

ORNL

OAK RIDGE  
NATIONAL  
LABORATORY

LOCKHEED MARTIN

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

MANAGED BY  
LOCKHEED MARTIN ENERGY SYSTEMS, INC.  
FOR THE UNITED STATES  
DEPARTMENT OF ENERGY

UCN-13673 (36-6-85)

RECEIVED  
APR 02 1996  
OSTI

Crusader Automated Docking  
System Phase III Report

W. B. Jatko  
J. S. Goddard  
R. K. Ferrell  
S. S. Gleason  
J. S. Hicks  
V. K. Varma

INSTRUMENTATION  
&  
CONTROLS DIVISION

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from the Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831; prices available from (615) 576-8401, FTS 626-8401.

Available to the public from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, VA 22161.

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Instrumentation and Controls Division

**CRUSADER AUTOMATED DOCKING SYSTEM PHASE III REPORT**

W. B. Jatko  
J. S. Goddard  
R. K. Ferrell  
S. S. Gleason  
J. S. Hicks

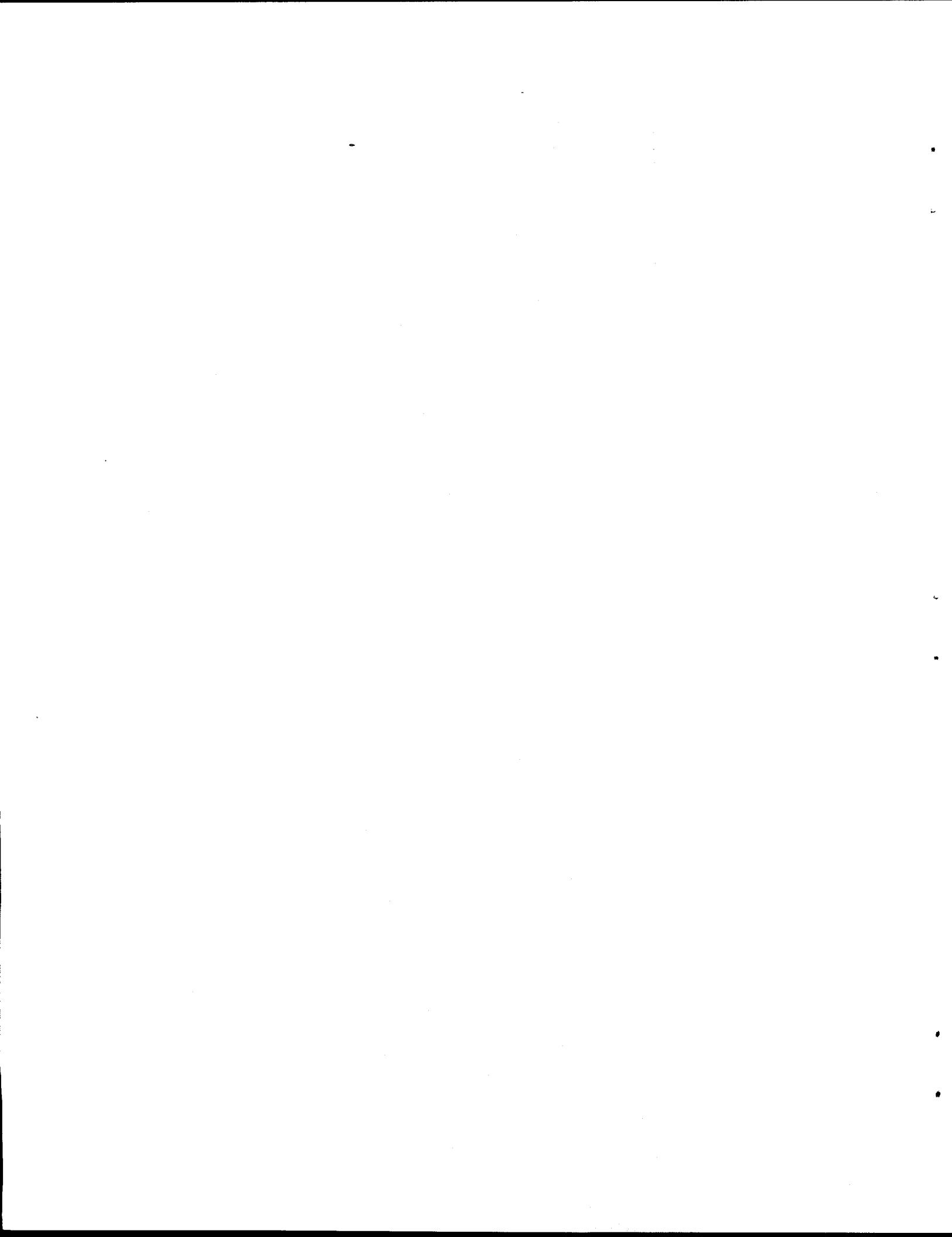
Instrumentation and Controls Division

V. K. Varma  
Robotics and Process Systems Division

Date Published — March 1996

Research sponsored by the Army's Project Manager, Crusader under the Interagency Agreement 1892-A078-A1 between the Department of Energy and the Armament Research, Development and Engineering Center at Picatinny Arsenal.

Prepared by  
OAK RIDGE NATIONAL LABORATORY  
Oak Ridge, Tennessee 37831-6285  
managed by  
LOCKHEED MARTIN ENERGY RESEARCH CORP.  
for the  
U.S. DEPARTMENT OF ENERGY  
under contract DE-AC05-96OR22464



## **DISCLAIMER**

**Portions of this document may be illegible  
in electronic image products. Images are  
produced from the best available original  
document.**



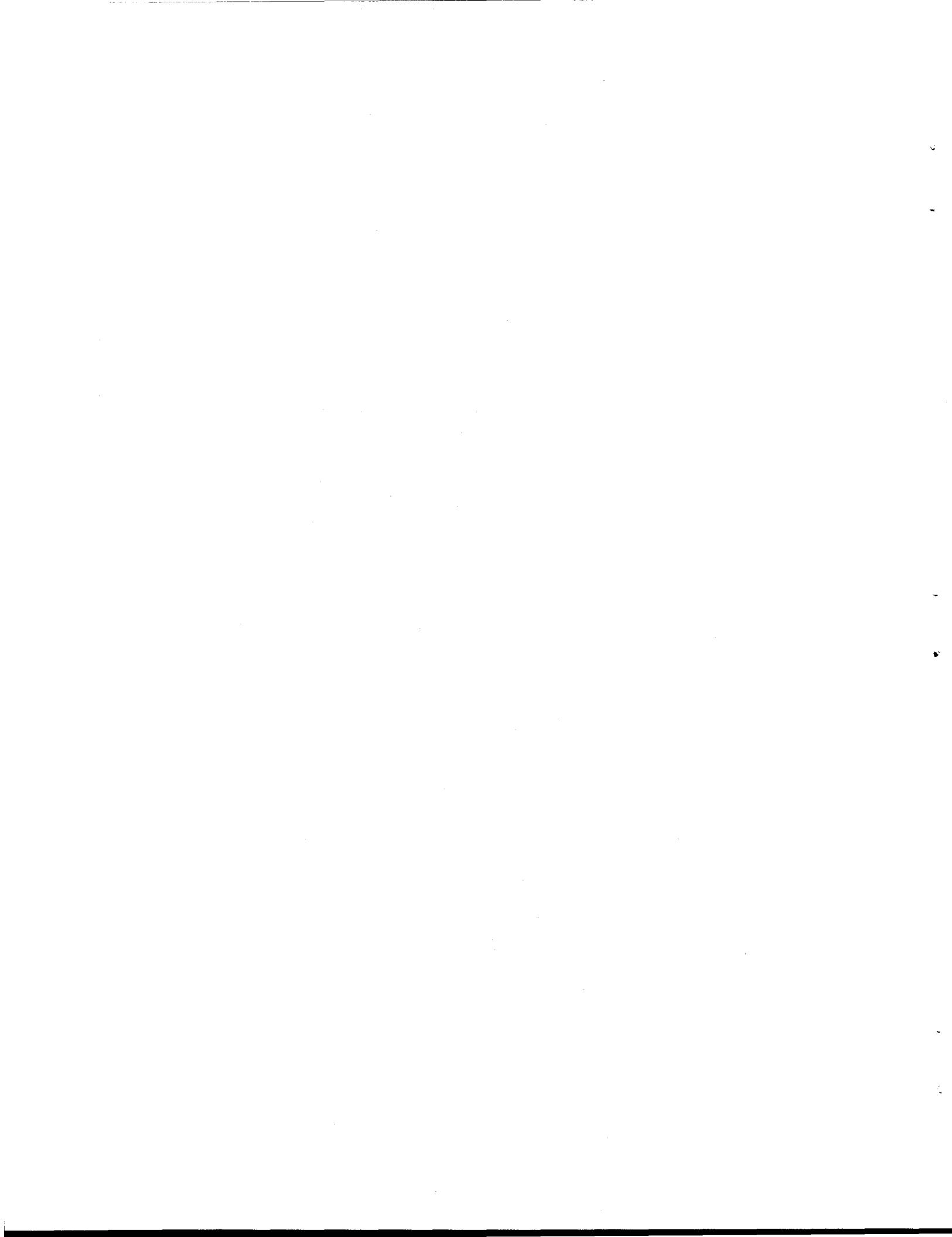
## CONTENTS

LIST OF FIGURES .....	v
ABSTRACT .....	vii
EXECUTIVE SUMMARY .....	ix
1. INTRODUCTION .....	1
1.1 SCOPE .....	1
1.2 ACKNOWLEDGMENTS .....	1
1.3 CRUSADER PROGRAM .....	1
1.4 ORNL AUTODOCKING PROGRAM .....	1
2. FUNCTIONAL DESCRIPTION .....	3
2.1 DEFINITION OF DOCKING .....	3
2.2 CADS OVERVIEW .....	4
3. OPERATIONAL DESCRIPTION .....	8
3.1 TARGET ILLUMINATION .....	8
3.2 IMAGE FORMATION .....	9
3.2.1 Optical Calibration .....	9
3.3 TARGET ACQUISITION .....	9
3.4 POSE DETERMINATION .....	12
3.4.1 Perspective Projection .....	12
3.4.2 Feature Matching .....	14
3.4.3 Pose Calculation .....	15
3.5 POSE INTERFACE .....	16
3.5.1 Gauge Position Indicators .....	17
3.5.2 Graphical Pose Widget Display Window .....	18
3.5.3 Live Video Window .....	20
3.5.4 Control Panel .....	20
3.6 REFERENCES .....	20
4. RESOURCE REQUIREMENTS .....	21
4.1 PROCESSING .....	22
4.2 COMPONENT LIST .....	22
4.3 POWER .....	22
4.4 VOLUME .....	23
4.5 WEIGHT .....	23
4.6 OPERATING TEMPERATURE .....	23
4.7 STORAGE TEMPERATURE .....	23
4.8 HUMIDITY (NONCONDENSING) .....	23
5. SOFTWARE ARCHITECTURE .....	24
5.1 ARCHITECTURE .....	24
5.2 ESTIMATED CODE .....	24
5.3 TARGET ILLUMINATION .....	24

5.4 TARGET ACQUISITION .....	25
5.5 POSE DETERMINATION .....	27
5.5.1 Initialization .....	27
5.5.2 Feature Point Preprocessing .....	28
5.5.3 Unit Transformations .....	28
5.5.4 Convex Quadrilateral Grouping .....	28
5.5.5 Initial Pose Calculation .....	28
5.5.6 Match Verification .....	28
5.5.7 Pose Calculation .....	29
5.5.8 Pose Reporting .....	29
5.6 POSE INTERFACE .....	29
5.6.1 Initialization of GUI .....	29
5.6.2 Pose Calculation and Interval Timer .....	30
5.6.3 GUI Update .....	30
5.6.4 Operation .....	30
5.6.5 Network Interface .....	31
<b>6. PERFORMANCE SUMMARY .....</b>	<b>32</b>
6.1 DESCRIPTION OF ENVIRONMENTAL CONDITIONS .....	32
6.2 UPDATE RATE .....	32
6.3 MEASUREMENT ACCURACY .....	32
6.4 CONTINUOUS OPERATION .....	33
6.5 ADVERSE WEATHER .....	33
<b>7. DIAGNOSTICS, MAINTAINABILITY, AND PRODUCIBILITY .....</b>	<b>34</b>
7.1 DIAGNOSTICS .....	34
7.2 MAINTAINABILITY .....	34
7.3 PRODUCIBILITY .....	34
<b>8. INTERFACE DEFINITIONS .....</b>	<b>35</b>
<b>9. TECHNOLOGY RISK ASSESSMENT .....</b>	<b>36</b>
9.1 RECOMMENDATIONS .....	36
9.1.1 Application Guidelines .....	36
9.1.2 Protection from the Environment .....	36
9.1.3 Packaging .....	37
APPENDIX D.2: PARKING ASSIST SIMULATION .....	38
APPENDIX D.3: MAADS AUTODOCKING DEMONSTRATION .....	42

## LIST OF FIGURES

Fig. 1.	A candidate conceptual design for artillery resupply.....	2
Fig. 2.	Elements of an autonomous docking system.....	3
Fig. 3.	CADS hardware block diagram.....	6
Fig. 4.	Diagram mapping CADS functions to corresponding hardware modules.....	7
Fig. 5.	Block diagram of the digital oscillator and lamp driver.....	8
Fig. 6.	Variation in pixel intensity over sequential video frames.....	10
Fig. 7.	Block diagram of the digital demodulator.....	11
Fig. 8.	Perspective model for pose determination.....	13
Fig. 9.	Photograph of the operator's GUI screen for the CADS.....	17
Fig. 10.	Photograph of the graphical pose widget.....	19
Fig. 11.	The CADS is self-contained in the 21-slot VME chassis. ....	21
Fig. 12.	Signal flow diagram for the digital demodulator.....	26
Fig. 13.	Photograph of the input signal to the demodulator and the resultant output.....	27
Fig. 14.	The sensor pose coordinate system. ....	28
Fig. 15.	GUI flow chart .....	29
Fig. 16.	Network interface flow chart.....	31
Fig. 17.	Pose measurement statistics for simulated environmental conditions.....	33
Fig. 18.	CADS interface diagram. ....	35
Fig. A.1.	Volume describing the extent of the boom's reach.....	38
Fig. A.2.	Top view diagram of the RSV docked with the SPH.....	39
Fig. A.3.	Screen output of parking simulation program.....	40
Fig. A.4.	Side view of the MAADS boom approaching the simulated SPH docking port.....	42
Fig. A.5.	Flow chart of the MAADS control architecture .....	43
Fig. A.6.	View facing the simulated SPH docking port. ....	43
Fig. A.7.	View looking into the MAADS boom. ....	43
Fig. A.8.	MAADS boom about to dock with the SPH port.....	44



## ABSTRACT

The U.S. Army is developing the next generation of battlefield artillery vehicles, including an advanced, self-propelled howitzer and a companion resupply vehicle. The resupply vehicle is intended to rendezvous with the howitzer near the battlefield and to upload ammunition to the howitzer. The Army has recommended that the vehicles incorporate robotics to increase safety, by allowing the crew to remain inside their vehicles during resupply operations.

Oak Ridge National Laboratory has developed an autonomous docking system for a 6-D.F. robotic, ammunition transfer arm. The docking system augments the operator's abilities by determining the position and orientation (pose) of a docking port. The pose is the location of the x, y, and z reference axes in 3-D space; and the orientation is the rotations — roll, pitch, and yaw — about those axes. By precisely determining the pose of the docking port, the robot can be instructed to move to the docking position without operator intervention.

The system uses a video camera and frame grabber to digitize images of the special docking port. Custom algorithms were developed to recognize the port in the camera image, to determine the pose from its image features, and to distribute the results to the robot control computer. The system is loosely coupled to the robot and can be easily adapted to different mechanical configurations. The system has successfully demonstrated autonomous docking on a 24-in. table-top robot and a 12-ft ammunition resupply robot. The update rate, measurement accuracy, continuous operation, and accuracy with obstructed view have been determined experimentally.



## EXECUTIVE SUMMARY

### INTRODUCTION

The Crusader program is the Army's project to develop the next generation artillery system. There are two vehicles in the Crusader concept, a self-propelled howitzer (SPH) and its companion ammunition resupply vehicle (RSV). In a typical resupply mission, the RSV leaves the ammunition depot to rendezvous with the SPH near the battlefield. The RSV parks just behind the SPH and extends a robotic resupply arm to the SPH. The resupply arm mates with the docking port and transfers ammunition to the SPH. When the transfer is complete, the arm is retracted and the RSV returns to the depot. This chapter summarizes the development of the Crusader Automated Docking System (CADS) at Oak Ridge National Laboratory (ORNL). The CADS is a laboratory prototype intended to demonstrate the feasibility of autonomous docking (autodocking) for Crusader resupply missions.

The ORNL autodocking program began in June 1993 with a 3-month feasibility study to determine if autonomous docking was possible using commercially available hardware. ORNL engineers recognized that the additional hardware needed to support autodocking on a robotic arm was small. The engineers devised an architecture that uses a digital image processing system, called the CADS, to calculate the position and orientation (pose) of a special docking port.

The CADS itself does not initiate or control any robotic arm movements during autodocking. Its sole function is to supply the port's pose to the external control system, which uses the pose to move the robotic arm. By separating the path planning process from the motion control function, the CADS can be easily adapted to different mechanical arms.

The CADS prototype was initiated in February 1994, using a single video camera and a unique target to simulate the docking port. The first remote pose measurement was demonstrated in June 1994. The CADS, modified for increased accuracy and reliability, was completed in June 1995. The CADS was mounted on a small, commercial robot in an experiment to emulate resupply docking. The demonstration of small-scale autodocking proved that the CADS concept was sound. Next, the CADS was integrated with a full-scale artillery ammunition transfer arm and demonstrated fully autonomous docking and resupply on November 28, 1995.

### CADS OVERVIEW

The CADS is a vision-based, noncontact measurement system that calculates the pose of an artillery receiving port with respect to its camera. The CADS has five main subsystems: (1) target illumination, (2) image formation, (3) target acquisition, (4) pose determination, and (5) pose interface, summarized below.

**Target illumination** — An array of light-emitting diodes (LEDs) marks the structural vertices of the docking port in the video image. The target pose is computed from the image coordinates of these markers.

**Image formation** — The video camera captures a sequence of 20 images containing the docking port and its surrounding area. The image sequence is the raw material from which the CADS computes the port pose. A conventional video camera is currently used, but other models, including infrared, could be substituted.

**Target acquisition** — The LED markers are filtered from the image sequence and analyzed to verify that each object is a valid target marker. A list of coordinates is generated from the valid target markers in the filtered image.

**Pose determination** — The target's pose is calculated from the list of marker coordinates. The pose is found by aligning a computer model of the target to match the target markers seen by the camera. Because the target has more features than needed, an accurate pose can be measured even if some markers are missing or obscured.

**Pose interface** — The CADS interacts with the outside world through the pose interface subsystem. Pose data can be sent to a user-selectable computer, typically the robot control system, and to a local display screen, over a standard network connection. When the control system requests pose data, the CADS returns the current target position and orientation. A graphical interface provides an operator control panel on the local video monitor. The visual display has six simulated gauges to indicate the pose parameters, while additional video windows contain a control panel, a live image from the camera, and a unique docking icon that indicates all six pose parameters in a single object.

A series of experiments was conducted to characterize the system performance under adverse conditions, such as rain and fog. The update rate, measurement accuracy, continuous operation, and obstructed view have been characterized to date. It takes less than 1.1 s to make a new pose measurement. The measurement error was less than 2% of the expected value for all test conditions. The system has run continuously, making over 235,000 pose measurements, without a failure. The system will operate if the target is partially obscured, although the accuracy may be degraded.

## AUTODOCKING DEMONSTRATIONS

In addition to the tests to quantify its performance, the CADS also underwent a two-phased operational test designed to show autodocking under ideal conditions. In the first test, small-scale autodocking, the CADS was mounted onto a commercial 6-D.F. robot manipulator arm with a 24-in. reach and an internal control system. A miniature port with six LED markers was fabricated to simulate the SPH port. In the autodocking sequence, the CADS computed an initial estimate of the port's position and orientation. Using this pose estimate, the control system moved the arm to a point lying on the port's center axis and 10 in. away from contact with the port. From the new position, a second, more accurate pose measurement was made. The arm was then docked with the port using the second pose measurement.

In the second test, full-scale autodocking, the CADS was mounted onto the ORNL ammunition transfer boom. The ORNL boom is a custom-designed 6-D.F. robot articulated conveyer with a 12-ft reach. The boom has a complementary receiving port with eight LED markers. The autodocking sequence used in the full-scale demonstration was identical to that in

the small-scale demonstration. The elapsed time in autodocking is limited by the distance to the port and the velocity of the robot. For full-scale autodocking, the typical time from start to completed docking was 40 s.

## PARKING SIMULATION

A software tool was developed to better understand problems likely to arise during a resupply operation. Because the joints of the resupply arm have fixed mechanical limits, the arm can reach only a finite number of points, which defines the docking envelope of the arm. The port must be within the docking envelope, or the arm will not be able to couple mechanically to the port. The situation is compounded because the envelope varies with the orientation of the port. This could make parking the RSV difficult if the driver has to guess when RSV heading and distance will allow docking.

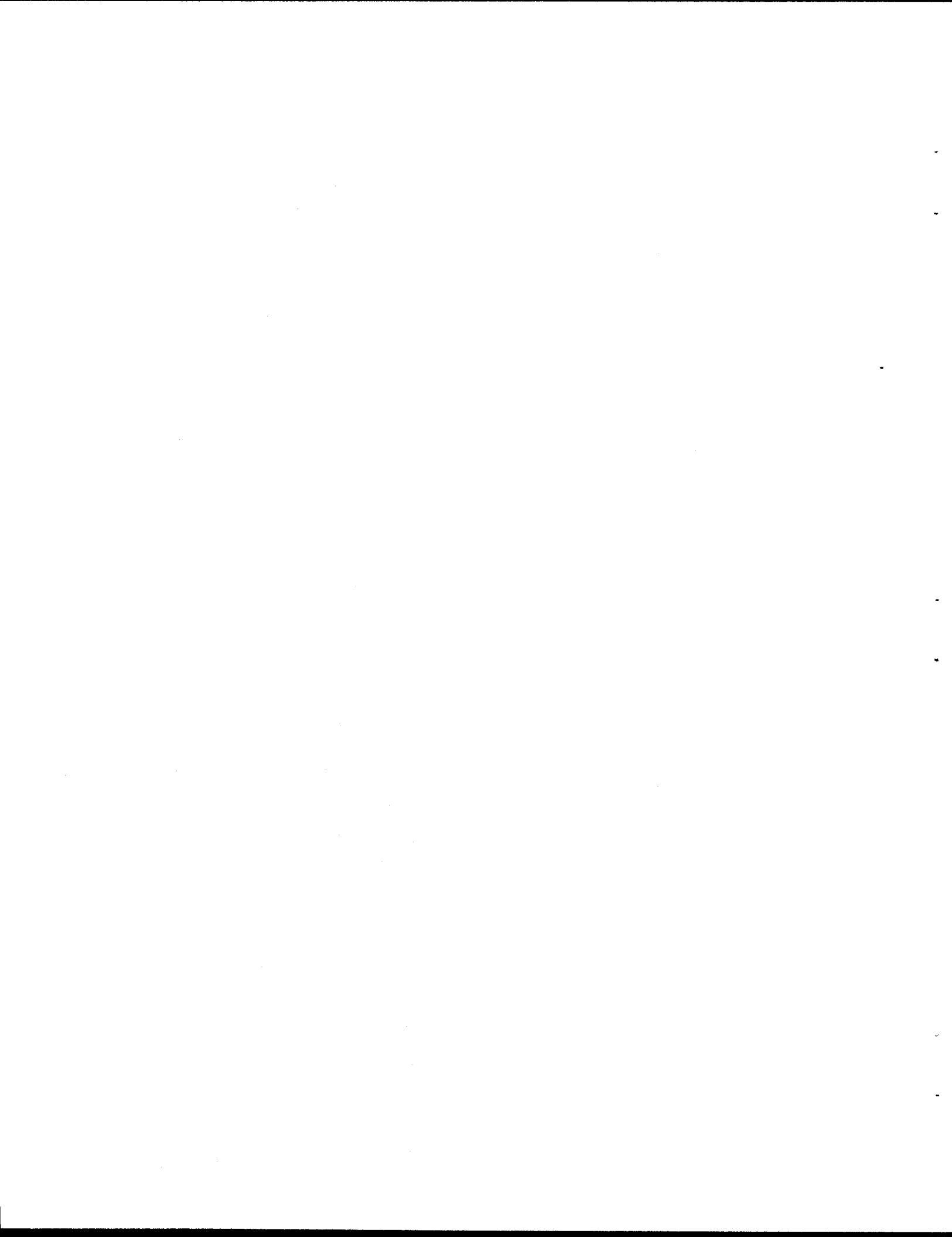
At present, the software tool only runs on a workstation but could be incorporated into a future CADS unit to assist the crew in parking the RSV within reach of the resupply arm. This feature would significantly improve the driver's ability to position the vehicle for docking and hence reduce the total time needed to reload the SPH.

## RECOMMENDATIONS

The CADS engineers believe that the CADS laboratory prototype has accomplished its goal to confirm that autodocking for Crusader resupply missions is possible. The present CADS has been successful in all laboratory tests but is not suitable for extended operation under the expected operating conditions. The engineers recommend that the next logical step in the CADS program is to develop a field prototype to continue the evolution toward eventual installation on a military vehicle.

The field prototype would be optimized for pose determination functions. The present CADS is a general-purpose configuration intended for algorithm development. The field prototype could be built with custom hardware to minimize the space and power requirements. The field prototype would be hardened for environmental extremes that would be expected in field use. The camera and optics will be encased in a protective enclosure to protect them from dust and moisture.

The present CADS uses illuminating markers attached on or near the port to identify the receiving port. Although the markers are low power and would not be active except during docking, they impose an additional requirement on the SPH. The markers could be removed by converting the CADS to passive port recognition. Passive recognition would require replacing the existing 2-D camera with a 3-D version and developing a new algorithm to interpret the 3-D image data. This change to the CADS is significant, but it would eliminate the need for any active components on the SPH during autodocking.



## 1. INTRODUCTION

### 1.1 SCOPE

This report documents the Crusader Automated Docking System (CADS) as of November 1995. CADS is a laboratory prototype implemented at the Oak Ridge National Laboratory (ORNL) which is intended to demonstrate the feasibility of autonomous docking under controlled conditions. Additional development would be required to deploy the system for military applications.

### 1.2 ACKNOWLEDGMENTS

The authors wish to thank the U.S. Army's Project Manager, Crusader, for their support of the ORNL autonomous docking program. This program has demonstrated that "hands-off" docking is feasible given the present level of technology. The program has been a collaborative effort between the Instrumentation and Controls Division and the Robotics and Process Systems Division at ORNL.

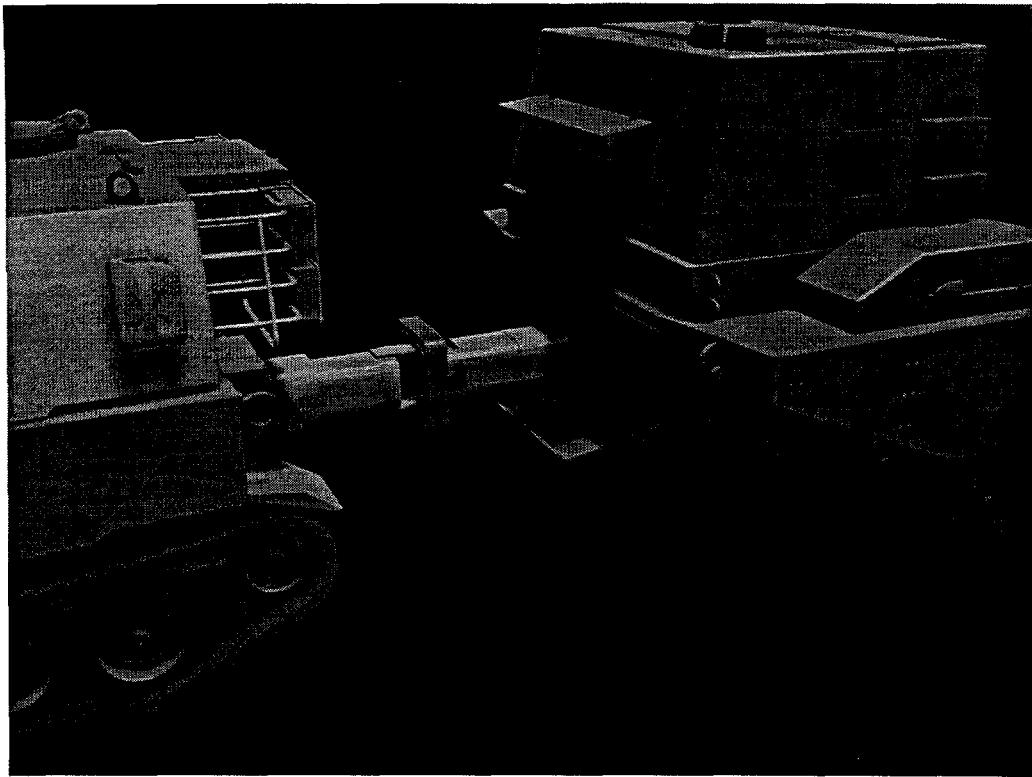
### 1.3 CRUSADER PROGRAM

The U.S. Army is developing the next generation of battlefield artillery vehicles. The development program includes an advanced, self-propelled howitzer (SPH) and a companion resupply vehicle (RSV). The RSV is intended to rendezvous with the SPH near the battlefield and to replenish ammunition, fuel, and other consumable material.

The Army has recommended that the vehicles incorporate new and innovative technologies to improve performance and safety. One conceptual design proposes a robotic resupply boom on the RSV to upload supplies to the SPH, illustrated in Fig. 1. The resupply boom would normally be retracted within the RSV during transit. When the RSV is within docking range of the SPH, the boom would be extended and linked with a receiving port on the SPH. To reduce the crew's exposure to small arms fire or nuclear, biological, and chemical hazards, they would remain inside the RSV during the resupply operation.

### 1.4 ORNL AUTODOCKING PROGRAM

The ORNL autonomous docking program began in June 1993 with a 3-month feasibility study to determine if autonomous docking was possible using existing hardware. Based on the information gathered in this study, a configuration with a single video camera, mounted on the end of the resupply arm, along with a known, unique target, mounted on or near the docking port, was recommended for further development. The initial CADS prototype was built in 1994 using the recommended configuration. The basic measurement functionality for autonomous docking was demonstrated in June 1994. The CADS was further refined to increase the accuracy and reliability. The improved CADS was completed in June 1995, when it successfully demonstrated autonomous docking with a small-scale robotic arm. In October 1995 the CADS was integrated with the full-scale robotic ammunition transfer arm. Full-scale autonomous docking was demonstrated on November 28, 1995.



**Fig. 1. A candidate conceptual design for artillery resupply.** The RSV (right) is resupplying the SPH (left) through the robotic transfer arm that bridges the two vehicles.

## 2. FUNCTIONAL DESCRIPTION

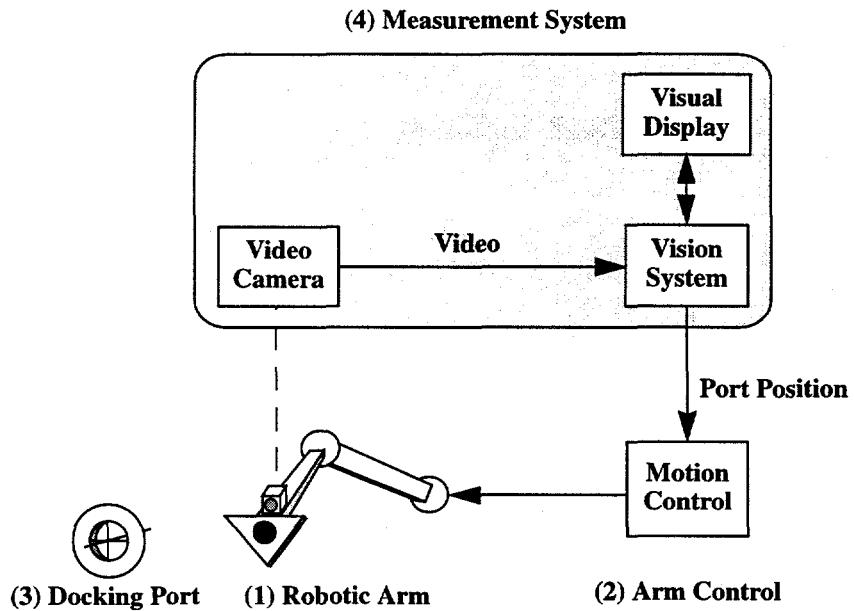
### 2.1 DEFINITION OF DOCKING

Before beginning a functional description of the CADS, it is useful to examine the components needed for autodocking. We first define docking to be the procedure of linking a resupply vehicle with a target vehicle by means of a resupply arm. A well-known example of docking is aircraft refueling whereby one aircraft is refueled by another during flight.

A docking system requires three essential elements: (1) a robotic arm that transfers the supplies to the docking port, (2) an arm control system to move the arm to the desired location, and (3) a docking port that mechanically interfaces with the arm. A block diagram of a docking system is shown in Fig. 2.

When an operator controls the motion of the arm, we call this manual docking. Manual docking is greatly dependent on the skill of the operator to manipulate the arm into position. Computer simulations at the National Aeronautics and Space Administration<sup>1,2</sup> (NASA) have shown that computer-assisted docking can improve the ability of the operator to dock safely and quickly.

We can expand our docking system to include computer-assisted, or autonomous, docking by adding a measurement system (item 4 in Fig. 2) that determines the location of the docking port.



**Fig. 2. Elements of an autonomous docking system.** The four major components needed to autonomously dock are shown.

The autonomous docking system could be designed to have several levels of autonomy from the highest level of controlling the robotic arm without operator involvement to the lowest level of serving as a simple video monitor.

The lowest level of the hierarchy is manual operation with vision assist. In this mode, the operator would be responsible for all aspects of arm control. The autodocking system would be inactive except for the image acquisition sensor, which would show the docking approach as seen from the arm-mounted camera.

The next level is a semiautonomous mode where the operator is in the loop. In this mode, the operator would be in full control of the arm and would use the autodocking system to provide direction cues. This could take the form of direction arrows on a cathode-ray tube (CRT) screen advising the operator to move in a particular direction in order to align the end effector of the arm with the docking port.

The highest level of the hierarchy is complete autonomy intended for unsupervised operation. At this level, the operator has no involvement after the autodocking sequence has been initiated. The autodocking system is responsible for identifying the docking port, determining the distance to the port and the angular orientation of the port, and supplying the six pose parameters (x, y, and z coordinates and roll, pitch, and yaw angles) to the control system. The control system then calculates the path to the port and moves the robotic arm along that path. Optionally, the arm may stop at intermediate points along the path to allow the autodocking system to recompute the pose.

## 2.2 CADS OVERVIEW

In the Crusader application, the CADS is the measurement system component in Fig. 2. The CADS provides positional information that enables the robotic control system to precisely align the arm with the docking port. In this architecture, the CADS itself does not perform any control function during autodocking. In order to control the robot arm, the CADS would have to know the kinematics of the arm being driven. The CADS was designed to be independent of the control functions in order to be easily adapted to different mechanical arms.

The CADS is a vision-based, noncontact measurement system designed to accurately measure the position and orientation (pose) of an artillery receiving port. The system measures the pose of the target port with respect to a vision sensor. Robustness and measurement quality estimates have been incorporated into the design to give high accuracy as well as to prevent the reporting of large position errors.

The CADS was intended to ultimately be used in a military vehicle. As such, the requirements for field operation were considered during the CADS design and implemented to the extent possible. Some requirements, such as power, weight, and volume, were undefined at the time and were not considered. The CADS was not built for harsh environments, but the CADS design could be hardened for all-weather operation. For example, it uses a low-cost video camera, but the measurement algorithm should work equally well with a military-grade infrared camera. This version of the CADS is a development system. It uses off-the-shelf components that were selected to expedite development of the pose algorithms. Future versions of the CADS should include hardware designed specifically for pose determination, which would minimize the present power and volume requirements.

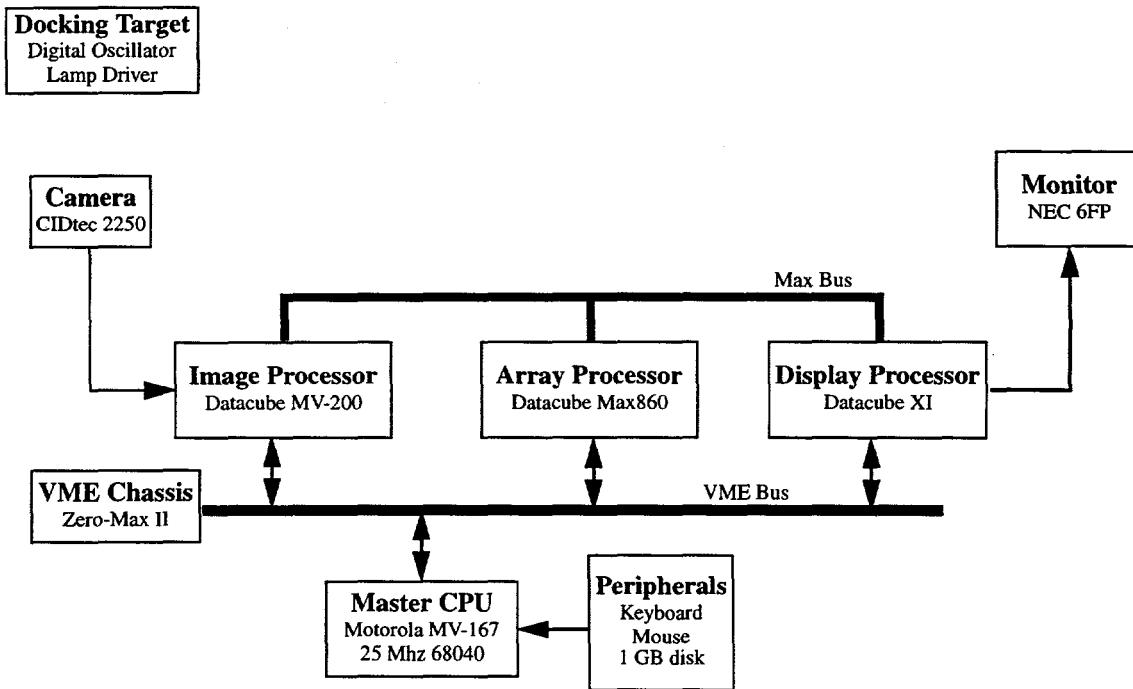
The CADS uses a video camera mounted on the robotic arm to view the docking port. The docking port must be equipped with identification markers to distinguish the intended target from other objects in the field-of-view. The CADS could be modified to eliminate the need for the identification markers given further development. In theory, the port could be identified from its intrinsic features rather than from the identification markers. Each surface on the port will have an edge boundary that delineates it from the background. The pose could be determined from these surface edges instead of the markers. In practice, however, the surface edges can be difficult to extract, particularly under unfavorable lighting conditions. An alternative method of eliminating the port markers is to use a 3-D range camera in place of the 2-D video camera. When the CADS was designed, range cameras lacked sufficient resolution for the autodocking application, but recent improvements in technology have increased the resolution to the point that range cameras could be considered in future systems.

The CADS communicates with the robot control system via a client-server communications model over an Ethernet socket connection. Pose data from the CADS can be sent to the control system or displayed locally on a CRT screen. The output of the CADS is a pose vector containing the **x**, **y**, and **z** position coordinates; pitch, roll, and yaw angles; and a confidence factor. The pose vector is updated continuously, with the most recent value saved in memory. A new pose vector is stored every 1.1 s during the continuous update mode. When the control system requires a measurement, it sends a pose request to the CADS via the Ethernet link. The CADS returns the most recently stored pose vector. The entire request-reply cycle takes only a few milliseconds to complete.

The CADS consists of the following hardware components that are also shown in Fig. 3.

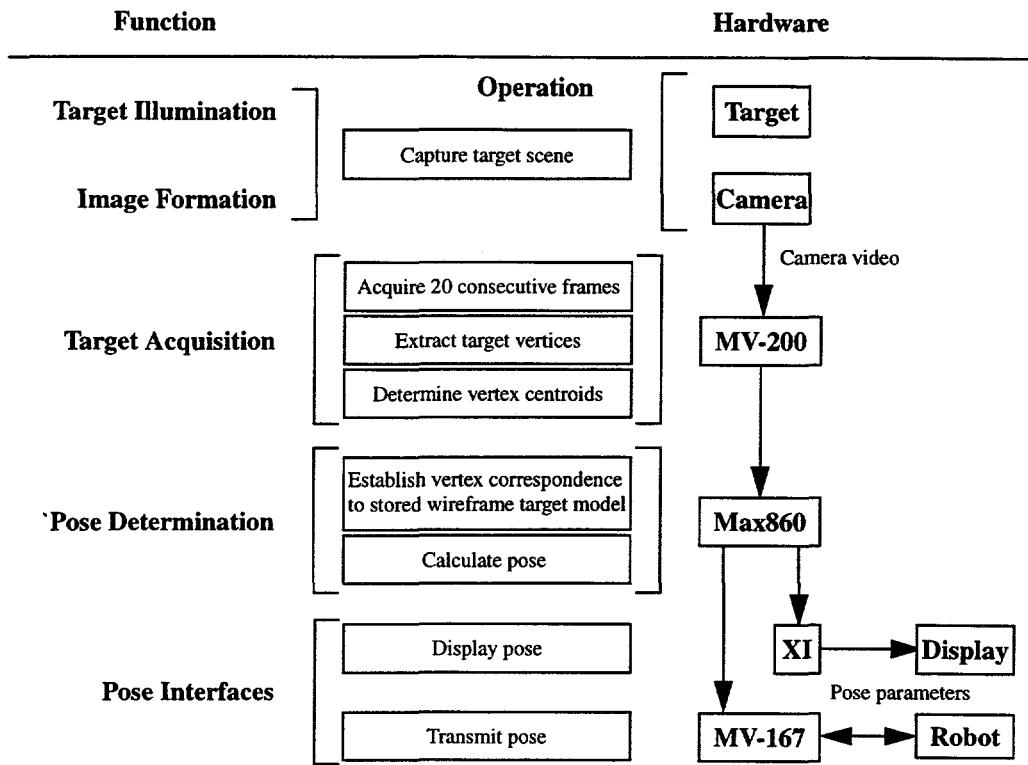
- **Target** — An array of light-emitting diodes (LEDs) serves as fiducial markers in the video image. The markers identify structural vertices on the docking port and are used by the pose determination algorithm to calculate the position and orientation of the port.
- **Camera** — A conventional, charge-injection device (CID) video camera generates an image of the target. The CADS currently uses a CIDtec Model 2250D camera but can be easily adapted to other types of cameras, including infrared.
- **Image processor** — The heart of the CADS imaging system is a Datacube MV-200 with 36 MB of image memory. The target acquisition function is performed on the MV-200 using the internal, high-speed image pipeline. A sequence of 20 image frames from the camera is digitized and stored on the MV-200. The coordinates of the center of each target marker are calculated with a digital demodulator algorithm. The image processor can transfer images to the array processor and display processor via the MaxBus™ at 20 MB/s.
- **Array processor** — The numerical processor in the system is a Datacube Max860 board. The Max860 has two Intel i860 floating point processors and 16 MB of memory that perform the recursive mathematical operations of the pose determination algorithm.
- **Display processor** — A Datacube XI board is used to generate the X11 compatible graphical user interface shown on the display monitor. The XI board combines the computer-generated graphics with live video from the camera. The pose display functions are performed on this board.

- **Display monitor** — A conventional 21-in. personal computer-grade color monitor, Model NEC-6FP, is used to display the graphical interface generated by the XI board.
- **Master CPU** — A Motorola MVME-167 board is the central processor unit. This board is based on the Motorola 68040 microprocessor and has 16 MB of memory. The CPU board monitors and coordinates the operations of all other boards. Additionally, the CPU board can perform the functions of the array processor or display processor, although at a slower throughput rate.
- **Chassis** — All board components in the CADS are VME compatible. The boards and disk drives are housed in a Zero brand 21-slot VME enclosure.
- **Peripherals** — The CADS needs several ancillary devices to support its operation including a keyboard, mouse, and disk drive.



**Fig. 3. CADS hardware block diagram.** This schematic shows the electrical connections of the major CADS components.

The CADS performs five basic functions: (1) target illumination, (2) image formation, (3) target acquisition, (4) pose determination, and (5) pose interface with external devices. In Fig. 4, each function is mapped with a corresponding set of operations and the hardware component that normally executes each operation. For example, the target acquisition function is comprised of three operations: acquire 20 consecutive video frames, extract the vertices from the target markers in the image, and calculate the centroids of each vertex. The target acquisition operations are done on the MV-200 image processor board.



**Fig. 4. Diagram mapping CADS functions to corresponding hardware modules.**  
Each function in the left column is implemented by the adjacent operation(s) in the middle column. The operations are performed on the hardware module in the right-hand column.

MAXbus, MaxVideo, and ImageFlow are trademarks of Datacube, Inc. UNIX is a trademark of AT&T. LynxOS is a trademark of Lynx Real-Time Systems, Inc. X Windows is a trademark of Massachusetts Institute of Technology.

### 3. OPERATIONAL DESCRIPTION

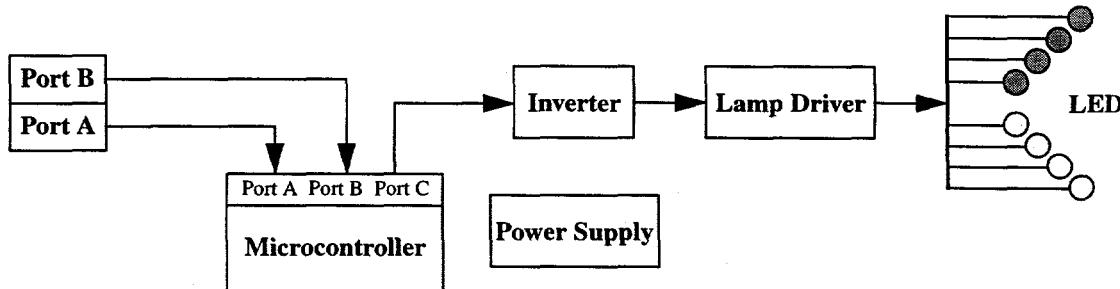
#### 3.1 TARGET ILLUMINATION

The CADS requires an array of LEDs mounted along the perimeter of the docking port to identify itself. The LEDs are strobed at 12 Hz to mark the major vertices of the structure. The LEDs are driven by a custom-designed microprocessor board shown in Fig. 5, that controls the flash duration and frequency of the LEDs. The CADS will recognize the LEDs without the need for an external communication link between the CADS and the target.

During initialization, the system reads a target data file from the disk containing a description of the LED coordinates. The pose algorithm uses this information to create an internal wire-frame model of the target which is matched to the camera image. The target geometry can be reprogrammed to accommodate changes in the LED layout.

The LED flasher is based on the 8-bit Motorola 68HC705C8 microcontroller, which controls the flashing of the LEDs. The ability to change the firmware provides great flexibility in the way the LEDs can be made to flash without the need to change the hardware itself. Also, the supporting circuitry around the microcontroller is minimal and leads to a straightforward, simple design.

The microcontroller operates at a fixed frequency controlled by an external 4.0-MHz crystal. This crystal provides the accurate and stable time base for generating accurate flash rates. The 68HC705C8 is powered by a 5-V voltage regulator. Ports A and B are input ports that set the number of LEDs in operation. The remaining port C is configured as an output port to drive the LEDs. Port C on the microcontroller lacks sufficient power to drive the LEDs directly. Therefore, the microcontroller drives a transistor-transistor logic (TTL) inverter, which, in turn, drives a special TTL interface chip designed to handle higher currents and voltages than standard TTL output circuitry.



**Fig. 5. Block diagram of the digital oscillator and lamp driver.** The LED flash frequency and duration are determined by the microcontroller integrated circuit.

### 3.2 IMAGE FORMATION

The scene of the docking port and surrounding area is captured by the image formation function. Image formation is implemented by the video camera, lens, and digitizer on the image processor board. The camera uses an area array sensor with a spatial resolution of 512 by 512 pixels and a progressive (noninterlaced) scan. The camera has an 8-mm, fixed-focal-length, C-mount lens. Although a visible spectrum camera was used, the pose algorithms could also operate with infrared images.

#### 3.2.1 Optical Calibration

To extract an accurate 3-D position from 2-D image coordinates, the camera's internal geometric and optical characteristics must be known. The parameters required for pose computation are

- $f$  – the effective focal length of the camera,
- $K$  – the first-order radial lens distortion coefficient,
- $C_x, C_y$  – the  $x$  and  $y$  coordinates of the optical center axis, and
- $s_x$  – the uncertainty factor for scale of the horizontal scan line.

To perform the calibration, a specialized target with a large number of precisely spaced points is imaged. Next, the coordinates of the target points in the image are found to subpixel accuracy. These image coordinates are correlated with the real world position of those target points. This information is then provided to the camera calibration routine, which calculates the camera parameters based on the target data and the camera model.<sup>3</sup>

The calibration target is a grid pattern with horizontal spacing of 1.5 cm and vertical spacing of 1.49 cm. Once the image points have been correlated with real world coordinate points, the focal length, radial distortion, and optical center are calculated for the camera and lens arrangement.

### 3.3 TARGET ACQUISITION

The goal of the target acquisition function is to isolate the LED fiducial markers on the docking target and determine the pixel coordinates of each marker. The list of pixel coordinates is then supplied to the subsequent pose determination function. The LEDs are first segmented from the image background using a digital demodulation technique. The resultant image is thresholded and labeled. The centroid of each object is determined from the moments of the binary image. An analysis of each object is then performed to verify that it is a valid target marker.

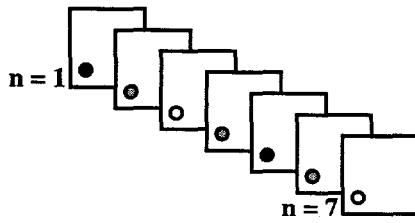
The digital demodulator is necessary because the LED flash frequency is different from the camera sampling frequency. The LEDs are strobed to increase the target detectability in the presence of background clutter and noise. The 12-Hz LED frequency was chosen to minimize modulation artifacts with the 30-Hz video sampling rate.

In the first CADS implementation, the target consisted of statically illuminated infrared LEDs placed in a known geometric pattern. An optical infrared bandpass filter was placed over the camera lens. A simple global threshold was used to segment the LEDs from the background. This method proved to be highly sensitive to variations in ambient illumination and reflections. Because the CADS might be subjected to extreme variations in background illumination, a more robust method of target identification was desired.

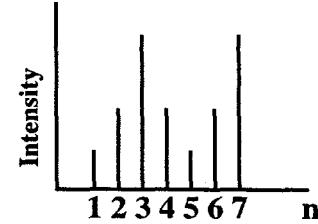
The improved technique for target identification developed for CADS version 2 reduced the possibility of other light sources or reflections being falsely detected as a part of the target. Because LEDs were already used in the pose determination, the most practical option for improving the target identification was to distinguish the presence of the LEDs temporally. A digital signal processing (DSP) technique using a target composed of LEDs flashing at a reference frequency was selected based upon (1) the desire for minimal modifications to the existing docking port, (2) the desire for no special communications between docking RSV and the port on the SPH, and (3) the desire for the vision system to provide an image useful to both operating personnel as well as to the automated system.

For the purpose of pose determination, the DSP technique requires that the target and the camera be stationary at the time that images are acquired. If motion is very limited or slow, the accuracy of the resulting pose calculation may be degraded.

In order to understand the DSP technique, consider the signal series obtained when a pixel is focused on an LED flashing at the desired frequency. In subsequent frames, the pixel intensity will alternately be high and low, representing the LED illumination. A pixel that represents a flashing LED marker, shown as the circle in Fig. 6(a), will be changing at a predetermined frequency, while other pixels in the consecutive camera frames will be relatively unchanging over time. The DSP algorithm treats the set of consecutive images as a time-ensemble. The time-ensemble in Fig. 6(b) shows the variation in intensity for a single pixel over seven frames. Let  $x(n)$  represent the intensities for a pixel location, where  $n$  is the frame number. If  $x(n)$  varies at the reference frequency  $\omega_0$ , then the Fourier transform of  $x(n)$  will have a strong response at the corresponding digital frequency.



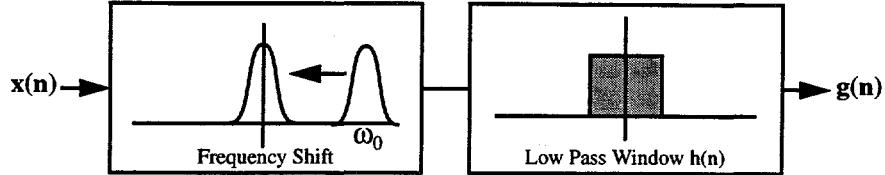
(a) Consecutive image frames taken from a stationary camera



(b) Intensity variations in a single pixel over time

**Fig. 6. Variation in pixel intensity over sequential video frames.** In (a), the small circle represents a single pixel in each of the  $n$  camera frames. The camera is pointed such that a flashing light is illuminating the single pixel but not the surrounding pixels. The time-variation of the pixel value is shown in (b).

The DSP algorithm must suppress any signal content outside the reference frequency  $\omega_0$ . In this manner, if the pixel is not changing at the reference frequency, its output will be attenuated. One technique for accomplishing the desired operation is to shift the frequency of interest at  $\omega_0$  and then to filter the signal as shown in Fig. 7.



**Fig. 7. Block diagram of the digital demodulator.** The figure illustrates the method to isolate the flashing LED markers in the camera video.

If  $x(n)$  has a Fourier transform  $X(\omega)$ , then the shifted frequency can be given by the Fourier pair:

$$X(\omega + \omega_0) \Leftrightarrow x(n)e^{-j\omega_0 n} . \quad (1)$$

Let  $h(n)$  be the time-domain representation of the low-pass windowing function and let  $H(\omega)$  be its Fourier transform. We filter  $x(n)$  by multiplying in the frequency domain,

$$X(\omega + \omega_0)H(\omega) \Leftrightarrow [x(n)e^{-j\omega_0 n}] \otimes [h(n)] , \quad (2)$$

where  $\otimes$  denotes the convolution operator.

The exponential term in the right-hand side of the above equation can be expanded into its cosine and sine components to give the following relationship:

$$X(\omega + \omega_0)H(\omega) \Leftrightarrow x(n)[\cos(\omega_0 n) - j\sin(\omega_0 n)] \otimes h(n) . \quad (3)$$

The magnitude of  $X(\omega)H(\omega)$  is a measure of the energy in  $x(n)$  at  $\omega_0$ . The magnitude will be large if the pixel intensity is varying at the reference frequency. If the magnitude of  $g(n)$  represents an LED flashing at the reference frequency  $\omega_0$ , then  $g(n)$  will be large. Pixels that are unchanging or varying at another frequency will be suppressed and eliminated from the resultant image. The magnitude of  $g(n)$  is given by the equation,

$$|g(n)| = \sqrt{[x(n)\cos(\omega_0 n) \otimes h(n)]^2 + [x(n)\sin(\omega_0 n) \otimes h(n)]^2} . \quad (4)$$

If  $g(n)$  represents an LED flashing at the reference frequency, it will be passed unattenuated. Pixels that are unchanging or varying at some other frequency will be suppressed and eliminated from the image.

### 3.4 POSE DETERMINATION

The pose determination function calculates the position and orientation of the target based on the image coordinates of the LEDs marking the docking port vertices. The coordinates of the LED markers are calculated by the target acquisition function and passed to the pose determination function.

The pose determination function stores a wire-frame, descriptive model of the target in memory. The model is a representation of the size and shape of the physical target. The model is created from a data file on disk during system initialization. The data file contains a list of the LED coordinates as measured on the target. The vertices in the stored wire-frame model are rotated and scaled to match the LEDs in the image. The match is accomplished through a least-squares regression to minimize the spatial errors. After the regression has converged, the target position and orientation parameters are available.

A number of possible pose determination methods were identified as a result of the literature survey performed during the initial phase of the autodocking task. Several candidate methods that had demonstrated good experimental results were examined in more detail. Monocular or single camera methods predominated in the literature, and many were shown to produce accurate results in a laboratory setting. These included a 4-point planar target method, an  $n$ -point model-based method, a line-based method, and a circular feature method. Desirable characteristics of a method is that it operate reliably over distances from 12 ft to 4 in.; be highly accurate, especially at close distances; be robust so that extraneous features do not cause erroneous readings; and provide redundancy so that an accurate pose can be measured even with missing features.

Point-based features were chosen for the conceptual design primarily for robustness and performance over large distance ranges. In particular, a target with discrete circular LEDs in a specific 3-D arrangement has been designed. The pose determination method that has been developed calculates the pose from these patterns of lights using a single camera. As described in Sect. 3.3, image processing is used to extract the centroid of each of the lights.

An advantage to using circular lights is that the centroid location calculation is accurate even with poor focus. A simple lens can be used to cover the entire distance range while set at one focus distance. With the 3-D environment, even if good focus is present on one feature, the other features will likely be out of focus. As a result, there is little advantage in using an autofocus lens, which adds complexity and cost.

#### 3.4.1 Perspective Projection

The process of determining pose from a single camera image necessitates reconstructing the 3-D coordinates of an object from the corresponding 2-D image coordinates. The difficulty of this process is that the transformation is not unique. That is, an infinite number of points in the 3-D coordinate frame will map to one point in the image.

The transformation problem of pose determination is illustrated in Fig. 8. A perspective projection model, commonly referred to as the pinhole lens model, is used where the center of the lens is the origin of the camera coordinate frame  $R_c$ . A point  $p$  on the object has a coordinate of  $x_c$

along the  $x$  axis of  $R_c$ . Point  $p$  is projected onto the image plane at  $q$ , with the coordinate value of  $x_i$  in the image coordinate frame. The value for  $x_i$  can be found from the geometry of similar triangles,

$$x_i = x_c \left( \frac{f_e}{z_c} \right), \quad (5)$$

where  $f_e$  is the effective focal length of the camera lens and  $z_c$  is the object coordinate in the camera  $z$  axis.

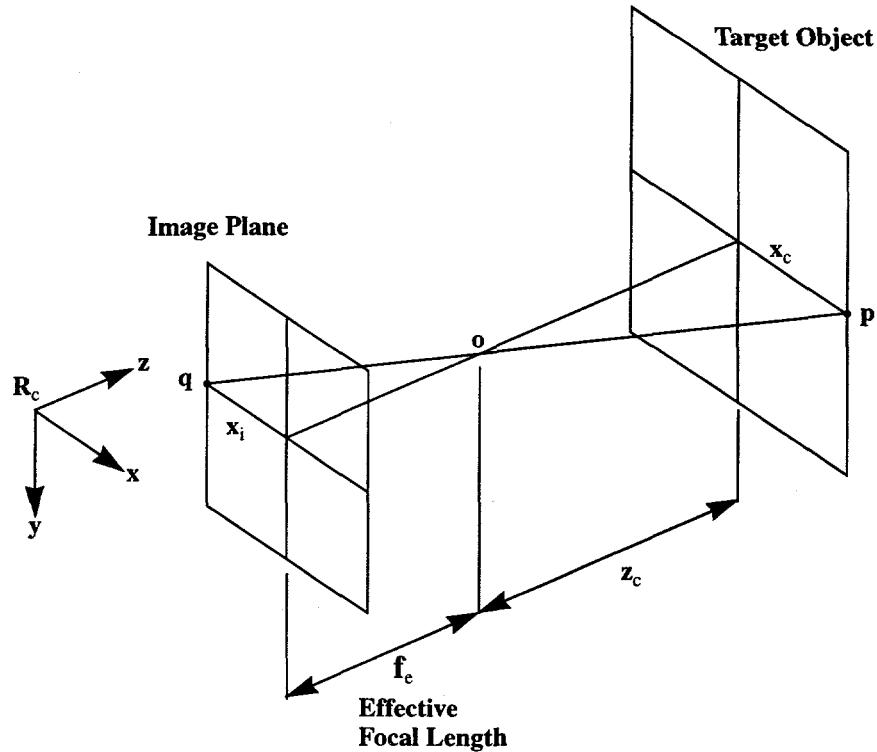


Fig. 8. Perspective model for pose determination.

The same reasoning can be extended to the  $y$  axis to yield

$$y_i = y_c \left( \frac{f_e}{z_c} \right). \quad (6)$$

The projection transformation is noninvertible; that is, the relations are unique for 3-D to 2-D, but the 3-D coordinates cannot be determined uniquely from the 2-D image coordinates. This can be seen when you consider that any point along line  $op$  in the figure will map to image point  $q$ .

From a single intensity image of the target, the 2-D feature coordinates in the image plane can be extracted. These coordinates, along with the geometric model of the target and the intrinsic camera parameters, provide the necessary information for pose calculation. Camera calibration performed off-line prior to the pose measurement is used to determine the intrinsic camera parameters such as pixel spacing and focal length. The relation between the 3-D camera coordinates and the 2-D image is based on the assumed perspective projection vision model.

While the calculation of the image coordinates is straightforward given the 3-D coordinates and the camera parameters, the inverse operation of determining the 3-D coordinates from the 2-D image plane coordinates is more difficult since the correspondence of 2-D to 3-D coordinates is not unique. Therefore, multiple features must be used when the pose calculation is made from a single image.

### 3.4.2 Feature Matching

A point method based on a model of a polyhedral object is used to calculate the position and orientation (pose) of a target with respect to a camera. The model of the object is represented as surface faces, each of which is a convex quadrilateral with four vertices. Adjacent faces, then, share two vertices. From an image of the target object, points are extracted that represent the vertices of the object. These image points are then matched to the target vertices based on convexity and adjacency constraints. Extraneous points at random locations that are not part of the target may also be detected. The extraneous points will not, in general, provide a fit to the model and will be rejected. After the point correspondence is made, the pose is calculated using a least-squares-fit algorithm.

Two basic operations are performed: matching and pose calculation. A process of hypothesis and verification similar to that described by Lowe<sup>4</sup> is applied to perform these two operations. For the first operation, an initial match on one face of four points is found. The pose for this face is calculated, and then the hypothesis is extended to additional feature points. Choosing the points for this initial face requires identifying the subsets of four image points that form the vertices of convex quadrilaterals. After perspective projection, these points on the 3-D object form a convex quadrilateral in the image plane. Sorting the points into subsets of possible faces greatly reduces the search time.

Four coplanar points have been shown to be the minimum necessary to obtain a unique pose. A direct method using this approach has been implemented and tested. This method is fast and computationally exact whenever the centroids are determined perfectly. The direct algorithm determines a unique measurement for the four coplanar points in each face assuming the perspective vision model. The algorithm first calculates the 3-D coordinates of the points relative to the camera using only the image point locations and the dimensions of the face points from the target model. Then, a rotation matrix and a translation are found that transform the 3-D target point coordinates in the base, or world, reference frame to the camera reference frame. Using quaternions to represent this rotation, the best, or optimum, rotation in a least-square sense that transforms from world to camera coordinates is calculated. The translation values are the difference between the centroid of the world points and the centroid of the scaled and rotated camera referenced points. The method, however, has little redundancy and loses accuracy whenever substantial noise is present in the image centroids.

An error measure is computed from the sum of the distances between the calculated camera-referenced face points and the transformed world target points. This error measure is an indicator of the difference in geometric shape between the derived points and the transformed face points and can be used as an indicator of an incorrect point assignment as well as a measurement quality indicator.

Using the direct method, an evaluation is made of the error value returned. If the error is small, then the pose is judged valid for the face. From this assumed match and tentative pose, additional points are projected to the image plane, and the distances between these projected points and actual image features are measured. For those points in the vicinity of a true imaged point, a match is declared. This process continues for all remaining model points. As many matches as possible are made since the resulting error is inversely proportional to the number of matched points.

### 3.4.3 Pose Calculation

The iterative pose calculation method uses  $k$  pairs of matched points that has Newton's method as its foundation. Newton's method for solving the scalar function,  $f(x) = 0$ , is given as

$$x_i = x_{i-1} - \frac{f(x_{i-1})}{f'(x_{i-1})} . \quad (7)$$

From this expression, it is seen that the method is iterative and uses the previous iteration and the derivative of  $f$  with respect to  $x$  to calculate the next iteration estimate.

The desired result in the pose determination problem is to solve for the three translation values and the three orientation angles. To accomplish this, the model points are projected to 2-D image space using perspective projection, as discussed above. Then the errors based on  $(u, v)$  coordinates in the image plane are calculated. The partial derivatives of  $u$  and  $v$  with respect to each pose parameter are found next. The error component in the  $u$  direction is measured from the image data, and it is given by

$$e_u = \frac{\partial u}{\partial x} \Delta x + \frac{\partial u}{\partial y} \Delta y + \frac{\partial u}{\partial z} \Delta z + \frac{\partial u}{\partial \alpha} \Delta \alpha + \frac{\partial u}{\partial \beta} \Delta \beta + \frac{\partial u}{\partial \phi} \Delta \phi , \quad (8)$$

where the delta components represent the unknown corrections to the pose parameters. Note that an initial pose estimate is required in order to calculate the delta values. A similar expression for the  $v$  direction is also found.

For  $n$  points the partial derivatives can be represented by a Jacobian matrix  $\mathbf{J}$  of dimension  $2n$  by 6. The matrix equation is

$$e = \mathbf{J}c , \quad (9)$$

where  $\mathbf{J}$  is the Jacobian as given above,  $\mathbf{c}$  is the unknown corrections vector of length 6, and  $\mathbf{e}$  is the error vector of length  $2n$  which contains the measured values from the image. This equation is overdetermined since the minimum number of points  $n$  is 4. To fully use this information, a least-squares solution is calculated as follows:

$$\mathbf{c} = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \mathbf{e} . \quad (10)$$

The corrections vector  $\mathbf{c}$  is used to update the present pose estimate. This process continues iteratively until the error vector magnitude becomes less than a preset value. In practice, the algorithm converges with any valid initial pose estimate. The number of iterations, however, can become large for some positions and orientations. This problem has been solved by adding variable damping to the Jacobian matrix components, which causes the convergence to be linear initially for the large error components and then become quadratic as the errors become small. The damping factor initially is relatively large and is reduced as the result converges. The result converges generally in six to eight iterations. If the algorithm does not converge because of mismatched points or noise, a maximum number of iterations is reached with an error status.

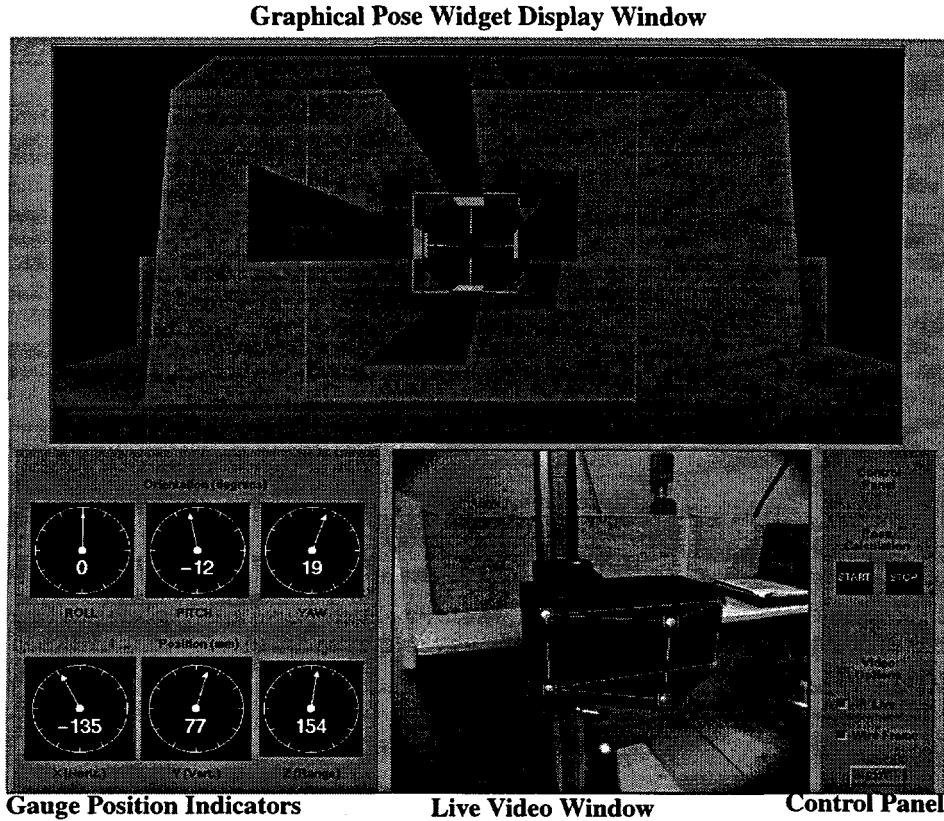
### 3.5 POSE INTERFACE

An interface was developed to present the data calculated by the pose determination function to the operator on a local video screen. The display provides simulated gauges to indicate the position (horizontal, vertical, and range) and orientation (roll, pitch, and yaw) of the target. Additional video windows contain a system control panel, a live image from the arm-mounted camera, and a unique docking icon that indicates all six pose parameters in a single object.

When the RSV is docking with the SPH for the purposes of ammunition transfer, the RSV operator may desire feedback on the status of the docking process. Whether the docking is performed in autonomous or manual mode, it is important that the information describing the position of the arm be presented in such a way that the operator can quickly determine the docking status. In some instances, the docking procedure may be a combination of the manual and autonomous modes. In a manual mode the operator will be actively using the positional information to guide the arm to the SPH port. In an autonomous mode, the operator will only monitor the positional information to determine, for example, if manual intervention becomes necessary because of an undetected obstacle. In either situation, the operator will need to have information presented in a form that is easily and quickly interpreted.

The graphical user interface (GUI) constructed for the CADS system was designed to display information about the position of the 6-D.F. robotic arm relative to the resupply port on the RSV. Two main issues regarding human interpretation of such information guided the design of the GUI. The first limitation is the human's inability to focus on more than one area of the screen at a time. Displaying the position information to the operator only as six different positional numbers is overwhelming because only one pose coordinate can be monitored at a time. The second limitation that influenced the design of the GUI is the human's lack of accurate depth perception. A graphical pose widget was designed and implemented which provides the operator with all six positional numbers, including an intuitive indication of the range (depth) of the arm relative to the reload port of the SPH.

A photograph of the GUI layout is shown in Fig. 9. There are four main modules within the GUI to aid in the operator's interpretation of the position of the robotic arm relative to the SPH reload port. These four areas are the gauge position indicators, the graphical pose widget display window, the live video window, and the control panel.



**Fig. 9. Photograph of the operator's GUI screen for the CADS.** The GUI can provide visual feedback to the operator during either automatic or manual docking.

### 3.5.1 Gauge Position Indicators

One way to show all 6 D.F. on the screen is with the position instruments and text displays as shown in the lower-left area of Fig. 9. The top row of gauges provides orientation information (roll, pitch, yaw) about the SPH docking port relative to the camera. The bottom row of gauges shows the position (horizontal, vertical, range) of the port relative to the camera in 3-D space. There is also a text display at the center of each gauge that gives the numerical value of the gauge reading.

The border around each gauge can be either red or green. If the particular degree-of-freedom is within a predefined range for successful docking, then that gauge border will be green. If that degree-of-freedom is outside the range for successful docking, then the border will be red. This color indication is easier to interpret than trying to check each gauge by reading them individually.

The gauges are helpful in situations where the arm is very close to the port, and only one or two of the degrees-of-freedom needs slight adjustment. Although the gauges provide all of the necessary pose information, it is difficult to watch all of the readouts at the same time. More importantly, gauges and numbers do not provide an intuitive feel for the actual position of the robotic arm relative to the port. For this reason a graphical pose widget was developed that intuitively indicates pose information.

### 3.5.2 Graphical Pose Widget Display Window

As shown in Fig. 9, the entire upper-half of the GUI is dedicated to the graphical pose widget. An enlarged view of the graphical pose widget is shown in Fig. 10. Each component labeled in Fig. 10 is described in the following sections.

The discussion that follows assumes an arm-mounted camera. All pose measurements are made in the coordinate frame of the camera as the reference. However, because the camera is affixed to the arm, the arm coordinate frame can be made to match the camera frame. Thus, for docking, the camera is synonymous with the endpoint of the arm.

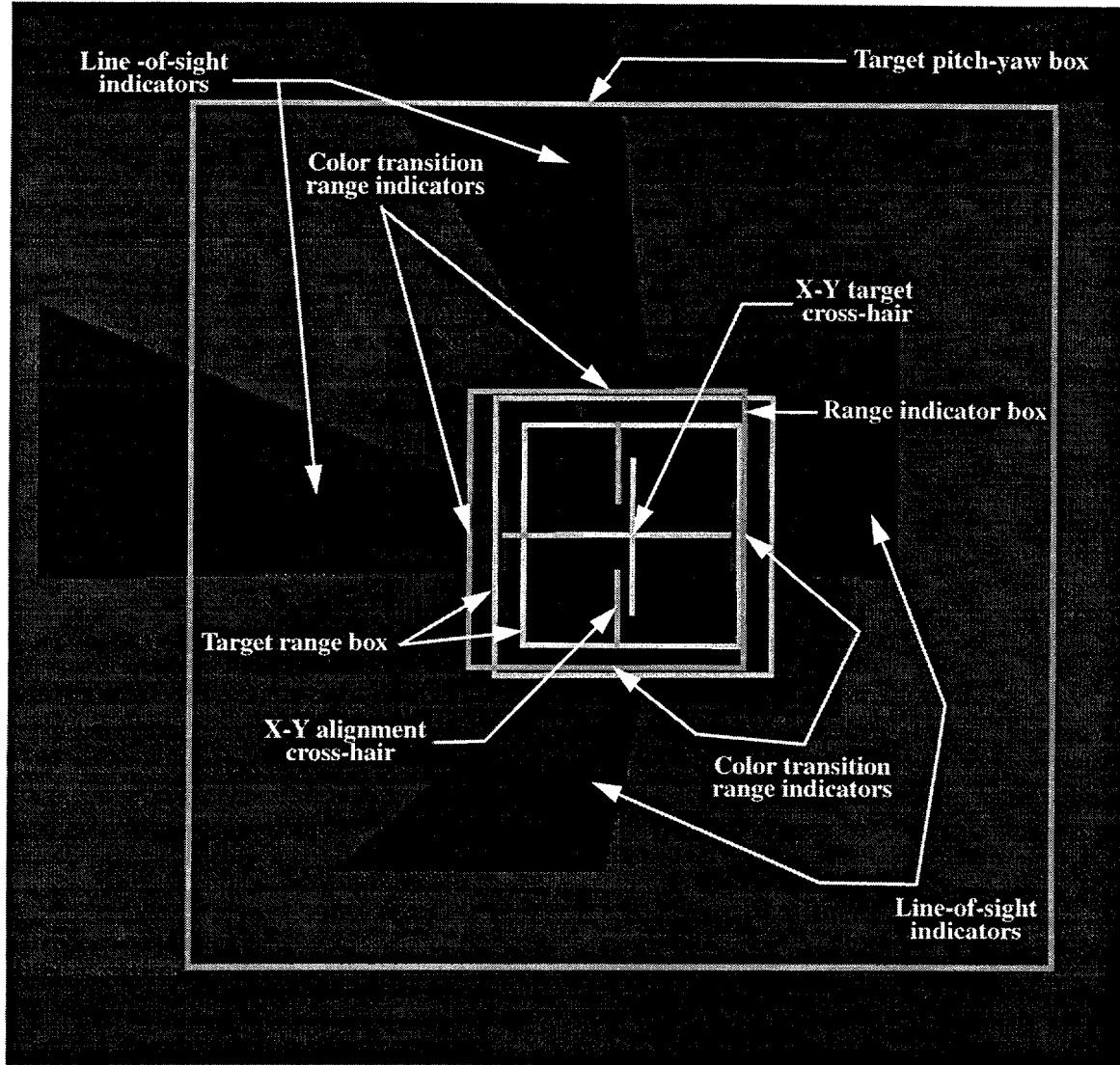
*X-Y target cross-hair*. This is a fixed, red cross-hair on the center of the docking port that serves as a target for proper x-y positioning. It represents the center of the docking port in the x-y plane.

*X-Y alignment cross-hair*. This green cross-hair moves with the camera and thus the robotic arm. When this cross-hair is aligned with the X-Y target cross-hair, then the line-of sight of the camera is aligned with the center of the port. In Fig. 10, the target and alignment cross-hairs are very close to being aligned, so the camera is aiming almost directly at the center of the port.

*Line-of-sight indicators*. These four arms of the “cross” provide information on the roll, pitch, and yaw of the camera relative to the port. The four pieces can be thought of as the four sides of the interior of a tunnel which indicates the line-of-sight path from the camera to the center of its field-of-view. In Fig. 10, the horizontal line-of-sight indicators are skewed to the left, indicating a significant positive yaw orientation. Note how the left horizontal arm is much wider than the right and how the upper and lower arm of the cross originate near the center of the port, but then lean to the left. This indicates that the camera is positioned to the left of the port and is pointing to the center of the port. The pitch is positive relative to the port, indicated by the slight skew in the y-direction. Variations in the roll orientation are indicated by the rotation of the entire cross (including the X-Y alignment cross-hair) about its center. The roll value indicated in Fig. 10 is zero.

*Color transition range indicators*. The outer segment of the line-of-sight cross arms is red, while the inner segment is green. The color transition range indicators are the red-to-green transition in each arm of the cross. As the camera gets closer to the port, the color transition point moves in toward the center of the cross, resulting in more of the cross becoming red. As the arm moves out away from the port, the color transition border moves out toward the ends of the cross, resulting in more of the cross becoming green.

*Target range box*. This fixed-position, red box indicates the desired range for docking. When the four color transition range indicators align with the edges of the target range box, the arm is the proper distance from the port for docking.



**Fig. 10. Photograph of the graphical pose widget.** The widget is indicating that the camera is to the left and slightly above the port.

**Target pitch-yaw box**. This green box serves as a target for the operator when adjusting the pitch and yaw angular positions of the robotic arm. When all four of the line-of-sight indicators fall inside this target pitch-yaw box, the arm is within acceptable orientation for docking.

**Range indicator box**. This green box gets larger as the operator moves the camera away from the port and gets smaller as the arm moves in closer to the port. As illustrated, the borders of this box are coincident with the color transition range indicators. The size of this box changes with the range of the target.

### 3.5.3 Live Video Window

The live video window is located in the bottom-center of the GUI. This window provides a booms-eye view of the outside world. Using this display, the operator can observe things that would not be evident in the simulated display, such as weather conditions or obstacles between the end of the robotic arm and the resupply port.

Because this live image is also used by the autodocking system for autonomous pose determination, the display can be used to verify the function of the pose determination algorithm. In response to an automatic pose measurement, a green overlay is generated and overlaid on top of the live image. Refer to Fig. 9 for an example of this overlay. The overlay indicates the number of valid, four-point facets found in the pose determination function. Notice that three facets are formed by the overlay. If the automatic pose determination algorithm is successful, at least one overlay facet will be generated. If the algorithm is unsuccessful, none of the facets will be generated.

### 3.5.4 Control Panel

The bottom-right portion of the GUI layout is the control panel where the operator can set various parameters of the GUI. The first set of buttons labeled "START" and "STOP" allows the operator to start and stop the automatic pose determination algorithm. If the pose determination algorithm is not engaged, all of the position gauge indicators will be blank, and no graphical pose widget will be displayed. The live video image will be active, but there will be no overlay. After the operator starts the pose determination, all of the GUI pose indication functions will begin displaying information.

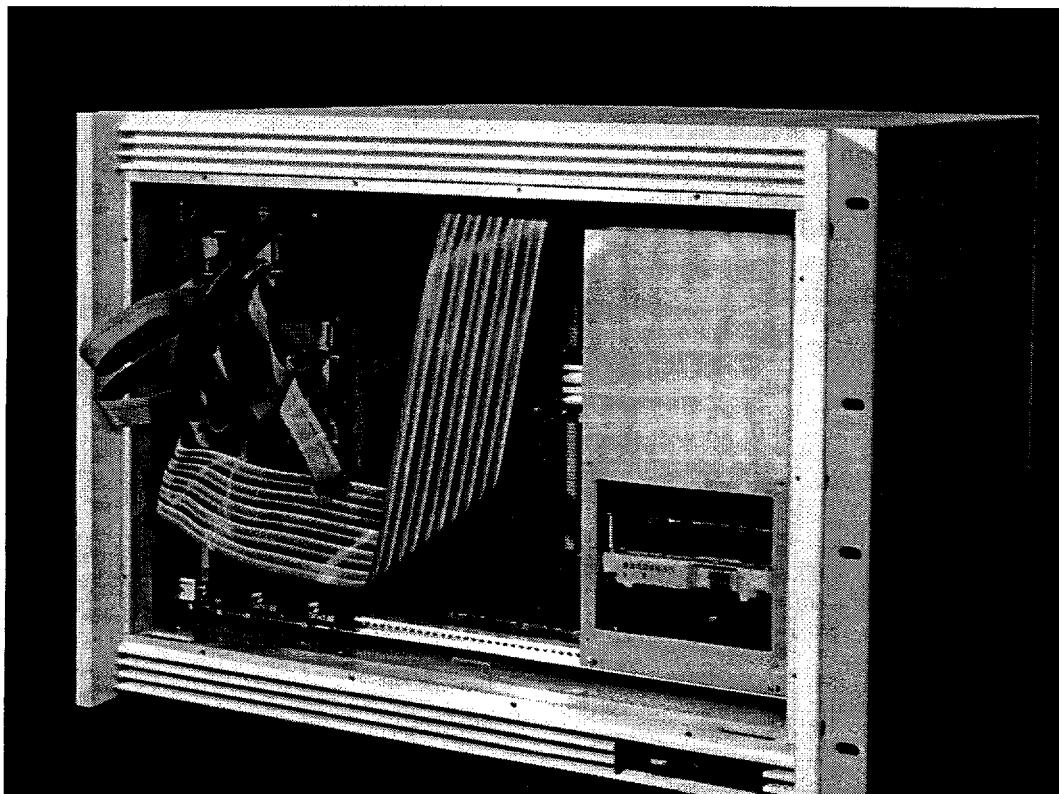
There are two video option toggle buttons. The first allows the operator to turn the live image display on or off. The second toggle button gives the operator the option of turning the wire-frame overlay on or off. The final button on the control panel labeled "EXIT" will terminate the GUI.

## 3.6 REFERENCES

1. C. L. Leonard and E. V. Bergmann, "A Survey of Rendezvous and Docking Issues and Developments," *Orbital Mechanics and Mission Design: Proc. of AAS*, report No. AAS 89-158, April 24-27, 1989.
2. J. D. Michael, "A Study of Autonomous Rendezvous and Docking Systems," Marshall Space Flight Center, Third Annual Technical Review, Large Space Systems Technology, Nov. 16-19, 1981.
3. Roger Y. Tsai, "A Versatile Camera Calibration Technique for High-Accuracy 3-D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses," *IEEE J. Robotics Autom.* RA-3(4), 323-44 (August 1987).
4. David G. Lowe, "Three-Dimensional Object Recognition from Single Two-Dimensional Images," *Artificial Intelligence*, 31, 355-95 (1987).

#### 4. RESOURCE REQUIREMENTS

The physical configuration of the CADS consists of the camera, chassis, and display monitor. The four electronic printed circuit boards are contained in an industry-standard, VME enclosure, as shown in Fig. 11. The boards in the system are: (1) the MV-167 CPU, (2) the MV-200 image processor, (3) the Max860 array processor, and (4) the XI display processor. The CPU and array processor are single-slot boards, while the image processor and display processor take two slots each. The chassis houses the two system hard disks and power supply. The hardware components of the CADS are also described in Sect. 2.2.



**Fig. 11. The CADS is self-contained in the 21-slot VME chassis.** The four CADS circuit boards are located on the left side of the enclosure, while the system disk drives can be seen on the right side.

Because the current version of the CADS was intended for a development environment, the resource requirements are higher than for a fielded system. For example, the chassis is oversized to accommodate additional modules should the need arise during development. For a fielded unit, a smaller chassis would reduce the space and power requirements. The camera and monitor were also selected for development applications. The CADS could be easily reconfigured to accept a different camera or monitor.

#### 4.1 PROCESSING

The pose measurement functions require a high data throughput rate. A set of special-purpose imaging boards is used in the CADS to perform the 2-D signal processing operations. The boards are state-of-the-art commercial products employing the latest integrated circuit technology.

The primary imaging board is a Datacube MV-200 single-board image processor. This board is a double-height VME form factor with built-in data acquisition, frame buffer with 36 MB of memory, and special-purpose hardware accelerators. The board provides a pixel-processing pipeline at a continuous rate of 20 million pixels per second. The special-purpose accelerators perform the digital filtering needed to identify and locate the flashing lamps on the docking port.

The Datacube Max860 is the system array processor and handles the iterative pose computation. The Max860 is a vector processor based on two Intel 40-MHz i860 RISC processors and pipeline floating point processor with 16 MB of memory. The Max860 is compatible with the other Datacube boards and can transfer images at up to 60 MB/s.

The display functions are handled by the Datacube XI 24-bit color graphics display controller. The XI board is a two-slot, double-height VME board based on the Texas Instruments 40-MHz 34020 graphics processor. The board is capable of accepting incoming images at 20 million pixels per second. The XI links the frame buffer to an X11R5 compatible window manager. The spatial resolution of the XI is  $1280 \times 1024$  pixels.

The host CPU is a Motorola MV-167 single-board computer. The MV-167 is a double-height VME board with a built-in Ethernet interface. The CPU uses a 25-MHz, Motorola 68040 microprocessor with 16 MB of memory and is rated at 28 MIPS and 3.5 MFLOPS. The function of the CPU board is to coordinate and control all of the CADS functions and data flow. Normally, the CPU initiates data processing that is performed on other boards. The CPU can assume the tasks of the imaging boards, but at a lower speed.

#### 4.2 COMPONENT LIST

CPU board.....	Motorola MVME-167
Image processor board .....	Datacube MV-200
Array processor board.....	Datacube Max860
Display processor board .....	Datacube XI 24-bit color + 8-bit overlay
Camera .....	CIDTEC 2250 $512 \times 512$ array
Monitor .....	NEC 6FGp 21-in. color
Chassis .....	Zero Max-II 21 slot VME
Target.....	Custom LED driver
Peripherals.....	1-GB disk, keyboard, mouse

#### 4.3 POWER

The power requirements for the CADS is estimated by adding the manufacturer's rated power consumption for each hardware component. The power supply rating represents the maximum power available from the chassis.

Power consumption .....	32 A @ 5 Vdc
.....	1 A @ -12 Vdc
.....	4.4 A @ 12 Vdc
Power supply capacity.....	100 A @ 5 Vdc
.....	12 A @ 12 Vdc
.....	10 A @ -12 Vdc
Camera .....	1.0 A @ 15 Vdc
Monitor .....	2 A @ 120 Vac

#### 4.4 VOLUME

Chassis (in.) .....	17.0 W × 22.7 H × 28.2 D
Camera head (in.) .....	1.9 W × 1.9 H × 1.9 D
Camera control unit (in.).....	7.7 W × 1.5 H × 9.4 D
Monitor (in.) .....	20.1 W × 20.7 H × 22.4 D

#### 4.5 WEIGHT

Chassis .....	80 lb
Camera (minus lens) .....	2 lb
Monitor .....	80 lb

#### 4.6 OPERATING TEMPERATURE

Chassis .....	0° to 55°C
Camera .....	0° to 50°C
Monitor .....	0° to 40°C

#### 4.7 STORAGE TEMPERATURE

Chassis .....	-40° to 100°C
Camera .....	-25° to 85°C
Monitor .....	-20° to 60°C

#### 4.8 HUMIDITY (NONCONDENSING)

Chassis .....	10%–90% R.H.
Camera .....	0%–95% R.H.
Monitor .....	30%–80% R.H.

## 5. SOFTWARE ARCHITECTURE

### 5.1 ARCHITECTURE

The CADS is presently configured as a self-sufficient software development system. The utilities such as editor, compiler, and debugger that are needed for code development are resident on the CADS disk. Additional utilities such as an image viewer are also found on the disk. The philosophy behind the CADS configuration was to provide tools to aid in the development and evaluation of the image processing and pose calculation software. Once we have confidence in the algorithms, the development environment could be deleted, significantly reducing the size and complexity of the system.

The CADS software architecture consists of three major components: (1) the operating system, (2) the development resources, and (3) the CADS code. The operating system is the LynxOS™, a UNIX™ variant with a real-time kernel. The windowing system is based on X11R5 and Motif. The development environment includes the GNU utilities, a C compiler and debugger, and other tools installed locally. Additional development utilities include the ImageFlow™, a C function library that provides the programming tools to control the image processor, array processor, and display processor boards. ImageFlow controls the data flow within each board and to other boards over the MAXbus or the VME bus. All functions that handle interaction between the host CPU and the image boards are also provided by the ImageFlow software. The ImageFlow functions are heavily used within the locally developed CADS code.

The CADS algorithm functions, described in Sect. 2.2, can be divided into two software modules, (1) the pose module needed to measure the pose of the port and (2) the GUI module to show the pose results to the operator on the display monitor. The GUI module is an optional feature on the CADS. The GUI is extremely helpful for computer-assisted manual docking, where the operator can see the present distance from the arm to the port. The CADS could be reconfigured to exclude the GUI, using only the network interface to send pose data to the robotic control system.

### 5.2 ESTIMATED CODE

The CADS pose measurement algorithm and GUI are written in the GNU-C language. There are ~25,000 lines of code in the pose measurement module and an additional 8,000 lines of code in the graphical interfaces. These estimates are only for the CADS code written by ORNL and do not include precompiled libraries linked during compilation, notably X11R5, Motif, and Datacube ImageFlow. The main executable program is called *dash* and is 2.8 MB in size.

### 5.3 TARGET ILLUMINATION

The assembly-coded firmware program starts by clearing the RAM memory and setting ports A and B as inputs and port C as an output. It then reads ports A and B to determine the switch settings. Port A is used to select which of the eight LEDs is enabled to flash in the flash mode. Only LEDs whose respective switches were closed at power up will flash. The flash loop is composed of two timing loops that are responsible for turning the LEDs on and off at fixed time

intervals. The flash rate is fixed by the crystal and constant data values in the firmware. The current flash rate is 12 Hz with a 10% on duty cycle. This means that the LEDs are on for 8 ms and then off for 75 ms.

There are two modes of operation. The first mode is the normal flash mode and is used to flash the LEDs that are selected by closed switches on port A. There are eight dip switches on port A corresponding to the eight LEDs. If a switch position on port A is closed, the corresponding LED will flash in the flash mode. If the switch is open, the LED will stay off.

The other mode is a diagnostic mode and is governed by the switch settings on port B. Port B is used to select groups of four or all eight LEDs to remain on steadily without flashing. If switch 1 on port B is closed, the inner four LEDs will stay lit. If switch 2 on port B is closed, the outer four LEDs will come on and stay on. If switch 3 on port B is closed, all the LEDs will light. Only the first three switches on port B are of any consequence to the operation of the flasher circuit. This mode may be useful for testing purposes.

Switches on either port A or B can be changed while the circuit is flashing, but nothing will change until the microcontroller is reset by cycling the power.

#### 5.4 TARGET ACQUISITION

The sampling rate of the camera is 30 frames per second; then, by the Nyquist criteria, the highest frequency that can be detected by the system is 15 Hz. The actual flash rate selected for the target LEDs was 12 Hz, corresponding to a digital frequency of

$$\omega = 2\pi fT = 2\pi \left( \frac{12}{30} \right) , \quad (11)$$

where  $f$  is the flash rate and  $T$  is the sampling period.

The Hamming windowing function was selected as the low-pass filter  $h(n)$  to be convolved with the frequency-shifted signal  $x(n)$ . This window function was chosen for its sharp frequency attenuation beyond the frequencies of interest. The Hamming window was implemented in the following form:

$$h(n) = 0.54 - 0.46 \cos\left(2\pi \frac{n}{N-1}\right), \quad 0 < n < N-1 \quad (12)$$

where  $N$  represents the total number of samples (images) in the set. A sample size of 20 images was selected to provide ~4-Hz total bandwidth. This corresponded to a total image acquisition time of 670 ms. Greater resolution can be obtained with more sampled images but would take longer to acquire and process the additional samples.

Note that as the result of Eq. (12), the  $\cos(\omega_0 n) \otimes h(n)$  term in Eq. (4) is constant for any given frame number  $n$ . Therefore, those terms can be predetermined for each frame and stored in a lookup table designated  $K_c(n)$ . Similarly, the  $\sin(\omega_0 n) \otimes h(n)$  terms are stored in lookup table  $K_s(n)$ . The convolution operation can be written as the sum-of-products given by

$$x(n) \cos(\omega_0 n) \otimes h(n) = \sum_{n=0}^{N-1} x(n) K_c(n) \quad (13)$$

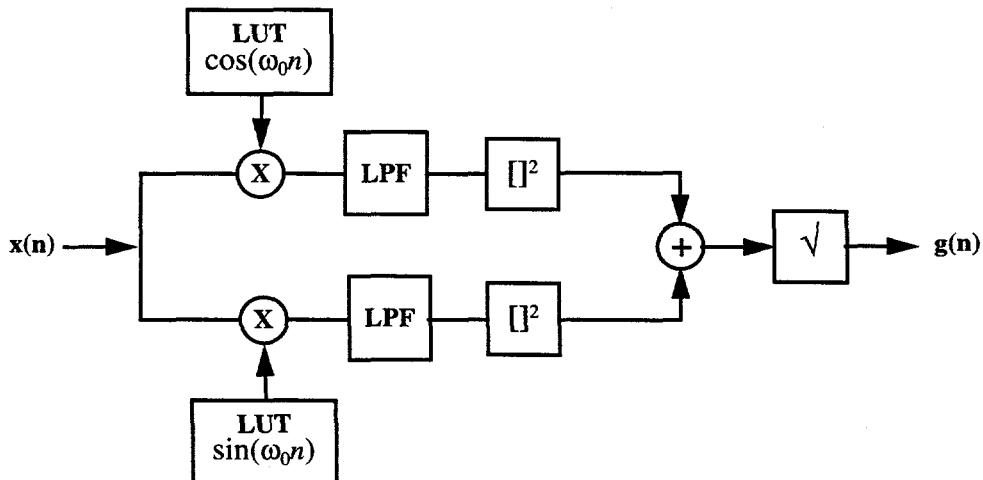
and

$$x(n) \sin(\omega_0 n) \otimes h(n) = \sum_{n=0}^{N-1} x(n) K_s(n) \quad (14)$$

We can then rewrite Eq. (4) to the computationally simpler form,

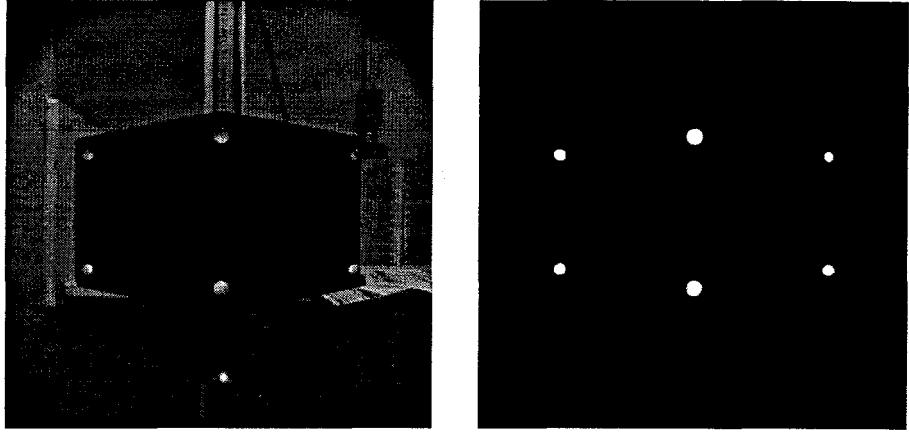
$$|g(n)| = \sqrt{\left[ \sum_{n=0}^{N-1} x(n) K_c(n) \right]^2 + \left[ \sum_{n=0}^{N-1} x(n) K_s(n) \right]^2} \quad (15)$$

This observation allowed the filter described by Eq. (15) to be implemented in the image pipeline with the following sequence: (1) multiply each image frame by the predetermined sine and cosine constants for that frame, (2) sum the results into separate sine and cosine accumulators, (3) square the contents of each accumulator, (4) add the contents of the two accumulators, and (5) take the square root of the accumulator sum. The filter function is implemented by the block diagram in Fig. 12.



**Fig. 12. Signal flow diagram for the digital demodulator.** The block diagram above implements the demodulator function given by Eq. (15).

The sensor used in this particular camera cannot be electronically shuttered. This means that the sensor integrates the incident illumination over the entire frame period. If the target LED duty cycle were 50%, then the LED would be on for  $\sim 40$  ms. By comparison, the camera sampling period is  $\sim 30$  ms. If the LED period is greater than the camera sampling period, some LED illumination periods could overlap two successive camera frames. The digital method works best if the LED is clearly defined as either on or off for each camera frame. To minimize the degrading effects, the LED duty cycle was set to 10%. The results of the demodulator function are shown in the photographs of Fig. 13.



**Fig. 13. Photograph of the input signal to the demodulator and the resultant output.** The photograph on the left is a single frame  $x(19)$  of a six-point target, while at right is the resultant image  $g(20)$  after running the algorithm described by Eq. (15).

## 5.5 POSE DETERMINATION

The algorithm for pose determination is based on the theory given in Sect. 3.4. This algorithm solves the nonlinear pose calculation problem using Newton's method by linearizing the pose parameters about the present iteration and forms a more accurate estimate of the minimum least-squares error solution. Several preprocessing steps are initially done to the point features before the principal algorithm is applied. The entire algorithm is given below.

### 5.5.1 Initialization

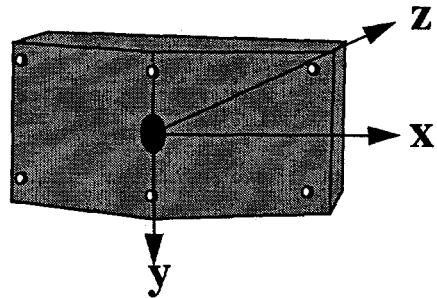
The geometric dimensions and parameters of the target model are read from the file labeled *target.dat*. These values are stored in a memory for use by the matching and pose calculation programs. In addition, camera parameters that include size and number of sensor elements are read from the *camera.dat* file. Optical calibration parameters are also read from the file *calib.dat*. The calibration parameters include the lens focal length, radial lens distortion coefficient, and the optical center location in terms of pixels.

### 5.5.2 Feature Point Preprocessing

At least four feature points must be present, and all points must have areas within a preset limit of each other. In addition, the points must have an aspect ratio and fill ratio that are consistent with those of an elliptical object. A practical upper limit on the number of points to analyze is set at 25.

### 5.5.3 Unit Transformations

The pixel coordinates of the feature points are converted from pixels to sensor coordinates in millimeters. The sensor coordinate reference frame is shown in Fig. 14 imposed on the six-point target. The coordinate frame has its origin at the optical center with positive x to the right looking from the back of the camera toward the front. Positive y is down, and positive z is in front of the camera. This step also adjusts the coordinates using the radial lens distortion coefficient.



### 5.5.4 Convex Quadrilateral Grouping

The next step is to group the feature points into sets of four which form convex quadrilateral shapes in the image plane. These groupings provide a set of potential matches to the quadrilateral faces of the geometric model.

Fig. 14. The sensor pose coordinate system. The sensor coordinate reference frame is shown imposed on a six-point simulated docking target.

### 5.5.5 Initial Pose Calculation

For each face on the geometric model, each quadrilateral group is matched against the face. The pose is calculated initially using the fast direct method to give a go/no-go decision as to whether the match is potentially good. The first potential match that is found is then used as the hypothesis. A more accurate pose is calculated for verification using the least-squares algorithm.

### 5.5.6 Match Verification

From the assumed match of a set of image points and model points with the calculated pose, the set of matched points is extended. This is performed by projecting each remaining model point using perspective projection into the image plane. For a match to exist, there must be a corresponding feature point within some neighborhood of the imaged model point. As many additional model points as possible are matched in this way. If no additional matching points are found, the potential matching group of four is discarded, and the process is continued for the remaining candidates. At least one additional matching point must be found for the model recognition to be declared a success. One exception to this is if there are only four image feature

points and these points produce a valid pose with the four-point matching face. This matching process enhances robustness by permitting some of the points in the target to be missing while at the same time rejecting any points that do not fit the model.

### 5.5.7 Pose Calculation

Using all matched points from the previous step, the pose is calculated using the iterative least-squares error algorithm. As discussed above, this algorithm requires an initial pose estimate. Several initial poses were tested, and as long as a valid pose was set, little differences were encountered in the number of iterations or in the result. The algorithm runs until the correction between iterations is below a limit value or the squared error does not change sufficiently. A maximum number of iterations guarantees a result. The result is valid if the sum of the squared errors in the distances between the projected model points and the image points is less than a preset value.

### 5.5.8 Pose Reporting

The pose is calculated as a four-by-four transformation matrix that contains both rotation and displacement. This matrix is converted to the three displacements and three orientation angles before reporting. The pose calculation is performed continuously, with the new calculation starting as soon as the previous calculation has ended.

## 5.6 POSE INTERFACE

The GUI developed for the CADS is implemented on the Datacube MaxTD host CPU using X-windows (X11R5) protocol and the Motif widget set. A high-level software functional flow diagram is shown in Fig. 15.

### 5.6.1 Initialization of GUI

The first step in the initialization of the GUI is creating and realizing all of the motif widgets that make up the GUI simulated dashboard display. The simulated image of the SPH port is mapped to the graphical pose widget window as a backdrop. Finally, the Datacube MV-200 is initialized, and a program is started which pipes live image data from the camera to the Datacube XI 24-bit graphics accelerator module. These live image data are then sent to the live video window of the GUI.

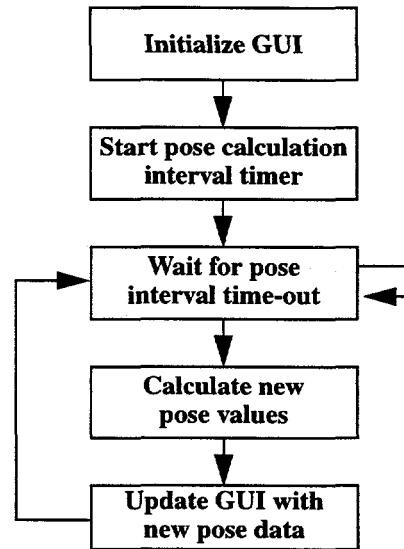


Fig. 15. GUI flow chart.

### 5.6.2 Pose Calculation and Interval Timer

The continuous cycle of pose calculation and subsequent GUI update is controlled by a software interval timer that can be adjusted to increase or decrease the GUI update rate. A software routine available within the X protocol is used to generate a software interrupt after some specified interval. When this software interrupt occurs, an X event is generated which is then captured and handled by the X event handler. When the X event handler receives this event, it makes a call to the pose calculation procedure described in Sect. 5.5. When the new pose calculation is complete, the GUI update function is called. After the GUI update is complete, the interval time is reset to generate another interrupt after the specified interval. This cycle continues until the operator either presses the “STOP” button to stop the pose calculation or presses the “EXIT” button to terminate the GUI application.

### 5.6.3 GUI Update

The GUI update consists of changing three areas of the GUI: (1) the six position and orientation gauges, (2) the graphical pose widget, and (3) the live video image overlay.

*Gauges.* The position and orientation gauge update takes the raw pose data describing the 6-D.F. pose and makes a separate call to the update routine for the corresponding gauge widget.

*Graphical pose widget.* Each time a new pose value is calculated, the previous graphical pose widget must be erased, and the new widget must be drawn to reflect the new pose. To preserve the underlying simulated SPH port image, the top two bit planes of the pseudocolor colormap were reserved for the red and green colors of the widget. Colormap entries 64 through 127 were reserved for green, and entries 128 through 255 were reserved for red. Using this technique, the simulated SPH image was preserved in the lower six bit planes within the colormap and did not have to be redrawn after erasing the graphical pose widget. This enhances the update speed of the GUI by 50%. Based on the new pose data, the geometry of the widget is recalculated before the new widget is mapped to the screen.

*Live image overlay.* The last item on the GUI that must be updated after a new pose calculation is the live image overlay. Using the hardware overlay capability of the XI 24-bit graphics accelerator module, the wire-frame overlay is erased and redrawn into XI memory after each pose calculation.

### 5.6.4 Operation

To start the GUI, perform the following steps:

- Move to the directory where the executable is located by ‘cd /home/cads/gui/build’.
- To start the GUI in an idle state (no initial pose calculation), type ‘dash’.
- To start the GUI with immediate pose calculations, type ‘dash -start’.
- If necessary, start the pose calculations and subsequent GUI updates by pressing the “START” button. To pause the pose measurement, press the “STOP” button.

There are two video option toggle buttons to change the appearance of the live image display window. To display the live image, arm the toggle button labeled "Live" by selecting it with the left mouse button. To turn off live image display, disarm the toggle button by selecting it again with the left mouse button. Similarly, the wire-frame overlay can be displayed or not using the "Wire Frame" toggle button. Arm this button by selecting it with the left mouse button to display the wire frame. Disarm the toggle button to turn off the wire-frame display. Note that the wire frame cannot be displayed without the live image also being displayed. To terminate the GUI, and the pose determination algorithm, press the "EXIT" button with the left mouse button.

### 5.6.5 Network Interface

Another task of the pose display function is communication with external devices, such as the control system of the robotic arm. The CADS does not actively control the arm; its sole purpose is to inform the control system of the position and orientation of the docking port. The control system has the responsibility to control the arm based on data from the CADS. The CADS and the control system communicate through a Unix socket protocol in a client-server model. When the CADS starts, it assumes the role of client by attempting to establish connection with a predefined host server. If the host is available, an Internet protocol socket stream is established between the CADS and the host. Sockets of this type are full-duplex byte streams, similar to pipes. A stream socket must be in a connected state before any data may be sent or received on it. The connection is created with a connect call. Once connected, data may be transferred using read and write calls. When a session has been completed, a close is called.

When a request for pose data is received from an external device, the CADS will reply with the most recent pose measurement. Normally, the pose measurements are made continuously, so the most recent measurement was made within 1100 ms. A pose request may be received at any point in the measurement cycle; therefore, the response latency is minimal.

There are two modes of network operation: active and listen modes. The network mode does not affect the pose measurement function. The pose data will continue to be displayed locally regardless of the network status. The active mode begins once a host is acknowledged on the socket. In the active mode, a pose vector can be sent to the host. The listen mode occurs when the CADS is operating but there is no host on the socket. In the listen mode, the CADS will continue to try to connect to the host, sending out a "connect request" every 5 s.

The CADS tries to handle all eventualities during network operation. In the active mode, it generates a "heartbeat" signal to verify that it is still connected to the host. The heartbeat signal is broadcast to the host every 5 s. If the host is connected, it should respond with a heartbeat reply. If the host fails to respond to the heartbeat within the allotted period, the CADS will close its socket and return to the listen mode and attempt to make a new connection, as illustrated in Fig. 16.

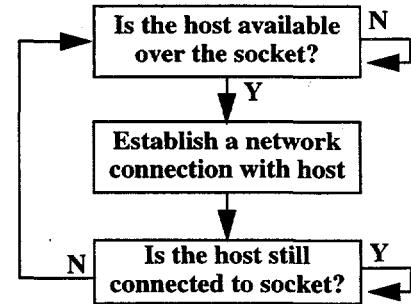


Fig. 16. Network interface flow chart.

## 6. PERFORMANCE SUMMARY

### 6.1 DESCRIPTION OF ENVIRONMENTAL CONDITIONS

A group of performance tests was conducted to characterize the baseline performance of the CADS and to make a qualitative assessment of performance under adverse conditions, notably rain and fog. The procedure included testing under ideal conditions to establish a performance baseline and under simulated adverse conditions for comparison. The baseline performance factors that were measured are update rate, measurement accuracy, continuous operation, and adverse weather. The adverse weather test conditions included rain, water droplets on the lens, fog, and vibration.

### 6.2 UPDATE RATE

The execution time required to calculate the docking pose is dependent on the two primary signal processing algorithms: target acquisition and pose determination. The target acquisition algorithm requires 20 consecutive video frames to extract the illuminated vertex markers from the background. For the camera sampling rate of 30 frames per second, the minimum time to capture 20 frames is 670 ms. The pose determination algorithm is recursive, so the time is dependent upon the number of iterations needed. The pose determination algorithm can be run on the master CPU or optionally on the arithmetic accelerator board. The pose determination normally takes 400 to 700 ms to complete. The typical total time to calculate target pose is 1300 ms using the CPU and 1000 ms using the accelerator.

### 6.3 MEASUREMENT ACCURACY

The absolute position accuracy of the CADS is not meaningful because of the difficulty in independently measuring the position of the docking target and the camera in 3-D space. The CADS measures the distance from the camera coordinate origin to the target coordinate origin. The camera coordinate origin lies inside the lens body and is physically inaccessible, and the target coordinate origin is a point in free space at the center of the docking aperture. The ethereal nature of the two origin points makes direct measurement impractical.

In practice, absolute positional accuracy has little meaning for docking. Of greater importance is the relative positional accuracy that can be measured indirectly by comparing the change in position reported by the CADS after a known change in the camera position. The measurement error  $\epsilon$  in percent of the actual value is given by

$$\epsilon = \frac{\rho(measured) - \rho(actual)}{\rho(actual)} \cdot 100 , \quad (16)$$

where  $\rho$  is the position coordinate of the camera.

The camera was positioned perpendicular to the target and moved in 5-mm increments toward the target. A total of 30 measurements was recorded, beginning at a distance of 400 mm and ending at 105 mm. The measurement error was computed according to the equation above. The CADS mean positional error was  $-0.3\%$ , the median error was  $-0.35\%$ , and the maximum error magnitude was  $0.9\%$ .

#### 6.4 CONTINUOUS OPERATION

The purpose of the continuous operation test was to operate the CADS over an extended period of time while recording the number of valid and invalid pose measurements that were made. The intent of this test was to determine the probability that a request for pose would be successful and therefore establish a baseline indication of short-term system reliability under ideal conditions. The system attempted 236,640 pose measurements over a 5-day period, with one invalid pose recorded during that period.

#### 6.5 ADVERSE WEATHER

Adverse weather conditions were simulated to make a preliminary estimate of the measurement accuracy and repeatability under actual outdoor conditions. The accuracy, given by measurement error in percent ( $\varepsilon$ ), describes how close the measured value is to the true value. The repeatability, given by the standard deviation ( $\sigma$ ), describes the variability of one measurement to another. For each weather condition, the standard deviation and  $z$  axis error were computed from 25 pose measurements of a stationary target, as shown in Fig. 17.

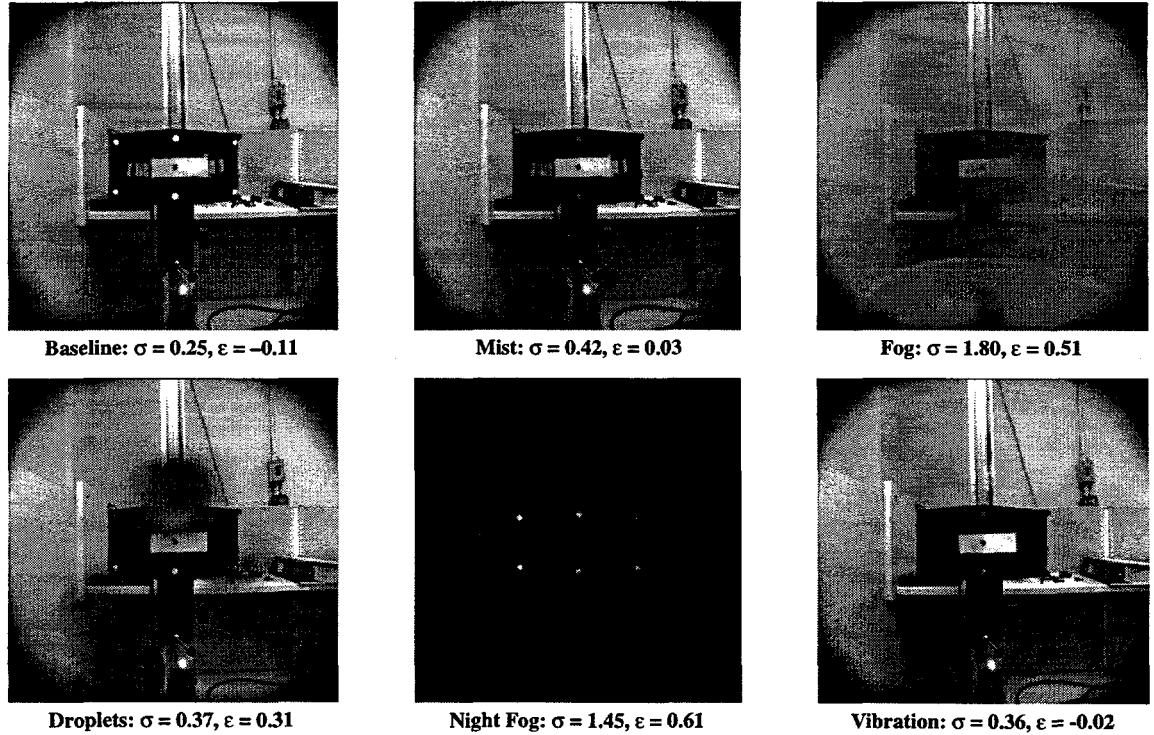


Fig. 17. Pose measurement statistics for simulated environmental conditions.

## **7. DIAGNOSTICS, MAINTAINABILITY, AND PRODUCIBILITY**

The current CADS is a laboratory prototype. Issues of diagnostics, maintainability, and producibility have not been considered beyond simple troubleshooting measures appropriate for a prototype system. The following section describes a preliminary diagnostic, maintenance, and production method.

### **7.1 DIAGNOSTICS**

An elementary diagnostics system could be incorporated into the CADS to detect gross abnormalities. The diagnostics approach would be to first verify that the CADS camera is operational and then check the signal processing boards. This could be accomplished by examining the dynamic range of intensity values from the camera output to verify the presence of an image. This type of self-test is useful to determine that the camera is working but cannot evaluate that the camera is pointing at a suitable target.

A calibrated test signal could be injected into the video stream in place of the camera. The test signal could be designed to elicit a predictable response from the CADS. The absence of the expected response to the test signal would be an indication of an abnormal condition requiring further investigation.

### **7.2 MAINTAINABILITY**

Few components in the CADS require periodic maintenance, notably, the optical surfaces of the camera, lens, and monitor. Maintenance tasks that may be necessary in the field include wiping the camera glass. The camera will probably be sealed in an environmental housing for field use and should not need additional care. There are no user serviceable components in the CADS chassis.

### **7.3 PRODUCIBILITY**

All components in the CADS are commercial, off-the-shelf items or could be fabricated using standard machining methods. The majority of components are electronic, and MIL versions are available. The primary producibility concern is the alignment and calibration of the camera. The position and orientation measurements are relative to the camera coordinate system and are independent of the resupply arm. On the other hand, arm motion and docking are based on the arm coordinate system. Therefore, a calibration is required to determine the position and orientation of the camera relative to the arm. The accuracy of autonomous docking is dependent on the precision of this calibration. Other factors affecting accuracy include calibration of the camera optical parameters such as focal length and distortion. Additionally, the dimensions of the port must be known a priori.

## 8. INTERFACE DEFINITIONS

The CADS has a single interface to the arm over an Ethernet local area network link, as illustrated in Fig. 18. The details of the network interface were given in Sect. 5.

The data sent to the host consist of a message header and pose vector. The message header has four fields, including the message type, data status, data size, and a blank field. The pose vector has six fields containing the x, y, and z coordinates and roll, pitch, and yaw orientation.

The CADS was designed for use with any robotic arm regardless of the specific geometry and kinematics. Because of the simplicity of the interface, the CADS can be easily adapted to different arm geometries. The CADS does not directly control arm motion; therefore, it does not require arm-specific data such as kinematics. This allows the CADS to switch from one arm configuration to another with minimal changes. For example, the CADS was deployed on a table-top robot for initial testing. The time required to reconfigure the CADS for the full-scale Modular Artillery Ammunition Delivery System (MAADS) arm was under 4 h.

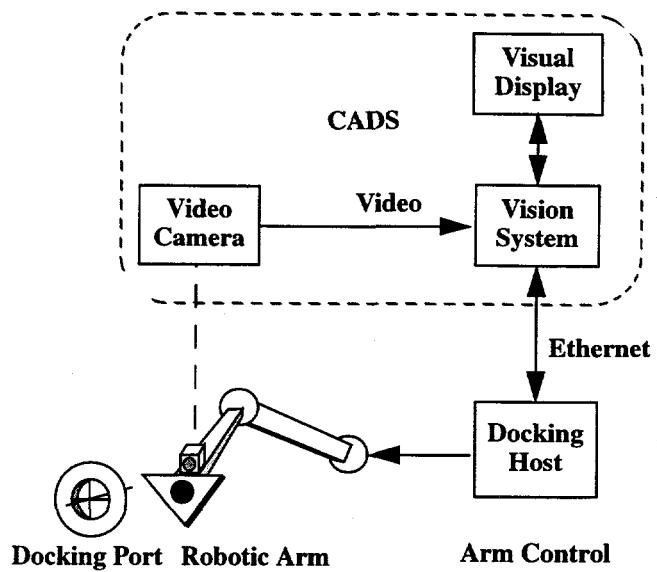


Fig. 18. CADS interface diagram.

## 9. TECHNOLOGY RISK ASSESSMENT

### 9.1 RECOMMENDATIONS

The CADS version 2 is a laboratory prototype intended to demonstrate that autodocking is technically feasible. The system has been tested under controlled conditions and has proven successful in the laboratory. The next phase of development for the CADS should focus on field applications.

The developers recommend that a field prototype of the CADS be developed. The field prototype should be targeted for use outside the laboratory and should incorporate the following items.

#### 9.1.1 Application Guidelines

The primary consideration in the development of the current CADS has been algorithm selection. The constraints imposed by the field operation were unknown at the time and were not considered in the CADS design. Some of the unresolved issues are operational, and other issues are vehicle specific. An application guidelines document that more clearly defines the objectives and constraints expected for Crusader autodocking should be created. The guidelines would be a living technical document that would evolve with the Crusader program. Key topics that should be addressed in the application guidelines are as follows:

- The selection of 2-D or 3-D image acquisition methods. The CADS presently uses a 2-D visual camera for image acquisition. The 2-D method requires illuminating LED markers on the SPH receiving port. If the SPH port markers are unacceptable in field use, additional algorithm development will be necessary to convert to a 3-D method such as a laser range camera. Laser range cameras are relatively recent innovations and have a higher development risk than the visual cameras. The 3-D method should be evaluated to determine its suitability for this application.
- Given that the 2-D image acquisition method is accepted, select a camera for use in the field prototype. The present options are a visual spectrum camera or an infrared camera. The visual camera is more easily interpreted by humans and would be an advantage if the operator participates in autodocking. The infrared system would be invisible to the naked eye during docking and could operate better in fog.
- The applications guidelines should define the maximum volume and weight restrictions for the CADS.

#### 9.1.2 Protection from the Environment

- A program to field test a CADS field prototype should be initiated. The present system has not been tested over the full range of environmental conditions that are anticipated in field use. Data gathered from the field test should be incorporated into the application guidelines.

- The current CADS camera is not environmentally hardened. The camera should be enclosed in an environmental housing to protect it from the elements.
- The system has not been tested over the full range of ambient lighting conditions. The camera may require a mechanism to automatically adjust the lens aperture for changing light conditions.

### **9.1.3 Packaging**

The electronic hardware should be optimized for pose determination. The boards in the present CADS configuration were intended to develop the pose algorithm. If the pose algorithm proves acceptable for field use, the algorithm could be implemented in custom hardware. Custom hardware would minimize the space and power usage of the CADS.

## APPENDIX D.2: PARKING ASSIST SIMULATION

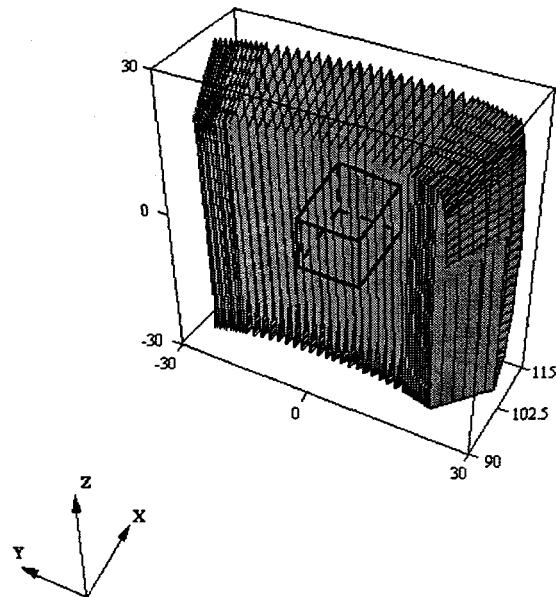
To assist the crew in parking the RSV within reach of the resupply arm, a parking assist tool that graphically shows the desired parking area for the vehicle was developed. The parking assist tool makes parking at the required standoff distance easier. Any decrease in the time needed to either park the RSV or dock with the SPH would increase the time available for ammunition transfer.

In the current concept pursued by ORNL, the mechanism that enables the RSV to transfer ammunition to the SPH is a 6-D.F. articulated boom. The available workspace envelope of the boom is defined by the mechanical limits of the various joints. For the ORNL boom, the available workspace is the shaded volume in Fig. A.1. This volume represents all of the points in space that are accessible by the boom's end effector when the base of the boom is fixed.

An additional constraint must be imposed upon the boom. Because the receiving port is fixed to the SPH vehicle and is not rotatable, the boom must be coaxial to the port upon docking. This requires that the boom end be aligned with the SPH axis before making contact with the port. When the additional constraint of coaxial orientation of the boom is imposed, the workspace volume reduces further to the parallelepiped shown inside the shaded region.

In practical terms, this means that the volume where the port can be positioned with respect to the RSV is limited. The port must lie within the parallelepiped of Fig. A.1 in order for the boom to successfully dock. When the RSV is being parked, the SPH port is stationary and the boom is mobile. For this case, the base point of the boom must lie within the constrained volume in order to dock.

Consider a schematic of the RSV and SPH during docking, as seen in Fig. A.2. The angle  $\gamma$  is measured counterclockwise from the RSV axis to the SPH axis.  $\alpha$  is the angle between the RSV axis and the first link of the boom, and  $\beta$  is the angle between the SPH axis and the first link. It can be shown that for docking to occur, the angles should satisfy the relationship  $-\beta \leq (\gamma \pm \alpha) \leq \beta$ . From this relationship, given  $\gamma$ , the approach angle of the RSV, then angles  $\alpha$  and  $\beta$  can be calculated. Observe that  $\alpha$  and  $\beta$  correspond directly to the joint limits of first and second linkages of the boom respectively. This allows one to predict whether the present heading of the RSV will permit docking with the SPH.



**Fig. A.1. Volume describing the extent of the boom's reach.** The shaded region is the boom's workspace constrained by the mechanical limits of the joints. The parallelepiped inside the shaded region is the workspace with the coaxial alignment constraint.

We make the assumption that the terrain does not change significantly over the small distance between the two parked vehicles. If the ground slope is relatively constant between vehicles, the vertical misalignment is negligible. Furthermore, the RSV driver has limited control over the height and pitch compared to the range and heading. Therefore, although the pitch and height displacement between the two vehicles will affect the size of the docking volume, they will be considered as constants.

A graphical display to guide the operator in parking the RSV with respect to the SPH was developed. The program runs on a Silicon Graphics workstation and is intended to graphically illustrate the constraints of parking and docking. The program also demonstrates a concept for using the CADS to also provide instrumentation for parking.

The parking assist program simulates a stationary SPH and an approaching RSV. The program allows the user to move the RSV forward or backward or make turns to position the RSV. The screen output of the parking simulation program is shown in Fig. A.3. In this figure, the lower object is the RSV, and the upper object is the SPH (with port). The smaller rectangle between the two vehicles is the calculated acceptable or safe parking zone for the RSV. The white triangle at the front of the RSV is the base point of the boom. The objective for simulated parking is to position the triangle within the safe parking area.

The acceptable parking area is the projection of the docking volume with the coaxial constraint onto the ground plane. The user can make turns (yaw adjustments) and can make lateral and approach distance adjustments. As the RSV approaches the SPH, the acceptable parking region is modified according to the  $\gamma$  provided by the simulated CADS. The safe parking zone in Fig. A.3 is determined by the approach angle of the RSV and by the extension capability of the boom. As  $\gamma$  increases, the horizontal extent of the safe parking zone decreases. In the limit, the safe parking zone reduces to a vertical line where the RSV must be placed in order to dock with the SPH.

A simulated CADS on the RSV can be activated in the program. An icon shows the field of view of the CADS camera mounted on the end of the boom. When the CADS is active, the program generates continuous pose measurements for the approach angle  $\gamma$  and the distance to the SPH.

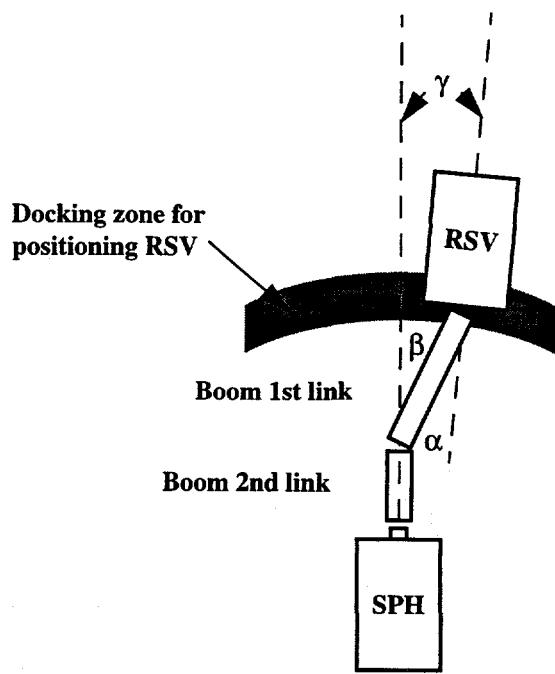
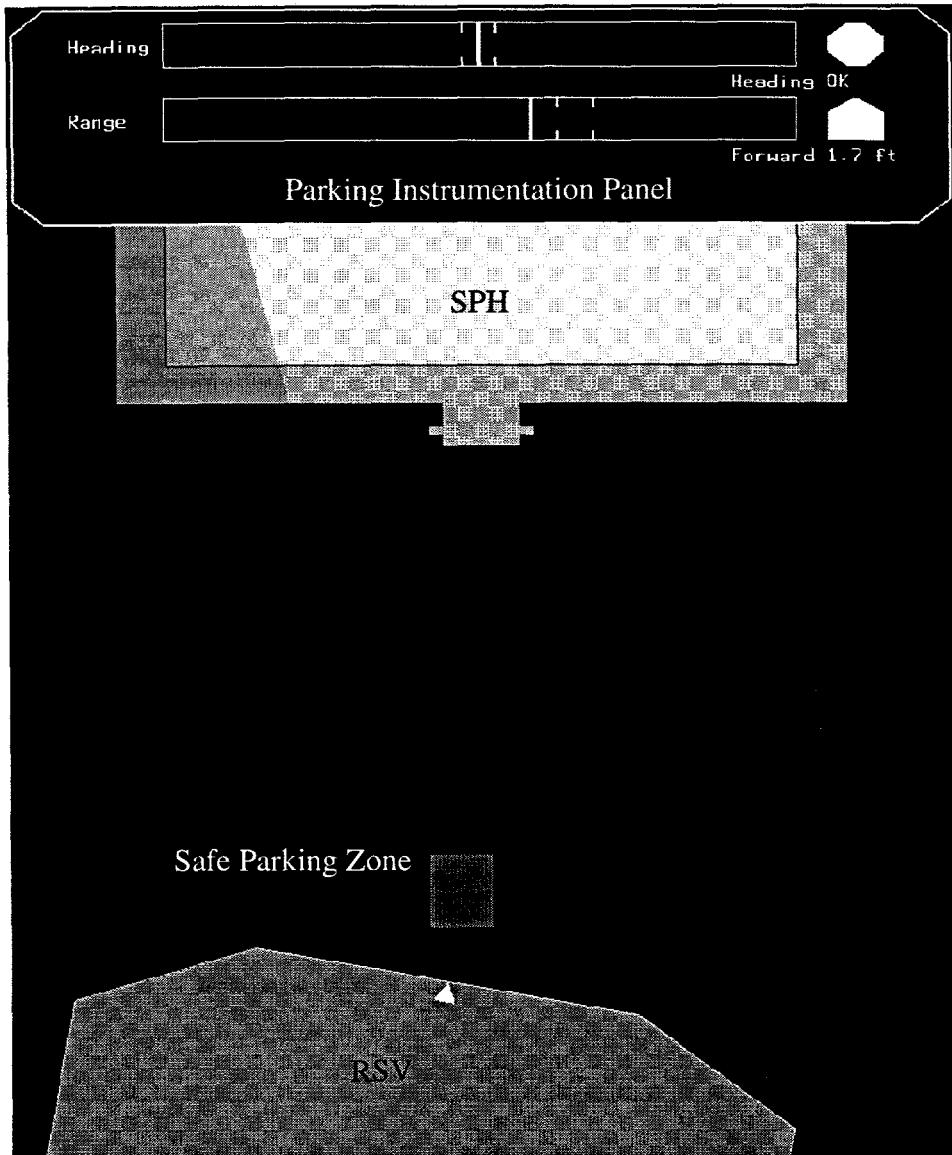


Fig. A.2. Top view diagram of the RSV docked with the SPH.



**Fig. A.3. Screen output of parking simulation program.**

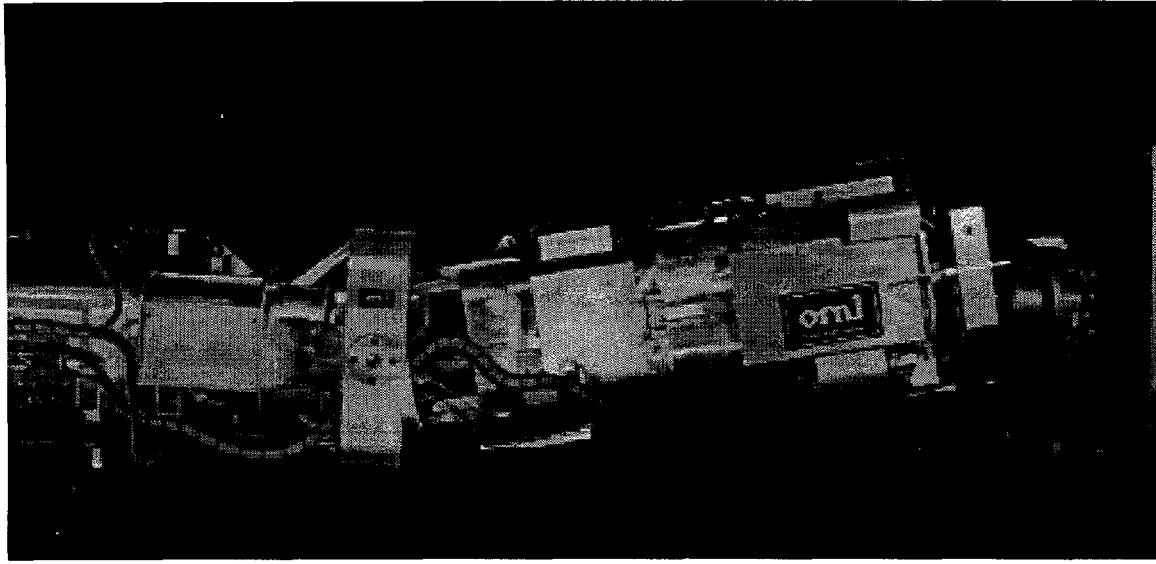
When the SPH port is within the camera field-of-view, the parking instrumentation panel is displayed on the upper portion of the screen. This panel consists of gauges to indicate the present heading of the RSV and the range to the SPH port. The gauges also show the minimum and maximum limits for acceptable docking. To the right of the gauges is an icon indicating the proper direction of travel for optimal parking. In Fig. A.3, the upper gauge shows that the heading is within acceptable limits but the RSV should advance 1.7 ft along the present heading. Tests with the parking simulation suggest that the preferred parking sequence is to stop the RSV when the port first comes in range of the camera. Then, the RSV heading should be rotated toward the center of the safe parking zone. The RSV can then be advanced until the boom base is within the parking zone.

If the operator advances the RSV without turning, the boom base (marked by a triangle) can be placed inside the parking zone. Once inside the area, a command to autodock will result in the program calculating the yaw angles of the joints and displaying the boom in docking configuration. Because of the previous assumption of small vertical gradient, only the yaw axis and extension of the boom are simulated.

The parking assist simulation could be implemented on the RSV to provide the driver with directions to reach the safe parking zone. This feature would significantly enhance the driver's ability to position the vehicle for docking and hence reduce the time for reloading.

### APPENDIX D.3: MAADS AUTODOCKING DEMONSTRATION

After the tabletop version of autodocking was successfully demonstrated, the CADS hardware and software were used to demonstrate full-scale autodocking on the MAADS boom. The MAADS boom is a 6-D.F. articulated transfer system. A photograph of the MAADS boom just prior to docking is shown in Fig. A.4.

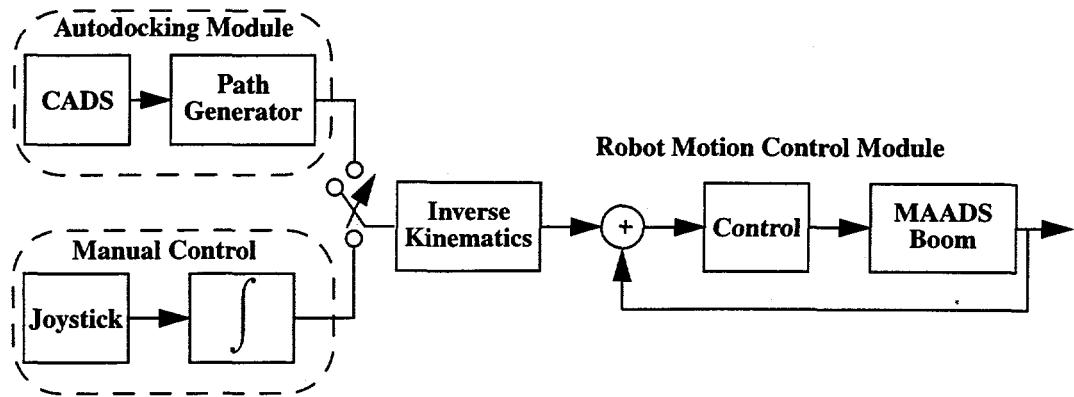


**Fig. A.4. Side view of the MAADS boom approaching the simulated SPH docking port.**

The incremental effort to incorporate autodocking into the MAADS boom was minimal. A software algorithm was created to convert the CADS pose vector to a desired path of travel for the boom. This path generator determines the exact path of motion needed to dock the boom from any random position in the docking envelope. The path directions are sent to the inverse kinematics algorithm, which, in turn, drives the individual MAADS linkages, as shown in Fig. A.5.

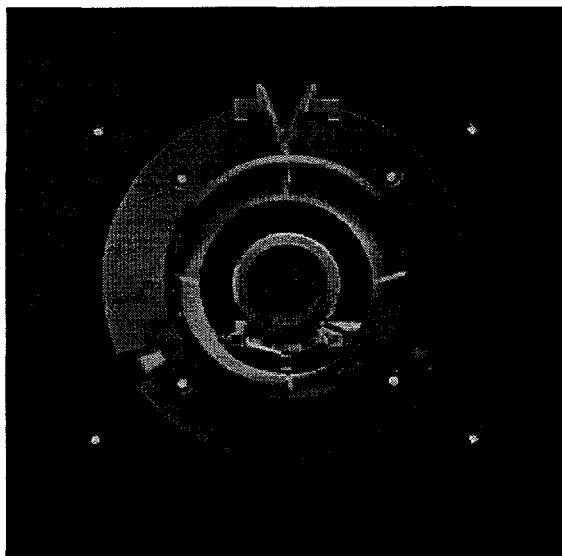
In the MAADS control architecture, the robot motion control module remained unchanged with the addition of the autodocking module. Either the manual joystick or the CADS can provide path data to the motion control module. The choice of manual or autonomous docking is software selectable by the operator.

Instead of the pentagonal shaped docking port used for the tabletop demonstration, a simulated SPH port (see Fig. A.6) was fabricated. The number of LED markers was increased to eight in order to make an accurate pose measurement from both the maximum and minimum camera range. The SPH port could be used for manual and autonomous docking without modification. The camera used for this demonstration was identical to the one used for the tabletop demonstration; so the intrinsic camera parameters were unchanged. The camera was mounted on the inside of the boom on a pivot arm that is rotated out of the path during ammunition transfer (see Fig. A.7).

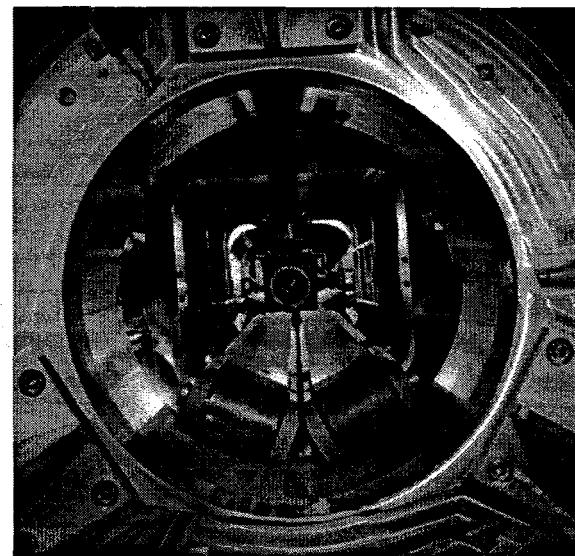


**Fig. A.5. Flow chart of the MAADS control architecture.**

To dock with the port, the location of the optical center of the camera with respect to the boom docking plane has to be accurately determined. This is because the CADS measures the position and orientation of the SPH port with respect to the camera. However, the robot motion control module needs the position and orientation of the port with respect to the contact tabs of the boom since that is the surface that will make first contact during docking. Therefore, all pose measurements must be converted to the boom's coordinate frame.



**Fig. A.6. View facing the simulated SPH docking port.** The array of eight LED target markers can be seen along the outer perimeter of the port.

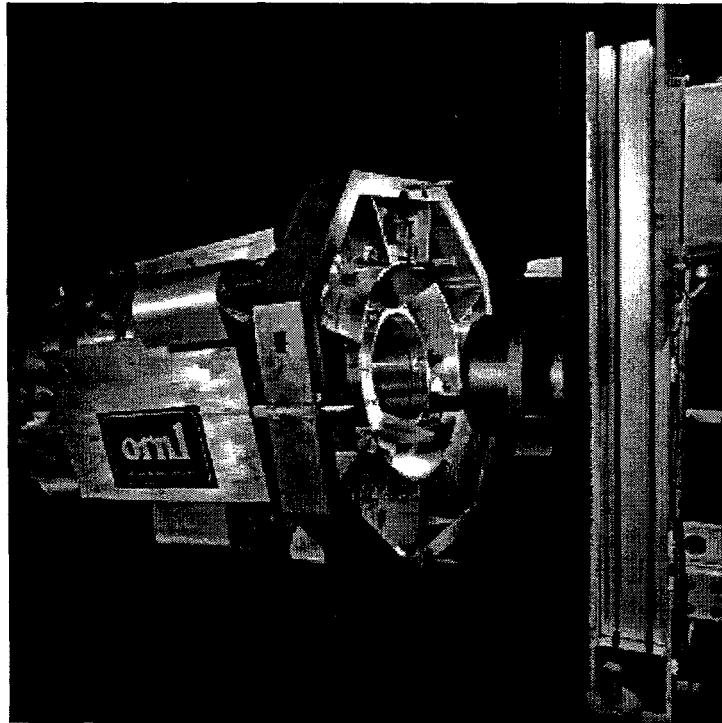


**Fig. A.7. View looking into the MAADS boom.** The camera used for manual and autodocking is located in the center of the boom.

To accomplish this, a coordinate transformation matrix is created. First, the geometric position of the LED lights with respect to the SPH port center is accurately determined. Starting from the docked position, the boom is retracted to bring the LED lights into the camera field of view. The position of the camera optical center with respect to the simulated SPH port center is just the pose measurement described in Sects. 3.4 and 5.5. Knowing the calculated position of the camera from the pose and the actual distance the boom (and thus the camera) has been retracted, the distance from the optical center of the camera to the boom mating face can be determined. The coordinate transformation matrix is retained until recalibration is needed. Normally, recalibration is necessary only if the camera position or orientation on the boom is changed.

Autodocking was accomplished in two stages. With the boom retracted, a pose measurement of the SPH port is obtained from the CADS. Because the measurement accuracy is proportional to the target range, it is desirable to take another pose measurement closer to the port. From the initial pose reading, a new position, directly in front of the port and approximately 10 in. away, is calculated for the second and more accurate pose measurement. The boom is then actuated to this new position, where the final pose measurement is taken. The coordinates of the port are computed from the pose measurement taken at the closer location. The boom can then be autodocked with the port. The camera must be kept fixed relative to the boom, during all pose measurements.

Fig. A.8 shows the MAADS boom about to autodock with the SPH port. The time required to complete an autodocking is less than 40 s from initiation of the autodocking command to camera retraction.



**Fig. A.8. MAADS boom about to dock with the SPH port.**

**INTERNAL DISTRIBUTION**

1-3.	W. B. Jatko	23.	L. M. Kyker
4.	R. K. Ferrell	24.	D. W. McDonald
5.	S. S. Gleason	25.	G. N. Miller
6.	J. S. Goddard	26.	C. E. Oliver
7.	J. S. Hicks	27.	K. E. Plummer
8.	K. W. Tobin	28.	S. L. Schrock
9.	G. T. Alley	29.	J. O. Stiegler
10.	J. H. Clift	30.	R. E. Uhrig
11.	B. G. Eads	31-33.	V. K. Varma
12.	D. C. Haley	34.	J. D. White
13.	W. R. Hamel	35-36.	Central Research Library
14.	J. H. Hannah	37.	Y-12 Technical Reference Section
15-17.	J. N. Herndon	38-39.	Laboratory Records
18.	R. A. Hess	40.	Laboratory Records- Record Copy
19.	J. M. Jansen	41.	ORNL Patent Section
20-22.	C. T. Kring	42.	I&C Division Publications Office

**EXTERNAL DISTRIBUTION**

43. C. M. Bloomhardt, Lockheed Martin Armament Systems, Lakeside Avenue, Burlington, Vermont 05401-4985.
44. M. Dolecki, Commander, USA TACOM, Attention: AMSTA-ZED (Mr. Matt Dolecki), Warren, Michigan 48397-5000.
45. N. F. Gravenstede, PM-AMMOLOG, HQ ARDEC, AMCPM-AL Picatinny Arsenal, New Jersey 07801-5001.
46. J. Fedewitz, PM-AMMOLOG, HQ ARDEC, AMCPM-AL Picatinny Arsenal, New Jersey 07801-5001.
47. T. Fitzgerald, Project Manager, FARV, Attention: SFAE-ASM-FR, Picatinny Arsenal, New Jersey 07806-5000.
48. G. L. Kent, PM-AMMOLOG, HQ ARDEC AMCPM-AL Picatinny Arsenal, New Jersey 07801-5001.
49. Ron Moore, The RM Group, Inc. 12024 Broadwood Drive, Knoxville, TN 37922.

50. David Norton, Houston Advanced Research Center, 4800 Research Forest Drive, The Woodlands, TX 77381.
51. M. M. Sevik, Carderock Division, Naval Surface Warfare Center, Code 70, Bethesda, MD 20084-5000.
52. A. Simon, U. S. Army, SIMCAR-FSS-DF, Attention: Alex Simon, Building 3159, Picatinny Arsenal, New Jersey 07806-5000.
53. Ernesto Suarez, Pratt & Whitney, P.O. Box 109600, Mail Stop 716-87, West Palm Beach, FL 33410-9600.
54. Office of Assistant Manager for Energy Research and Development, Oak Ridge Operations Office, Department of Energy, P.O. Box 2008, Oak Ridge, TN 37831-6269.
- 55-56. Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831.