unstructured Environments*. Proceedings of the 2002 IEEE International Conference on Robotics and Automation, May 11-15, 2002, Washington, D.C., U.S.A.
- Sujan, V.A. and Dubowsky, S. *Metric based dynamic parameter identification for mobile field manipulator systems*. Proceedings of the ASME 27th Biennial Mechanisms and Robotics Conference, September 29 – October 2, 2002, Montreal, Canada.

- Syromiatnikov, V.S. *Manipulator system for module re docking on the Mir Orbital Complex*. Proceedings of the 1992 IEEE International Conference on Robotics and Automation, 1992 Page(s): 913 -918 vol.1
- Takanishi, A.; Tochizawa, M.; Karaki, H.; Kato, I. *Dynamic Biped Walking Stabilized With Optimal Trunk And Waist Motion*. Proceedings of IEEE/RSJ International Workshop on Intelligent Robots and Systems '89. The Autonomous Mobile Robots and Its Applications. IROS '89. Page(s): 187 –192
- Tarabanis, K.A., Allen, P.K. and Tsai, R.Y. *A survey of sensor planning in computer vision*. IEEE Transactions on Robotics and Automation, Vol. 11 no. 1, Pp. 86-104. February, 1995.
- Thrun, S., Burgard, W., and Fox, D. *A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping*. Proceedings of the 2000 IEEE International Conference on Robotics and Automation. San Francisco, CA, April, 2000.
- Thrun, S, Dieter Fox, and Wolfram Burgard. *A probabilistic approach to concurrent mapping and localization for mobile robots*. *Machine Learning*, 31:29–53, 1998.
- Tornatis, N., Nourbakhsh, I., Siegwar, R. *Simultaneous localization and map building: a global topological model with local metric maps*. Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Volume: 1, 2001 Page(s): 421 -426
- Veloso, M., Stone, P. *Task decomposition and dynamic role assignment for real-time strategic teamwork*. Intelligent agents V - Proceedings of the Fifth International Workshop on Agent Theories, Architectures and Languages, Springer -Verlag, Heidelberg, 1999.
- Veloso, M., Stone, P., Bowling, M. *Anticipation: A key for collaboration in a team of agents*. Proceedings of SPIE Sensor Fusion and Decentralized Control in Robotic Systems II, volume 3839, Boston, September 1999.
- Victorino, A.C., Rives, P., Borrelly, J.-J. *Localization and map building using a sensor-based control strategy*. Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems. (IROS 2000). Volume: 2, Page(s): 937 -942 vol.2
- Walker, I.D.; Cavallaro, J.R. *The use of fault trees for the design of robots for hazardous environments*. Proceedings of the Annual International Symposium on Product Quality and Integrity, Reliability and Maintainability Symposium, 1996. Pages(s): 229-235

- Whitney, D.E. *Historical Perspective and State of the Art in Robot Force Control*. International Journal of Robotics Research, Vol. 6, No. 1, Spring 1987
- Xiao J. And Liu, L. *Contact states: Representation and recognizability in the presence of uncertainties*. Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Victoria, B.C., Canada. October, 1998
- Yamauchi, B., Schultz, A., Adams, W. *Mobile robot exploration and map-building with continuous localization*. Proceedings of the IEEE International Conference on Robotics and Automation, 1998. Volume: 4 Page(s): 3715 -3720 vol.4
- Yeh, E., Kriegman, D.J. *Toward selecting and recognizing natural landmarks* Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems, Volume 1 Page(s): 47 -53
- Yeo H., Suh, I.H., Yi, B-J., and Oh, S-R. *A single closed-loop kinematic chain approach for hybrid control of two cooperating arms with a passive joint: An application to sawing task*. IEEE Transactions on Robotics and Automation, Vol. 15, No. 1, Pp. 141-151. February 1999.
- Zhang L. and Xiao, J. *Derivation of Contact States from Geometric Models of Objects*. IEEE International Symposium on Assembly and Task Planning, Aug, 1995

Appendix A

Cooperative Mobile Robot System Dynamic Model

A.1. Model Predictive Control

In model predictive control disturbance rejection is accomplished by estimating the equivalent disturbance of a system based on the dynamic model of the system and the sensed disturbances. This is also known as feed-forward control as seen from the block diagram in figure A-1. Disturbance measurements are fed into a dynamic system model to account for changes caused by these disturbances. The resultant dynamic disturbance commands are fed forward and added to the basic controller commands to give the system control command. The basic joint level controller considered here is a PID controller. Such a system is dependent on an accurate dynamic model and low noise disturbance sensors, to function effectively. Degradation in the accuracy of the models and the disturbance measurements result in corresponding degradation of the controller.

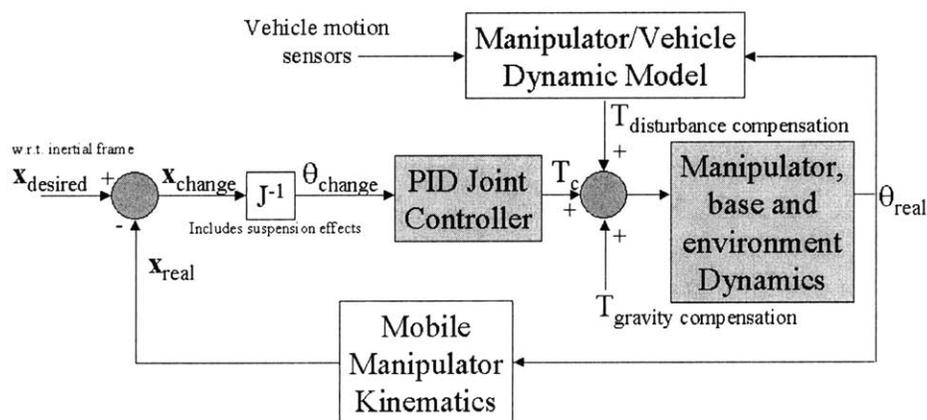


Figure A-1: Block diagram of linear feed-forward compensation for dynamic disturbance rejection

A.2. Development Of The Cooperative System Model

Figure A-2 shows two physically interacting cooperative robots working in an unstructured field environment (as described in the Chapter 1).

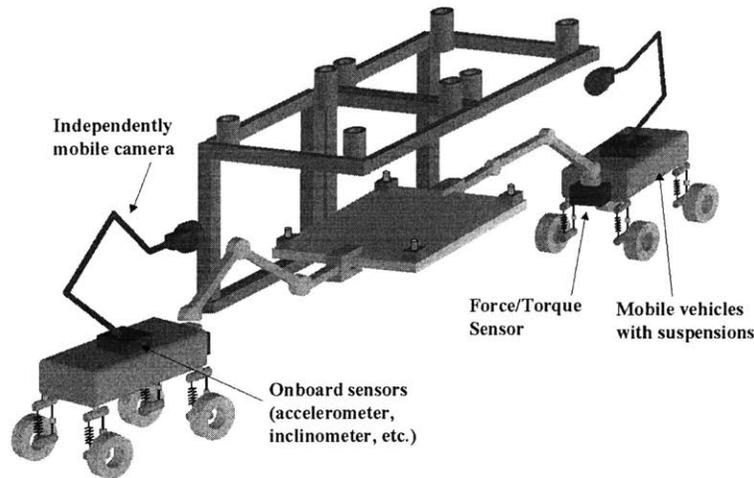
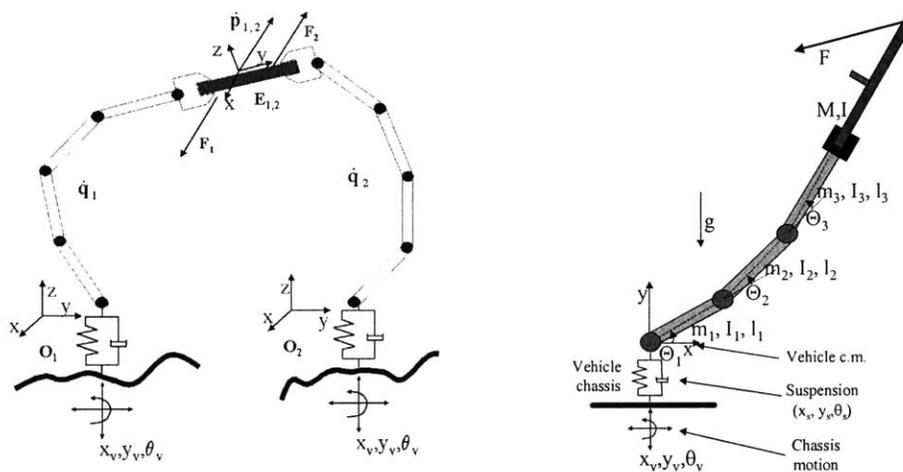


Figure A-2: Representative physical system

Using a Lagrangian formulation, the dynamic models of the systems and task are developed. These models account for robot base motion, compliance, and multi-robot interaction forces (see Figure A-3). This method can be readily extended to model the closed chain dynamics of multiple cooperating robots.



(a) Interacting mobile systems

(b) Individual robot system with environment interaction forces

Figure A-3: Cooperative robot modeling

Step 1: Reduction of suspension compliance system

The first step is to reduce the suspension system (Figure A-2) to a 6 DOF compliance module located at the vehicle base center of gravity (Figure A-3a). If the contributions to the suspension are known to occur only from the vehicle tires and a passive compliance element, then the combined 6 DOF stiffness at the center of gravity is given by:

$$\begin{aligned}
 K^{x,y,z} &= \sum_n \left(\frac{1}{K_{suspension}^{x,y,z}} + \frac{1}{K_{wheel}^{x,y,z}} \right)_n^{-1} \\
 K^{\theta_x} &= \sum_n \frac{K_n^z dz_n l_n \cdot \hat{z} + K_n^y dy_n dz_n}{\tan^{-1} \left(\frac{dy_n}{dz_n} \right)} \\
 K^{\theta_y} &= \sum_n \frac{K_n^z dz_n l_n \cdot \hat{z} + K_n^x dx_n dz_n}{\tan^{-1} \left(\frac{dx_n}{dz_n} \right)} \\
 K^{\theta_z} &= \sum_n \frac{K_n^y dy_n l_n \cdot \hat{y} + K_n^x dx_n dy_n}{\tan^{-1} \left(\frac{dx_n}{dy_n} \right)}
 \end{aligned} \tag{A-1}$$

Similar expressions may be derived for the 6 DOF damping terms.

Step 2: Robot model Lagrangian dynamics

In general :

$$Q_i - (\mathbf{J}^T \mathbf{F})_i = \sum_{j=1}^n H_{ij} \ddot{q}_j + \sum_{j=1}^n \sum_{k=1}^n h_{ijk} \dot{q}_j \dot{q}_k - G_i$$

where

Q_i = generalized force on joint i

\mathbf{F} = manipulator endpoint external force

$$\mathbf{H}_{n \times n} = \sum_{j=1}^n m_j \mathbf{J}_L^{(j)T} \mathbf{J}_L^{(j)} + \mathbf{J}_A^{(j)T} I_j \mathbf{J}_A^{(j)}$$

$$h_{ijk} = \frac{\partial H_{ij}}{\partial q_k} - \frac{1}{2} \frac{\partial H_{jk}}{\partial q_i}$$

$$G_i = \sum_{j=1}^n m_j \mathbf{g}^T \mathbf{J}_{Li}^{(j)}$$

$$\begin{aligned} \bar{\mathbf{v}}_{c_j} = \mathbf{J}_L^{(j)} \cdot \dot{\bar{\mathbf{q}}} &\Rightarrow \mathbf{J}_{Li}^{(j)} = \begin{cases} b_i & \text{for a prismatic joint} \\ b_i \times r_{i,c_j} & \text{for a revolute joint} \end{cases} \left\{ \begin{array}{l} b_i = 3 \times 1 \text{ unit vector} \\ \text{along joint axis } i \\ r_{i,c_j} = \text{vector of centroid} \\ \text{of link } j \text{ from } i^{\text{th}} \text{ frame} \end{array} \right. \\ \bar{\boldsymbol{\omega}}_{c_j} = \mathbf{J}_A^{(j)} \cdot \dot{\bar{\boldsymbol{\theta}}} &\Rightarrow \mathbf{J}_{Ai}^{(j)} = \begin{cases} 0 & \text{for a prismatic joint} \\ b_i & \text{for a revolute joint} \end{cases} \end{aligned} \quad (\text{A-2})$$

Treat $\bar{\mathbf{r}}_i$ and $\bar{\boldsymbol{\theta}}$ as manipulator joints.

$\mathbf{J}_A^{(i)}$ and $\mathbf{J}_L^{(i)}$ are defined from the manipulator Jacobian :

$$\mathbf{q} = [x_v \ y_v \ \theta_v \ \theta_s \ x_s \ y_s \ \theta_1 \ \theta_2 \ \theta_3]^T \quad (\text{generalized variables})$$

Perturbation about an equilibrium state while considering small motions and substituting into the non - linear dynamic equations of motion :

$$\begin{aligned} q_i &= q_i^e + \Delta q_i \\ (\dot{q}_i^e + \Delta \dot{q}_i)(\dot{q}_j^e + \Delta \dot{q}_j) &= \dot{q}_i^e \dot{q}_j^e + \dot{q}_i^e \Delta \dot{q}_j + \dot{q}_j^e \Delta \dot{q}_i + \Delta \dot{q}_i \Delta \dot{q}_j \\ \Rightarrow \mathbf{Q} - \mathbf{J}^T \mathbf{F} + \mathbf{G} - \widehat{\mathbf{H}}(\mathbf{q}, \dot{\mathbf{q}}^e) &= \mathbf{H} \Delta \dot{\mathbf{q}} + \widehat{\mathbf{h}}(\mathbf{q}, \dot{\mathbf{q}}^e) \Delta \dot{\mathbf{q}} \end{aligned}$$

Alternatively, consider Computed Torque Techniques, where :

$$Q_i - (\mathbf{J}^T \mathbf{F})_i = \sum_{j=1}^n H_{ij} \ddot{q}_j + \sum_{j=1}^n \sum_{k=1}^n h_{ijk} \dot{q}_j \dot{q}_k - G_i$$

$$Q_i^c \equiv Q_i - (\mathbf{J}^T \mathbf{F})_i - \sum_{j=1}^n \sum_{k=1}^n h_{ijk} \dot{q}_j \dot{q}_k + G_i$$

$$\Rightarrow \mathbf{Q}^c = \mathbf{H} \ddot{\mathbf{q}}$$

As long as the left hand terms can be computed, the equations have been linearized

Converting to state space form :

$$\bar{\dot{\mathbf{x}}} = \mathbf{A}\bar{\mathbf{x}} + \mathbf{B}(\bar{\mathbf{u}} - \hat{\mathbf{F}}[\ddot{x}_v \quad \ddot{y}_v \quad \ddot{\theta}_v]^T)$$

$$\bar{\mathbf{y}} = \mathbf{C}\bar{\mathbf{x}} + \mathbf{D}\bar{\mathbf{u}}$$

where

$$\bar{\mathbf{x}} = [\theta_s \ x_s \ y_s \ \theta_1 \ \theta_2 \ \theta_3 \ \dot{\theta}_s \ \dot{x}_s \ \dot{y}_s \ \dot{\theta}_1 \ \dot{\theta}_2 \ \dot{\theta}_3]^T$$

$$\bar{\mathbf{u}} = [Q_4^c \ Q_5^c \ Q_6^c \ Q_7^c \ Q_8^c \ Q_8^c]^T = [\tau_{\theta_s}^c \ F_{x_s}^c \ F_{y_s}^c \ \tau_6^c \ \tau_7^c \ \tau_8^c]^T$$

$$\mathbf{H} \equiv \begin{bmatrix} \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \hat{\mathbf{H}}_{6 \times 6} \end{bmatrix} & \mathbf{0}_{6 \times 6} \\ \hat{\mathbf{F}}_{6 \times 3} & \hat{\mathbf{H}}_{6 \times 6} \end{bmatrix} \quad \mathbf{h} \equiv \begin{bmatrix} \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \hat{\mathbf{h}}_{6 \times 6} \end{bmatrix} & \mathbf{0}_{6 \times 6} \\ \hat{\mathbf{f}}_{6 \times 3} & \hat{\mathbf{h}}_{6 \times 6} \end{bmatrix}$$

$\hat{\mathbf{H}}$ is the non - singular arm inertia matrix

$$\mathbf{A} = \begin{bmatrix} \begin{bmatrix} \mathbf{0}_{6 \times 6} & \mathbf{I}_{6 \times 6} \\ \mathbf{0}_{6 \times 6} & \mathbf{0}_{6 \times 6} \end{bmatrix} & \mathbf{0}_{6 \times 6} \\ \mathbf{0}_{6 \times 6} & \mathbf{0}_{6 \times 6} \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} \begin{bmatrix} \mathbf{0}_{6 \times 6} \\ \hat{\mathbf{H}}^{-1}_{6 \times 6} \end{bmatrix} & \mathbf{0}_{6 \times 6} \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} \mathbf{I}_{6 \times 6} & \mathbf{0}_{6 \times 6} \end{bmatrix} \quad \mathbf{D} = \begin{bmatrix} \mathbf{0}_{6 \times 6} \end{bmatrix} \quad (\text{A-3})$$

Step 3: Stability of local linear controller for position control

Robot arm dynamics :

$$\begin{bmatrix} \tau_4 \\ \tau_5 \\ \tau_6 \end{bmatrix} = \hat{\mathbf{H}} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} + \hat{\mathbf{h}} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} + \hat{\mathbf{F}} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta}_0 \end{bmatrix} + \hat{\mathbf{f}} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta}_0 \end{bmatrix} + \begin{bmatrix} G_4 \\ G_5 \\ G_6 \end{bmatrix}$$

$$\hat{\boldsymbol{\tau}} = \hat{\mathbf{H}}\ddot{\mathbf{q}} + \hat{\mathbf{h}}\dot{\mathbf{q}} + \hat{\mathbf{F}}\ddot{\mathbf{d}} + \hat{\mathbf{f}}\dot{\mathbf{d}} + \hat{\mathbf{G}} \quad (\text{A-4})$$

with PD control + gravity compensation + feed forward terms

$$\hat{\boldsymbol{\tau}} \equiv \mathbf{K}_p(\hat{\mathbf{q}}_r - \hat{\mathbf{q}}) + \mathbf{K}_D(\dot{\hat{\mathbf{q}}}_r - \dot{\hat{\mathbf{q}}}) + \hat{\mathbf{F}}\ddot{\mathbf{d}} + \hat{\mathbf{f}}\dot{\mathbf{d}} + \hat{\mathbf{G}} = -\mathbf{K}_p\tilde{\mathbf{q}} - \mathbf{K}_D\dot{\tilde{\mathbf{q}}} + \hat{\mathbf{F}}\ddot{\mathbf{d}} + \hat{\mathbf{f}}\dot{\mathbf{d}} + \hat{\mathbf{G}}$$

Consider the Lyapunov function candidate :

$$V(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} \tilde{\mathbf{q}}^T \mathbf{K}_p \tilde{\mathbf{q}} + \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{H} \dot{\mathbf{q}}$$

Since \mathbf{K}_p and \mathbf{H} are symmetric + ve definite $\Rightarrow V > 0$ (except when $\mathbf{q} = \mathbf{q}_d$)

$$\begin{aligned} \dot{V} &= \tilde{\mathbf{q}}^T \mathbf{K}_p \dot{\tilde{\mathbf{q}}} + \dot{\mathbf{q}}^T \mathbf{H} \dot{\mathbf{q}} + \frac{1}{2} \dot{\mathbf{q}}^T \dot{\mathbf{H}} \dot{\mathbf{q}} \\ &= \tilde{\mathbf{q}}^T \mathbf{K}_p \dot{\tilde{\mathbf{q}}} + \dot{\mathbf{q}}^T (\boldsymbol{\tau} - \mathbf{h}\dot{\mathbf{q}} - \hat{\mathbf{F}}\ddot{\mathbf{d}} - \hat{\mathbf{f}}\dot{\mathbf{d}} - \mathbf{G}) + \frac{1}{2} \dot{\mathbf{q}}^T \dot{\mathbf{H}} \dot{\mathbf{q}} = \tilde{\mathbf{q}}^T \mathbf{K}_p \dot{\tilde{\mathbf{q}}} - \dot{\mathbf{q}}^T (\mathbf{K}_d \dot{\tilde{\mathbf{q}}} + \mathbf{h}\dot{\mathbf{q}}) - \dot{\mathbf{q}}^T \mathbf{K}_p \tilde{\mathbf{q}} + \frac{1}{2} \dot{\mathbf{q}}^T \dot{\mathbf{H}} \dot{\mathbf{q}} \\ &= -\dot{\mathbf{q}}^T (\mathbf{K}_d + \mathbf{h}) \dot{\mathbf{q}} + \frac{1}{2} \dot{\mathbf{q}}^T \dot{\mathbf{H}} \dot{\mathbf{q}} = -\dot{\mathbf{q}}^T \mathbf{K}_d \dot{\mathbf{q}} + \frac{1}{2} \dot{\mathbf{q}}^T (\dot{\mathbf{H}} - 2\mathbf{h}) \dot{\mathbf{q}} = -\dot{\mathbf{q}}^T \mathbf{K}_d \dot{\mathbf{q}} \leq 0 \end{aligned}$$

If $\dot{V} = 0 \Rightarrow \dot{\mathbf{q}} = 0$

$$\therefore \ddot{\mathbf{q}} = \mathbf{H}^{-1} (\boldsymbol{\tau} - \mathbf{h}\dot{\mathbf{q}} - \hat{\mathbf{F}}\ddot{\mathbf{d}} - \hat{\mathbf{f}}\dot{\mathbf{d}} - \mathbf{G}) = \mathbf{H}^{-1} (-\mathbf{K}_p \tilde{\mathbf{q}} - \mathbf{K}_d \dot{\tilde{\mathbf{q}}} - \mathbf{h}\dot{\mathbf{q}}) = -\mathbf{H}^{-1} \mathbf{K}_p \tilde{\mathbf{q}}$$

$\therefore \ddot{\mathbf{q}} \neq 0$, if $\mathbf{q} \neq \mathbf{q}_d$

Therefore, Lyapunov asymptotic stability applies

Step 4: Classical Controllability and Observability

$$\begin{aligned} \mathbf{T} &= [\mathbf{B} \mid \mathbf{A}\mathbf{B} \mid \mathbf{A}^2\mathbf{B} \mid \dots \mid \mathbf{A}^{n-1}\mathbf{B}] \\ \Theta &= [\mathbf{C}^T \mid \mathbf{A}^T\mathbf{C}^T \mid (\mathbf{A}^T)^2\mathbf{C}^T \mid \dots \mid (\mathbf{A}^T)^{n-1}\mathbf{C}^T] \end{aligned}$$

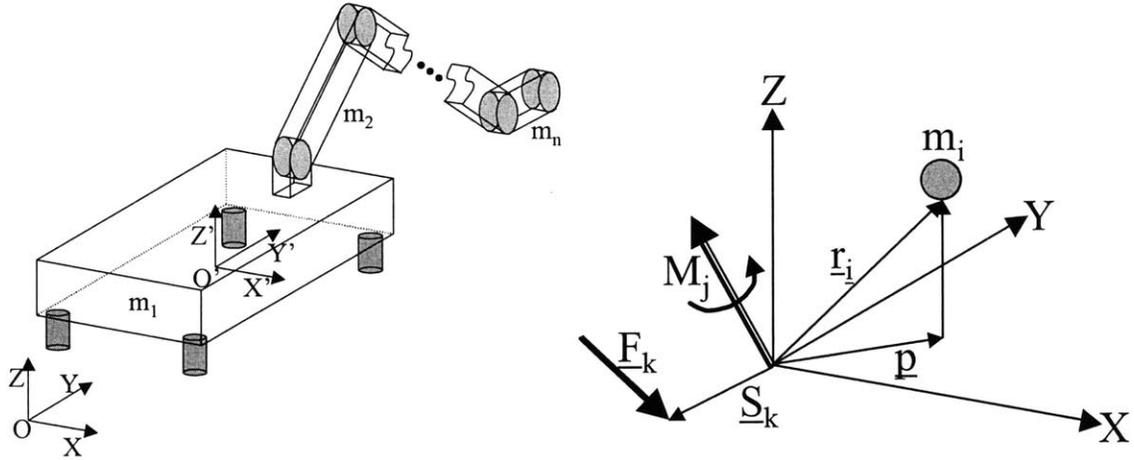
plugging in the previous expressions for A, B, C and n = 6 states and simplifying, it can be seen that the rank of T and Θ is 6.

The system is both Controllable and Observable

(a general solution, but proved for the manipulator in question)

A.3. Dynamic Tip-Over Stability

Once a dynamic model of the robotic system(s) have been set up, the controller needs to maintain tip-over stability. This is achieved by limiting the motion of the dynamic zero-moment point to lie within the vehicle footprint (see Figure A-4) [Takanishi]. Using d'Lambert's principle, the forces/torques on each individual mass particle is evaluated. The X and Y components of the zero moment point are given by Equation A-5.



(a) General robot system

(b) Force and moments on an individual mass element

Figure A-4: Dynamic tip-over stability

$$\begin{aligned}
 \bar{T} &= \sum_j \bar{M}_j - \sum_i m_i (\bar{r}_i - \bar{p}) \times (\ddot{\bar{r}}_i + \bar{G}) + \sum_k (\bar{S}_k - \bar{p}) \times \bar{F}_k \\
 X_{zmp} &= \frac{\sum_{i=1}^n m_i (\ddot{z}_i + g_z) x_i - \sum_{i=1}^n m_i (\ddot{x}_i + g_x) z_i + \sum_j M_{y_j} + \sum_k (z_{S_k} F_{x_k} - x_{S_k} F_{z_k})}{\sum_{i=1}^n m_i (\ddot{z}_i + g_z) - \sum_k F_{z_k}} \\
 Y_{zmp} &= \frac{\sum_{i=1}^n m_i (\ddot{z}_i + g_z) y_i - \sum_{i=1}^n m_i (\ddot{y}_i + g_y) z_i + \sum_j M_{x_j} + \sum_k (y_{S_k} F_{z_k} - z_{S_k} F_{y_k})}{\sum_{i=1}^n m_i (\ddot{z}_i + g_z) - \sum_k F_{z_k}}
 \end{aligned} \tag{A-5}$$

m_i : mass of particle i

$\bar{r}_i = [x_i, y_i, z_i]$: position vector of particle i

$\bar{p} = [x_p, y_p, 0]$: position vector of p

$\bar{G} = [g_x, g_y, g_z]$: gravitational acceleration vector

$\bar{T} = [T_x, T_y, T_z]$: total torque acted on p

$\bar{M}_j = [M_{x_j}, M_{y_j}, M_{z_j}]$: external moment j

$\bar{F}_k = [F_{x_k}, F_{y_k}, F_{z_k}]$: external force k

$\bar{S}_k = [x_{S_k}, y_{S_k}, z_{S_k}]$: position vector where external force k is put

A.4. Cooperative Task Execution—Robotic Assembly

Using the above dynamic models and a model predictive control architecture, multiple robots can cooperatively execute an assembly task. Here a planar model of this problem is developed. Here robotic assembly is an insertion task. Researchers have developed several approaches to the robotic insertion problem including motion in direction of least resistance, perturbation methods, petri-nets and event based approaches, remote compliance center modeling contact state identification [Giraud, Hirai, Kang-1, Kang-2, Kitagaki, Kittipongpattana, Lee, McCarragher, Nelson, Shimokura, Whitney, Xiao, Zhang]. Here the insertion problem is addressed by identifying the contact point (based on measured forces/torques) and formulating a motion plan in the direction of least resistance. Figure A-5 show an example of the six possible environment interaction states of a rectangular object at a similarly shaped insertion site.

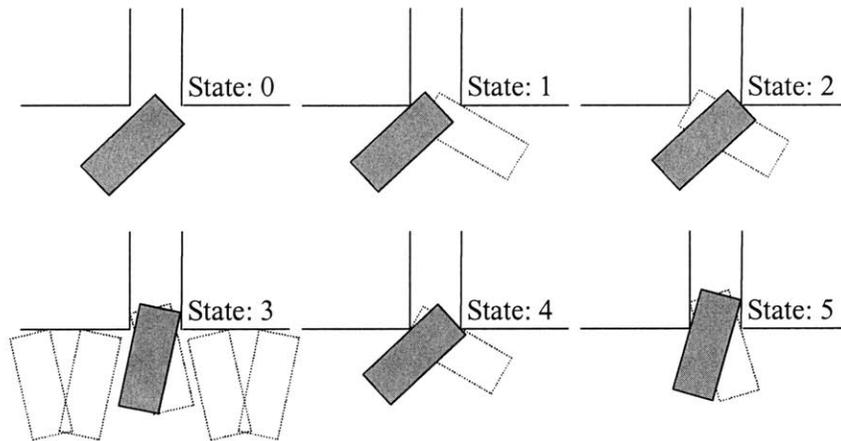
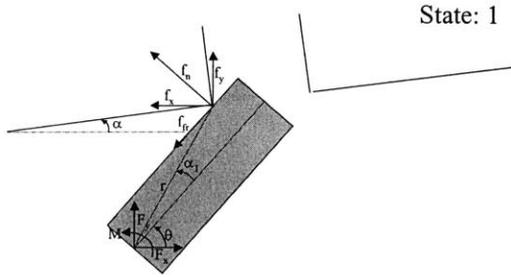


Figure A-5: Environment contact states

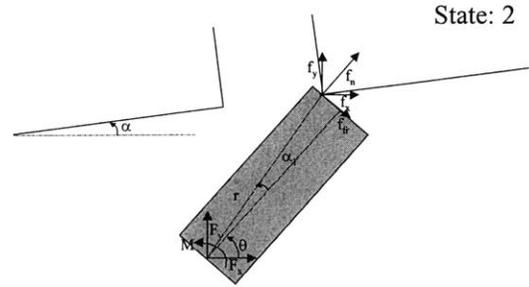
Figure A-6 show the relation between interaction forces, the contact point(s) and the measured forces/torques (F_x , F_y and M) for the different contact states seen in Figure A-5. The measured forces/torques are evaluated from force/torque sensor readings of all cooperating robots. Note that although multiple contact points cannot be uniquely identified, yet the motion plan is valid. Figure A-7 shows the error in location of the contact point as a function of sensor noise.



State: 1

$$\begin{aligned}\sum \mathbf{F} = 0 &= \mathbf{F}_x + \mathbf{F}_y + \mathbf{f}_n + \mathbf{f}_{fr} \\ \sum \mathbf{M} = 0 &= \mathbf{M} + \mathbf{r} \times \mathbf{f}_n + \mathbf{r} \times \mathbf{f}_{fr} \\ \frac{-f_x}{f_y} &= \tan \theta\end{aligned}$$

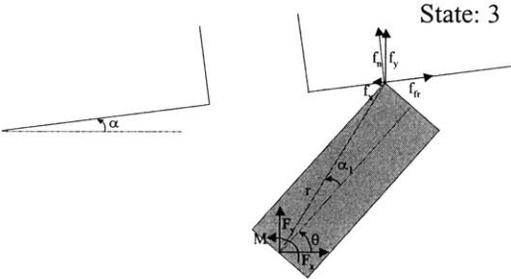
(a)



State: 2

$$\begin{aligned}\sum \mathbf{F} = 0 &= \mathbf{F}_x + \mathbf{F}_y + \mathbf{f}_n + \mathbf{f}_{fr} \\ \sum \mathbf{M} = 0 &= \mathbf{M} + \mathbf{r} \times \mathbf{f}_n + \mathbf{r} \times \mathbf{f}_{fr} \\ \frac{f_y}{f_x} &= \tan \theta\end{aligned}$$

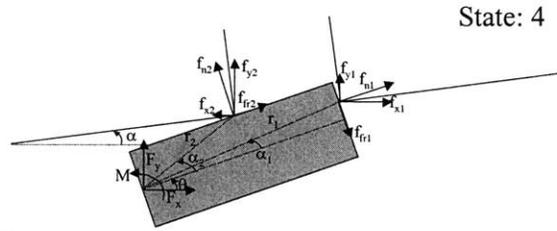
(b)



State: 3

$$\begin{aligned}\sum \mathbf{F} = 0 &= \mathbf{F}_x + \mathbf{F}_y + \mathbf{f}_n + \mathbf{f}_{fr} \\ \sum \mathbf{M} = 0 &= \mathbf{M} + \mathbf{r} \times \mathbf{f}_n + \mathbf{r} \times \mathbf{f}_{fr} \\ \frac{-f_x}{f_y} &= \tan \alpha\end{aligned}$$

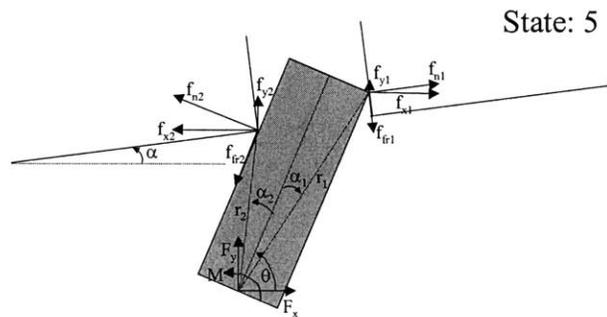
(c)



State: 4

$$\begin{aligned}\sum \mathbf{F} = 0 &= \mathbf{F}_x + \mathbf{F}_y + \mathbf{f}_{n1} + \mathbf{f}_{fr1} + \mathbf{f}_{n2} + \mathbf{f}_{fr2} \\ \sum \mathbf{M} = 0 &= \mathbf{M} + \mathbf{r}_1 \times \mathbf{f}_{n1} + \mathbf{r}_1 \times \mathbf{f}_{fr1} + \mathbf{r}_2 \times \mathbf{f}_{n2} + \mathbf{r}_2 \times \mathbf{f}_{fr2} \\ \frac{f_{y1}}{f_{x1}} &= \tan \theta & \frac{-f_{x2}}{f_{y2}} &= \tan \theta\end{aligned}$$

(d)



State: 5

$$\begin{aligned}\sum \mathbf{F} = 0 &= \mathbf{F}_x + \mathbf{F}_y + \mathbf{f}_{n1} + \mathbf{f}_{fr1} + \mathbf{f}_{n2} + \mathbf{f}_{fr2} \\ \sum \mathbf{M} = 0 &= \mathbf{M} + \mathbf{r}_1 \times \mathbf{f}_{n1} + \mathbf{r}_1 \times \mathbf{f}_{fr1} + \mathbf{r}_2 \times \mathbf{f}_{n2} + \mathbf{r}_2 \times \mathbf{f}_{fr2} \\ \frac{f_{y1}}{f_{x1}} &= \tan \alpha & \frac{-f_{x2}}{f_{y2}} &= \tan \theta\end{aligned}$$

(e)

Figure A-6: Relation of interacting forces, contact points and measured forces/torques

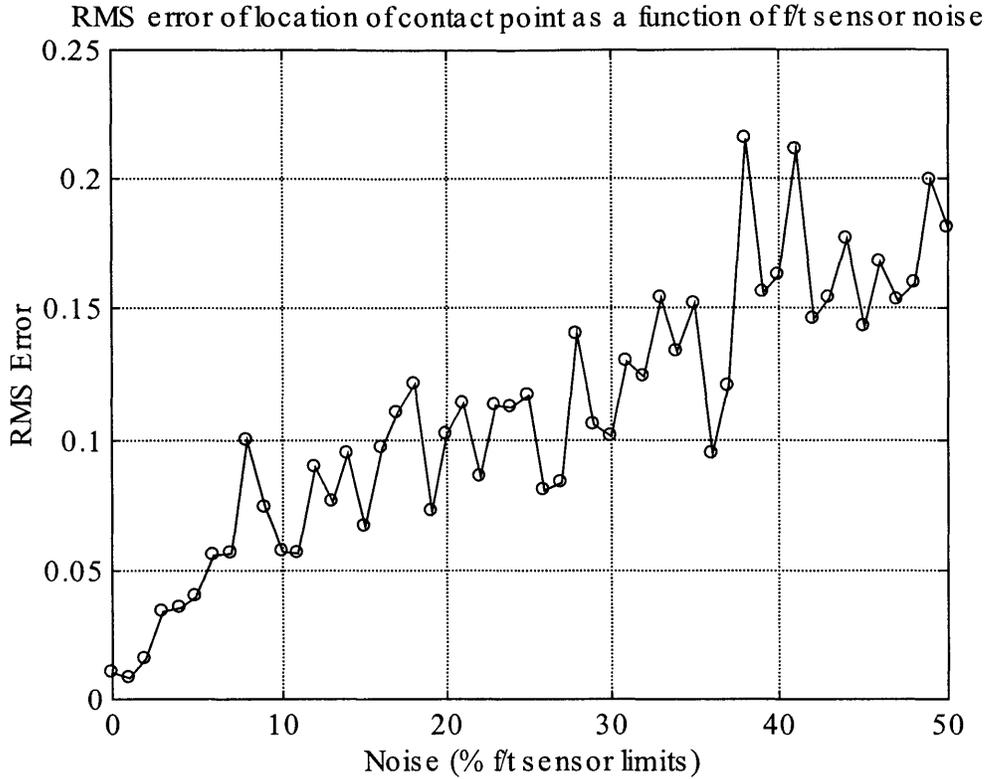


Figure A-7: Error of contact point location w.r.t. sensor noise

Two mobile robots (see Figure A-9) cooperate using a model predictive master-slave hybrid position-force control architecture (see Figure A-8), to insert a segment in a truss stage. Figures A-10, A-11 and A-12 show simulation results of the joint positions and forces felt by the cooperating robots during task execution.

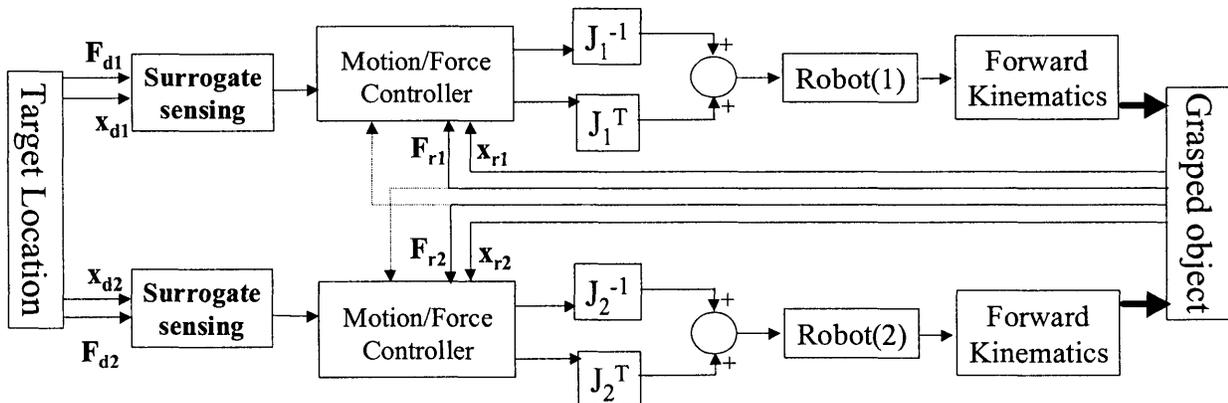


Figure A-8: Decentralized cooperative control architecture using surrogate sensing

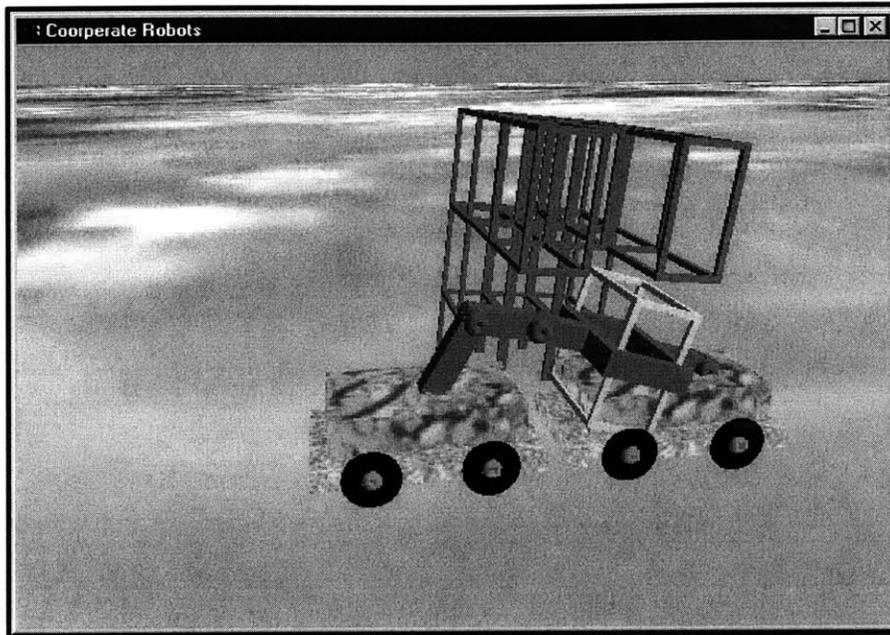


Figure A-9: Physically cooperating mobile robots

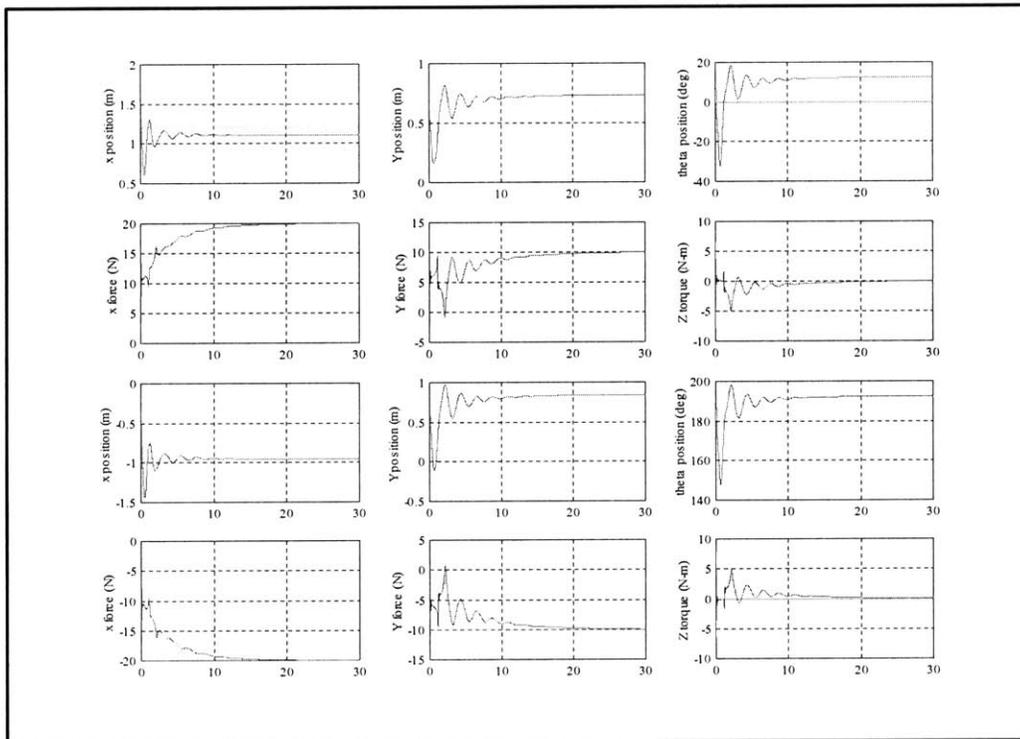


Figure A-10: Endpoint position and force of master robot (top) and slave robot (bottom)

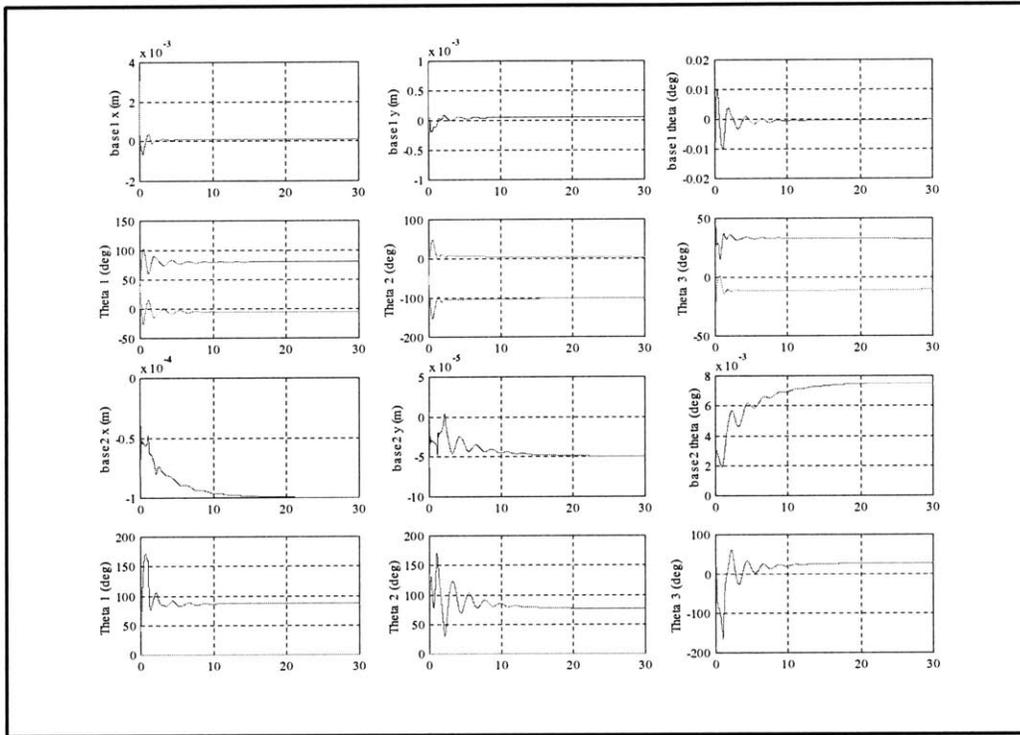


Figure A-11: Arm joint and base position of master robot (top) and slave robot (bottom)

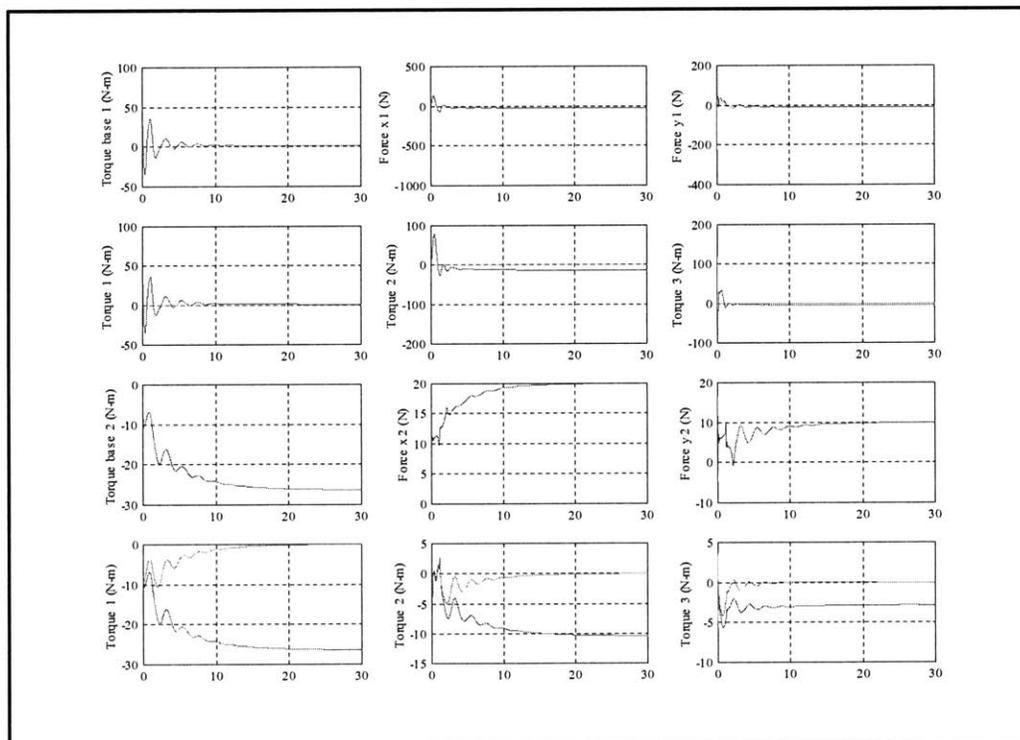


Figure A-12: Arm joint and base forces of master robot (top) and slave robot (bottom)

Appendix B

Newton-Euler Equations of Motion of Mobile Robot

B.1. Mobile robot dynamic model

This appendix presents the dynamic model of the mobile robot presented in figure 2-2.

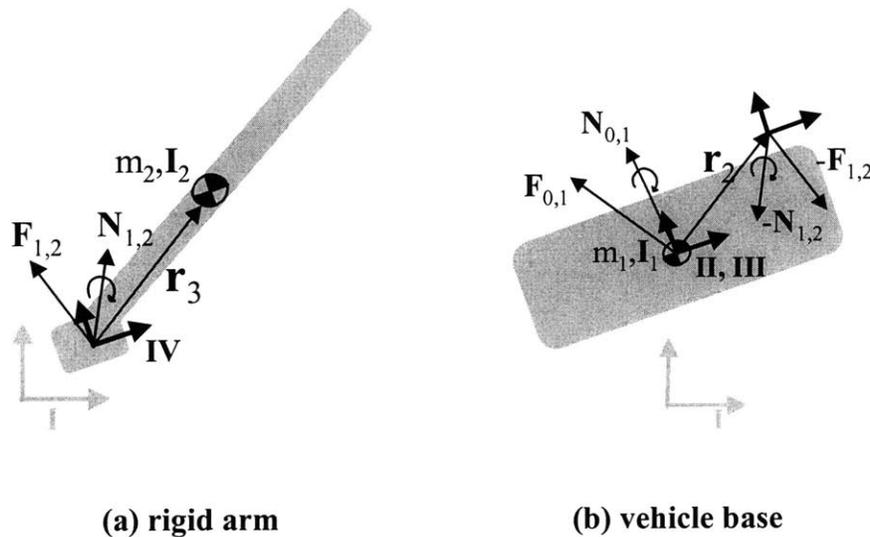


Figure B-1: Force/moment balance

The spatial interaction forces/moments of the rigid arm are:

$$\begin{aligned}
 (\mathbf{F}_{1,2})_{IV} + (m_2 \mathbf{g})_{IV} &= (m_2 \mathbf{a}_{cm_2})_{IV} \\
 (\mathbf{N}_{1,2})_{IV} - (\mathbf{r}_3 \times \mathbf{F}_{1,2})_{IV} &= \left(\frac{\partial (\mathbf{I}_2^{cm} \boldsymbol{\omega}_2^{cm})}{\partial t} + \boldsymbol{\omega}_2^{cm} \times (\mathbf{I}_2^{cm} \boldsymbol{\omega}_2^{cm}) \right)_{IV} \\
 (\mathbf{F}_{1,2})_{IV} \text{ and } (\mathbf{N}_{1,2})_{IV} &\Rightarrow \text{from arm base F/T sensor}
 \end{aligned} \tag{B-1}$$

where F_{12} and N_{12} are the reaction forces and moments, m_2 and I_2 are the arm mass and inertia tensors, \mathbf{a}_2 and $\boldsymbol{\omega}_2$ are the arm linear acceleration and angular velocity vectors. The spatial

interaction forces/moments of the rigid base are:

$$\begin{aligned}
 (\mathbf{F}_{0,1})_{\Pi} + (-\mathbf{F}_{1,2})_{\Pi} + (m_1 \mathbf{g})_{\Pi} &= (m_1 \mathbf{a}_{cm_1})_{\Pi} \\
 (\mathbf{F}_{0,1})_{\Pi} &= (m_1 \mathbf{a}_{cm_1})_{\Pi} - (-\mathbf{F}_{1,2})_{\Pi} - (m_1 \mathbf{g})_{\Pi} \\
 (\mathbf{N}_{0,1})_{\Pi} + (-\mathbf{N}_{1,2})_{\Pi} - (\mathbf{r}_2 \times \mathbf{F}_{1,2})_{\Pi} &= \left(\frac{\partial(\mathbf{I}_1^{cm} \boldsymbol{\omega}_1^{cm})}{\partial t} + \boldsymbol{\omega}_1^{cm} \times (\mathbf{I}_1^{cm} \boldsymbol{\omega}_1^{cm}) \right)_{\Pi} \\
 (\mathbf{N}_{0,1})_{\Pi} &= \left(\frac{\partial(\mathbf{I}_1^{cm} \boldsymbol{\omega}_1^{cm})}{\partial t} + \boldsymbol{\omega}_1^{cm} \times (\mathbf{I}_1^{cm} \boldsymbol{\omega}_1^{cm}) \right)_{\Pi} - (-\mathbf{N}_{1,2})_{\Pi} + (\mathbf{r}_2 \times \mathbf{F}_{1,2})_{\Pi}
 \end{aligned} \tag{B-2}$$

where F_{01} and N_{01} are the reaction forces and moments, m_1 and I_1 are the base mass and inertia tensors, \mathbf{a}_1 and $\boldsymbol{\omega}_1$ are the base linear acceleration and angular velocity vectors.

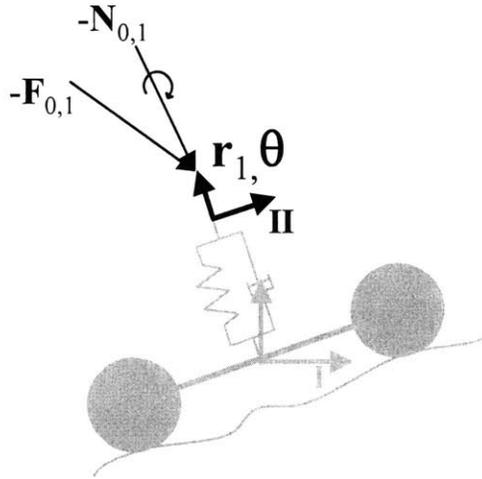


Figure B-2: Force/moment balance of compliance module

Finally, the spatial interaction forces/moments of the compliance module are given by:

$$\begin{aligned}
 d(-\mathbf{F}_{0,1})_{\Pi} &= \mathbf{b}_r^T \cdot d\dot{\mathbf{r}}_1 + \mathbf{k}_r^T \cdot d(\mathbf{r}_1 - \mathbf{r}_1^0) \\
 d(-\mathbf{N}_{0,1})_{\Pi} &= \mathbf{b}_\theta^T \cdot d\dot{\boldsymbol{\theta}} + \mathbf{k}_\theta^T \cdot d(\boldsymbol{\theta} - \boldsymbol{\theta}^0)
 \end{aligned} \tag{B-3}$$

where k_r and k_θ are the translational and rotational stiffness coefficients, \mathbf{b}_r and \mathbf{b}_θ are the translational and rotational damping coefficients. Using Equations B-1, B-2 and B-3 a set of 6 dynamic equations is obtained (forces and moments in 3D):

$$\begin{aligned}
d(-\mathbf{F}_{0,1})_{\text{II}} &= \mathbf{b}_{\bar{r}_1}^T \cdot d\dot{\mathbf{r}}_1 + \mathbf{k}_{\bar{r}_1}^T \cdot d(\mathbf{r}_1 - \mathbf{r}_1^0) \\
\Rightarrow d(-\mathbf{F}_{0,1})_{\text{II}} - \mathbf{b}_{\bar{r}_1}^T \cdot d\dot{\mathbf{r}}_1 - \mathbf{k}_{\bar{r}_1}^T \cdot d\mathbf{r}_1 &= 0 \\
\Rightarrow d(-m_1\dot{\mathbf{r}}_1)_{\text{II}} + d(-\mathbf{F}_{1,2})_{\text{II}} + d(m_1\mathbf{R}_0^{-1}\mathbf{g}) - \mathbf{b}_{\bar{r}_1}^T \cdot d\dot{\mathbf{r}}_1 - \mathbf{k}_{\bar{r}_1}^T \cdot d\mathbf{r}_1 &= 0 \\
\Rightarrow -m_1d(\ddot{\mathbf{r}}_1)_{\text{II}} + d(-\mathbf{F}_{1,2})_{\text{II}} + m_1d(\mathbf{R}_0^{-1}\mathbf{g}) - \mathbf{b}_{\bar{r}_1}^T \cdot d\dot{\mathbf{r}}_1 - \mathbf{k}_{\bar{r}_1}^T \cdot d\mathbf{r}_1 &= 0 \\
\Rightarrow m_1(d(\mathbf{R}_0^{-1}\mathbf{g}) - d(\ddot{\mathbf{r}}_1)_{\text{II}}) - \mathbf{b}_{\bar{r}_1}^T \cdot d\dot{\mathbf{r}}_1 - \mathbf{k}_{\bar{r}_1}^T \cdot d\mathbf{r}_1 &= d(\mathbf{F}_{1,2})_{\text{II}}
\end{aligned} \tag{B-4}$$

$$\begin{aligned}
d(-\mathbf{N}_{0,1})_{\text{II}} &= \mathbf{b}_{\bar{\theta}}^T \cdot d\dot{\boldsymbol{\theta}} + \mathbf{k}_{\bar{\theta}}^T \cdot d(\boldsymbol{\theta} - \boldsymbol{\theta}^0) \\
\Rightarrow d(-\mathbf{N}_{0,1})_{\text{II}} - \mathbf{b}_{\bar{\theta}}^T \cdot d\dot{\boldsymbol{\theta}} - \mathbf{k}_{\bar{\theta}}^T \cdot d\boldsymbol{\theta} &= 0 \\
\Rightarrow d(-(\mathbf{I}_1\ddot{\boldsymbol{\theta}} + \dot{\boldsymbol{\theta}} \times (\mathbf{I}_1\dot{\boldsymbol{\theta}}))_{\text{II}} + (-\mathbf{N}_{1,2})_{\text{II}} - (\mathbf{r}_2 \times \mathbf{F}_{1,2})_{\text{II}}) - \mathbf{b}_{\bar{\theta}}^T \cdot d\dot{\boldsymbol{\theta}} - \mathbf{k}_{\bar{\theta}}^T \cdot d\boldsymbol{\theta} &= 0 \\
\Rightarrow -\mathbf{I}_1d(\ddot{\boldsymbol{\theta}})_{\text{II}} - d(\dot{\boldsymbol{\theta}} \times (\mathbf{I}_1\dot{\boldsymbol{\theta}}))_{\text{II}} - d(\mathbf{N}_{1,2})_{\text{II}} - d(\mathbf{r}_2 \times \mathbf{F}_{1,2})_{\text{II}} - \mathbf{b}_{\bar{\theta}}^T \cdot d\dot{\boldsymbol{\theta}} - \mathbf{k}_{\bar{\theta}}^T \cdot d\boldsymbol{\theta} &= 0 \\
\Rightarrow -\mathbf{I}_1d(\ddot{\boldsymbol{\theta}})_{\text{II}} - d(\dot{\boldsymbol{\theta}} \times (\mathbf{I}_1\dot{\boldsymbol{\theta}}))_{\text{II}} - d(\mathbf{r}_2 \times \mathbf{F}_{1,2})_{\text{II}} - \mathbf{b}_{\bar{\theta}}^T \cdot d\dot{\boldsymbol{\theta}} - \mathbf{k}_{\bar{\theta}}^T \cdot d\boldsymbol{\theta} &= d(\mathbf{N}_{1,2})_{\text{II}}
\end{aligned}$$

B.2. Alternate derivation

This appendix also presents an alternative derivation for the dynamic equations of motion for a spatial (3D) mobile robot system. The key difference between this derivation and one derived in Section B-1 is the location of an intermediate reference frame (Frame II). However, this small change leads to final equations of motion that are incompatible with the available sensory data for the model based dynamic parameter identification problem—an important consideration when using model based identification methods. The system represented in Figure 2-1, is again reduced to three components: a rigid arm, a rigid vehicle body and a compliance module (see Figure B-3).

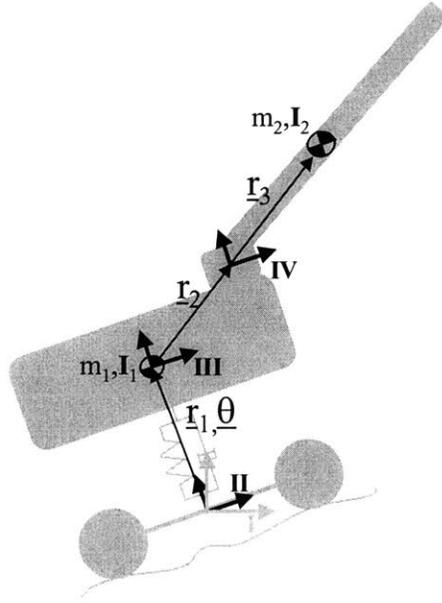
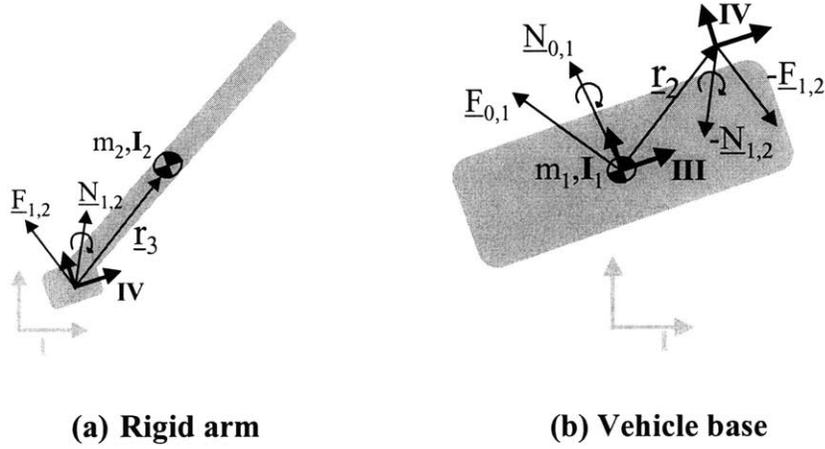


Figure B-3: Representation of the simplified mobile robot



(a) Rigid arm

(b) Vehicle base

Figure B-4: Force/moment balance

The spatial interaction forces/moments of the rigid arm (Figure B-4a) are:

$$\begin{aligned}
 (\bar{F}_{1,2})_{IV} + (m_2 \bar{g})_{IV} &= (m_2 \bar{a}_{cm_2})_{IV} \\
 (\bar{N}_{1,2})_{IV} - (\bar{r}_3 \times \bar{F}_{1,2})_{IV} &= \left(\frac{\partial (\mathbf{I}_2^{cm} \bar{\omega}_2^{cm})}{\partial t} + \bar{\omega}_2^{cm} \times (\mathbf{I}_2^{cm} \bar{\omega}_2^{cm}) \right)_{IV} \\
 (\bar{F}_{1,2})_{IV} \text{ and } (\bar{N}_{1,2})_{IV} &\Rightarrow \text{from arm base F/T sensor}
 \end{aligned} \tag{B-5}$$

where $\underline{F}_{1,2}$ and $\underline{N}_{1,2}$ are the reaction forces and moments, m_2 and \mathbf{I}_2 are the arm mass and inertia

tensors, \underline{a}_2 and $\underline{\omega}_2$ are the arm linear acceleration and angular velocity vectors. The spatial interaction forces/moments of the rigid base (Figure B-4b) are:

$$\begin{aligned}
(\bar{F}_{0,1})_{\Pi} + (-\bar{F}_{1,2})_{\Pi} + (m_1 \bar{g})_{\Pi} &= (m_1 \bar{a}_1^{cm})_{\Pi} \\
(\bar{F}_{0,1})_{\Pi} &= (m_1 \bar{a}_1^{cm})_{\Pi} - (-\bar{F}_{1,2})_{\Pi} - (m_1 \bar{g})_{\Pi} \\
&= \left(m_1 \left(\ddot{\bar{r}}_1 + 2 \left(\dot{\bar{\theta}} \times \bar{v}_{rel} \right) + \dot{\bar{\theta}} \times \left(\dot{\bar{r}}_1 \right) + \ddot{\bar{\theta}} \times \bar{r}_1 \right) \right)_{\Pi} - (-\bar{F}_{1,2})_{\Pi} - (m_1 \mathbf{R}_0^{-1} \bar{g}) \\
&= \left(m_1 \left(\ddot{\bar{r}}_1 + 2 \left(\dot{\bar{\theta}} \times \left(\dot{\bar{r}}_1 + \dot{\bar{\theta}} \times \bar{r}_1 \right) \right) + \dot{\bar{\theta}} \times \left(\dot{\bar{r}}_1 \right) + \ddot{\bar{\theta}} \times \bar{r}_1 \right) \right)_{\Pi} - (-\bar{F}_{1,2})_{\Pi} - (m_1 \mathbf{R}_0^{-1} \bar{g}) \quad (\text{B-6}) \\
(\bar{N}_{0,1})_{\Pi} + (-\bar{N}_{1,2})_{\Pi} - (\bar{r}_2 \times \bar{F}_{1,2})_{\Pi} &= \left(\frac{\partial (\mathbf{I}_1^{cm} \varpi_1^{cm})}{\partial t} + \varpi_1^{cm} \times (\mathbf{I}_1^{cm} \varpi_1^{cm}) \right)_{\Pi} \\
(\bar{N}_{0,1})_{\Pi} &= \left(\frac{\partial (\mathbf{I}_1^{cm} \varpi_1^{cm})}{\partial t} + \varpi_1^{cm} \times (\mathbf{I}_1^{cm} \varpi_1^{cm}) \right)_{\Pi} - (-\bar{N}_{1,2})_{\Pi} + (\bar{r}_2 \times \bar{F}_{1,2})_{\Pi} \\
&= \left(\mathbf{I}_1 \ddot{\bar{\theta}} + \dot{\bar{\theta}} \times (\mathbf{I}_1 \dot{\bar{\theta}}) \right)_{\Pi} - (-\bar{N}_{1,2})_{\Pi} + (\bar{r}_2 \times \bar{F}_{1,2})_{\Pi}
\end{aligned}$$

where \underline{F}_{01} and \underline{N}_{01} are the reaction forces and moments, m_1 and \mathbf{I}_1 are the base mass and inertia tensors, \underline{a}_1 and $\underline{\omega}_1$ are the base linear acceleration and angular velocity vectors.

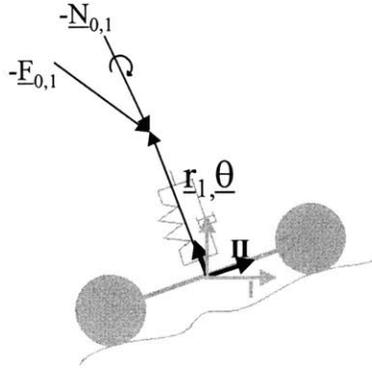


Figure B-5: Force/moment balance of compliance module

Using Equations B-3, B-5, B-6 a set of 6 dynamic equations is obtained (forces/moments in 3D):

$$\begin{aligned}
d(-\bar{F}_{0,1})_{\Pi} &= \bar{b}_{\bar{r}_1}^T \cdot d\dot{\bar{r}}_1 + \bar{k}_{\bar{r}_1}^T \cdot d(\bar{r}_1 - \bar{r}_1^0) \\
\Rightarrow d(-\bar{F}_{0,1})_{\Pi} - \bar{b}_{\bar{r}_1}^T \cdot d\dot{\bar{r}}_1 - \bar{k}_{\bar{r}_1}^T \cdot d\bar{r}_1 &= 0 \\
\Rightarrow d \left(- \left(m_1 \left(\ddot{\bar{r}}_1 + 2 \left(\dot{\bar{\theta}} \times \left(\dot{\bar{r}}_1 + \dot{\bar{\theta}} \times \bar{r}_1 \right) \right) + \dot{\bar{\theta}} \times \left(\dot{\bar{r}}_1 \right) + \ddot{\bar{\theta}} \times \bar{r}_1 \right) \right)_{\Pi} + (-\bar{F}_{1,2})_{\Pi} + (m_1 \mathbf{R}_0^{-1} \bar{g}) \right) & \quad (\text{B-7}) \\
- \bar{b}_{\bar{r}_1}^T \cdot d\dot{\bar{r}}_1 - \bar{k}_{\bar{r}_1}^T \cdot d\bar{r}_1 &= 0 \\
\Rightarrow m_1 \left(d(\mathbf{R}_0^{-1} \bar{g}) - d \left(\ddot{\bar{r}}_1 + 2 \left(\dot{\bar{\theta}} \times \left(\dot{\bar{r}}_1 + \dot{\bar{\theta}} \times \bar{r}_1 \right) \right) + \dot{\bar{\theta}} \times \left(\dot{\bar{r}}_1 \right) + \ddot{\bar{\theta}} \times \bar{r}_1 \right)_{\Pi} \right) - \bar{b}_{\bar{r}_1}^T \cdot d\dot{\bar{r}}_1 - \bar{k}_{\bar{r}_1}^T \cdot d\bar{r}_1 &= d(\bar{F}_{1,2})_{\Pi}
\end{aligned}$$

$$\begin{aligned}
& \mathbf{d}(-\bar{N}_{0,1})_{\Pi} + \mathbf{d}(\bar{\mathbf{r}}_1 \times -\bar{F}_{0,1}) = \bar{\mathbf{b}}_{\theta}^T \cdot \mathbf{d}\dot{\bar{\theta}} + \bar{\mathbf{k}}_{\theta}^T \cdot \mathbf{d}(\bar{\theta} - \bar{\theta}^0) \\
\Rightarrow & \mathbf{d}(-\bar{N}_{0,1})_{\Pi} + \mathbf{d}(\bar{\mathbf{r}}_1 \times -\bar{F}_{0,1})_{\Pi} - \bar{\mathbf{b}}_{\theta}^T \cdot \mathbf{d}\dot{\bar{\theta}} - \bar{\mathbf{k}}_{\theta}^T \cdot \mathbf{d}\bar{\theta} = 0 \\
\Rightarrow & \mathbf{d}\left(-\left(\mathbf{I}_1 \ddot{\bar{\theta}} + \dot{\bar{\theta}} \times \left(\mathbf{I}_1 \dot{\bar{\theta}}\right)\right)_{\Pi} + \left(-\bar{N}_{1,2}\right)_{\Pi} - \left(\bar{\mathbf{r}}_2 \times \bar{F}_{1,2}\right)_{\Pi}\right) \\
& + \mathbf{d}\left(\bar{\mathbf{r}}_1 \times \left(-m_1 \left(\ddot{\bar{\mathbf{r}}}_1 + 2\left(\dot{\bar{\theta}} \times \left(\dot{\bar{\mathbf{r}}}_1 + \dot{\bar{\theta}} \times \dot{\bar{\mathbf{r}}}_1\right)\right) + \dot{\bar{\theta}} \times \left(\dot{\bar{\theta}} \times \dot{\bar{\mathbf{r}}}_1\right) + \ddot{\bar{\theta}} \times \bar{\mathbf{r}}_1\right)_{\Pi} + \left(-\bar{F}_{1,2}\right)_{\Pi} + \left(m_1 \mathbf{R}_0^{-1} \bar{\mathbf{g}}\right)_{\Pi}\right) \quad (\text{B-8}) \\
& - \bar{\mathbf{b}}_{\theta}^T \cdot \mathbf{d}\dot{\bar{\theta}} - \bar{\mathbf{k}}_{\theta}^T \cdot \mathbf{d}\bar{\theta} = 0 \\
\Rightarrow & -\mathbf{I}_1 \mathbf{d}\left(\ddot{\bar{\theta}}\right)_{\Pi} - \mathbf{d}\left(\dot{\bar{\theta}} \times \left(\mathbf{I}_1 \dot{\bar{\theta}}\right)\right)_{\Pi} - \mathbf{d}\left(\bar{\mathbf{r}}_2 \times \bar{F}_{1,2}\right)_{\Pi} - \mathbf{d}\left(\bar{\mathbf{r}}_1 \times m_1 \bar{\mathbf{a}}_{cm1}\right)_{\Pi} + \mathbf{d}\left(\bar{\mathbf{r}}_1 \times m_1 \mathbf{R}_0^{-1} \bar{\mathbf{g}}\right)_{\Pi} \\
& - \bar{\mathbf{b}}_{\theta}^T \cdot \mathbf{d}\dot{\bar{\theta}} - \bar{\mathbf{k}}_{\theta}^T \cdot \mathbf{d}\bar{\theta} = \mathbf{d}\left(\bar{N}_{1,2}\right)_{\Pi} + \mathbf{d}\left(\bar{\mathbf{r}}_1 \times \bar{F}_{1,2}\right)_{\Pi}
\end{aligned}$$

Note the last equation (B-8) require measurements of absolute positions ($\underline{\mathbf{r}}$) with respect to the nominal “spring” position. In general this is difficult to measure and incompatible with the onboard sensors.

Appendix C

Loss-less Image Compression

C.1. Run Length Encoding—The General Idea Of Compression

Data files frequently contain the same character repeated many times in a row. Digitized signals can have runs of the same value, indicating that the signal is not changing. Run-length encoding is a simple method of compressing these types of files.

A *compression program* is used to convert data from an easy-to-use format to one optimized for compactness. Likewise, an *uncompression program* returns the information to its original form. Here only compression is addressed (since a measure on the information present after compression is required, with no needs for decompression of the data). Decompression techniques can be inferred from the compression methods or refer to [Smith-2] for complete descriptions.

There are many different forms of compression which have been classified in various ways. One way to classify these is: Lossless vs. lossy. A lossless technique means that the restored data file is *identical* to the original. This is absolutely necessary for many types of data, for example: executable code, word-processing files, tabulated numbers, etc. In comparison, data files that represent images and other required signals do not have to be kept in perfect condition for storage or transmission. All real world measurements inherently contain a certain amount of noise. If the changes made to these signals resemble a small amount of additional noise, no harm is done. Compression techniques that allow this type of degradation are called lossy. This distinction is important since lossy techniques are much more effective at compression than

lossless methods. The higher the compression ratio, the more noise added to the data. In this appendix, a few common methods of lossless compression are briefly described. A thorough review is beyond the scope of this thesis, but can be found in [Smith-2].

C.2. Simple ccompression

In digitized (binary) input data, each time a zero is encountered, *two* values are written to the output file. The first of these values is a zero, a flag to indicate that run-length compression is beginning. The second value is the number of zeros in the run. If the average run-length is longer than two, compression will take place. However, many single zeros in the data can make the encoded file larger than the original.

C.3. More efficient case—Huffman coding

This method is named after D. A. Huffman, who developed the procedure in the 1950s. The histogram of a data file may show a large percentage of certain symbol(s) occurring more often than others. This can be used to make an appropriate compression scheme for this file. The idea is to assign frequently used characters fewer bits, and seldom used characters more bits. In mathematical terms, the optimal situation is reached when the number of bits used for each character is proportional to the logarithm of the character's probability of occurrence. This is achieved by generating a binary tree (i.e. any node can only have two branches) where each leaf (terminal node) is assigned a unique symbol from the data file. Each branch of this tree is assigned a value of 0,1. The code for any given symbol is found by reading the 0's and 1's from the root to the leaf of the tree. The key is to set up the branching of the tree appropriately. This is done as follows:

- (i) initialization: put all nodes (symbols at this point) in an open list
- (ii) Repeat the following until the open list has only one node left

- (a) from the open list pick two nodes having the lowest probabilities of occurrence and place them into two leaf locations
- (b) create a parent node of them in the tree
- (c) assign the sum of the children's probabilities of occurrence to the parent node and insert the parent node label into the open list
- (d) assign code 0,1 to the two branches of the parent node in the tree and delete the children from the open list

C.4. Very efficient case—LZW coding

LZW compression is named after its developers, A. Lempel and J. Ziv, with later modifications by Terry A. Welch. It is the foremost technique for general purpose data compression due to its versatility. Typically, you can expect LZW to compress text, executable code, and similar data files to about one-half their original size. LZW also performs well when presented with extremely redundant data files, such as tabulated numbers, computer source code, and acquired signals. Compression ratios of 5:1 are common for these cases. LZW compression uses a code table. A common choice is to provide 4096 entries in the table. In this case, the LZW encoded data consists entirely of 12 bit codes, each referring to one of the entries in the code table. Decompression is achieved by taking each code from the compressed file, and translating it through the code table to find what character or characters it represents. Codes 0-255 in the code table are always assigned to represent single symbols from the input file. The LZW method achieves compression by using codes 256 through 4095 to represent *sequences* of bytes. The longer the sequence assigned to a single code, and the more often the sequence is repeated, the higher the compression achieved. Although this is a simple approach, there are two major obstacles that need to be overcome: (1) how to determine what sequences should be in the code

table, and (2) how to provide the decompression program the same code table used by the compression program. The LZW algorithm exquisitely solves both these problems. When the LZW program starts to encode a file, the code table contains only the first 256 entries, with the remainder of the table being blank. This means that the first codes going into the compressed file are simply the single symbols from the input file being converted to 12 bits. As the encoding continues, the LZW algorithm identifies repeated sequences in the data, and adds them to the code table. Compression starts the second time a sequence is encountered. The key point is that a sequence from the input file is not added to the code table until it has already been placed in the compressed file as individual characters (codes 0 to 255). This is important since it allows the decompression program to reconstruct the code table directly from the compressed data, without having to transmit the code table separately. The actual LZW algorithm is summarized in a flowchart in figure C-1. One drawback of the LZW algorithm is its computational complexity. However, several commercial software packages are available that have optimized this algorithm to make it computationally efficient.

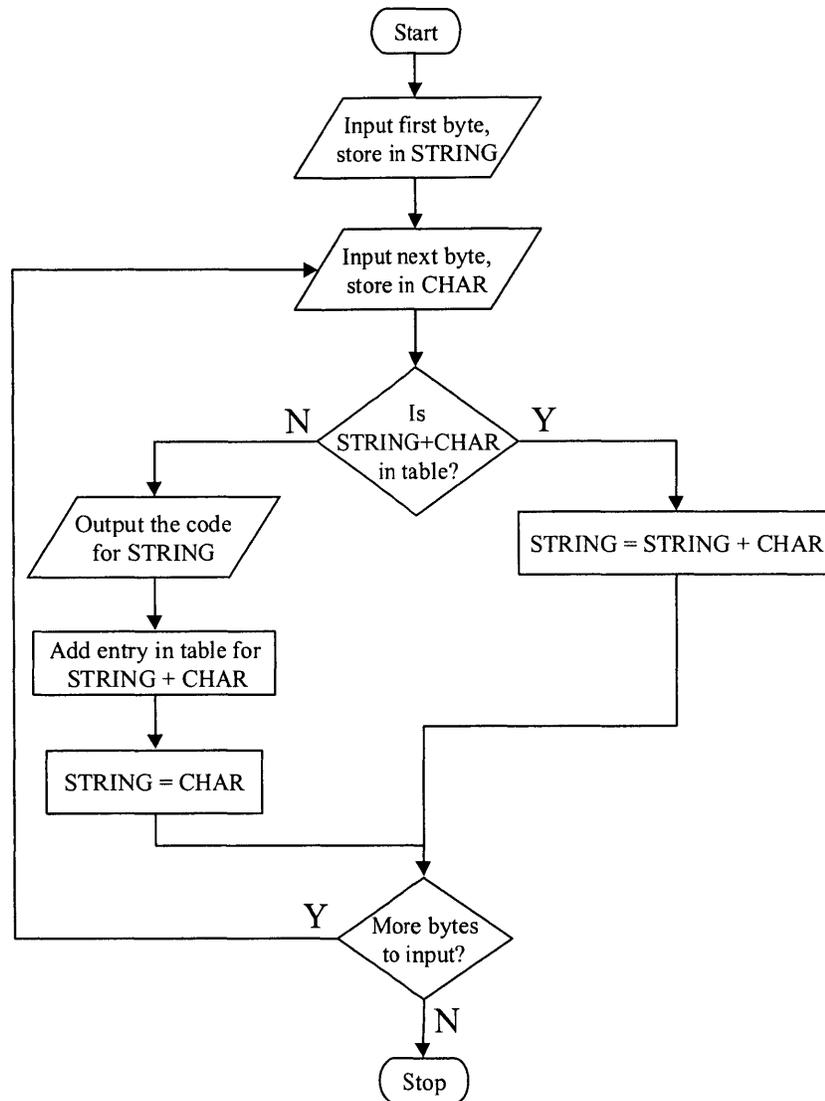


Figure C-1: LZW compression flowchart

C.5. Lossless JPEG compression

This compression method is a special case of the JPEG where there is no loss. It uses a predictive method. A predictor combines the values of up to three neighboring pixels to give the predicted value for the current pixel. The encoder compares this prediction with the actual pixel value at the given position, and encodes the difference (prediction residual) losslessly. Since it uses only previously encoded neighbors, certain pixels will use their original values for encoding.

C.6. Compression comparison results

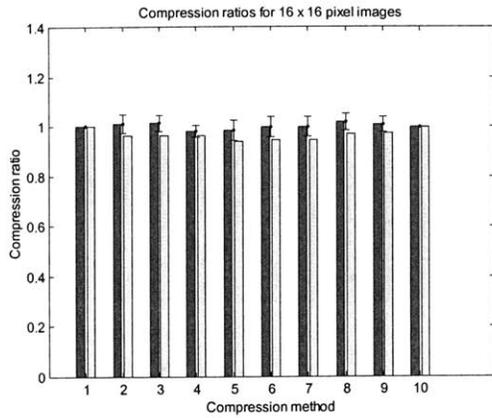
9 techniques for data compression are compared here:

Method 1	No compression
Method 2	Predictor: $I(u,v) = I(u,v) - I(u,v-1)$
Method 3	Predictor: $I(u,v) = I(u,v) - I(u-1,v)$
Method 4	Predictor: $I(u,v) = I(u,v) - I(u-1,v-1)$
Method 5	Predictor: $I(u,v) = I(u,v) - (I(u,v-1)+I(u-1,v)-I(u-1,v-1))$
Method 6	Predictor: $I(u,v) = I(u,v) - (I(u,v-1)+(I(u-1,v)-I(u-1,v-1))/2)$
Method 7	Predictor: $I(u,v) = I(u,v) - (I(u-1,v)+(I(u,v-1)-I(u-1,v-1))/2)$
Method 8	Predictor: $I(u,v) = I(u,v) - (I(u-1,v)+I(u,v-1))/2$
Method 9	Predictor: $I(u,v) = I(u,v) - (I(u-1,v)+I(u-1,v-1)+I(u,v-1))/3$
Method 10	Huffman coding

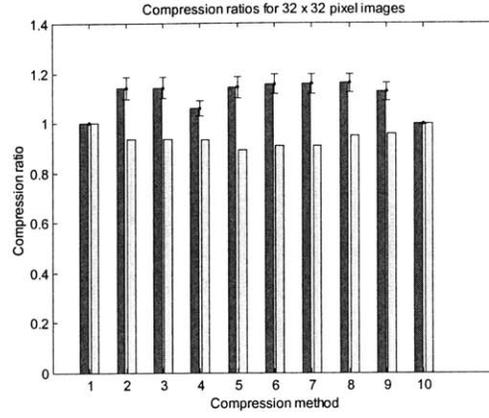
- 500 different images were coded by each of the methods (image sizes considered: 16^2 , 32^2 , 64^2 , 128^2 , 256^2 pixels)
- 100 pure noise images were also coded by each of the methods
- Comparison of information content before and after coding give the compression ratio i.e. number of bits needed to represent a pixel before and after compression using the Shannon entropy function
- From table C-1 and Figure C-2 predictor method 5 appears to give the highest compression ratios and is extended to the information theoretic vision planning algorithm in chapter 3.

Table C-1: Comparison of compression methods on 2-D images

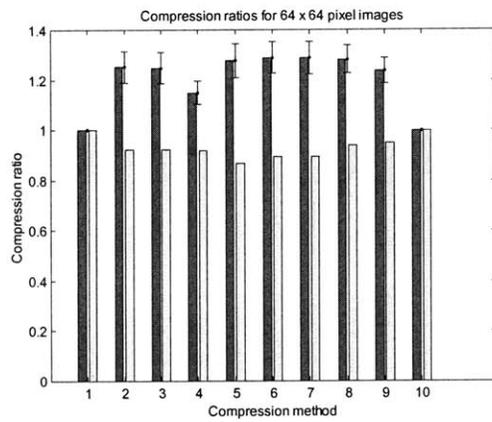
	Compression ratio (CR \pm σ)				
	Image: 16^2 pixels	Image: 32^2 pixels	Image: 64^2 pixels	Image: 128^2 pixels	Image: 256^2 pixels
Method 1	1.000 \pm 0.000	1.000 \pm 0.000	1.000 \pm 0.000	1.000 \pm 0.000	1.000 \pm 0.000
Method 2	1.009 \pm 0.038	1.142 \pm 0.045	1.253 \pm 0.063	1.375 \pm 0.089	1.531 \pm 0.146
Method 3	1.013 \pm 0.033	1.142 \pm 0.044	1.247 \pm 0.064	1.358 \pm 0.091	1.488 \pm 0.147
Method 4	0.980 \pm 0.025	1.060 \pm 0.031	1.149 \pm 0.046	1.250 \pm 0.066	1.348 \pm 0.101
Method 5	0.983 \pm 0.041	1.143 \pm 0.042	1.277 \pm 0.068	1.416 \pm 0.111	1.723 \pm 0.234
Method 6	0.996 \pm 0.041	1.156 \pm 0.041	1.289 \pm 0.063	1.427 \pm 0.098	1.646 \pm 0.184
Method 7	0.997 \pm 0.039	1.157 \pm 0.039	1.288 \pm 0.064	1.422 \pm 0.101	1.619 \pm 0.181
Method 8	1.018 \pm 0.035	1.159 \pm 0.036	1.282 \pm 0.056	1.414 \pm 0.086	1.564 \pm 0.145
Method 9	1.008 \pm 0.031	1.126 \pm 0.035	1.236 \pm 0.052	1.358 \pm 0.077	1.480 \pm 0.123
Method 10	0.997 \pm 0.001	0.996 \pm 0.001	0.996 \pm 0.001	0.996 \pm 0.001	0.996 \pm 0.001



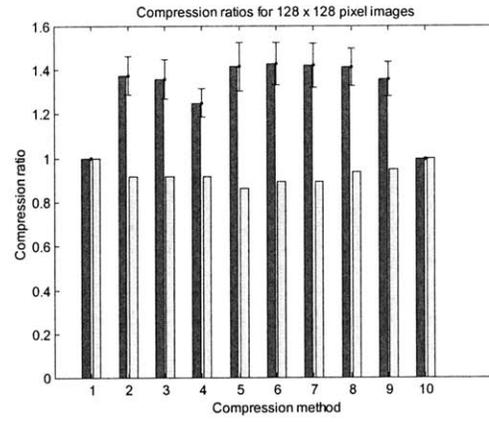
(a) 16² pixels



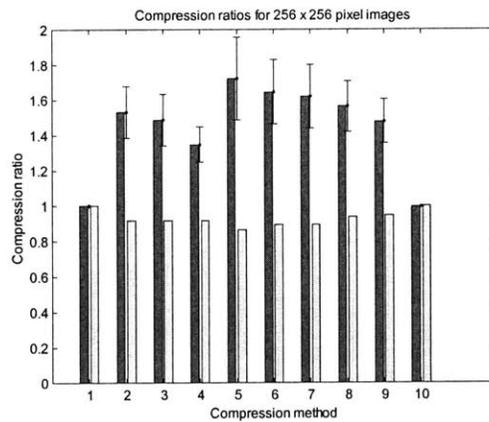
(b) 32² pixels



(c) 64² pixels



(d) 128² pixels



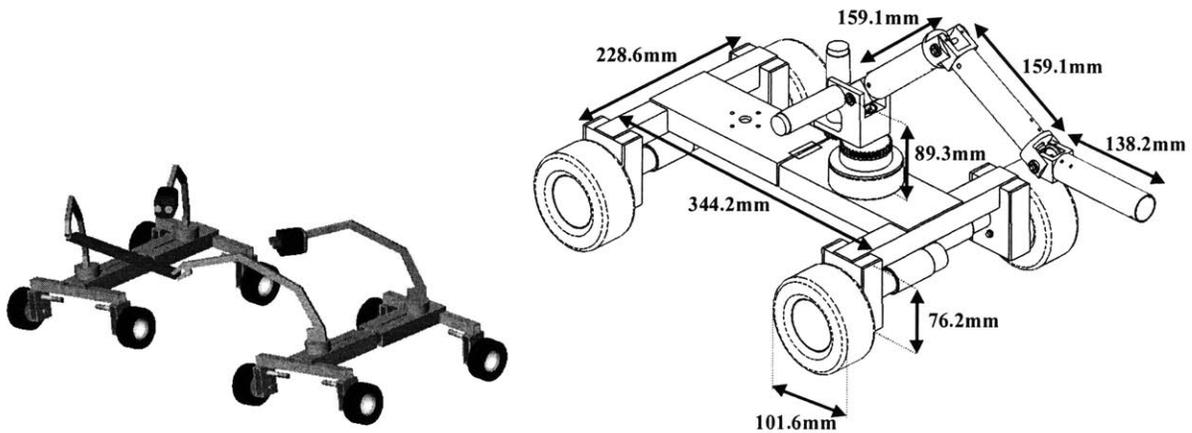
(e) 256² pixels

Figure C-2: Comparison of 2-D image compression methods

Appendix D

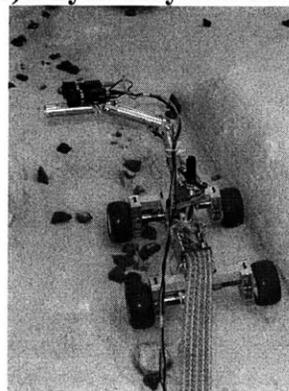
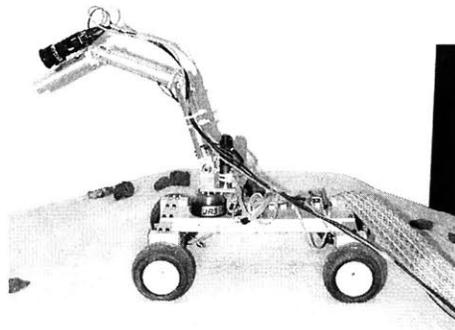
Experimental Cooperative Robot System

This section contains a brief description of the Field and Space Robotics Laboratory (FSRL) experimental cooperative robot system, which are used to experimentally validate much of the work in this thesis. The two FSRL robots were designed and built primarily by Grant Kristofek. A more detailed description of the rover can be found in (Kristofek, 2002). All power electronics and computer interfaces were developed by the author.



(a) Conceptual physical cooperation

(b) Physical system schematic



(c) Physical system implementation

Figure D-1: FSRL Experimental cooperative rover

Each of the FRSL experimental robots is a four-wheeled mobile robot with an articulated center (see Figure D-1). This passive joint allows all 4 wheels to maintain ground contact without the need for independent suspensions. The four wheels are driven by 12 V geared DC motors with a peak stall torque of 908 oz-in and angular velocity of 49 rpm at 208 oz-in. The resulting velocity of the rover is approximately 25 cm/sec at 208 oz-in motor torque. The robot is steered with skid-steering. Each system is also equipped with a 4 DOF arm that can control endpoint (x, y, z) position and angular pitch (see Figure D-2). Each joint of the arm is driven by 12 V geared DC motors with a peak torque of 1495 oz-in. The robot weighs (w/o arm) 3.74 kg and the arm weighs 0.76 kg.

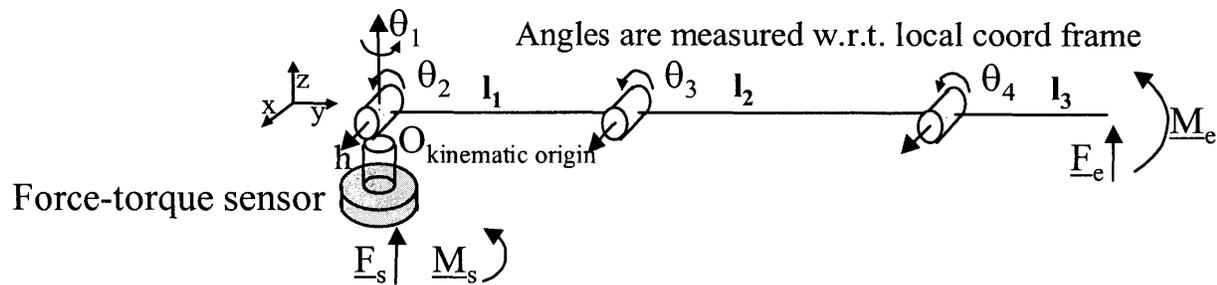


Figure D-2: Experimental system: 4 DOF manipulator kinematics

The rover has on-board interface card that links the robot sensors and vehicle/arm drive motors to a power amplifier card and a Pentium 166 MHz computer. An 8 axis motion control card provides the analog, digital and encoder interfaces required to perform closed loop control on the individual motors. Power to the motors is provided by an 8 axis PWM amplifier card built using the National Semiconductor's LMD18200T 3A 55V solid state amplifiers. The 18200T chips takes a low power PWM input signal and outputs a high power PWM signal (proportional in frequency and duty cycle as the input; amplitude given by the supply voltage, V_s). The low power PWM input signal is generated using a 555 timer chip. The frequency of the output is set up by a clock. The duty cycle of the 555 output is a controlled fraction of the clock duty cycle.

This control fraction is the command voltage/max. voltage (V_{cc}). The command voltage is supplied by the analog output of the motion control card, set by the control software. Although, a conventional quartz oscillator may be used as a clock, it is better to use another 555 timer chip to provide the clock input. This way the clock duty cycle may be selected close to 100%, giving a greater range of duty cycles for the PWM output from the second stage 555 timer chip. A single axis circuit is given in figure D-3.

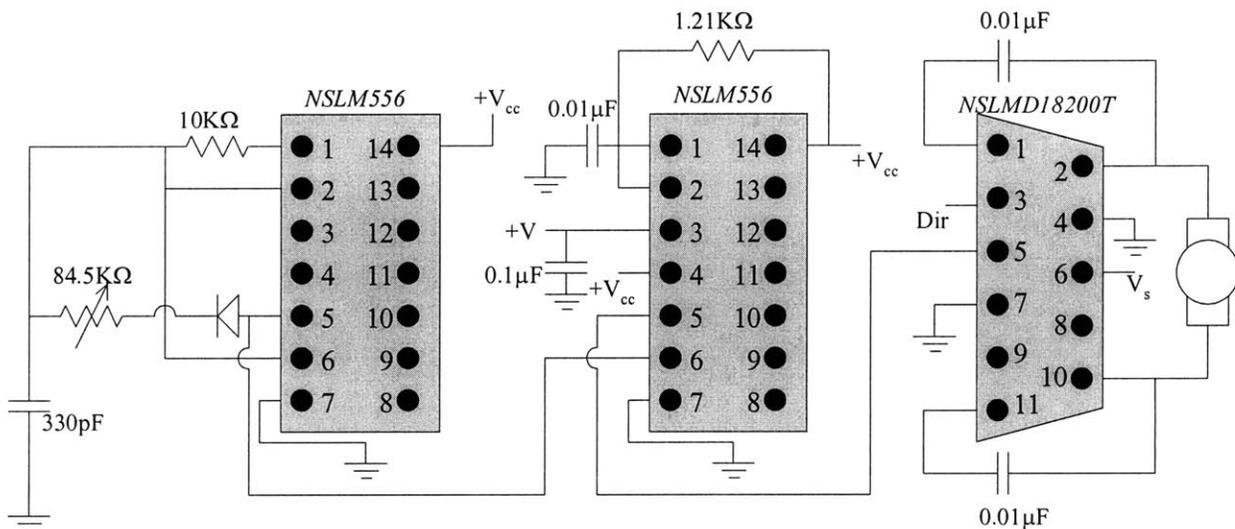


Figure D-3: PWM motor control circuit

The robot sensor suite is composed of tachometers to measure the wheel angular velocities, encoders to measure arm joint angles, a potentiometer to measure the articulated center angle, and a two-axis Crossbow CXTA02 inclinometer mounted to the robot body to determine roll and pitch relative to an inertial frame. A JR3-67M25A six-axis force/torque sensor is mounted under the robot arm to measure forces exerted on the robot body by the manipulator. Inter-rover communication is achieved using 9600 bps wireless modems. Additionally, one of the two robot systems is also equipped with a stereo vision system. An overview of the hardware architecture interface is seen in figure D-4.

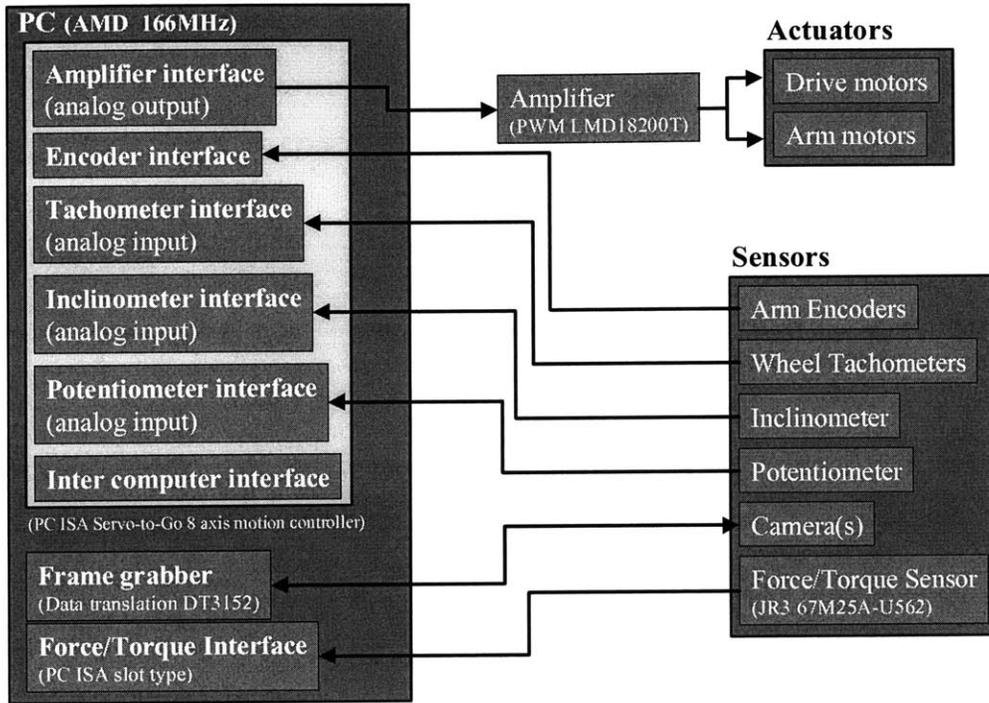


Figure D-4: Overview of experimental system hardware

Calibration is required for the Crossbow inclinometers since the factory power settings (8V-30Vdc) are different than the laboratory power settings (5Vdc). In the sensor's linear range ($\pm 20^\circ$) the output of the sensor (volts) is converted to a pitch/roll angle (radians) using the following:

$$\theta_{r/p} = \frac{V_{out} - \tilde{V}_{r/p}}{S_{r/p}} \quad (D-1)$$

where V_{out} is the current sensor reading, $S_{r/p}$ is the sensitivity of the roll/pitch axis and $\tilde{V}_{r/p}$ is the zero angle voltage for roll/pitch axis. The goal of calibration is to determine $\tilde{V}_{r/p}$ and $S_{r/p}$. Thus we need two angular readings $\theta_{r/p}$ and V_{out} for each axis, to give us two equations in two unknowns in each axis. Using these two readings the unknown constants can be solved. These two equations for each axis are obtained by tilting the robot/sensor system in the desired axis.

V_{out} is measured directly from the inclinometer. $\theta_{r/p}$ is measured using the force/torque sensor— at every new angle the force/torque sensor is reset to zero, a weight is added, and the resulting force components give direction of the gravity vector and $\theta_{r/p}$. Table D-1 give the results after calibration:

Table D-1: Inclinometer calibration data at room temperature

Calibration parameters	Robot 1		Robot 2	
	Roll axis	Pitch Axis	Roll Axis	Pitch Axis
$\tilde{V}_{r/p}$ (volts)	2.406	2.381	2.447	2.392
$S_{r/p}$ (mV/degree)	33.631	34.733	29.338	34.905

Inter-robot communication can be achieved in several ways. However, since this is not a critical part of this research, a simple method is implemented. Digital I/O data cards in each computer are linked using one byte and a handshaking protocol is set up. To save space, bits are used for both transmitting and receiving data. The “transmitting” robot issues a transmit command to the “receiving” robot and waits for an acknowledgement command. Upon receiving the acknowledgement command it starts transmitting the data byte-wise separated by an end-of-byte command. The last byte is followed by an end-of-data command. The “receiving” robot periodically checks its receiving port for transmit command. Upon receiving this, it sends out an acknowledgement command and starts reading the data. The end-of-byte commands help the “receiving” robot separate the data coherently. Additionally, this protocol must make sure that the “receiving” robot is able to read data at least as fast as it is being transmitted. This can be achieved in two ways. First, the “transmitting” robot can be forced to transmit at a slower rate, by empirically determining the receiving rate. This is not very robust. A second, slower but more

robust method requires sending a received-ok command from the “receiving” robot to the “transmitting” robot, between each byte. This simple protocol can be expanded for n-bytes.

Visual data is acquired by a stereo camera pair. The monochrome CCD cameras use 8.5mm lenses, with a 41.2° FOV. Each camera has a 1/3” interline transfer CCD, with 510x494 (H x V) pixels. Pixel dimensions are 9.6 x 7.5 μm (H x V).

Table D-2: Camera specifications

Camera supplier	P39,244 from Edmund Scientific, Barrington, NJ 08007-1380.
Imaging Device	½" interline transfer CCD
Pixels	510H x 492V
Pixel Size	12.7 x 9.8μm
Horizontal Resolution	>350 TV lines
Sensing Area	6.4 x 4.8 mm
Signal Format	EIA (RS-170);
Video Output	1V p-p, 75W; RCA connector
Lens Mount	C-mount
Flange Back Length	12.5 mm
Sync System	Internal
Min. Sensitivity	0.2 lux
S/N Ratio	>46 dB
Electronic Shutter Speed	N/A
Gamma	0.45
Auto Gain Control	27 dB
Power requirement	12V DC; 200mA max.
Storage/Operating Temp.	-30 to 70°C/-20 to 55°C
Dimensions	30.5 x 30.5 x 61 mm
Mounting	¼-20 TPI tapped
Weight	120g

For an ideal stereo camera pair (cameras aligned parallel), range resolution is governed by:

$$\Delta r = (r^2/(bf))\Delta d \quad (D-2)$$

where r is the distance to the target, b is the inter-camera baseline distance, f is the focal length, and Δd is the minimum disparity (e.g. 1 pixel). Thus for a desired resolution (Δr) of 0.1mm, at a range (r) of 100mm, with a focal length (f) of 8.5mm and minimum disparity (Δd) of 1 pixel (or about 0.0075mm), we get a required baseline distance of about 88mm. However, due to the properties of the stereo imaging software (Small Vision System™, Videre Design), the stereo baseline is set at 38mm.

Appendix E

Lightweight hyper-redundant binary mechanisms

E.1. Introduction

As shown in Chapters 3, 4 and 5, the environment and task modeling algorithms require significant motion of visual sensors. This includes both large and precise motion of the cameras as well as manipulator dexterity (i.e. hyper-redundancy in manipulator motion to get around obstacles). Here, a new design for a lightweight hyper-redundant manipulator based on binary actuators is presented. This mechanism is called the BRAID—Binary Robotic Articulated Intelligent Device. With a large number of binary actuators, the mechanism can approximate a continuous system in dexterity and utility. However, with its polymer construction and binary actuation the design is very lightweight and simple, appropriate for space exploration systems.

E.2. Background and Literature Review

In the robotic research community, efforts have been made to develop concepts of simple manipulators with good performance. An interesting example is binary manipulators [Ebert-Uphoff, Hughes, Lees]. In this concept, a manipulator is controlled by activating actuators that can assume only one of two states (“on” or “off”). The joint level control is very simple. By activating the given actuator in the system a discrete change in state is obtained. Often, the control does not require feedback sensors. The two states are the extreme positions of the actuator. This form of control has been classified as sensor-less manipulation [Erdmann, Goldberg, Lees]. As the number of binary actuators in the system increases, the capabilities of the device approach that of a conventional continuous manipulator. However, this leads to

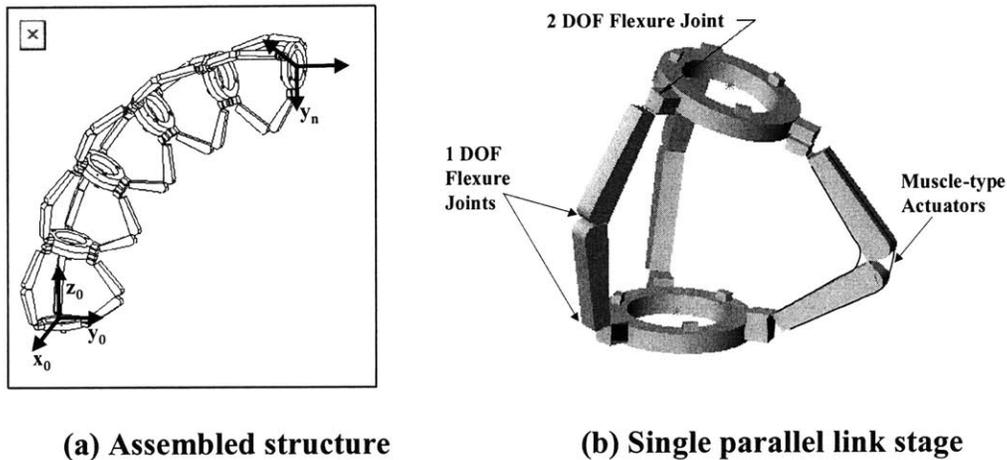
mechanisms with complex system kinematics. Studies of the kinematics and control of such “hyper-redundant” manipulators, both with and without binary actuation have been performed [Chirikjian, Ebert-Uphoff, Huang, Hughes, Kwon]. A single, yet lightweight, robust and simple device that could perform a number of these tasks would be highly desirable. It would need to have fine motion resolution, a large motion workspace, high accuracy, multiple degrees of freedom, control simplicity, and have a small stowed volume.

This appendix presents the design of an element that is intended to meet these requirements and overcome the challenges. This device, called a Binary Robotic Articulated Intelligent Device (BRAID), consists of compliant mechanisms with large numbers of embedded actuators and is a step toward practical implementation of binary devices for space robotic systems. Several potential application concepts for the BRAID element have been proposed [Lichter-1, Lichter-2, Sujan]. In some ways it resembles deployable systems that have been used in the past for space applications as: deployable booms, solar arrays, antennas, articulated masts and others [Dotson, Gantes, Meguro, Syromiatnikov]. However, these are not controllable and usually constructed from heavy and complex components, such as gears, motors, cables, etc., although there are some notable exceptions [Darby, Lichter, Pellegrino]. This appendix addresses the design issues, system kinematics, and practical implementation concerns that go into developing such a system.

E.2. BRAID Design

The BRAID mechanism is made of a serial chain of parallel stages (see Figure E-1(a)). Each three DOF stage has three flexure-based legs, each with muscle type binary actuator. In the experimental system described here these are shape memory alloy (SMA) actuators. Muscle actuation allows binary operation of each leg. Figure E-1(b) shows one stage of the BRAID

element. Each leg has three flexure joints—two one DOF joints and one three DOF joint. This results in five axes per leg: three in parallel, the fourth orthogonal to the first three and the fifth orthogonal to the fourth. Coupling the three legs together (symmetrically 120° apart) gives the parallel link stage three DOF mobility (vertical translation, pitch, and yaw). However, in the physical implementation of the design the fifth DOF in each leg was removed, as this motion is small and can be accommodated by elastic deflections.



(a) Assembled structure (b) Single parallel link stage
Figure E-1: BRAID design concept

In the BRAID application large ranges of motion and low stiffnesses in the axes of rotation are desired, while maintaining high stiffnesses in all other axes. Repeated bending of a flexure can cause fatigue failure. The relationship between performance and fatigue life can be estimated to first order by considering a simple beam of thickness t , with Young’s modulus E , bent elastically to a radius of R . Surface strain and maximum elastic stress is given by:

$$\varepsilon = \frac{t}{2R} \quad \text{and} \quad \sigma = E \frac{t}{2R} \tag{E-1}$$

This stress must not exceed the fatigue yield strength of the material, σ_f . The minimum bend radius is given by:

$$R \geq \frac{t}{2} \left(\frac{E}{\sigma_f} \right) = \frac{t}{2} \left(\frac{1}{M} \right) \tag{E-2}$$

Materials that can be bent to the smallest radius or the ones where M (defined in Equation E-2) is maximized are desirable because they give the largest range of motion. Literature suggests the best choices are polymeric materials and elastomers with M equal to 3×10^{-2} [Ashby]. Materials such as polyethylene, polypropylene and nylon fall into this category. For comparison, for spring steel M equals 0.5×10^{-2} (which would be appropriate when high stiffness and small range of motion is desired). An ultra high molecular weight polyethylene is chosen here, based on its machinability, fatigue life, stiffness, weight, and cost. Detents help lock each binary leg into a discrete state, providing more accurate and repeatable positioning. They also eliminate the need for power while the BRAID is stationary.

E.3. Kinematics

E.3.1. Forward kinematics

A BRAID based system has further challenges, due to the complexity of its kinematics. A typical four-by-four homogeneous transformation matrix is formulated as a combination of a rotation matrix and a translation vector of one coordinate frame with respect to another. The kinematic variables are three rotational and three translational variables (six DOF). In general, given the four by four transformation matrix $A_{i-1,i}$, of the i^{th} coordinate frame with respect to the $i-1^{\text{th}}$ coordinate frame, one can derive the forward kinematics of the entire n -staged system. A_{0n} defines the forward kinematics from base to end-effector of the entire system and is given by:

$$A_{0n} = A_{0,1}A_{1,2}A_{2,3} \cdots A_{n-1,n} = \prod_{i=1}^n A_{i-1,i} \quad (\text{E-3})$$

where $A_{i-1,i}$ is given by:

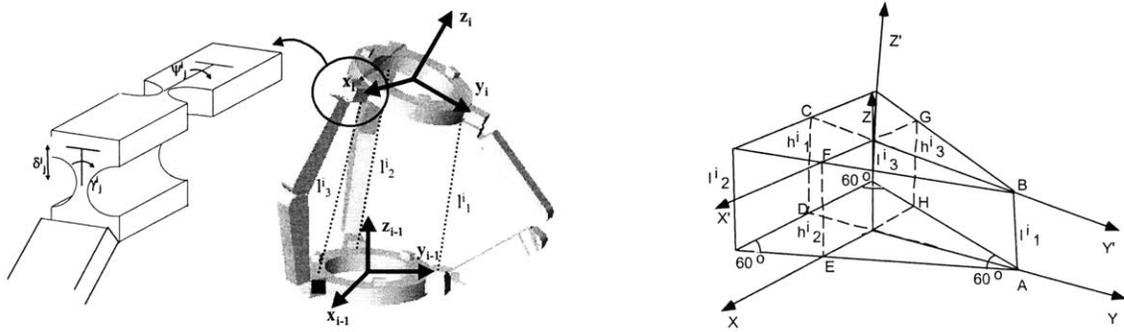
$$A_{i-1,i} = g_{01}^i g_{12}^i g_{23}^i = \begin{bmatrix} R_{xy}^i & \vec{0} \\ \vec{0}^T & 1 \end{bmatrix} \begin{bmatrix} x^i \\ y^i \\ z^i \\ 1 \end{bmatrix} = \begin{bmatrix} R_3 & x_e \\ \vec{0}^T & 1 \end{bmatrix} \begin{bmatrix} y_e \\ z_e \\ 1 \end{bmatrix} \quad (\text{E-4})$$

and rotation matrix R_{xy}^i is given by:

$$R_{xy}^i = \mathfrak{T}(\theta_x^i, \theta_y^i) \quad (\text{E-5})$$

where θ_x^i, θ_y^i are rotation of i^{th} frame about $i-1^{\text{th}}$ frame x, y axes

In a single parallel link stage of the BRAID system, the three legs are positioned about the vertices of two equilateral triangles (see Figure E-2). Based on the joint configuration of each leg, the single stage has three degrees of freedom—pitch (θ_P) and yaw (θ_Y) and a vertical (z) translation (coupling effects lead to non-independent motions in the x and y directions as well).



(a) Physical parallel link stage (b) diagrammatic representation
Figure E-2: i^{th} parallel link stage

In this formulation the leg lengths, l_j^i , are the control variables. The relationship between these leg lengths and the pitch, yaw, and vertical translation of the i^{th} coordinate frame with respect to the $i-1^{\text{th}}$ coordinate frame can be formulated. From Figure E-2:

$$\gamma_1^i = \theta_y^i \quad \gamma_2^i = \gamma_{2_x}^i + \gamma_{2_y}^i \quad \gamma_3^i = \gamma_{3_x}^i + \gamma_{3_y}^i \quad (\text{E-6})$$

where

$$\gamma_{2_x}^i = \sin(\pi/6)\theta_x^i \quad (\text{E-7})$$

$$\gamma_{3_x}^i = -\sin(\pi/6)\theta_x^i \quad (\text{E-8})$$

$$\frac{\delta_3^i \sin \gamma_{3_y}^i}{\cos(\pi/3)} = r \cos(\pi/6) - r \cos(\pi/6) \cos(\theta_y^i) - \delta_1^i \sin(\gamma_1^i) \quad (\text{E-9})$$

$$\frac{\delta_2^i \sin \gamma_{2_y}^i}{\cos(\pi/3)} = r \cos(\pi/6) - r \cos(\pi/6) \cos(\theta_y^i) + \delta_1^i \sin(\gamma_1^i) \quad (\text{E-10})$$

$$\psi_1^i = 0 \quad \psi_2^i = \psi_{2_x}^i + \psi_{2_y}^i \quad \psi_3^i = \psi_{3_x}^i + \psi_{3_y}^i \quad (\text{E-11})$$

where

$$\psi_{2_x}^i = -\frac{\pi}{6} \sin(\theta_x^i) \quad (\text{E-12})$$

$$\psi_{3_x}^i = \frac{\pi}{6} \sin(\theta_x^i) \quad (\text{E-13})$$

$$\psi_{2_y}^i = \frac{\pi}{6} \sin(\theta_y^i) \quad (\text{E-14})$$

$$\psi_{3_y}^i = \frac{\pi}{6} \sin(\theta_y^i) \quad (\text{E-15})$$

where r is the radius of the coupling disks in each parallel link stage. The deflection parameters $(\delta^i, \gamma^i, \psi^i)$ give us the coupled x^i and y^i translation of the i^{th} stage:

$$x^i = -\delta_1^i \sin(\gamma_1^i) \quad (\text{E-16})$$

$$y^i = -\left(\frac{r}{2} - \frac{\delta_3^i \sin \gamma_{3_x}^i}{\cos \pi/6} - \frac{r}{2} \cos \theta_x^i \right) \quad (\text{E-17})$$

The vertical projections (b_2^2 and b_3^2) of the leg lengths, l_2^i and l_3^i , are found from the following:

$$(l_2^i)^2 = a_2^2 + b_2^2 = \left(\frac{\delta_2^i \sin \gamma_{2_x}^i}{\cos \pi/6} \right)^2 + \left(\frac{\delta_2^i \sin \gamma_{2_y}^i}{\cos \pi/3} \right)^2 + b_2^2 \quad (\text{E-18})$$

$$(l_3^i)^2 = a_3^2 + b_3^2 = \left(\frac{\delta_3^i \sin \gamma_{3_x}^i}{\cos \pi/6} \right)^2 + \left(\frac{\delta_3^i \sin \gamma_{3_y}^i}{\cos \pi/3} \right)^2 + b_3^2 \quad (\text{E-19})$$

Figures E-3 and E-4 shows the projections of sections ABCD and EFGH defined in Figure E-2.

Using these projections, the relationship between the desired unknowns, z^i , θ_x^i , θ_y^i and the known link lengths l_1^i, l_2^i, l_3^i can be established.

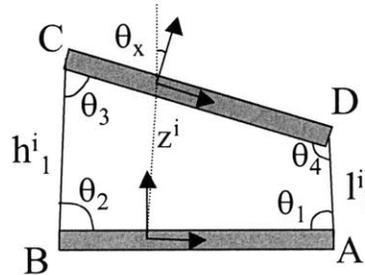


Figure E-3: Projection of section ABCD from Figure E-3

First from Figure E-3, h_1^i can be found as:

$$h_1^i \sin \theta_2 = \frac{b_2 + b_3}{2} \quad (\text{E-20})$$

A relationship between the angles θ_1 , θ_2 , θ_3 , θ_4 , and θ_x , can also be established:

$$\theta_3 = \text{acos} \left(\frac{(h_1^i)^2 + \left(\frac{3}{2}r\right)^2 - \beta^2}{2(h_1^i)\left(\frac{3}{2}r\right)} \right) \quad \text{where } \beta^2 = \left(\frac{3}{2}r - l_1^i \cos \theta_1\right)^2 + (l_1^i \sin \theta_1)^2 \quad (\text{E-21})$$

$$\theta_2 = \theta_2' + \theta_2'' \quad \text{where } \theta_2' = \text{asin} \left(\frac{l_1^i \sin \theta_1}{\beta} \right) \quad \text{and } \theta_2'' = \text{asin} \left(\frac{\frac{3}{2}r \sin \theta_3}{\beta} \right) \quad (\text{E-22})$$

$$\theta_x^i = \pi - \theta_3 - \theta_2 \quad (\text{E-23})$$

Also from Figure E-3 we have:

$$h_1^i \sin \theta_2 - l_1^i \sin \theta_1 = \frac{3}{2}r \sin \theta_x \quad (\text{E-24})$$

Using Equations E-18, 19, 20, 21, 22, 23 and Equation E-24, we get one equation in two unknowns (θ_x and θ_y). A similar equation can be derived using the geometry in Figure E-4.

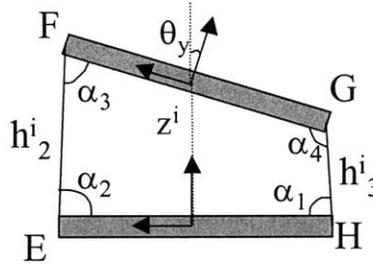


Figure E-4: Projection of section EFGH from Figure E-3

From Figure E-4, h_2^i and h_3^i are found as follows:

$$h_2^i \sin \alpha_2 = \frac{2}{3}b_2 + \frac{1}{3}l_1^i \sin \theta_1 \quad (\text{E-25})$$

$$h_3^i \sin \alpha_1 = \frac{2}{3}b_3 + \frac{1}{3}l_1^i \sin \theta_1 \quad (\text{E-26})$$

Once again, a relationship between the angles α_1 , α_2 , α_3 , α_4 , and θ_y , can be established:

$$\alpha_3 = \text{acos} \left(\frac{(h_2^i)^2 + (2r \tan \pi/6)^2 - \beta^2}{2(h_2^i)(2r \tan \pi/6)} \right) \quad \text{where } \beta^2 = (2r \tan \pi/6 - h_3^i \cos \alpha_1)^2 + (h_2^i \sin \alpha_1)^2 \quad (\text{E-27})$$

$$\alpha_2 = \alpha_2' + \alpha_2'' \quad \text{where } \alpha_2' = \text{asin} \left(\frac{h_3^i \sin \alpha_1}{\beta} \right) \quad \text{and } \alpha_2'' = \text{asin} \left(\frac{2r \tan \pi/6 \sin \alpha_3}{\beta} \right) \quad (\text{E-28})$$

$$\theta_y^i = \pi - \alpha_3 - \alpha_2 \quad (\text{E-29})$$

Also from Figure E-4 we have:

$$h_2^i \sin \alpha_2 - h_3^i \sin \alpha_1 = \frac{2}{\sqrt{3}} r \sin \theta_y \quad (\text{E-30})$$

Using Equations E-18, 19, 25, 26, 27, 28, 29 and Equation E-30, we get one equation in two unknowns (θ_x and θ_y). Equations E-24 and E-30 give two independent equations in two unknowns. However, both are highly non-linear transcendental equations and can only be solved numerically. A Newton-Raphson algorithm is implemented to solve for the unknowns, θ_x and θ_y .

The vertical translation can be solved using solutions for θ_x and θ_y and equation E-31:

$$z^i = h_2^i \sin \alpha_2 - \frac{1}{\sqrt{3}} r \sin \theta_y = h_1^i \sin \theta_2 - \frac{1}{2} r \sin \theta_x \quad (\text{E-31})$$

This is the general solution for the BRAID system for the given leg lengths. Hence, $A_{i-1,i}$ is only a function of the variable leg lengths of the i^{th} stage ($l_{1,2,3}^i$). However, for a binary system, since only two leg lengths need to be considered, given one of the two lengths (i.e. the maximum or open leg length) the second one is then a well defined function of the first. The minimum value of the leg length l_1^i , given the maximum value of h_1^i (from Figure E-3) is found by solving:

$$r = (r + l_1^i) \cos \theta_1 = (r + l_1^i) \cdot \left(\frac{(3r/2)^2 + (3r/2 + l_1^i)^2 - (h_1^i)^2}{3r(3r/2 + l_1^i)} \right) \quad (\text{E-32})$$

The minimum value of the leg length h_1^i given the maximum value of l_1^i is found by solving:

$$\frac{r}{2} = \left(\frac{r}{2} + h_1^i \right) \cos \theta_2 = \left(\frac{r}{2} + h_1^i \right) \cdot \left(\frac{(3r/2)^2 + (3r/2 + h_1^i)^2 - (l_1^i)^2}{3r(3r/2 + h_1^i)} \right) \quad (\text{E-33})$$

The minimum leg length h_2^i given max h_3^i (from Figure E-4) is found by solving:

$$\frac{r}{\sqrt{3}} = \left(\frac{r}{\sqrt{3}} + h_2^i \right) \cos \theta_3 = \left(\frac{r}{\sqrt{3}} + h_2^i \right) \cdot \left(\frac{\left(2 \frac{r}{\sqrt{3}} \right)^2 + \left(2 \frac{r}{\sqrt{3}} + h_2^i \right)^2 - (h_3^i)^2}{2 \left(2 \frac{r}{\sqrt{3}} \right) \cdot \left(2 \frac{r}{\sqrt{3}} + h_2^i \right)} \right) \quad (\text{E-34})$$

Since every leg in the system can be in only one of two states (binary), each leg length can have

only one of two values. Hence, each term of $A_{i-1,i}$ can have only 8 different discrete values corresponding to the 8 possible states of a single BRAID stage. Figure E-5 shows the workspace generated for a 5 stage BRAID element. The workspace consists of 2^{15} unique states.

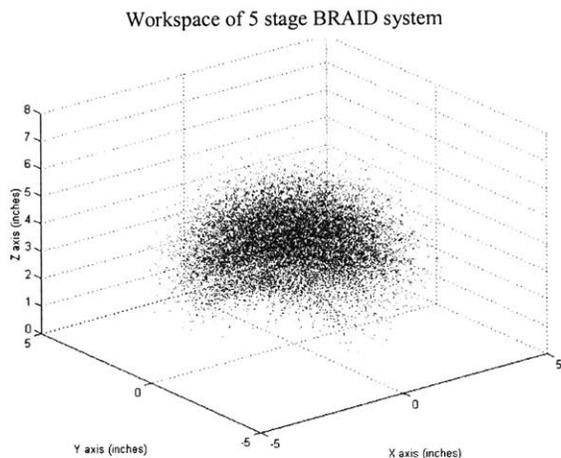


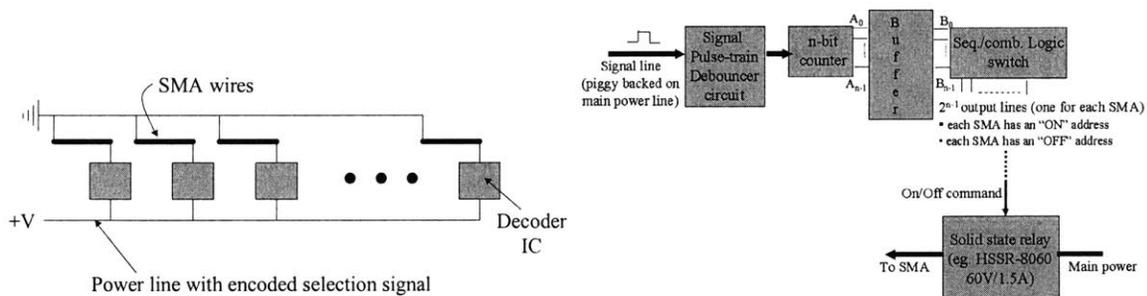
Figure E-5: Workspace of 5 stage BRAID element (BRAID element base center=origin)

E.3.2. Inverse kinematics

The inverse kinematics problem cannot be expressed in a closed form solution. Exhaustive search methods may prove appealing for systems with few stages (less than 5), but become impractical for larger systems. For every additional stage there is about an order of magnitude increase in the number of states in the workspace. Two possible efficient search methods for the inverse kinematics problem are explored: a genetic search and a combinatorial heuristic search algorithm. The search metric is to minimize the error between the end-effector and desired pose. Performance of the two search methods is quantified on a stochastic basis using a Monte Carlo method. 1000 target points are selected randomly within the volume of a binary workspace cloud. Each target is given a random orientation. The inverse kinematics for each target point is then solved and the solution times, displacement and angular errors are computed and recorded. These results have been previously published [Lichter-2].

E.4. Actuator control

A hyper-redundant BRAID would need a large number of actuators. In future, the actuators would be expected to be polymer-based smart materials, such as conducting polymers and electrostrictive polymers [Madden, Pelrine]. In the near term, shape memory alloys (SMAs) are being used as surrogate muscle actuators. These alloys can be used as actuators, as the ratio of the deformation stress to the recovery stress can be higher than 10 to 1 [Gilbertson]. To actuate the BRAID element, the actuators need to be triggered selectively, as required by the inverse kinematics. Conceptually, such a binary control is simple requiring no sensory feedback. However, a large number of actuators can rapidly make the physical realization of such a system difficult, if each actuator requires unique power supply lines. A multitude of wiring introduces possibility for error and would result in additional weight and volume, large external forces, and complexity. The BRAID uses a more compact and efficient form of supplying power and control (see Figure E-7). A common power line and ground are provided to all the actuators. Each actuator has a small “decoder” chip that can be triggered into either binary state by a carrier signal “piggybacked” on the power line. The carrier signal is a sequence of pulses that identifies a unique address in the form of a binary word for the actuator that requires toggling. This architecture reduces the wiring of the entire system to only two wires (see Figure E-7).



(a) Overview of actuator control electronics (b) Power/control bus decoder architecture
Figure E-7: SMA power and control bus

The signal (consisting of a sequence of pulses) is extracted from the main power line by a

simple Thevenin voltage divider. This is driven through a de-bouncer circuit to remove noise. A counter adds the number of pulses in the pulse train. The output of the counter forms the binary address of the SMA to be triggered (on or off). A buffer between the counter output and the latching circuit prevents intermediate count values to accidentally trigger the wrong SMA.

The buffer can be implemented in several ways. Two possibilities are described here. First, a RC delay circuit can be used. The resistor (R) and capacitor (C) values can be changed so as to adjust the rise time ($t = 0.63 R C$), to allow sufficient time for the entire signal to be processed by the counter before TTL (or CMOS) voltage thresholds are reached. Hence, the output of the counter affects the latching circuit only after a delay equal to the rise time. To be practical (allowing for variations in signal transfer time), this may require large values for the resistance and capacitance terms, making the circuit bulky.

A second option for the buffer is to use a series of flip-flops triggered by an end-of-pulse-train flag, added to the signal line in the form of a voltage spike. By introducing another voltage divider between the signal line and the buffer, all address pulses may be ignored as they would lie below the TTL (or CMOS) voltage threshold level. The buffer would be triggered by the end-of-pulse-train flag. For example, for TTL thresholds (1.3V), the signal line with address pulses would peak at 2V. A 2.5:1 voltage divider would force the buffer to see a 0.8V (<TTL thresh.) signal. The buffer trigger pulse would peak at 5V on the signal line. This would force the buffer to see a 2V (>TTL thresh.) signal, thus triggering it. This end-of-pulse-train flag can also be used to reset the counter. In testing, the second method proves to be more reliable and is implemented. After the buffer acquires the signal, a latching circuit decodes the address using standard combinatorial logic, and latches it to the appropriate actuator trigger line using sequential logic (see Figure E-8). For simplicity in fabrication each decoder chip can be identical, having as

many output lines as there are actuators. However, for any decoder chip, only one line is connected, thus providing unique addressing.

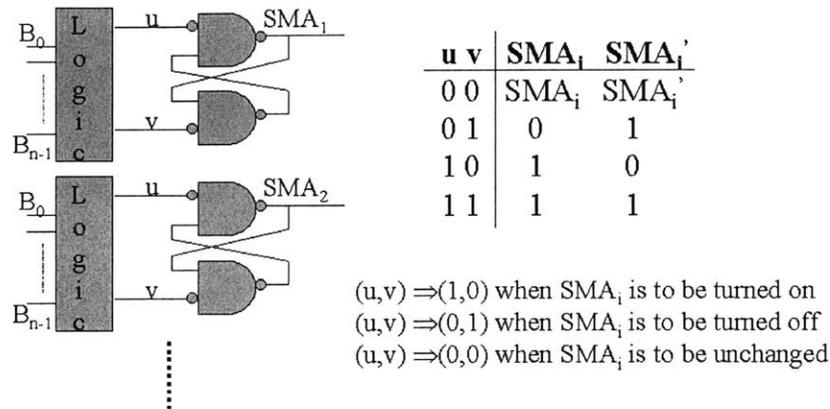


Figure E-8: SMA power bus address decoding and latching electronics

E.5. Experimental system

The experimental system constructed is shown in Figure E-9. It consists of five parallel link stages. With binary control this structure has $2^{3 \times 5}$ (or 32768) possible states giving the device suitable freedom for a number of applications. For other applications this could be extended to 10 or 20 stages giving $2^{3 \times 20}$ (approximately 10^{18}) possible states. While this closely approximates a continuous workspace, it leads to some interesting inverse kinematic problems due to the hyper-redundancy of the system, (see Section E.3).

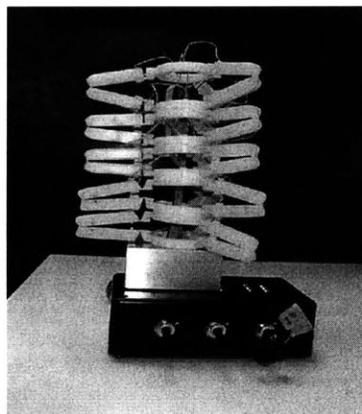


Figure E-9: Experimental platform of BRAID