A11103 458210

NISTIR 4463

NIST
PUBLICATIONS

SHORT-TERM EVOLUTION FOR THE FLIGHT TELEROBOTIC SERVICER

Ronald Lumia

U. S. DEPARTMENT OF COMMERCE
National Institute of Standards
and Technology
Robot Systems Division
Intelligent Controls Group
Bldg. 220/B124
Gaithersburg, MD 20899

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary

NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
John W. Lyons, Director

QC
100
.U56
#4463
1990
C.2

NISTIR 4463

SHORT-TERM EVOLUTION FOR THE FLIGHT TELEROBOTIC SERVICER

Ronald Lumia

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards
and Technology
Robot Systems Division
Intelligent Controls Group
Bldg. 220/B124
Gaithersburg, MD 20899

December 1990

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary

NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
John W. Lyons, Director

SHORT TERM EVOLUTION FOR THE FLIGHT TELEROBOTIC SERVICER

Intelligent Controls Group
Robot Systems Division

Principle Author: Ron Lumia
Date: Nov. 6, 1990

Document number: ICG-#25
Document version: 2.0
Document approval: _____

Revisions
1. Revision Author:
   Revision Date:
   Purpose:

Scope of the Document

This document identifies near term technology developments which would have significant impact on the evolution of the FTS toward autonomous operation. Analysis of anticipated FTS tasks is used to identify operations that might be performed autonomously rather than in a purely teleoperated fashion. Alternative techniques for automating these operations are then described. A discussion of FTS long term evolution is included as an appendix.

Keywords: space telerobotics, teleoperation, space station servicing, control system architecture, robot visual sensing, robot programming

TABLE OF CONTENTS

1.EXECUTIVE SUMMARY 4

1.1.FTS Technology Development Policy 4
1.2.Purpose 4
1.3.Approach 4
1.4.Targets for Automation Development 5

2.INTRODUCTION 7

3.APPROACH TO EVOLUTION 9

4.AREAS FOR AUTOMATION 13

5.NEAR-TERM AUTOMATION POSSIBILITIES 21

6.CONCLUSION 27

APPENDIX A 29

A.1.Introduction 30

A.2.Current Technology 30

A2.1.Teleoperation 30
A2.2.Robotics 31

A.3.Evolution of FTS Capabilities 34

A3.1.Telepresence versus Autonomy 34
A3.2.Realistic Goals 36

A.4.NASREM Development 38

A4.1.Servo Level Task Decomposition 38
A4.2.Primitive Task Decomposition 39
A4.3.Elemental Move Task Decomposition 41
A4.4.Task Level Task Decomposition 42
A4.5.Data Acquisition Sensory Processing 42
A4.6.Low-Level Sensory Processing 43
A4.7.Intermediate-Level Processing 44
A4.8.High-Level Sensory Processing 44
A4.9.Servo Support World Modeling 45
A4.10.Primitive Support World Modeling 46
A4.11.Elemental Move Support World Modeling 46
A4.12.Task Support World Modeling 47
A4.13.Data Acquisition Support World Modeling 47
A4.14.Low-Level Support World Modeling 47
A4.15.Intermediate-Level Support World Modeling 47
A4.16.High-Level Support World Modeling 48
A4.17.Servo Operator Interface 48

A4.18.Primitive Operator Interface 49
A4.19.E-move Operator Interface....49
A4.20.Task Operator Interface 49
A4.21.Programmer Interface 50
A4.22.Interface Development....50
A.5.Robot Metrology 50
A.6.Research 51
A6.1.Motor Behavior....51
A6.2.Machine Vision....53
A6.3.World Modeling....54
A.7.Conclusion 55
A.8.References 56

1. EXECUTIVE SUMMARY

The primary function of the Flight Telerobotic Servicer (FTS) Project is the development and deployment of a space qualified telerobot to support Space Station Freedom assembly, maintenance, and servicing. The requirements for the FTS machine combine the capabilities of currently existing robotic and teleoperated systems with control capabilities that are not operational today. This represents a wide spectrum of requirements ranging from good repeatability and accuracy for autonomous applications to advanced human-machine interface capabilities for teleoperation applications. A dexterous space telerobot of this nature has never been built. However, the basic technology for such a machine is in hand. The initial FTS Project technical challenge is, therefore, the integration of state-of-the-art technology elements into a reliable, spaceflight quality system.

The current state of technology dictates that the early operation of the FTS system must rely heavily on teleoperation techniques since the working environment cannot be as methodically structured as that of a factory floor. However, there are technologies, such as advanced computer vision and autonomous planning techniques, currently in the research phase which could greatly enhance the FTS capabilities to perform autonomously in a less structured work environment. Therefore, a specific requirement on the initial FTS design is that it have the capability to evolve as new technology becomes available.

1.1. FTS Technology Development Policy

The FTS Project does not directly sponsor technology programs. The primary sources of new telerobotic technology for use by the FTS are the Space Station Advanced Development Program, an element of the Office of Space Station (OSS), and the Office of Aeronautics and Space Technology (OAST) telerobotic technology development and demonstration program. The FTS Project will work directly with these organizations to establish its technology requirements. Technology elements produced under the sponsorship of these organizations will then be incorporated into the FTS as part of a planned evolution process. This process includes the use of the Development Integration and Test Facility (DITFAC) at the Goddard Space Flight Center. This facility will allow the integration and test of new technology elements in the context of the FTS architecture and FTS applications. Elements which are selected to be incorporated as upgrades to the FTS flight system will be implemented as flight qualified hardware as part of FTS development costs.

1.2. Purpose

This document represents an initial attempt to identify near term technology developments which would have significant impact on the evolution of the FTS toward autonomous operation. It is the FTS Project intent that the objectives and requirements put forth in this document be incorporated into the telerobotic technology development project planning activities of OSS and OAST.

1.3. Approach

A top-down approach was taken to the identification of high potential payoff technology developments needed for the evolution of the FTS toward autonomy. This approach was based on the analysis of FTS task scenarios.

The FTS Mission Utilization Team has performed a detailed, step-by-step analysis and scripting of a group of representative FTS tasks. These task scripts were examined with the objective of

characterizing types of task steps or sub-tasks that were common to all. As a result, "generic" sub-task definitions were developed (see figure 3). The actual scripted task steps were then mapped into the generic sub-task definitions, and a tally was done to see which tasks were most frequently performed. This process revealed that the sub-tasks performed most frequently are: (1) orient torso, (2) move arms to vicinity of work, (3) attach, and (4) detach. A further breakdown of these generic sub-tasks reveals that they have certain elements in common: alignment, path planning, and designation techniques.

Three primary task states were formulated from the above analysis:

(1) Path Planning - In this state the system and/or operator must plan and execute a path or trajectory through relatively large volumes of free space. This is thought of mostly in terms of end-effector motion and is primarily concerned with translation as opposed to orientation. Upon completion of activity in this state, the end-effector will be within the cone of acceptance for the non-contact alignment procedures.

(2) Non-Contact Alignment - In this state the system and/or operator must accomplish alignment of the end-effector with an object, fixture, or target with which it is to attach. The motion in this state is within a much smaller volume than for the path planning state and is primarily in orientation rather than translation. Upon completion of activities in this state, the end-effector is properly situated for the contact planning and control procedures to be activated.

(3) Contact Planning and Control - In this state, the system and/or operator bring the end-effector into contact with the environment, forming a closed kinematic chain. Motion is through very small volumes and compliance control (active, passive, or both) is required. Control stability is particularly critical in this state.

1.4. Targets for Automation Development

The FTS Project advocates a pragmatic approach to evolution toward autonomy. General solutions to problems such as image processing and path planning are not required before useful autonomous capabilities can be realized on the FTS. Selective structuring of the environment and engineering solutions to specific problems are appropriate steps to take until more elegant approaches become available. It is also very important to make efficient use of the operator. The system must allow the operator to move about from teleoperation to various levels of supervisory control smoothly and easily.

The three states discussed above (path planning, non-contact alignment, and contact planning and control) represent high priority areas in which automation could have significant positive impact on FTS operations. Clearly, unconstrained, autonomous path planning with its incumbent complex image processing is still beyond the state of the art. However, vision processing and path planning are not difficult for the human operator. The key then is to develop techniques that allow the operator to perform these functions in as painless and natural a manner as possible without requiring him to laboriously teleoperate each motion. He must have simple methods by which he can assess the situation, develop a plan, and impart his desired plan of action to the telerobot control system. Techniques for target and path designation using video overlays combined with a touch screen or a mouse appear to hold promise in this area.

For non-contact alignment and contact planning and control, various autonomous techniques are possible with today's basic technology. Such techniques will typically involve three basic com-

ponents: (1) sensor(s), (2) target, and (3) control algorithm(s). These three components can be integrated to form a system that provides the telerobot with a well defined capability. Such a system can be developed independently of the FTS prime contract by using the NASREM architecture to define where it will eventually interface. It can be breadboarded and delivered to the FTS DITFAC for assessment and testing.

2. INTRODUCTION

The approach to FTS evolution uses task analysis to identify the high pay-off areas where automation could be most effective. It also attempts to establish some short term achievable goals. While it is always desirable to develop a generic solution to a problem, it may not be technically feasible to expect a completely general solution. The goal for the FTS is not necessarily generic solutions. At this stage of FTS evolution, task structuring is permissible and engineering solutions are appropriate. The FTS evolutionary process should make appropriate use of the operator. Humans are excellent at image processing and planning. Consequently, FTS will not attempt to totally automate these difficult capabilities too early. In making the best use of the operator, however, there will be a conscious attempt to automate tedious sub-tasks. Furthermore, a serious investigation into methods for the operator to easily designate desired actions will be explored. Above all, the operator should be able to interact painlessly yet effectively with the FTS. It should even be fun to use. It is anticipated that the operator will be interacting with the system at various levels of the NASREM architecture during a task depending on the FTS competence. Figure 1 shows a typical sequence of task steps, along with the NASREM level, during which the operator interacts with the FTS during the execution of a task. The figure shows how the operator interacts with the FTS through different levels of NASREM. Certain actions, such as orienting the torso, require teleoperation while other actions, such as attaching the bolt runner, are more autonomous. As the FTS evolves, and greater levels of autonomy are implemented, the operator will interact at higher levels of the NASREM hierarchy, and thereby act more as a supervisor.

There is a basic dichotomy in the evolution of FTS activity from the operator's perspective. The FTS can move toward full telepresence or full autonomy. Reaching either extreme, full telepresence or full autonomy, is a long term, possibly unattainable objective. However, it is instructive to examine the possibilities. A system that consists only of telepresence implies that the human remains in the loop for all task steps. In a telepresence system, the human operator performs the tasks, his "presence" being translated to the remote worksite via the technology. This capability is very useful and even necessary for certain applications, especially where the environment is relatively unknown or unstructured. The development of systems which pursue true "telepresence" where the operator is immersed in sensory feedback will require much R & D. For more details, see Appendix A - FTS Long Term Evolution.

The nature of the FTS role on Space Station (i.e., assembly, maintenance, inspection, etc.) does not involve tasks of a scientific or totally unstructured nature where the human operator is actually performing some sort of investigation. Therefore, the FTS is choosing to pursue autonomy for several reasons. First, the repetitive chores performed by the robot are less onerous to the operator if the FTS had some form of autonomous capabilities. Second, the operator does not give up anything since he may break into any level of the NASREM architecture to take control at will. Third, time delays may preclude telepresence for useful applications. For example, in remote satellite servicing, the time delays incurred between the operator and the FTS preclude the use of force reflection and other sophisticated telepresence techniques. An autonomous mode of operation may be the only viable alternative. There may be only a sparse data return, e.g., one image per minute. A telepresence strategy of move and wait would be too slow to be a realistic option.

One of the reasons for choosing the NASREM architecture as the functional architecture for the FTS is that the system will be able to evolve with advances in technology. There are two basic types of improvement for the FTS:

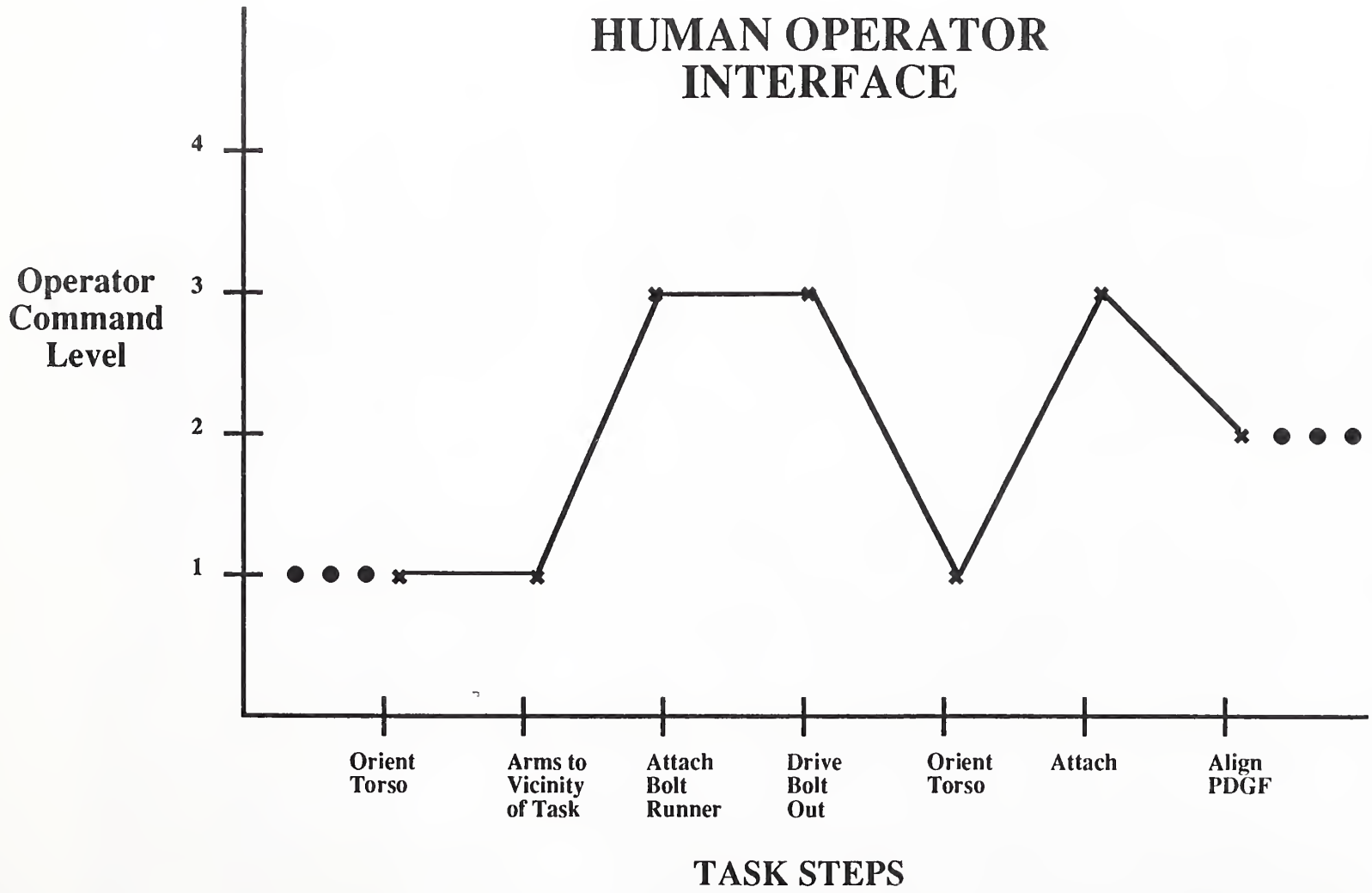


Figure 1: Human Interacts with Different Levels of NASREM During Task Execution

1. Human uses the operator interface at higher level

   The human operator can interface with the NASREM hierarchy at any level. As the system becomes more sophisticated, the operator can interact at higher levels. Interaction at higher levels allows the user to act more and more as a supervisor.

2. Enhance performance at a particular level

   There are two ways to enhance the performance at a particular level. The first way is to substitute superior but functionally equivalent algorithms inside the system, e.g., install a new control algorithm. The second way to enhance operation is to improve the way in which the operator interfaces with the system, making the interaction more natural.

Looking at several areas of improvement, each can be classified to belong to one or both of these evolution types:

AREA OF IMPROVEMENT	TYPE
Simplify operator's job	1,2
Improve robustness/probability of success	1,2
Improve speed/efficiency/productivity	2
Perform a new task	1
Improve safety (e.g., reflexive obstacle avoidance)	2

The original attempt at developing an approach to evolution focused on long term evolution and is included in Appendix A. The result was a generic view and while that is useful, it was perceived as not specific enough to serve as a guide for what to do next. Throughout this report, there will be references to particular sections of this long term evolution approach (Appendix A) for the interested reader. The rest of the report is organized as follows:

- Approach to Evolution

  The approach to determining high pay-off automation tasks is discussed. This involves presenting the analysis for FTS tasks along with the metrics which can be used to identify the areas which benefit most from automation.

- Areas for Automation

  Several flow charts are presented which describe in detail certain technological alternatives in automation.

- Near-term Automation Possibilities

  A near-term set of tasks which could benefit from automation is presented.

3. APPROACH TO EVOLUTION

The Mission Utilization Team (MUT) on the FTS project has developed a task scripting methodology and has scripted several representative FTS tasks with the expected variations. The script-

ed tasks, which were taken from the FTS RFP, include:

- -- Structural Interface Adapter (SIA) installation
  - FTS attached to mobile servicing center
  - FTS attached to astronaut translation device/ SIA outside truss
  - FTS attached to astronaut translation device/ SIA inside truss
- -- Installation of radiator panels to Electrical Power System (EPS) pallet
  - Diagonal truss removed
  - Diagonal truss not removed
- -- Diagonal truss member removal and installation
- -- Utility tray connection - electrical and fluid
  - Separate cables and connectors
  - Ribbon cables and multiple connectors
- -- Inspection tasks
  - Breakout box
  - O-ring
  - End of habitat module

These tasks are intended to be representative of the range of work which the FTS can perform rather than the limit of its capabilities. Figure 2 illustrates an example script for the highest level description of the EPS radiator panel installation with diagonal truss removal.

The method used to ascertain the best candidates for automation is based on these task analyses. The procedure is as follows:

1. Create a generic task
   The generic task is shown in Figure 3. It is clear that not all FTS tasks require each of the steps in the figure, and in that case the null step is used.

2. Map task scripts into the generic task
   The MUT was requested to take each of the task scripts and perform a mapping into the generic task description.

3. Summary of the task steps
   The steps for the complete task were catalogued. An example for steps 1 and 2 of the radiator panel installation is shown in Figure 4. This procedure was repeated for each of the tasks already scripted by the MUT. The final tally sheet, shown in Figure 5, shows tasks and how many times each operation was performed expressed as a percent of the total operations. The higher the percentage the more often the operation was done.

EPS radiator panel installation takes place in seventeen steps:

1. Unstow FTS from shelter.
2. Unstow PDGFs from shelter and store in payload bay.
3. Move to IEA bay.
4. Install PDGFs.
5. Detach diagonal truss member.
6. Position FTS for radiator panel installation.
7. Unstow magazine from payload bay.
8. Position magazine for radiator panel installation.
9. Attach radiator panel guide.
10. Install panels.
11. Stow magazine in payload bay.
12. Detach FTS from PDGF.
13. Attach diagonal truss.
14. Detach PDGFs from nodes and store in bay.
15. Position orbiter to shelter.
16. Stow PDGFs.
17. Stow FTS.

Figure 2. EPS Radiator Panel Installation with Diagonal Truss Removal

RETRIEVE ROBOT

TRANSPORT WORK SYSTEM (e.g. RMS, OMV, MSC) ACTIVE

FTS ACTION IS POSITIVE RELEASE OF THE GRAPPLE FIXTURE

TEST ROBOT

FTS IS ACTIVE SYSTEM

DELIVER ROBOT WORK SYSTEM

TRANSPORT WORK SYSTEM (e.g. RMS, OMV, MSC) ACTIVE

FTS ACTION IS A POSITIVE "GRAB" OF GRAPPLE FIXTURE

ORIENT TORSO

FTS POSITIONING SYSTEM IS ACTIVE SYSTEM

REQUIRES: OBSERVE, DESIGNATE, AND PLAN PATH, UTILIZATION OF FTS POSITIONING SYSTEM TO APPROACH ZONE OF MANIPULATION

MOVE ARMS TO VICINITY OF WORK

FTS MANIPULATOR ARMS ARE ACTIVE SYSTEMS

REQUIRES: OBSERVE, DESIGNATE, AND PLAN PATH, MOVE END EFFECTORS TO VICINITY OF WORK

ATTACH

FTS END EFFECTOR AND ARMS ARE ACTIVE SYSTEMS

REQUIRES: ITERATE UNTIL SEATING VERIFIED: OBSERVE/ SENSE, DESIGNATE, AND PLAN PATH, ALIGN (MAY USE VISION FORCE FEEDBACK, GUIDES, ETC.), MOVE

DO WORK

FTS IS ACTIVE SYSTEM

REQUIRES: OBSERVE, DESIGNATE ACTIVITIES FOR COMPLETION OF TASK AND PLAN PATH REQUIRED TO COMPLETE TASK

WILL INCLUDE ONE OR MORE OF THE FOLLOWING: MOVE, INSERT, DRIVE, PUSH, PULL, TWIST, TURN, ENGAGE, DISENGAGE, COUNT TURNS, FOLLOW, DEPLOY, LOCK, ALIGN

DETACH

FTS MANIPULATOR ARMS AND END EFFECTORS ARE ACTIVE SYSTEMS

REQUIRES: OBSERVE, DESIGNATE, AND PLAN RETRACT PATH, ALIGN/ NULL STORED ENERGY, DISENGAGE END EFFECTOR, RETRACT

MOVE ARMS TO SAFE POSITION

FTS MANIPULATOR ARMS ARE ACTIVE SYSTEMS

REQUIRES: OBSERVE, DESIGNATE, AND PLAN PATH, MOVE

Figure 3. Generic Task Definitions

The result shown in Figure 5 uses the metric of frequency. It is argued that any task which is performed frequently is a good candidate for automation. While this is a very useful metric, other metrics, such as power consumption, could certainly be used. Another common metric is time. However, it is not clear that time is a good metric. For remote operations where time delays can be significant, it is more important to perform the task reliably. Even when time delays are not significant, it is undesirable to automate complex operations that are rarely performed because the effort may not justify the payoff. It could be much more profitable to automate the most frequently performed tasks simply because those tasks will generate the most operator boredom and fatigue.

Based on the results of the task analysis, several conclusions may be drawn. First, it is clear that the "do work" part of the generic task represents only a small percentage of the activities required in the execution of a task since it is different for each FTS task. As a result, "do work" has not been identified as a high pay-off candidate for automation in the short term. However, the task analysis points out that there are several elements in common with all tasks: alignment, path planning, and designation techniques.

Figure 6 illustrates three task states: path planning, non-contact alignment, and contact planning and control. The purpose of path planning is to move a robot through large volumes of space, primarily in translation, so that it will be within the cone of acceptance for the non-contact alignment procedures to take over. Similarly, the non-contact alignment procedure moves the robot through a much smaller volume, primarily in orientation, so that a sensor is at a known position with respect to a target. This motion is sufficient for the contact planning and control procedures to finish the task where manipulator motion causes contact with the environment, creating closed kinematic chains. Clearly, during contact, stability is crucial. For example, consider the robot autonomously mating a connector. First, a path plan is executed to bring the robot arm close to the goal. Then, a non-contact alignment is performed to line up the two sides of the connector so that a straight insertion is possible. Finally, the contact planning and control procedure would perform a contact alignment using appropriate sensors (probably a force/torque sensor). Note that each stage of the process is handing off a problem from the domain of one sensor or sensor suite to the next. All tasks have elements from each box shown in Figure 6. Some of the capabilities could be pre-taught or simplified in some other way depending on the pre-structuring of the task and workpieces. The next section will expand the task states shown in Figure 6 by looking at the technological alternatives for each of the three boxes: non-contact alignment, path planning, and contact planning and control.

4. AREAS FOR AUTOMATION

In this section, three flowcharts are presented (Figures 7, 8, and 9) to illustrate the proposed alternative technology paths to meet FTS near term needs. There is an attempt to build a bridge from required capabilities to options of what to work on next. There are two fundamental issues involved with the FTS system:

- Operator specifies task

  The operator must tell the system what he wants it to do. This requires techniques to transfer information from the operator's mind into the system.

- System performs task

EPS RADIATOR PANEL INSTALLATION TASKS: WITH DIAGONAL TRUSS REMOVAL

TASK 1	RETRIEVE ROBOT	TEST ROBOT	DELIVER ROBOT WORK SYSTEM	ORIENT TORSO	MOVE ARMS TO VICINITY OF WORK	ATTACH	DO WORK	DETACH	MOVE ARMS TO SAFE POSITION	COMMENT	TOTAL STEPS
1d	1										
1e		1									
TOTAL	1	1	0	0	0	0	0	0	0		2

TASK 2	RETRIEVE ROBOT	TEST ROBOT	DELIVER ROBOT WORK SYSTEM	ORIENT TORSO	MOVE ARMS TO VICINITY OF WORK	ATTACH	DO WORK	DETACH	MOVE ARMS TO SAFE POSITION	COMMENT
2b				1	1					
2c						1				
2d							1			Drive bolt out.
2e							1			Move node fitting.
2f								1		
2g	0	0	0	2	2	2	4	2	0	Repeat 2b-2f 2 times.
2h				1	1					
2i						1				
2j					1					
2k						1				
2l							1			Unlatch and separate joint.
2m								1		
2o					1					
2p						1				
2q					1		1			Align PDGF.
2r							1			Engage and latch PDGF.
2s								1		
2t								1	1	
2u	0	0	0	1	4	3	3	3	1	Repeat 2h-2t
TOTAL	0	0	0	5	11	9	12	9	2	48

Figure 4. MUT Version of EPS Radiator Installation Task

SUMMARY (PERCENT)

TASK	CASE	RETRIEVE ROBOT	TEST ROBOT	DELIVER ROBOT WORK SYSTEM	ORIENT TORSO	MOVE ARMS TO VICINITY OF WORK	ATTACH	DO WORK	DETACH	MOVE ARMS TO SAFE POSITION
SIA INSTALL	1	2.5	2.5	2.5	15.6	20.5	18.0	12.3	18.0	8.2
	2	1.6	0.8	1.2	8.9	24.4	20.9	16.7	20.9	4.7
	3	1.9	1.5	1.0	9.2	23.8	22.3	13.1	22.3	4.9
	4	1.7	1.2	1.7	13.3	22.5	17.3	17.9	17.3	6.9
PALLET INSTALL INSIDE/ AWP	1	2.2	0.7	5.8	8.0	24.8	17.5	15.3	17.5	8.0
	2	1.9	0.0	4.9	8.0	25.3	17.3	17.9	17.3	7.4
	3	2.2	0.7	5.8	8.0	24.8	17.5	15.3	17.5	8.0
	4	1.9	0.0	4.9	8.0	25.3	17.3	17.9	17.3	7.4
PALLET INSTALL OUTSIDE/ AWP	1	1.8	0.9	4.4	11.5	20.4	19.5	13.3	19.5	8.8
	2	1.3	0.0	2.2	7.1	24.1	22.3	15.6	22.3	4.9
	3	1.4	0.0	3.6	11.5	21.6	18.7	16.5	18.7	7.9
	4	1.5	0.5	2.5	6.6	23.7	23.2	13.6	23.2	5.1
ORU CHANGE-OUT	1	3.1	0.5	5.2	9.9	20.9	16.2	18.3	16.2	9.4
	2	2.3	0.4	3.4	9.0	23.7	14.7	20.7	14.7	11.3
RADIATOR WITHOUT TRUSS REMOVAL	1	0.3	0.2	0.3	7.5	25.8	15.9	28.1	15.9	6.0
RADIATOR WITH TRUSS REMOVAL	1	0.3	0.2	0.3	7.6	27.0	16.1	27.2	16.1	5.3

Figure 5. Task Summary

TASK STATES

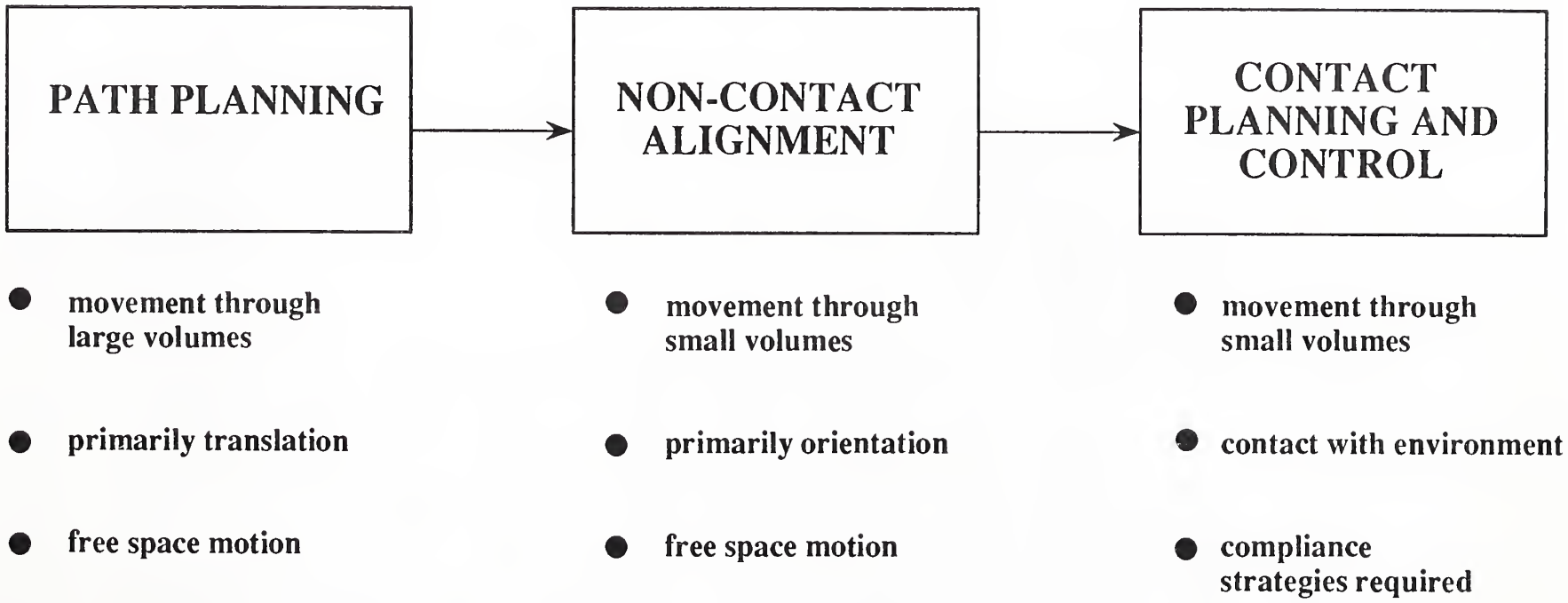


Figure 6: Task States

The system must have the ability to execute the given task.

In general, the human processes visual data, decides what to do, and then must convey his desires to the telerobot system. The FTS starts the task and reports when it is done. Certain methods already exist to perform some tasks autonomously. Others need to be developed. It is desirable for the operator to interact at higher levels of the hierarchy or at least to improve the fashion of interaction at a given level. The interaction must be painless and natural; the operator should not need to type lists of numbers, or perform complex operations on mechanical programming devices. Figures 7-9 provide an overview of potential solutions to the three primary steps of alignment, path planning, and contact/control. Using "decision tree" type process flow diagrams, one can outline the basic technological splits for each step in the robot control process as a function of "less" vs. "greater" environmental structuring, and subsequent "less" vs. "greater" certainty with which the robot can cope with the environment. The technological requirements in terms of hardware and software increase as the "uncertainty" factor increases.

Figure 7 illustrates the process of non-contact alignment. The purpose of the diagram is to illustrate alternatives and trade-offs rather than particular algorithms. The initial condition is that the *target* is in the camera field of view and that the *target's* orientation is obtainable from the camera image. Once this initial condition is satisfied, the system is capable of moving the sensor with respect to the target until a pre-defined geometric relationship is satisfied. Consequently, non-contact target motion is predominately in orientation rather than in translation. Alignment can be performed using pre-defined targets or work directly with objects. The alternatives are a trade-off between more structuring of the environment i.e., with greater certainty or more complex sensory processing i.e., with greater potential uncertainty. The result is that there is a target/algorithm/sensor triplet which is chosen. This is analogous to the world model/task decomposition/sensory processing triplet of the NASREM architecture.

Path planning alternatives are shown in Figure 8. In path planning, the goal is to move through a large volume of space primarily in translation in order to place a sensor close enough to a target to perform the previously described alignment procedure. First, the operator must designate a goal position, which could be in Cartesian space or joint space. Either the system can achieve the goal position completely autonomously or some combination of man and machine is required. Path planning is a very hard problem when attempting a completely autonomous solution. However, path planning for humans is nearly trivial under most circumstances. Certain routes within the flow chart allow a human-assisted path planning capability. The operator needs to view the scene and decide a prudent course of action. He needs the capability to designate "via points" for the robot if the autonomous system is unable to generate an acceptable path. As technology advances, it may be possible to perform path planning completely autonomously. Currently, some combination of human and machine planning appears most useful.

The process of contact planning and control is illustrated in Figure 9. Whenever contact is made with the environment, the robot or the environment must comply. The simplest way to achieve this compliance is to use passive devices such as a remote center compliance end effector. However, to use such a device effectively implies a certain amount of structuring of the environment. Even if a passive device is used, it is also possible to have other sensors, such as a wrist force/torque sensor to check whether the motion is proceeding "normally." This added complexity is compensated by greater safety and reliability. Active compliance can be performed by humans under teleoperation (with bilateral force reflection) or by autonomous control algorithms.

NON-CONTACT ALIGNMENT

(assumes motion will be within a small volume and primarily in orientation)

(primary source of data is a camera image and a low level vision system exists)

(initial condition is that target is in field of view and sufficiently large for processing)

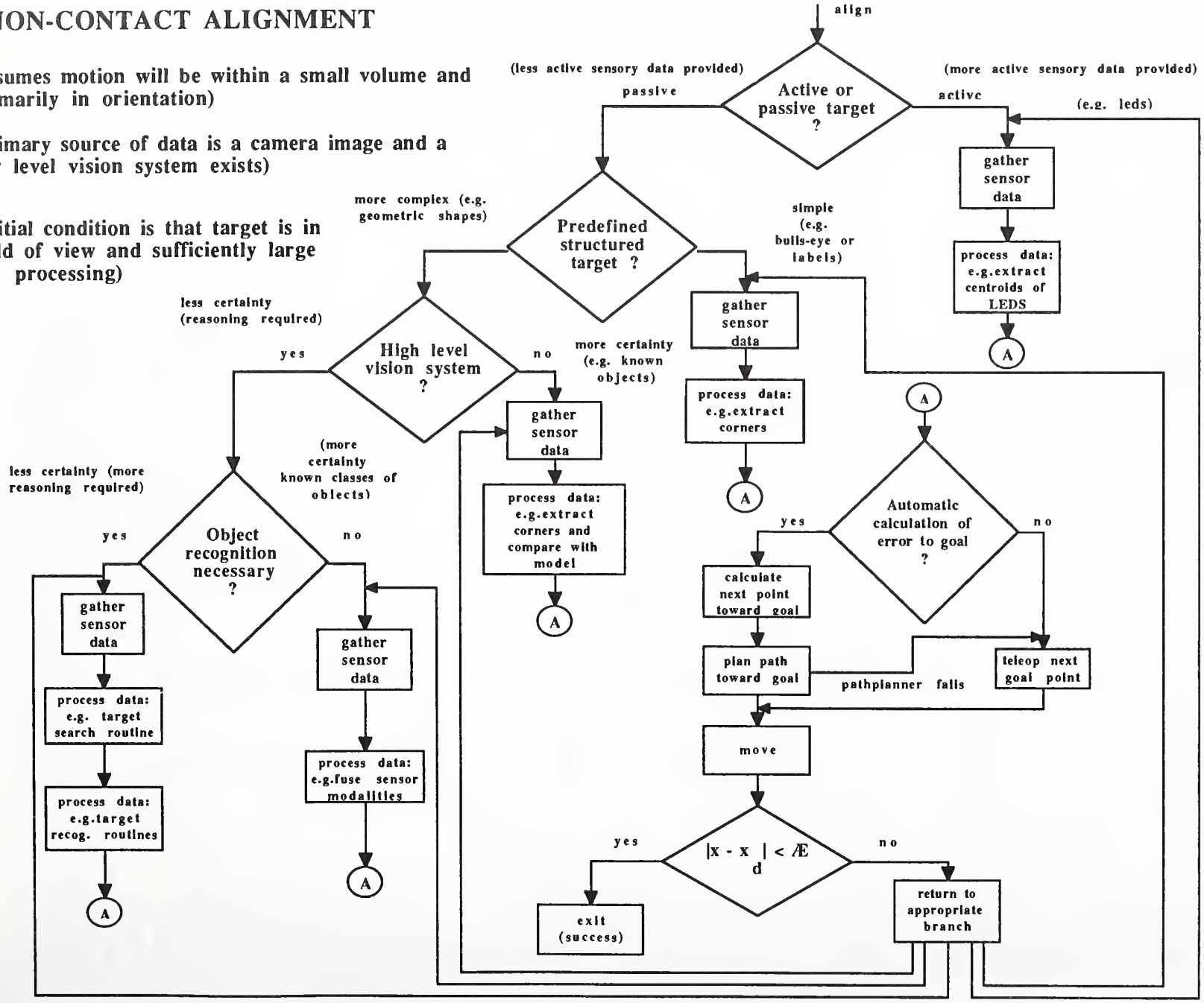


Figure 7: Non-Contact Alignment

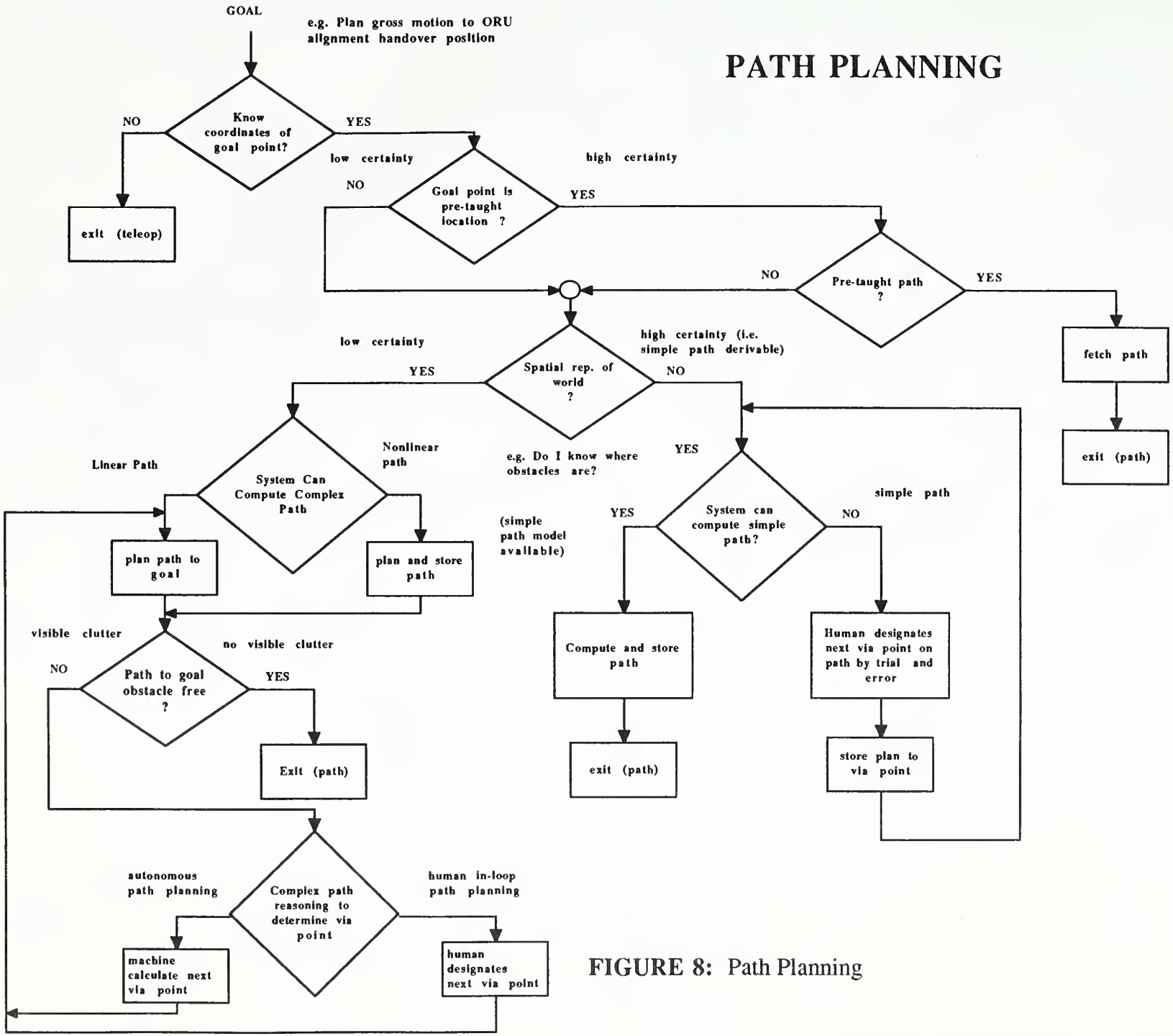


FIGURE 8: Path Planning

CONTACT PLANNING AND CONTROL

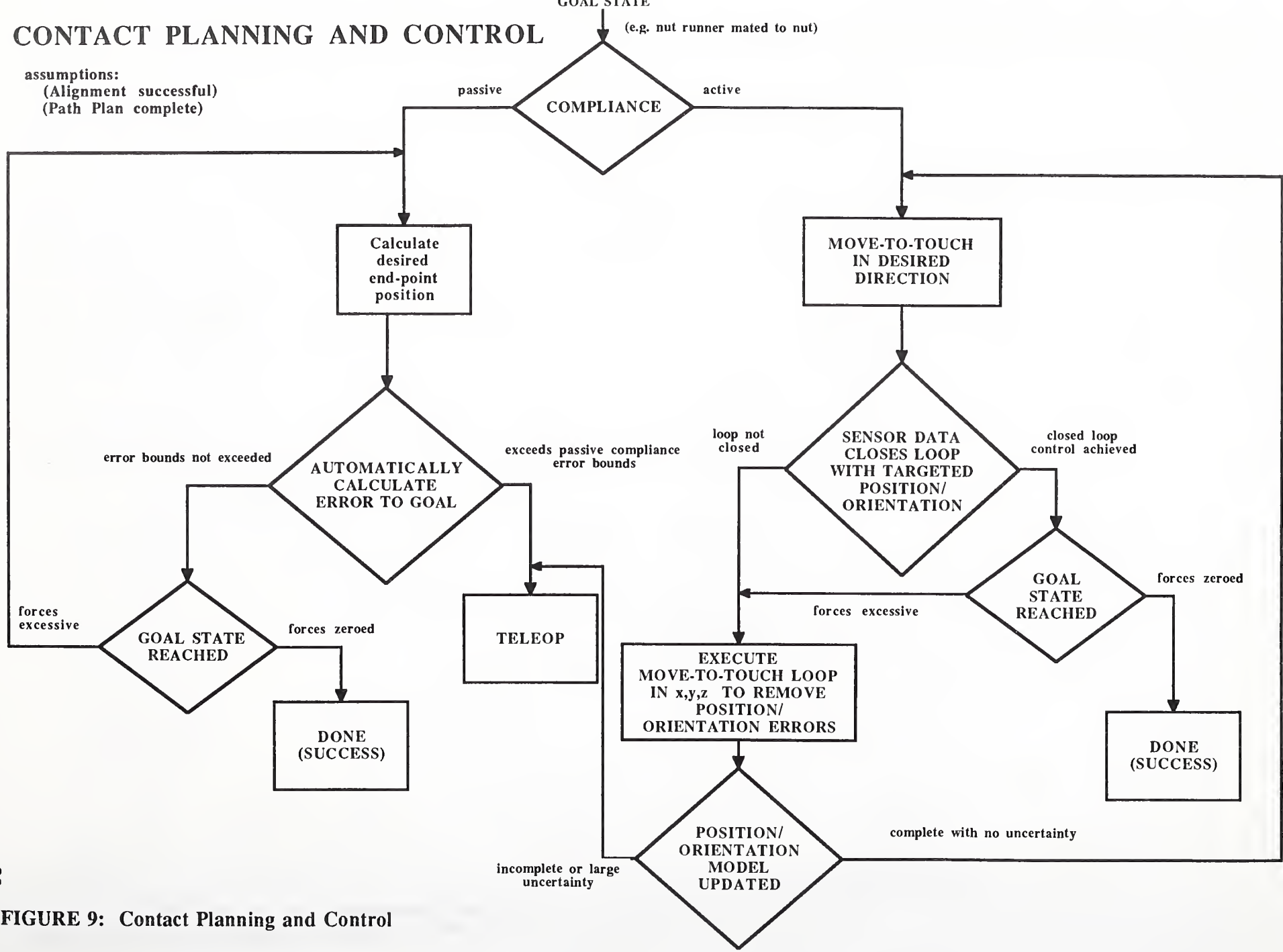


FIGURE 9: Contact Planning and Control

5. NEAR-TERM AUTOMATION POSSIBILITIES

A justification for automating the actions of alignment and path planning has been presented. However, depending on the time frame for development, different options could be chosen from the diagrams. For this report, a two year time frame is considered to be near term since it implies that all needed technologies must be ready for flight by 1991. Two areas are suggested as appropriate for automation.

TASK I. Autonomous Alignment Techniques

Non-contact alignment

Contact planning and control

Requires:

A target/ sensor/ control algorithm combination, or triplet that works in conjunction to achieve to the desired capability.

Figure 10 is provided in view of the 1991 near horizon for FTS operations. It suggests the appropriate technology required for non-contact alignment by the darkened path. It is assumed that a passive target or label is attached to objects of interest. Appropriate sensors are used so that robust control algorithms can perform the non-contact alignment automatically.

The concept is illustrated by considering a demonstration that was performed at GSFC by NIST personnel in 1987. It consisted of a robot with a camera that had an algorithm which allowed it to follow an active target comprised of four LEDs. The target was allowed to move anywhere in the workspace of the robot. The camera, mounted on the robot arm, fed information to a sensory processing algorithm which could extract the centroids of the LEDs and calculate the equation of the plane of the target. Since the LED spacing was known, all 6 degrees of freedom of the model could be determined. Then, a control algorithm for a motion which minimizes the error between the current location of the robot and the goal location, was executed. For this example, the goal was to place the robot end effector three inches away from the target and normal to it. This illustrates one particular target/algorithm/sensor triplet. This particular approach used an active target, but the path through Figure 10 requires a passive target. This is more difficult from a sensor processing perspective but highly desirable from an implementation perspective, because passive targets are less intrusive to the design of the workpieces. It should be noted that an alternate plan through the "more complex shapes" side of the technology flow diagram might be possible. In the near term, simple geometric shape modeling could be employed with a high level vision system to do object designation and verification. The list of potential technologies associated with alignment are as follows:

1. Near term (state-of-the-art)
   - Object labels
   - Teleoperation/autonomous verification of simple convex geometric shapes
   - Acquisition/tracking of stationary or slowly moving objects
   - High resolution / zoom cameras
   - Proximity/ranging sensing (e.g. radar, laser ranging)

NON-CONTACT ALIGNMENT

(assumes motion will be within a small volume and primarily in orientation)

(primary source of data is a camera image and a low level vision system exists)

(initial condition is that target is in field of view and sufficiently large for processing)

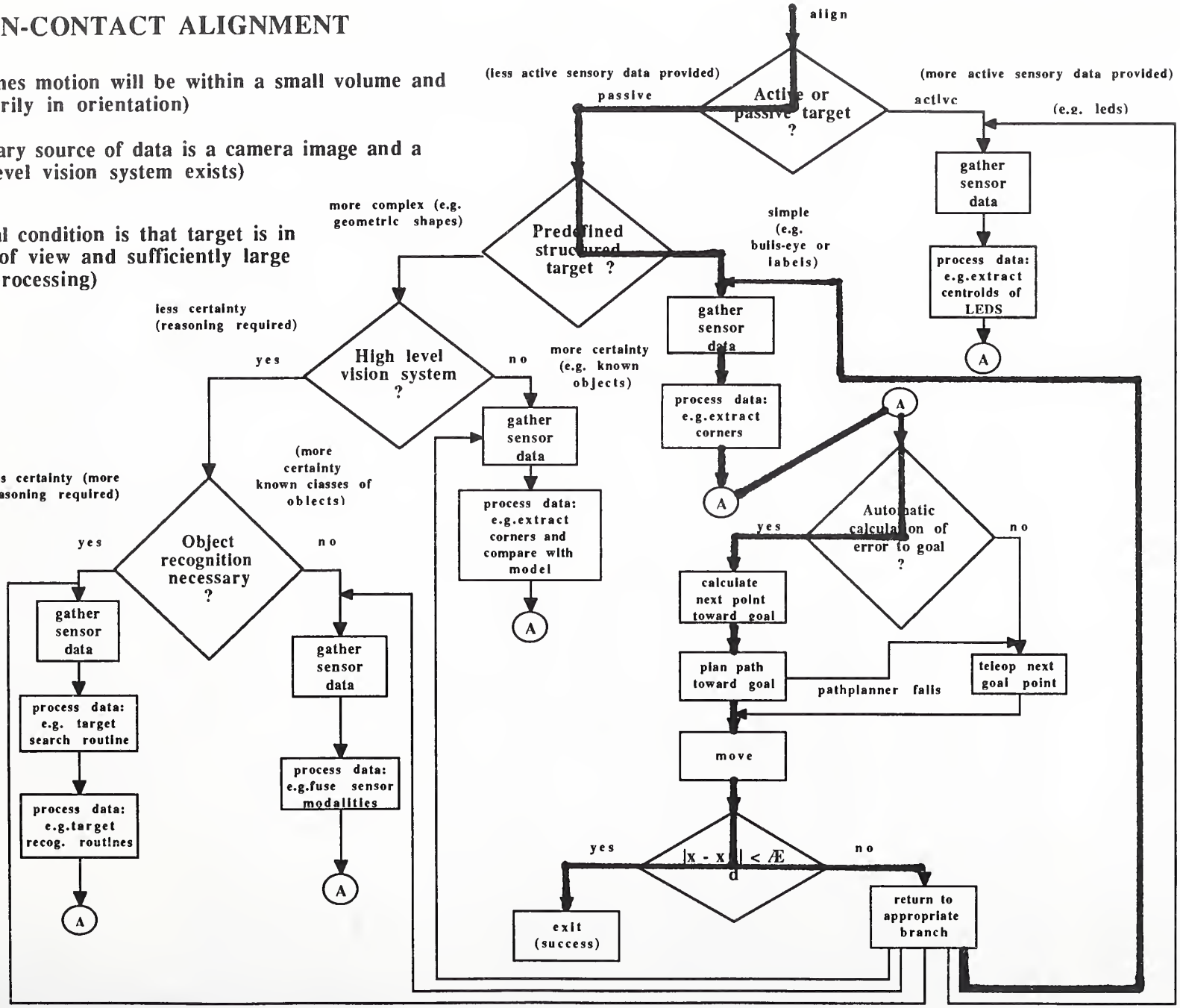


Figure 10: Non-Contact Alignment Trade-off Between Automation and Human Assistance

- Centroid determination using photo-emitting/sensitive devices
- Alignment by touch

2. Mid - far term technologies
   - Object models for concave and convex curved objects
   - Shape sensing with conversion to approximate object models
   - Autonomous verification of complex and concave objects
   - Real time world model updates
   - Autonomous acquisition/tracking of rapidly moving objects
   - High resolution full field zoom cameras
   - Fused vision sensing with uncertainty (object occlusion/ambiguities)

Figure 11 suggests the appropriate technology for contact planning and control. It would be desirable to develop techniques for active compliance because the environmental structuring is minimized. It is understood that the "passive compliance" side of the technology flow diagram will be employed in the near term. Furthermore, with passive compliance, techniques must be available to verify success. A more detailed description of the capabilities associated with the long term evolution is presented in sections 3.2 and 4 of Appendix A. The associated technologies are summarized as follows:

1. Near term single and dual arm control
   - Passive compliance devices
   - Model driven position control
   - Teleoperation teach pendant position control
   - Force/torque sensing with feedback to operator
   - Autonomous force/torque control
   - Pre-determined autonomous compliance macros
   - Operator assisted shared control

2. Mid-far term single and dual arm control technologies
   - Autonomous robust model driven active compliance
   - Real-time dynamic damping of contact induced arm /object oscillations
   - Contact control with time delay
   - Autonomous compliance with operator shared control

TASK II. Path Planning -- With Human Interaction

Designation of:

Where

Object

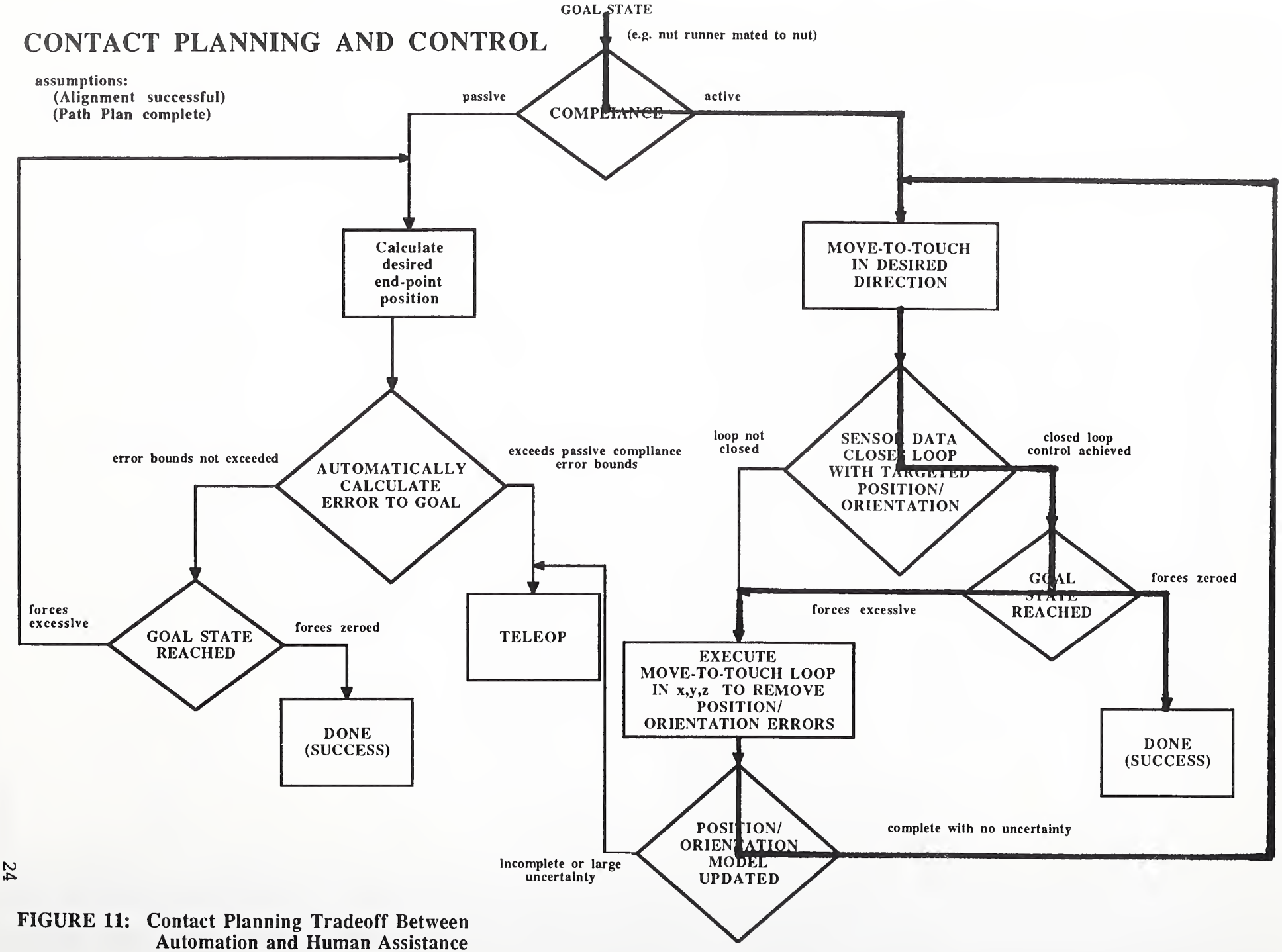


FIGURE 11: Contact Planning Tradeoff Between Automation and Human Assistance

Target

Via point

(This tells the system where the designated thing is.)

What

Object

Target

(This tells the system what the designated thing is.)

Figure 12 illustrates a particular choice for human assisted path planning. In this scenario, there is no stored spatial representation of the FTS workspace so that totally autonomous path planning cannot be performed. However, the system can plan straight lines from the current location to the goal location. The operator first designates a goal point for the FTS and the system attempts to plan a path to the goal autonomously. If the operator determines that the path is unacceptable for any reason, he designates via points through which the FTS must travel. Eventually, the system generates the last segment of the trajectory and the path is completely planned. This method uses the human for the vision processing and path decision processes but uses the machine for straight line planning between points. Certainly if the goal point location has a pre-taught path, then that technology branch is also available in the near term.

Path planning needs simple, easy to use techniques for designation. These techniques are probably visual with appropriate feedback, e.g.,

-- point to something on a screen

-- draw a path on a screen

-- overlay a model on a screen

An example of a designation technique is given in Figure 13. The camera and laser are attached to same pan/tilt platform but the laser has an independent pan mechanism. The operator designates the desired target with a cursor on his (2-D) screen and clicks a mouse. The camera mechanism then moves autonomously such that the centroid of the clicked object is in the center of the field of view. The operator then pans the laser (only pan is required because the camera and laser are on the same platform, making their boresights coplanar) to the target and clicks the mouse. The system reads the laser pan angle and calculates the range through simple trigonometry. The desired location or object has now been designated to the system in x, y, and z. This particular technique eliminates the need for ranging devices and has camera calibration capabilities intrinsic to it. A summary of required technologies is provided in the following list:

1. Near term (state-of-the-art)
   - Pre-taught path plan

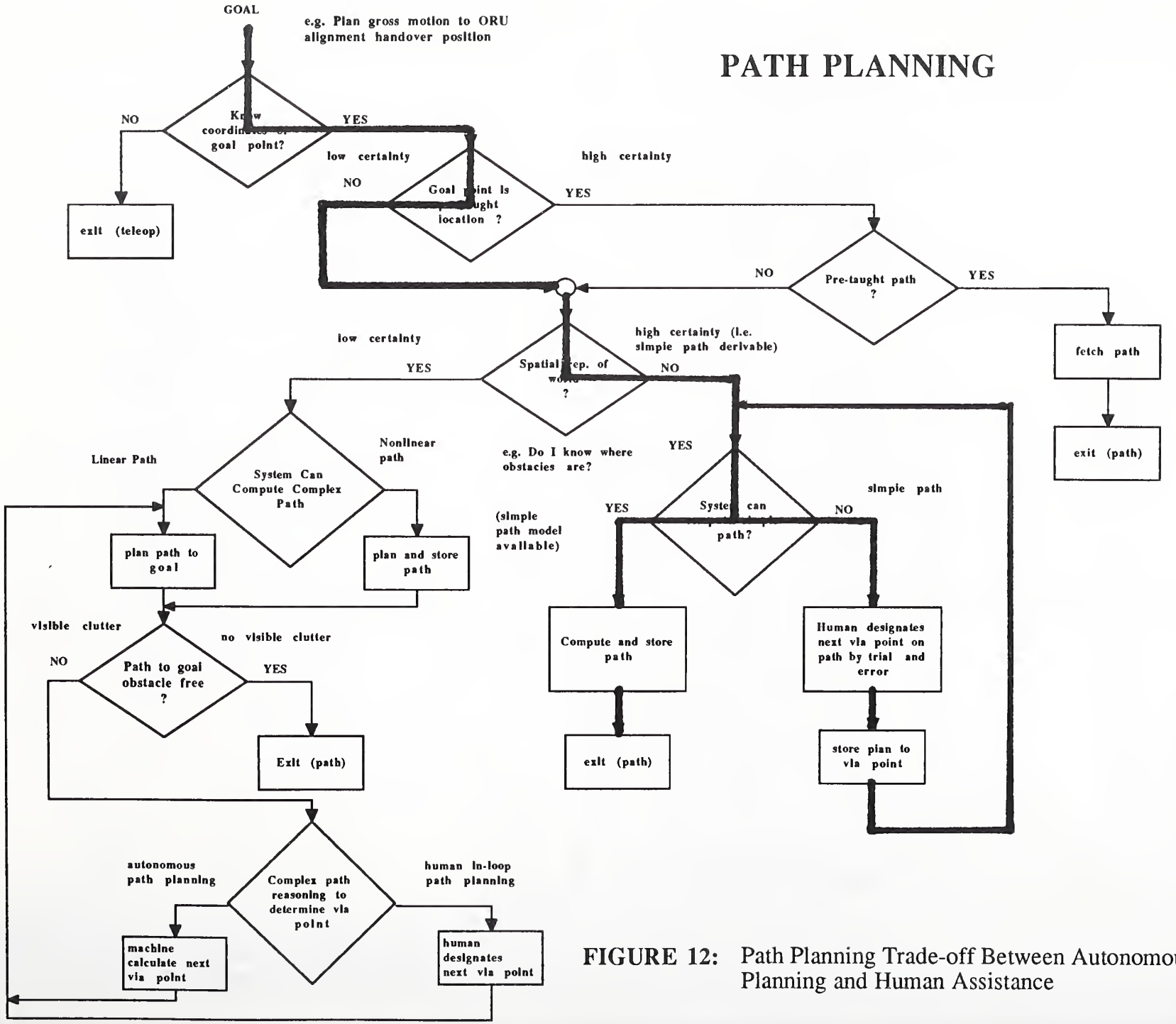


FIGURE 12: Path Planning Trade-off Between Autonomous Planning and Human Assistance

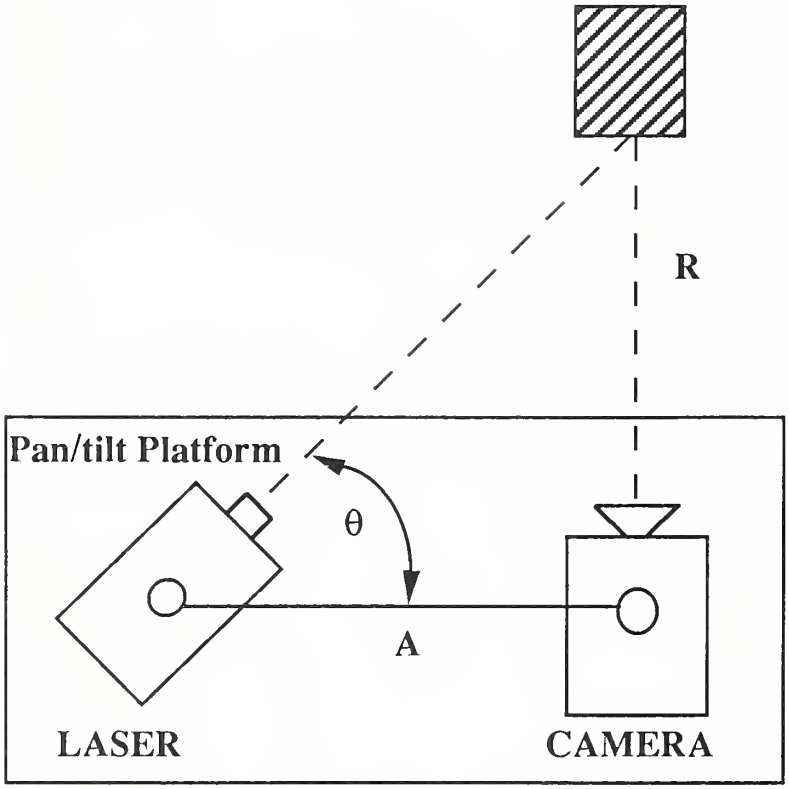
- Off-line spatial planning (operator designate/update)
- Off-line autonomous spatial planning when environment known
- Model driven collision detection
- Operator derived collision detection by touch

2. Mid-far term technologies
   - Model driven on-line task/spatial planning
   - On-line knowledge capture of new objects and interface with world model
   - Collision avoidance with the environment known or derived

A more detailed description of the capabilities associated with the long term evolution is presented in sections 3.2 and 4 of Appendix A. As stated in the introduction, the general solution is not sought but specific robust implementations in the two areas proposed. These techniques must be proven reliable in space-like conditions. Of particular importance is the ability to perform in the extremely high contrast and variable lighting conditions of space.

6. CONCLUSION

This report described the process followed for the identification of task which were high payoff candidates for automation. Two specific tasks, path planning and alignment, were described in detail for near term development. The technological trade-offs for the tasks attempt to take into account a reasonable balance between machine and human capabilities, taking advantage of each. A long term view of FTS evolution is presented in Appendix A.



$R = A \tan \theta$

- Camera & laser on same pan/tilt platform
- laser has independent pan mechanism
- operator designates desired target with cursor and clicks mouse
- camera moves to center target
- operator pans laser to target & clicks mouse
- system reads laser pan angle and calculates range

Figure 13: A Laser Designation Example

APPENDIX A:

FTS Long Term Evolution

Scope of the Document

This document describes the long term evolution of the FTS. The FTS is to evolve from a simple teleoperated system to an increasingly autonomous robot. Discussed are the requirements for increasing the FTS capabilities in the areas of teleoperation and autonomous operation. The feasibility of developing specific capabilities for the long and short term is discussed as well.

A.1. Introduction

The FTS has been conceived as an aid for the construction and maintenance of the Space Station and its support devices. The plan is for the FTS to evolve from an initially teleoperated system to a more complex system with increased autonomy and significantly increased capabilities. Ideally, the FTS would eventually be able to carry out tasks such as truss assembly and "EVA retrieval" [1] with no human intervention. Such a system would be an obvious asset in accomplishing the objectives of the Space Station.

Much of the evolution of FTS capabilities depends upon the application of existing technologies to the FTS problem domain. However, some capabilities will be attainable only by solving some fundamental problems in the field of robotics. The solution of fundamental problems requires what will be termed *research*, while the application and enhancement of existing technology will be referred to as *development*. In order to see which FTS capabilities fall into which category, it is necessary to examine the capabilities provided by current technology, as well as the overall desired capabilities for the FTS. This discussion is followed by a description of specific development issues for the FTS.

A.2. Current Technology

A2.1. Teleoperation

A variety of teleoperation technologies are currently available for use with the FTS. All teleoperation systems involve the remote control of one or more manipulators by one or more input devices. Various input devices have been devised in the last 30 years, along with a variety of algorithms to control these devices and the manipulator system.

Of the many types of input devices, the earliest ones were devices kinematically identical to the remote manipulator. These devices are replica *master* arms which are duplicates of the *slave* arms. A fixed-gain PD position control loop on the slave arm makes it follow the reference position of the master arm as the master arm is moved by the human operator. More recent developments include input devices such as exoskeletons that fit around the operator's arm, as well as joysticks and hand controllers for commanding the Cartesian position and orientation of the slave arm.

Cartesian controllers allow the operator to control the end effector position and orientation only. When the controlled device (slave arm) has redundancy, the operator must be provided with an additional input modality for controlling the redundancy. This makes controlling redundant systems more difficult with these input devices. There should really be some "automatic" way for the redundancy to be controlled. In addition, joysticks may not give as natural an interface as hand controllers, since joystick position corresponds to slave velocity, while hand controller position corresponds directly to slave position.

Exoskeleton systems provide a natural way for the operator to handle redundant degrees of freedom, e.g. seven-axis arms or articulated end effectors. In some instances the exoskeleton may not take full advantage of the robot's capabilities, as when the robot arm has considerably more freedom of movement than the human arm. However exoskeleton systems may be particularly useful for controlling articulated end effectors. This may be the only reasonable way to currently control these devices. Current technology consists of gloves (and even body suits) that are equipped with sensors to indicate the human operator's complete position. Unfortunately, these systems can-

not provide force-feedback.

Force-feedback control can greatly improve the effectiveness of a teleoperation system [2]. This type of control algorithm allows the operator to "feel" the resistance to movement in the slave arm, e.g. when the slave arm is in contact with the environment. The algorithm provides sensory feedback through the operator's own kinesthetic senses. To provide this capability, however, the input device cannot be a passive sensor, rather it must be actively actuated and controlled. Some early exoskeletal arm controllers were constructed to provide force-feedback, but these systems were quite bulky. More recently Cartesian hand controllers that can provide force-feedback have been developed [2].

Having "natural" kinesthetic feedback can greatly improve the operator's ability to perform teleoperation tasks. However, the most essential sensing for teleoperation is visual feedback. Visual feedback to the operator is normally provided by multiple cameras mounted near the workspace of the slave arm. The operator has several monitors that display the camera images. A problem with this approach is that it is difficult to judge distances from the monitor images. In addition, if the operator's hands are involved with the input devices, it may be difficult for the operator to change the camera views. (Sometimes camera view changes are handled through voice commands.) Recently much work has been done toward developing a helmet-mounted display that provides stereo *telepresence* to the operator. With this type of system the operator could possibly see the worksite as though he/she were actually present. The helmet can be attached to sensors so that the operator can adjust the camera view by moving his/her head in a natural way.

A full telepresence system would give the human operator the sense of being present at the remote worksite. In the ideal system, the operator would have the complete sense of working remotely through the slave manipulators, getting elaborate sensory feedback from the work site through some type of exoskeleton device. Although current technology would seem to support some amount of telepresence, much more work is required. A full telepresence system is far beyond the state of the art. This is discussed further in section 3.1, "Telepresence versus Autonomy."

A2.2. Robotics

Currently, robotics capabilities are quite limited. In fact, some have argued that the term *robot* is not appropriate for the devices that have been developed to date [3]. A *robot* would be something capable of useful, autonomous actions in *unstructured* environments, that typically operates for long periods of time without human intervention. No such device has ever been developed.

A2.2.1. Programming and Control

What has been developed during the last 25 years are sophisticated *automatic transfer devices*. These devices began as position-controlled manipulators which could be programmed to move to various positions within a fixed work space. Position control in these early devices was achieved via fixed-gain PID control technology, each axis of the device controlled by a separate PID servo loop. With appropriate position sensors this control technology is capable of positioning the manipulator to a point repeatedly to small tolerances. Thus, the device was useful for some jobs involving repeated movements to the same place if there was no need for any variation. (True robots would rarely find this type of behavior useful.)

These simple automatic transfer devices are programmed by a human operator moving the de-

vice to each position and storing the position in computer memory. The stored position, or taught point, can later be recalled and used as a reference for the position control loop. The manipulator moves to the taught location whenever this is done. This type of programming is quite burdensome for the human operator, and techniques were developed for simplifying this task. One of the principal techniques is to have the computer generate intermediate goal positions, i.e. a trajectory, between widely spaced goal points. With this approach, the operator need only teach-program a few points for each movement of the device, e.g. the start point and end point. This is approximately the current technology of commercial "robots" used in industry. This type of automatic transfer device is useless for tasks that require variation in behavior.

Recently, much effort has gone into developing more sophisticated manipulator systems. These systems use the same automatic transfer device technology described above, with the additional feature that all aspects of the manipulator motion are to be generated by the computer. Thus, instead of having the operator teach the start and end points, the computer generates these automatically based on a computer program developed by the operator, who must now be a computer programmer. In order for this to work, the manipulator must be modeled in great detail so that the motions generated by the computer program will accurately position the manipulator. If the computer motions fail to achieve the task goal the system can only abort the task altogether, just as in the teach-programmed case. One might argue that a better computer program could generate new motions to handle failures. For example, if the manipulator drops a part it is trying to move, the computer program would recognize this failure and generate new motions to reacquire the part and continue with the task. Development of programs that are capable of dealing with a significant number of failures has proven infeasible [3], however. As a result these systems have only been useful for repetitive tasks in highly structured, non-varying environments.

Most recently detailed models of the environment, just as for the manipulator, have been incorporated into the automatic transfer device controller. The models allow the computer program to take into account the environment in generating manipulator motions and processing sensory information. The subproblems of this approach, such as computer vision and obstacle-avoiding path planning, have proven so difficult that useful systems with these general features have not been developed. In addition, the amount of modeling required for this type of approach is extreme and assures, for all practical purposes, that the system will be limited to known, structured environments.

Even the most sophisticated of the current automatic transfer devices still use the PID position control to perform all motions. PID position control is designed to reject all disturbances in moving to goal positions. This means that contact with the environment, a necessary function in almost all tasks, is problematic in the sense that positions must be programmed with excruciating precision so that large contact forces do not develop. If there is variation in the environment, the PID controller generates excessive contact forces and the task often fails. To deal with this problem, automatic transfer devices are often equipped with passive compliance at the end effector. This compliance must be specially designed for the specific task and its usefulness is sensitive to environmental variation.

Another problem with PID control is that it cannot take advantage of contact force information to adjust the motion appropriately. There have been efforts to develop more sophisticated control algorithms that use force sensing. Such algorithms, although useful for some tasks, greatly increase the burden on the programmer. The problem of programming position information is difficult enough, now the programmer must be able to program the contact forces as well. Also, the problem

of infeasibility of exception programming still remains [3].

As this discussion has indicated, attempts to increase the robotic capabilities have resulted in systems that are increasingly difficult to program. Each advancement has called for models with more exquisite precision and the need to program more and more details of the motion. This has lead to intractably hard problems for application programmers, e.g. exception handling. Thus, there is a need to develop techniques that reduce the amount of programming and modeling that needs to be done rather than increase it. Certain recent results, such as impedance control, indicate some possibilities for new approaches. These approaches are a step in the (right) direction of considering contact with the environment as an acceptable, commonplace, and necessary occurrence, rather than as an exceptional circumstance. Further long term research along these lines, including that which improves our understanding of how humans can interact so easily with their environment, is required before any significant improvement in autonomous manipulative capabilities can be expected.

A2.2.2. Machine Vision

The main reason that current approaches to robotics must rely on fixed, a priori models of the environment in selecting behavior is the limitations of current sensory processing techniques. Machine vision is an example of an area of sensory processing which has been widely investigated in the past 20 years and still has limited application.

To date, there has been much work in the areas of image filtering, image segmentation, feature grouping, and object recognition. A wide variety of software exists for both commercial and research purposes. The majority of this software concentrates on the lower levels of image filtering and segmentation. The methods include a wide variety of noise removal, enhancement, edge extraction, corner detection, and region segmentation techniques based on color, intensity, or texture. In addition, research in the areas of symbolic representation of features has been done in the areas of shape, optical flow, and stereo. Researchers have also investigated the use of model-based prediction to accomplish object recognition.

Probably one of the most significant advancements in machine vision in recent years has been the introduction of computers dedicated to the real-time processing of iconic image data. The development of specialized hardware for image processing, such as PIPE and PIFEX, resulted from the recognition that the same basic operations are involved in many image enhancement and image segmentation algorithms. For example, almost all vision systems use the same basic type of convolution algorithm to extract edges. Thus, many of the basic filtering and segmentation algorithms can be implemented on the special purpose hardware. The result has been to move the bottleneck for processing time in image understanding from the enhancement and segmentation phases to the higher levels.

The advent of new hardware systems has done little, however, for the advancement of the robustness of image understanding. Most systems concentrate on trying to recognize a single object instead of analyzing an entire scene. Commercially available systems tailor assumptions to a specific structured environment. For example, inspection systems rely on consistent backlighting or precise placement to verify products. Intelligent information extraction in unstructured environments still presents many problems. Inconsistent lighting (a particular problem in space,) fine texture, sensor noise, and occluded surfaces are just some of the reasons that misleading information is extracted. These problems surface in numerous ways from the lowest levels, where false seg-

mentation occurs, to the upper levels, where features are grouped incorrectly and objects are unrecognized. The ability to analyze an entire scene to determine what is present and the robot's current situation still remains far beyond the capabilities of computer vision systems.

The overall goal of an image understanding system is to provide consistent and accurate information about the environment. In order to accomplish this, the system needs to evolve from one that provides reliable results within a highly structured environment to a sophisticated system capable of operating in unstructured environments.

A.3. Evolution of FTS Capabilities

In terms of capabilities for the FTS, NASA must have a device which allows control of waist, arm, and finger motions to accomplish truss assembly, ORU changeout, connector mating, elementary repairs, and other such tasks as described in [1]. The FTS will have a pair of redundant manipulators which must be used together to accomplish some tasks. Thus, the FTS must be capable of dual-arm coordination, including preventing the arms from colliding with themselves. In addition, the FTS must be capable of certain dextrous manipulation as might be involved in repairing equipment, such as removing and inserting screws and bolts, or mating various types of connectors and locking them into position.

Work has already begun on a Demonstration Test Flight (DTF) system that will provide two redundant manipulators controlled by hand controllers. Obviously, the capabilities of this system will be limited and more advanced capabilities will be added later. However, there are several directions FTS evolution could take. Which direction will be most productive and profitable? In response to this question, it is enlightening to compare the two extremes of telepresence and autonomy. T*elepresence*, as described in section 2.1, is a teleoperation system in which the operator is given the sensation of actually being present at the remote worksite. A*utonomy* refers to a robotic system capable of acceptable of carrying out high-level commands without human intervention, essentially a *robot* as described in section 2.2.

A3.1. Telepresence versus Autonomy

The basic DTF system, consisting primarily hand controller teleoperation, is a concession to the state-of-the-art. It has been demonstrated that such a system can be built, e.g. the Shuttle Remote Manipulator System, and will be capable of performing useful work in a reliable fashion. The work that can be performed by this system will be limited, however. In particular, it may be difficult to use this type of system for tasks that require dexterity, such as putting a nut on a bolt. The basic system must evolve to increase its capabilities.

Figure 1 depicts two extreme directions in which the FTS could evolve, toward increased telepresence or toward increased autonomy. Reaching either extreme, full telepresence or a true robot, is a long term, possibly unattainable objective, however it is instructive to examine the possibilities. A system that consists only of telepresence implies that the human remains in the loop for all FTS tasks. In a telepresence system the human operator performs the tasks, his/her "presence" being translated to the remote worksite via the technology. Thus, with telepresence astronaut time is required to perform all functions of the FTS. In an autonomous system, on the other hand, the astronaut would not be required to be present during the FTS operation. (In fact, a truly autonomous system could probably be operated as easily from the earth as from on board the space station.) This

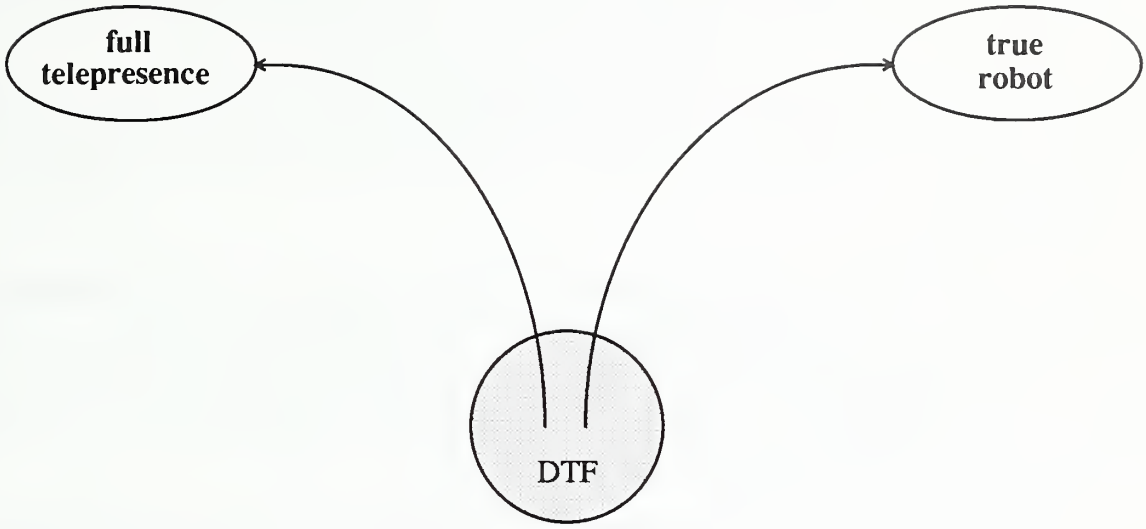


Figure 1. Telepresence versus Autonomy.

frees the astronaut to perform more creative, scientific functions, while the FTS performs the repair "chores" for the space station.

Development of telepresence requires work in many areas. Lighter weight, less bulky exoskeletons need to be developed as input devices. This problem is especially tricky for exoskeletons for the human hand, due to the complexity. Actuation is a real problem here, just as it is for the articulated slave hands themselves. Note that the use of exoskeletons (and telepresence in general) implies a large degree of anthropomorphism in the telerobot. This may not necessarily be desired, although it does ensure that an EVA astronaut will be able to perform the same task without the use of FTS.

Advanced input devices for telepresence will not be particularly useful without advanced sensory feedback systems, as well. As mentioned before, exoskeletons need to be equipped with force-feedback to let the operator detect environmental contact, stiffnesses, manipulability. These systems could benefit from tactile feedback as well. Tactile feedback appears to be a very long-range research topic. In terms of visual feedback, stereo helmet-mounted displays need to be improved for higher resolution stereo vision feedback.

If the path toward developing full telepresence seems arduous, the path toward full autonomy can only seem more difficult. As described in section 2.2, current technology is far from a true robot. New techniques need to be developed for robot control and programming, as well as sensing and world modeling, before an autonomous system can be developed.

To provide the FTS with sufficient capabilities in the short run, it is reasonable to rely on astronauts operating the FTS through teleoperation. Many of the FTS tasks could be accomplished by current teleoperation techniques, though frequently with some difficulty. Development of more advanced teleoperation systems, including some level of telepresence, would extend these basic

techniques and provide a reasonable level of functionality for the FTS for the short term. Eventually, techniques for autonomous control must also be developed so that astronauts will not have to spend extensive mission time performing teleoperated tasks. Automation of even some parts of the tasks, such as end effector changes, could relieve some of the burden on the operator in the short term. If, however, the FTS is to ultimately have capabilities similar to an EVA astronaut, i.e. unsupervised, autonomous operation in unstructured environments for long periods, then clearly much basic research is needed. Such a system would definitely require long term development.

A3.2. Realistic Goals

Figure 2 shows the areas of *development* (boxes) and the areas of *research* (ovals) that are involved in reaching certain levels of capabilities for the FTS. The figure starts from a basic DTF system which is almost exclusively teleoperated by hand controllers. Obviously, this system would have limited capabilities and would be difficult to use for complex tasks. Future developments would have to include progress in telepresence capabilities, as well as development of NASREM for automating the structured tasks of the FTS.

Near term projects in the figure are depicted by ovals and boxes with solid boundaries. The dashed projects represent necessary projects for the long term development of systems with truly advanced capabilities. The estimates of what constitutes near term and long term projects do not consider available resources, only that extensive new *research* is not needed to reach the short term milestones. It should be noted that the amount of work involved in (fully) developing even the short term capabilities is tremendous.

As depicted in the figure, development of a basic telepresence system with some limited autonomous capabilities should be possible in the short term (5 years). The autonomous capabilities would involve subtasks such as automatically changing end effectors, and locating and acquiring parts with special visual targets. It should be noted, however, that any estimated development times for limited capabilities depend on the degree of structuring applied to the tasks. For example, [1] describes a highly structured truss assembly task that would lie within the 5 year development period. However, an "EVA retrieval" task is described which lies outside the 5 year development period. The described task would apparently require grasping of moving, unknown objects while doing general obstacle avoidance. This task could probably also be achieved by the FTS using a type of butterfly net apparatus that was sufficiently compliant to not require extensive obstacle avoidance. This framework for the task might be realizable within the 5 year development period. The tasks capable of being developed in the short term depend on the degree to which they are structured for robot automation.

In order to achieve any autonomy, elements of the space station environment must be carefully structured. For example, objects to be viewed by the autonomous vision system must be painted with a flat finish to avoid specular reflection problems resulting from the uncontrolled lighting conditions. Parts to be mated must be properly chamfered and the end effectors must have the appropriate compliance for the mating tasks. It will be very difficult to automate operations that were designed primarily for human execution. Many things will need to be redesigned for robotic automation. This process is often called "Design for Automation" and is shown in the figure as a key development issue for the short term.

The best short term system for accomplishing the desired capabilities would be composed mostly of basic telepresence, with autonomous activities for tasks which can be appropriately

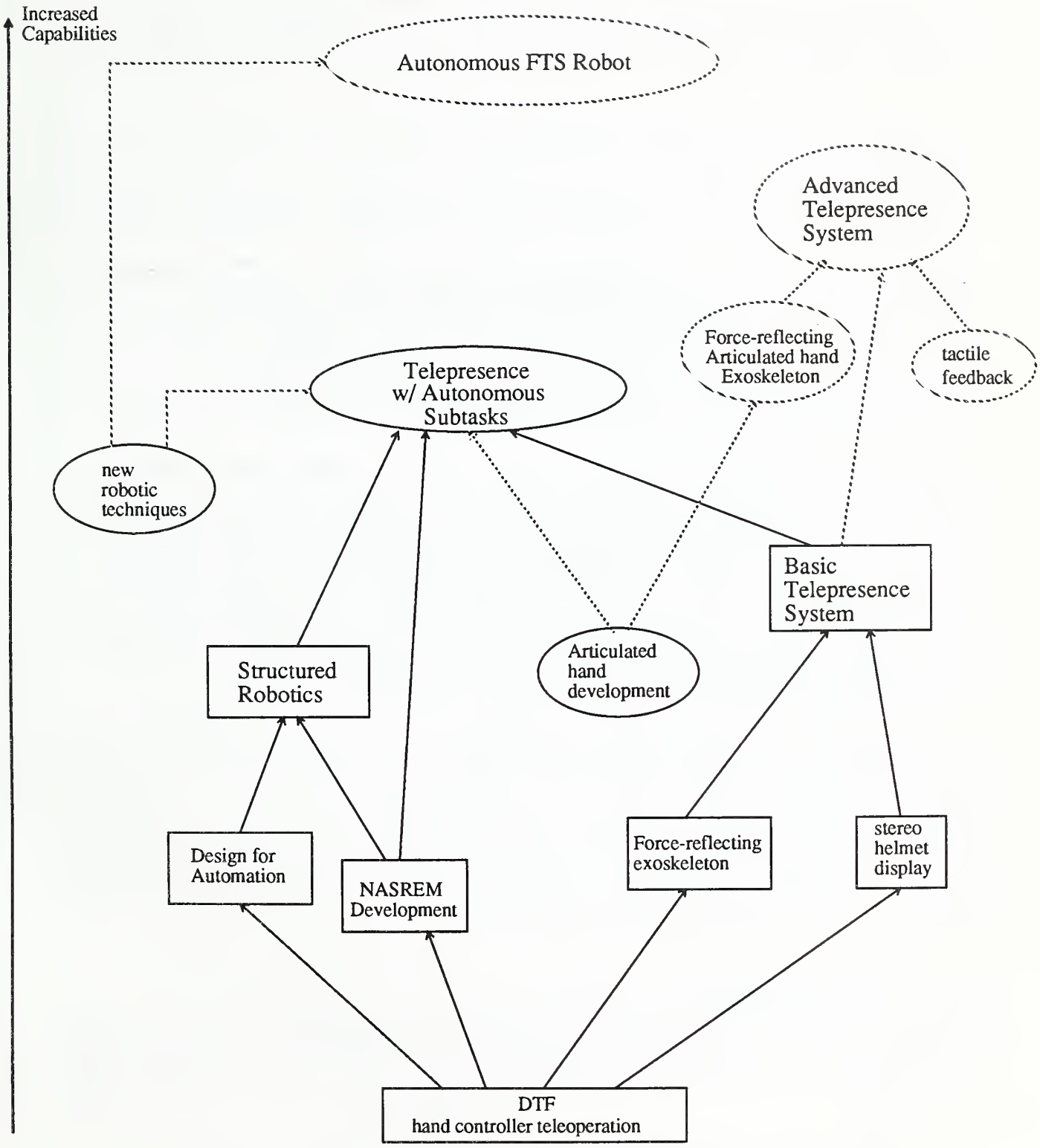


Figure 2. Systems with Increasing Capabilities

structured. Such a system could probably be developed without new *research* provided the tasks are properly structured. Therefore, it should primarily be a matter of *development*. A big part of the development would have to go toward improving the technology necessary for telepresence, as discussed before.

Among the tasks described for the FTS in [1] are a number of tasks that involve manipulations which have not yet been successfully developed for robots. One example of this type of task is the manipulation of flexible objects such as thermal blankets. Unlike rigid-body objects, the motions of flexible objects are often not predictable from models. For example, consider the task of shaking out a blanket to spread it on a bed. Since the environmental and initial conditions of the blanket cannot be known completely, the model cannot predict the blanket's complete motion and therefore cannot plan how to move the blanket such that the goal is achieved. Humans use sensory perception to determine the state of the blanket and adjust behavior accordingly. This is not currently possible with robot systems because of the limitations of robotic sensory processing. Thus, the robot cannot rely on the usual model-based approach to manipulating flexible objects. It is possible that the thermal blanket task could be structured so that the standard robotic techniques apply, but in general new techniques need to be developed for problems of this type.

All the systems in Figure 2 that need any degree of autonomy would obviously benefit from the development of new robotic techniques. In the current robot technology, a human operator would be required to monitor the system at all times in order to do exception handling, i.e. take over when there is a problem with the autonomous system. If this is not desired, or redesigning for automation is not desired, then research into new techniques would be necessary in order to achieve the desired functionality for the FTS. There is a trade-off between getting immediate results with approaches that can provide only limited capabilities, and making a commitment to long term research that has potentially big payoffs. Basic research has the added advantage that it offers more opportunities for developing technology that will be useful to American industry.

Since this basic research lies (for the most part) outside the established techniques for robotics, it is not shown as a part of the development of NASREM or any other current approach to the problem. It is certainly possible that such techniques could be incorporated with the NASREM approach, but the work is shown in the figure separately to emphasize the fact that it must be given special consideration as basic research. The problem of NASREM development is that of developing a telerobotic control system using the NASREM paradigm. This development is an important part of achieving the short term goals as depicted in the figure. The next section gives a detailed description of the elements of NASREM development.

A.4. NASREM Development

Figure 3 depicts the major components of a hierarchical NASREM control system for a telerobot. In order to achieve the desired functionality of the FTS each component must be developed to an appropriate level. The development required for each component in the figure is discussed in the following.

A4.1. Servo Level Task Decomposition

Servo level task decomposition is responsible for generating small motions of the controlled devices of the system, e.g. manipulators. These small motions move the system from one state to

another as commanded by the Primitive task decomposition component. Commercial systems rely almost exclusively on PID joint position control to implement this function. However, other approaches have been implemented recently in non-commercial systems. The motivation for a number of these approaches is described in [4].

- Compensation for the coupled, nonlinear rigid body dynamics and other dynamic effects. This "linearization and decoupling" is useful for supporting a number of advanced control techniques.

  Enabling Development - computer throughput, dynamic model and algorithm development

- Control of redundant degrees of freedom, including proper use of self-motion to optimize manipulability, torque distribution, and obstacle avoidance.

  Enabling Development - computer throughput, algorithm development

- Control in operational, or other coordinate systems, such as that of Cartesian force or visual space sensors. This requires sophisticated techniques for dealing with control near singularities.

  Enabling Development - computer throughput, sophisticated algorithms

- Stable, non-destructive control when in contact with objects in the environment. This includes force control, impedance control, and other constrained motion techniques.

  Enabling Development - development and evaluation of algorithms, development of models of contact tasks and closed kinematic chains

- Reflexive control features to ensure safety of system in areas such as servo error, motor burnout and over-heating, joint run-away, computer failure, and obstacle avoidance.

  Enabling Development - development and evaluation of algorithms, development of system models

- Ability for operator to specify in a natural way the performance and task domain, constraints which automatically utilize redundant DOF's. For example, reaching around obstacles and continuing to work without (a human) consciously controlling the configuration of robot to avoid obstacle.

  Enabling Technology - algorithms, human-machine interface for specification.

A4.2. Primitive Task Decomposition

The Primitive level task decomposition plans motions which are large in a dynamic sense. It plans "smooth" trajectories between two robot configurations. This planning includes dynamics of the robot and payload. Commercial systems employ only a very simple form of position trajectory planning that uses trapezoidal velocity profiles. Although many more methods for trajectory planning and Primitive level control have proven useful in experimental systems [5], these techniques are not currently employed commercially.

- Force Trajectory Planning - The combination of position and force control is essential for the completion of assembly operations.

  Enabling Development - appropriate representation for desired constrained motions, methods of determining what force trajectory will accomplish a particular task, a means

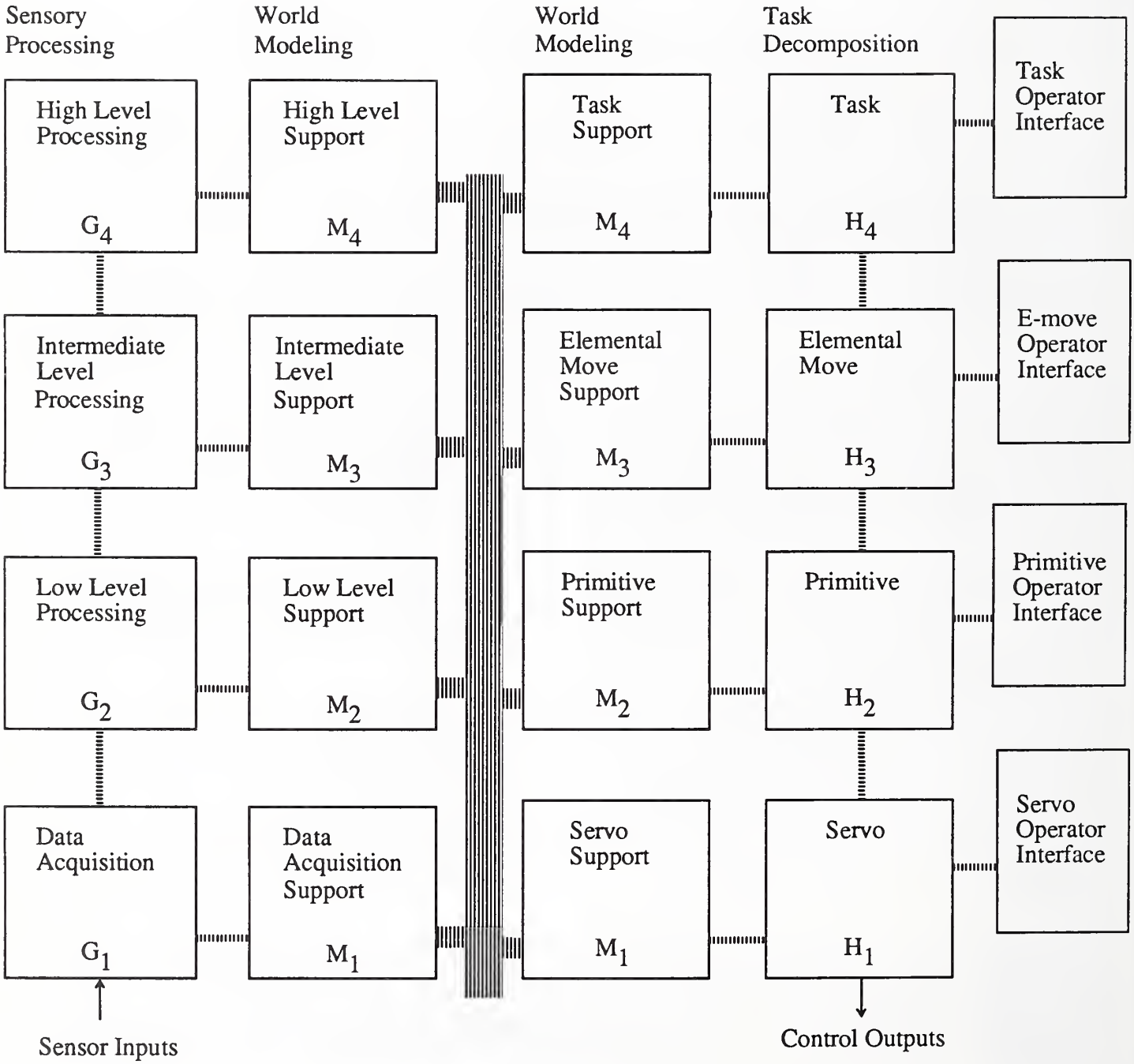


Figure 3. Major Components of Control Architecture.

of reasoning about object/task features to determine appropriate gains to command to servo.

- Dynamic coordination with other components of the system, as between two Primitive modules during dual-arm coordination.

  Enabling Development - algorithm development

- Sensory-Interactive Motions - Real-time trajectory generation and/or replanning based on sensory feedback, such as vision and proximity, to allow performance of tasks based on sensed data rather than (possibly incorrect) a priori knowledge. This provides for robustness of task execution amid incomplete or uncertain knowledge of the environment. Algorithms which make use of relative errors, without having to determine absolute positions of objects in the world, should also be relatively insensitive to camera calibration errors.

  Enabling Development - algorithm development, sensory-processing improvements, computer throughput.

- Consideration of Arm and Payload Dynamics - The manipulator dynamics must be considered in order for the full capabilities of the manipulator to be available and to prevent exceeding actuator limitations.

  Enabling Development - dynamic and actuator limit modeling, computer throughput

- Use of Objective Functions - Any particular motion may be performed in a variety of different ways, depending on what the objectives and constraints of the motion are. A number of variables, such as path tolerance, reaction forces exerted through the base, energy consumed, acceleration experienced by the handled object, and time taken to perform the motion, should be taken into account. The objectives and constraints also affect how redundant degrees of freedom may used to best advantage.

  Enabling Development - definition of objective functions and constraints, algorithms, computer throughput

- Human-machine Interface - The operator must be able to input destination and path constraints in a natural way, e.g. using graphics overlays, touch screens and other designation techniques. In order to phase in autonomous capabilities an operator interface must be provided which allows safe and convenient switching between teleoperated and autonomous activity. Also, the operator must be able to override the manipulator velocity at all times.

  Enabling Development - algorithms, displays and controls.

A4.3. Elemental Move Task Decomposition

The Elemental Move (E-move) task decomposition plans the gross and fine motions of the robot. This includes planning a path through a cluttered environment to produce a goal point, via points, and an acceptable error bound from a task description passed from the Task Level. The E-move level is the highest level which is robot dependent.

- Ability to coordinate two or more arms for the completion of a task, e.g. carry an object heavier than either arm could carry alone.

  Enabling Development - computer throughput, algorithms

- Ability to coordinate arm motion with end effector to accomplish the goals of the task. (There is a need to treat grasping as a more dynamic process, not just something that is done a the end of a manipulator motion, as when trying to grasp a moving object.)
  Enabling Development - computer throughput, algorithms
- Ability to determine collision free path from a starting point to a goal creating as many via points as necessary.
  Enabling Development - computer throughput, algorithms
- Ability to determine fine motion strategies from object and task models, and goal.
  Enabling Development - computer throughput, development of object and task models, algorithms
- Ability for human to easily enter collision free trajectories.
  Enabling Development - human-machine interface

A4.4. Task Level Task Decomposition

The Task level task decomposition module plans tasks in a geometric sense using CAD models. It operates from the kind of task commands that might be given to an astronaut rather than a robot. The Task level is the lowest level which is robot independent.

- Ability to perform geometric task planning. Using geometric models of objects and a specification of the goal, the system must be able to develop a robot independent sequence of appropriate actions.
  Enabling Development - algorithms, computer throughput
- Ability for human to specify goal state(s) in a natural fashion.
  Enabling Development - algorithms, computer throughput, human-machine interface
- Ability to produce output so that any robot can perform the task. For example, specification of compliance directions for mating, forces required for mating, strategies for insertion, etc., which (should be) are implemented on all robots.
  Enabling Development - algorithms, computer throughput, human-machine interface

A4.5. Data Acquisition Sensory Processing

The purpose of Data Acquisition sensory processing is to gather raw data from individual sensors and perform simple filtering on this data to remove noise or enhance the information in some way. A number of specialized hardware systems exist for performing this type of function for vision sensors such as PIPE, WARP, PIFEX, and DATACUBE. The types of algorithms used on this hardware for Data Acquisition would include grey-scale stretching, high and low pass filtering, and FIR and IIR filtering. Although these functions may exist commercially they need to be integrated with NASREM in several ways.

- Ability to gather data from each sensor with acceptable time delays. Since there is a Data Acquisition module associated with each sensor, there should be no impact in adding sensors to the system. This means that the requirements for acceptable time delays may necessitate special hardware for some sensors, e.g. vision.

Enabling Development - computer algorithms and hardware to support them

- Ability to perform filtering operations as necessary on the sensor data and send the enhanced data to either the world model or Low Level sensory processing as required by the data type.
  Enabling Development - algorithms
- Ability for the operator to specify a particular type of processing, or to request data from a specific instance of a sensor in a user-friendly way.
  Enabling Development - algorithms
- Ability to distinguish between noise and fine texture when filtering.
  Enabling Development - algorithms
- Better techniques for modeling sensors are needed for predictive based filtering algorithms.
  Enabling Development - algorithms, sensor models
- Rules for choosing appropriate techniques in a given environment.
  Enabling Development - algorithms, environment and sensor models
- Evaluation techniques that alter computational parameters or algorithms as required.
  Enabling Development - algorithms, environment and sensor models

A4.6. Low-Level Sensory Processing

The Low Level sensory processing module is responsible for segmenting array data from various sensors into regions or boundaries based on similarities or differences. A Low Level stage is associated with each camera, range array or tactile array sensor to expedite processing time. Existing hardware for a subset of segmentation techniques includes parallel processing systems and pyramid machines mentioned in section 4.5. Existing software techniques typically involve using boundary segmentation algorithms to detect edges or corners, or region segmentation algorithms using color, shape, texture, and motion. Some software packages exist but these need to be integrated with NASREM. Also, these techniques need to be improved to obtain more reliable and repeatable results.

- Ability to accept requests from either sensory processing system's Intermediate Level or the world model, and to accept filtered or enhanced data from Data Acquisition.
  Enabling Development - algorithms
- Ability to perform segmentation techniques as necessary and send the resultant information to either the world model or the Intermediate Level in an acceptable time period.
  Enabling Development - computer algorithms and appropriate hardware
- Ability to detect unexpected information as a means of early obstacle avoidance, i.e., observing the environment even when not trying to locate a specific object.
  Enabling Development - computer throughput, algorithms
- Ability to accept operator commands to perform specific types of segmentation.
  Enabling Development - algorithms, human-machine interface

- Ability to combine various techniques to substantiate results rather than depending on any single method. More work needs to be done on the intelligent combination of these methods.

  Enabling Development - computer throughput, algorithms

- Methods for evaluating results are needed which can alter parameters used in computations as required to improve output, or can choose a different algorithm to obtain better results in a given situation.

  Enabling Development - algorithms

A4.7. Intermediate-Level Processing

The Intermediate Processing is responsible for converting iconic information received from the Low Level to lists of features (curvilinear equations, corners, statistical measurements, etc.). Current approaches to this function emphasize the use of model-based techniques for grouping information into features or regions. Some specialized hardware exists for fast curve fitting (e.g., Hough transform), and connected component analysis. And, as in the Low Level, some software packages exist but these need to be integrated with NASREM and made more robust.

- Ability to accept requests from High Level Processing or the world model.

  Enabling Development - algorithms

- Ability to accept models or expectations from the world model to be used to guide algorithms and evaluate results.

  Enabling Development - algorithms, object and feature models

- Ability to extract symbolic features and their attributes within an acceptable time period.

  Enabling Development - computer throughput, algorithms

- Ability to merge results obtained from non-conflicting sensory information from multiple sources (e.g. cameras used for obtaining stereo information).

  Enabling Development - algorithms, object and sensor models

- Ability for operator to specify feature(s) to be extracted or window of interest either through a graphic display with a mouse or light pen, or by using ASCII strings on a keyboard.

  Enabling Development - algorithms, human-machine interface

- More work is needed in evaluating results and ways of altering input parameters to obtain better results.

  Enabling Development - algorithms

A4.8. High-Level Sensory Processing

The function of High Level Sensory Processing is to group features and to recognize and label objects based on information obtained from a single sensor. This is an area where a great deal of work is required to recognize objects independent of contextual information. Typical approaches to this problem are highly limited and can only be used to classify a small number of objects. Thus, there are a number of areas in which development needs to be done in order to meet the goals of

autonomous FTS operations.

- Ability to accept requests from the world model or the operator.
  Enabling Development - algorithms, human-machine interface
- Ability to accept operator input to focus attention on an object of interest.
  Enabling Development - algorithms, human-machine interface
- Ability to perform recognition and labelling tasks in a reasonable time period.
  Enabling Development - computer throughput, algorithms
- Ability to accept and use model information from the World Model for guiding processing and evaluating results.
  Enabling Development - algorithms, object models
- Ability to perform recognition when there are unexpected objects in the scene.
  Enabling Development - computer throughput, algorithms
- Ability to recognize partially occluded objects -- e.g. tracking of object (features) over time.
  Enabling Development - computer throughput, algorithms

A4.9. Servo Support World Modeling

The function of Servo Support world modeling is to support Servo task decomposition by computing model-based terms that are required for servo control functions. For example, Servo Support computes gravity compensation terms which can be added to the servo control in task decomposition to compensate for the effects of gravity. A more detailed description of the capabilities provided by Servo Support can be found in [4] and [6].

- Ability to support kinematic model of robot.
  Enabling Development - specific robot models, algorithms for redundant devices
- Ability to support dynamic model of robot. This capability will allow better performance of the robot.
  Enabling Development - specific robot models, computer throughput
- Estimation and learning of dynamic model of robot. This capability is required for unknown models or when there may be some variation in dynamic parameters.
  Enabling Development - specific robot models, computer throughput, algorithms
- Ability to easily add the kinematics and dynamics of a held object into the model of robot.
  Enabling Development - specific robot models, object models, algorithms
- Ability to provide feedback to Servo in the appropriate coordinate system.
  Enabling Development - computer throughput, algorithms
- Ability to compute local compensations to deal with torque distributions, joint limit avoidance, obstacle avoidance, and the self-motion of redundant mechanisms.
  Enabling Development - specific robot models, computer throughput, algorithms

A4.10. Primitive Support World Modeling

Primitive Support provides Primitive task decomposition with the required model-based information necessary for planning and executing the Primitive function. This information includes object positions and manipulation features such as friction and stiffness coefficients. A number of capabilities must be developed in order to support typical manipulation tasks.

- Ability to provide kinematic and dynamic models of the robot for use in planning efficient trajectories.

  Enabling Development - specific robot models, computer throughput, algorithms

- Ability to provide joint limit models for use in planning trajectories. (For a redundant manipulator, perhaps the inverse kinematics algorithm would use joint limits as the "extra" constraint.)

  Enabling Development - specific robot models, algorithms

- Ability to obtain specific object data, i.e. stiffness, friction, tolerances, from more general object data representations. This information would be used to aid Prim task decomposition in planning manipulations.

  Enabling Development - computer throughput, object models, algorithms

- Ability to provide Primitive task decomposition with data so that the manipulator can be servoed to a visual feature.

  Enabling Development - feature and object models, visual field kinematic models, algorithms

- Allow the operator to select and specify object parameters for use with Primitive.

  Enabling Development - algorithms, human-machine interface

A4.11. Elemental Move Support World Modeling

The function of Elemental Move Support is to provide the world modeling operations needed by Elemental Move task decomposition. Information provided by E-move Support includes environment geometry, geometric object models, and geometric models of the robots. E-move uses this information to plan safe paths for the robot's manipulators, and to plan fine motion and grasping strategies. Once a plan for a particular manipulation has been developed, world modeling may be involved in storing and accessing that plan.

- Ability to provide models of workspace and objects, e.g. spatial occupancy models, simplified CAD models.

  Enabling Development - computer throughput, workspace and object models, algorithms

- Ability to provide models of general robot capabilities such as manipulator payload limits and object masses.

  Enabling Development - workspace and object models, algorithms

- Allow operator to indicate elements of models to be considered in path planning.

  Enabling Development - workspace and object models, algorithms, human-machine interface

A4.12. Task Support World Modeling

The function of Task Support is to provide the world modeling operations needed by task decomposition. At this level, world modeling must deal with complete CAD representations of objects and the environment so as to support the geometric reasoning needs of task decomposition. Task Support must also assist in the generalized planning needs of the Task level, such that task decomposition or the operator can formulate (and store) plans for robot tasks such as ORU change-outs, etc.

- Ability to manipulate models of workspace and objects.
  Enabling Development - workspace and object models, algorithms
- Ability to support general planning and plans.
  Enabling Development - computer throughput, algorithms

A4.13. Data Acquisition Support World Modeling

Just as on the task decomposition side, there are world modeling functions to support sensory processing operations. The world modeling functions provide model-based information for use in sensory processing, such as model-derived templates or features for matching, as well as transforming data from sensory processing for storage in the global data system. At the lowest level, there is Data Acquisition Support to provide modeling functions related to the low-level processing of sensors. Data Acquisition uses models of sensors to transform sensory processing output into a form useful and understandable to the entire system. This is a general function of world modeling, to take the information processed on local data and provide global descriptions of the world that can be used for all elements of the system.

- Transform sensor readings to meaningful values (e.g.: strain gauge readings to forces and torques) based on models of the sensors.
  Enabling Development - algorithms, sensor models
- Maintain histories of sensor readings for filtering, prediction and signature analysis.
  Enabling Development - algorithms, data storage capabilities (large memories and possibly high-speed, off-line storage such as optical disks)

A4.14. Low-Level Support World Modeling

In Low Level Support world modeling, general object models are processed to produce iconic models suitable for matching at this level. This is useful when trying to servo rapidly to a visual feature. The matching can proceed more quickly by using an iconic "template" than in determining more global features before matching.

- Ability to provide templates for Low Level matching.
  Enabling Development - computer throughput, algorithms, camera and object models

A4.15. Intermediate-Level Support World Modeling

Intermediate Level Support provides model-based information for use in matching and "attention" information to guide the sensory processing at this level. Although this is generally the level at which most commercial systems have done model matching, (if at all,) much work needs to be

done to codify and evaluate appropriate techniques for use in FTS.

- Ability to provide "window of attention" to guide sensory processing (in choosing focal setting, etc.). For a teleoperated system, the "window of attention" may be read from an operator input device. The world model would translate (transform) the reading from the input device to a form usable by sensory processing.
  Enabling Development - sensor models, human-machine interface
- Estimate range to an object in order to guide range sensing.
  Enabling Development - object/environment models, algorithms
- Ability to provide 2D projections for model matching. These can vary from a simple 4 LED model to a boundary representation.
  Enabling Development - computer throughput, object models, algorithms
- Ability to select the most appropriate features for a particular matching problem. (This implies there are multi-feature models of things to be recognized.)
  Enabling Development - models, algorithms

A4.16. High-Level Support World Modeling

High Level Support extracts appropriate object models for the complex matching that is to be performed at this level. Based on the results of matching, world modeling updates the geometric knowledge of the world. This is a difficult function which is essential to many aspects of autonomous operation.

- Ability to update pose of an object based upon model and new sensory information.
  Enabling Development - object models, algorithms
- Ability to provide 3D models for matching.
  Enabling Development - computer throughput, object models, algorithms

A4.17. Servo Operator Interface

The function of the Operator Interface is to provide displays of simple robot motions and low level displays of raw or semi-raw data, such as force-torque sensor readings. Also to allow user specification of robot motion and selection of control and data-filtering methods. In particular, the operator must be able to "teleoperate" the FTS at this level with a hand controller or other input device. Although the basic capabilities for this function exist commercially much development is still going on in improving the usefulness of data feedback methods, as discussed earlier.

- Ability to display simple data visualizations, including real-time animations of simple robot representations and raw or low-level visualizations of sensory data.
  Enabling Development - computer throughput, display techniques and devices
- Ability to display and to integrate multiple displays and user input devices on a single workstation.
  Enabling Development - computer throughput; high resolution, deep pixel displays
- Ability to smoothly switch between autonomous, teleoperated, and shared control modes.

Enabling Development - algorithms, I/O devices

A4.18. Primitive Operator Interface

The Primitive Operator Interface functions to display more detailed robot motions and trajectories, higher order sensor data visualizations and to allow human to interact with discrete objects in the workspace. At this level, the operator is giving more "supervisory control" type commands which can executed by Primitive and Servo task decomposition.

- Ability to display solidly modeled workspaces and robots.
  Enabling Development - computer throughput, algorithms, solid models
- Visualization algorithms for the display of higher order sensory data and the hardware and software integration interfaces to facilitate communication and shared i/o responsibilities between specialized sensor hardware and graphical workstations.
  Enabling Development - computer throughput, algorithms
- Object orientated approaches for both programming and three-dimensional workspace object representation and its integration on graphical workstations will provide the human with increased ability to interact with workspace objects
  Enabling Development - computer throughput, algorithms

A4.19. E-move Operator Interface

This component functions to provide the user with a convenient and intuitive method for interacting with the workspace in three-dimensions. Also it provides user interaction with high order sensory visualizations. This level should allow some major elements of the FTS tasks to be automated such that the operator commands the element to be executed (possibly through a voice command,) and simple waits for the subtask to complete. An example of an appropriate subtask commanded at this level would be end effector or tool changes. This type user interaction will be an important part of the autonomous capabilities of FTS.

- Ability to interact with a three dimensional world display and have three dimensional pick devices.
  Enabling Development - computer throughput, environment models, algorithms, devices
- Provide the user with the ability to interact with higher order sensory visualizations.
  Enabling Development - computer throughput, algorithms
- Voice recognition and speech synthesis
  Enabling Development - computer throughput, algorithms
- Ability to smoothly switch between control modes.
  Enabling Development - algorithms, I/O devices

A4.20. Task Operator Interface

Task Operator Interface provides a high-level command interface for the FTS to provide interaction with the FTS as an integrated whole.

- Some kinds of "meta-displays" that allow the user to control or specify FTS as a unit.
  Enabling Development - computer throughput, algorithms

A4.21. Programmer Interface

Since it is not likely that a system can be developed in the near term that will not require extensive programming for autonomous tasks or subtasks, a useful programmer interface should be developed for the system. This will facilitate the adaptation of the FTS to new tasks and to modifications in old tasks and environments.

- At the highest level a Task Description Language should be provided to allow the programming of high-level tasks in a grammar that is logical and semantically consistent.
  Enabling Development - algorithms
- Methods for interacting with the world model through a CAD system and object-oriented database of objects. This should include a standard format for storing CAD data, creating new types of data, and visualizing data.
  Enabling Development - computer throughput, models, algorithms

A4.22. Interface Development

Between the NASREM elements described above, there are specific interfaces which allow the elements to communicate and thereby function as integral parts of the whole system. The nature of data comprising these interfaces is very important since it will determine what part of the robot control task can be done in each module. If an interface to a module is too simplistic, then the module will be limited in the types of algorithms that it can support. If an interface is too complex, it will be difficult to understand and maintain, and as a consequence, it will create problems in software integration.

Interfaces have been developed for some NASREM components. Task decomposition interfaces at the Primitive and Servo levels are described in [4] and [5]. These interfaces are defined so as to support the majority of the literature on robot control. At higher levels of the NASREM hierarchy, literature support decreases. Much more work (including experimentation) is required to develop interfaces at these levels. The same is true for higher interfaces in the sensory processing and world modeling hierarchies.

Experimental interfaces must be developed and tested in the lab before evolving to space applications. This will assure that the interfaces are sufficient for handling the vary of algorithms that might eventually be required in space applications. In addition, good "standard" interfaces will ease the integration of NASREM as the system is expanded, and as external systems are connected to it.

A.5. Robot Metrology

Another important area of development required for automation of robot tasks is robot metrology. Robot metrology refers to the measurement and characterization of robot kinematics and dynamics, and thus is essential for developing good models of manipulators and for evaluating manipulator motions.

As mentioned in section 2.2, extensive models of robots are required for programming. For absolute position accuracy, e.g. moving to a point based on a Cartesian model of the world, a precise kinematic model of the manipulator must be available. The only way to obtain a precise model is by measuring the Cartesian tip position of the manipulator at various joint positions and calculating the kinematic parameters.

One system for performing Cartesian measurements of the tip position is based on theodolite technology. New measurement systems using lasers are currently in development. The laser-based techniques that are being developed will allow more accurate measurements and will record the manipulator's dynamic behavior over time. Dynamic measurements, when compared with programmed motion, yield insight into control system performance. This will be useful for evaluating and comparing different control algorithms.

A.6. Research

As discussed in section 3, certain autonomous capabilities can only be achieved through basic research into fundamental problems of robotics. This section describes some kinds of research that might be considered in three principal areas, motor behavior, machine vision, and world modeling.

A6.1. Motor Behavior

Many of the basic limitations of current manipulator systems result from extremely poor motor capabilities. This has very little to do with "intelligence" in a human sense. Even in what would be considered "non-intelligent" animals, such as insects, there is sufficiently good motor behavior to accomplish complex tasks and achieve important goals. Not so for robots. This suggests that there is something fundamentally different in the way motor behavior is generated in animals and in robotic devices.

Consider the problem of manipulator trajectory formation as exemplifying the basic approach to robot control. Typically, start and goal points for a trajectory are known. Then a mathematical equation is obtained for the complete trajectory from start to goal, such that the position of the manipulator along the path at periodic time intervals is known explicitly. The trajectory is planned so that the motion will meet certain criteria for smoothness, duration, etc. Once the trajectory is "decomposed" into a sequence of positions, it can be "executed" by sending the positions one-by-one to a PID control loop. This approach forces the manipulator to be in a particular position at a particular time, with no allowance for the environmental situation at the time of movement. If the environmental situation is such that the trajectory cannot be executed exactly as planned, the resulting motion will most likely fail to achieve its objectives, e.g. smoothness [7]. This problem persists even when there is periodic replanning, since replanning takes time and the manipulator's state is continuously changing.

The problems created by a priori "decomposing" every action can be seen even more clearly at the higher levels. Suppose a command to "put part A in fixture B" is to be carried out by a manipulator controller. To do this, the module that accepts this command must produce a sequence of movements that result in the attainment of the goal. For example, the sequence might be {"approach A", "grasp A", "move to fixture B", "release A", "depart B"}. Ideally, this sequence should result in A being properly placed in fixture B, but what if at "release A" the part falls on the floor, or at "grasp A" the part is gripped crookedly. In these cases the goal is not achieved by the se-

quence, the decomposition fails, and the task fails. For a person carrying out the task "put part A in fixture B", it's not a matter of executing one sequence, it's more a matter of taking whatever actions are required to achieve the goal. If the part falls on the floor, a person automatically picks it up and tries again, perhaps being more careful the second time, so that ultimately the goal is achieved. Trying to program this type of behavior into a robotic device has proven infeasible with current approaches. It is impossible for a programmer to take into account all the things that could go wrong, the corrective actions for those failures, all things that could go wrong with those corrective actions, their corrective actions, and so on. This is the "exception handling" problem that was described in section 2.2.1.

Another approach to this "task decomposition" problem is to provide a motion technique that performs decomposition based on a task goal in real-time. For the trajectory generation example, this means providing ways for generating trajectories that meet appropriate criteria for smoothness, duration, energy consumption, etc., without planning the entire trajectory a priori. One possible method for this type of trajectory generation is suggested in [7]. As discussed above, in typical trajectory generation systems a complete path that has the desired smoothness properties would be planned before execution. In [7] it is shown that by creating an appropriate gain profile, smooth trajectories can be generated without "decomposing" the entire path into a sequence of positions beforehand. In this approach, a single gain profile produces smooth motions of a given duration over an infinite number of different trajectories. Since the position of the system is not specified except as a final goal (at the end point,) the trajectory can be adjusted to allow for the environmental situation and smoothness properties can still be maintained.

Similar approaches could be applicable to other problems in manipulator control. Gain modulating or impedance control techniques may be appropriate for creating behaviors that handle contact with the environment (including closed kinematic chains) in a useful manner. Manipulator control in these situations is still very much a research issue, although it would obviously be very useful to have techniques that provide good motor behavior in these cases. This would allow the robot to open doors, turn cranks, manipulate handles, etc., even when there is no detailed model of the positions and kinematics of these devises. Also, this kind of accommodation of contact forces would allow less destructive interaction with the environment, such that it may not always be necessary to avoid obstacles. Humans often use obstacles to guide hand and arm movements, (and to learn where they are,) rather than strictly avoiding them all together.

There is a need to proceed beyond a simple, linear PID position control loop. Linear techniques have dominated control approaches because of their analytic tractability. However, these techniques appear to be insufficient for handling the complex goals required of good motor behavior [8]. In addition, adaptive and learning techniques need to be studied further in order to provide control schemes which work well under unknown and changing conditions, such as variable loads and changes due to thermal fluctuations.

Algorithms that do not require a priori decompositions, but create motion using real-time feedback that results in goals being properly achieved, would be an outcome of basic research as described above. In order for these techniques to be generalized to higher level goals of the system, research must begin incorporating more advanced sensors into the system. Two important types of sensing for advanced motor behavior are tactile sensing and vision. Using these as a fundamental part of the control, much as joint position sensing is now the basis of control, is essential for behaviors such as "put part A in fixture B". Sensing like vision must be used in a continuous feedback

mode to monitor the environmental situation and modify the manipulation process. One important research topic in this line is servoing based on visual feedback. This allows tracking of moving objects in the visual field and helps remove uncertainty in grasping and assembly operations. This also leads into another important area of research, machine vision.

A6.2. Machine Vision

An essential component in achieving good motor behavior is machine perception. The system must be able to sense the aspects of the environment that it is to act upon and get feedback on what its doing. Within the general topic of machine perception, several areas need further research. One particularly important area is machine vision. In this area, existing methods need improvement or altogether new methods need to be introduced.

Existing machine vision techniques are not robust in unstructured environments, as discussed in section 2.2.2. In the short term, this problem can be reduced with the aid of a human operator. Given current technology, an operator can facilitate the correction of faulty information at the point at which it is produced. He/she may guide the type of segmentation used to yield more accurate features, or he/she may supply missing information such as the name of an unrecognized object.

An improvement on this mode of operation would provide for the acknowledgment and self-correction of false or inconclusive information. The research on model-based prediction is one attempt to imitate the way that humans process visual information and to guide processing based on current or previous knowledge. An additional enhancement is the intelligent combination of different types of sensory information, which can provide better reliability. Yet another important method that can impact robustness is learning, which has been demonstrated to be useful in pattern classification. The concept of enabling the system to dynamically converge on a correct interpretation is essential to advance the system towards autonomy.

Vision sensing can be integrated with other sensors to combine information about the environment. Multiple sensors promote greater flexibility with which to sample and understand the environment. For example, range to an object can be extracted from information from several different devices: a stereo pair of cameras, a moving camera, a tactile force sensor, or a laser ranging device. Since the knowledge from different sensor sources provides complimentary views of the surroundings, the issue of how to combine the data becomes important. The sensors have different accuracy limits and varied positions within the workspace which contribute to the reliability of the information they supply. Knowledge from the various sensors needs to be combined in a meaningful manner, and special consideration must be given when sensors provide conflicting information.

Not only does more research need to be done in sensor processing for individual sensors, but also in the area of how multiple sensors are combined to prevent the compounding of errors and to increase the robustness of the system. In an initial implementation, a human could perform the role of a supervisor to correlate information from multiple sensors and to plan which sensors are appropriate to use. Eventually, learning and goal planning must play significant roles in allowing the system to perform these functions autonomously.

An important area for machine vision research is that of providing <u>continual</u> feedback rather than occasional processing of images. The vision system will need to be capable of processing information quickly and continuously to support tasks such as visual servoing. This may require the ability to process sensor information based on context. For example, when playing ping-pong it is

not necessary to determine if the white blob coming toward you is a ping-pong ball, it is only necessary to determine its approximate position and velocity. The use of task context allows the amount of processing to be greatly simplified. Techniques need to be found for the use of context in machine vision.

In support of the research areas mentioned, there is a corresponding need for evolution of hardware that occurs. As new software develops and is tested, dedicated hardware is designed to reduce the computational time involved.

A6.3. World Modeling

Increasingly, advanced manipulator systems rely on internal models of the world. These models are used to plan motions and analyze sensory information. As the system is made more autonomous, there is more and more reliance on the internal world model. The models are generally precise, quantitative, geometric models of the manipulator system and of the objects in the manipulator's environment. As discussed in section 4, development of models of this type will be required for automation. However, it should be recognized that there are limitations to the quantitative approach.

The reliance on precise, quantitative models has created problems for system developers, as discussed in section 2.2. Development of models of the environment is a complex and time-consuming task. Even once geometric models are available, the manipulation of the models for use in sensory processing or motion planning can be a computationally intensive problem. And that is just for simple rigid-body models. Modeling complex objects in a quantitative, geometric manner, such as the blanket described in section 3.2, is impossible for all practical purposes. Suppose, for example, a robot was given the task of climbing a tree. There is no practical way for the robot to have a precise model of the tree, its complexity is too great. However, a truly autonomous system would be capable of dealing with this kind of complexity. This dilemma presents two topics for further research in world modeling.

One important area of research is acquiring models of sufficient complexity to allow the manipulator system to function successfully. Currently, manipulator systems use sensing only for immediate control objectives. For example, if a manipulator is to pick up a part, it asks the vision system to determine the location of the part. The vision system only looks for the part, it does not "see" any of the other aspects of its environment. Partly this is a limitation with machine vision, as described in the previous section, but also, the system does not maintain a continual internal model of the things it is perceiving. Also, the information from manipulator force and position sensors is never incorporated in model making, even though these sensors are rich in information about the locations of objects in the environment. The system simply does not maintain a central model which it is continually testing and updating using all available sensory information.

This brings up the second research topic. How is model information to be stored? There are obvious limitations with precise, quantitative, geometric models as described above. It does not make sense to maintain and update CAD-like models of the complete environment. Consider one's desk top as the environment. Does one maintain precise, quantitative models of the shapes and positions of all the items on one's desk? Certainly not. The information is much more qualitative in nature. Research needs to be conducted to find appropriate qualitative models for use with robots.

Qualitative models may be more procedural in nature, than declarative. In other words, an ob-

ject may be represented more in terms of how it is used than precisely what its geometry is. For example, when climbing a tree, a tree limb may be represented in terms of its appropriateness as a foothold or handhold, rather than as section of a cone with diameter X inches, flexibility Y, etc. This type of procedural representation of objects and related actions relates also to the representation of the "complex goals" required for advanced control techniques, as described in 6.1.

A.7. Conclusion

This document has discussed FTS evolution issues involving development and research. There is so much work relating to the FTS that could be done, one must consider which elements will be of most benefit to the project.

As discussed in section 3, of obvious short-term benefit is the development of teleoperation and telepresence systems. In considering the development of input devices to support teleoperation, NASA must determine what type of arrangement is most appropriate for the Space Station environment. Criteria for consideration of devices include ease of use, capabilities, size and weight, compatibility with the performance of other mission tasks (including eventual autonomous FTS control,) and the nature of the controlled device. The development and adaptation of teleoperation control algorithms and devices to operate in the space environment can be seen as a relatively high-payoff endeavor, providing a significant amount of capabilities for the investment.

To ease the burden on human operators teleoperating the FTS, those subtasks which are sufficiently structured should be automated. Increasing the autonomy of the system requires much of the NASREM development described in section 4. In addition, techniques for switching control between the operator and the autonomous system must be developed. However, there are some subtasks that will simply be easier for the autonomous system to perform than for the operator to do through teleoperation.

End effector and tooling changes are ideal candidates for this type of automation. End effector changes will require Prim and Servo task decomposition and world modeling, as well as some capabilities at the E-move level. Another opportunity for automation occurs when the system is being teleoperated. During teleoperation the autonomous part of the system prevents the manipulator from hitting obstacles or joint limits, or moves the self-motion of the manipulator to maximize manipulability. Such operations will require a fairly complete implementation of Level 1 of NASREM. A third candidate for automation is locating objects and servoing to them using some simple visual features. The LED docking demo demonstrated at Goddard is an example of the kind of operation this capability will enable. Visual servoing tasks will require implementation of the first three levels of the sensory processing hierarchy and the accompanying world modeling support models.

Finally, an investment in basic robotics research can be seen as good policy for long-term development issues. This research must go outside the established robotics research areas in order to find new techniques for robot control that will promote autonomy in the long term. Progress in research means that more FTS capabilities can be automated. For some tasks this may eventually mean the elimination of human operator intervention, thus freeing humans for more important activities.

A.8. References

[1] Zimmerman, W., Myers, J., Ruth, D., "Task Ranking for the Telerobot Demonstration," Contract 957908, Tasks 7 and 8, Final Report, July 1988.

[2] Bejczy, A. K., "Man-Machine Interface Issues in Space Telerobotics: A JPL Research and Development Program," Proceedings of the Workshop on Space Telerobotics, JPL, Pasadena, CA, July, 1987.

[3] Paul, R. P., "Robots and Teleoperation," IEEE Workshop on Shared Control, Philadelphia, PA, April, 1988.

[4] Fiala, J. C., "Manipulator Servo Level Task Decomposition," NIST Technical Note 1255, October, 1988.

[5] Wavering, A. J., "Manipulator Primitive Level Task Decomposition," NIST Technical Note 1256, October, 1988.

[6] Kelmar, L., "World Model: Level 1," Intelligent Controls Group working document, NIST, November, 1988.

[7] Fiala, J. C., "Generation of Smooth Trajectories without Planning," submitted to 1989 IEEE Robotics & Automation Conf.

[8] Franklin, J. A., Selfridge, O. G., "Some New Directions for Adaptive Control Theory in Robotics," 1988 NSF Workshop on Applications of Neural Networks to Robotics and Control.

NIST-114A
(REV. 3-89)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION OR REPORT NUMBER
NISTIR 4463

2. PERFORMING ORGANIZATION REPORT NUMBER

3. PUBLICATION DATE
December 1990

4. TITLE AND SUBTITLE

Short-term Evolution for the Flight Telerobotic Servicer

5. AUTHOR(S)

Dr. Ronald Lumia

6. PERFORMING ORGANIZATION (IF JOINT OR OTHER THAN NIST, SEE INSTRUCTIONS)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
GAITHERSBURG, MD 20899

7. CONTRACT/GRANT NUMBER

8. TYPE OF REPORT AND PERIOD COVERED

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP)

S/A

10. SUPPLEMENTARY NOTES

[ ] DOCUMENT DESCRIBES A COMPUTER PROGRAM; SF-185, FIPS SOFTWARE SUMMARY, IS ATTACHED.

11. ABSTRACT (A 200-WORD OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY OR LITERATURE SURVEY, MENTION IT HERE.)

This document identifies near term technology developments which would have significant impact on the evolution of the FTS toward autonomous operation. Analysis of anticipated FTS tasks is used to identify operations that might be performed autonomously rather than in a purely teleoperated fashion. Alternative techniques for automating these operations are then described. A discussion of FTS long term evolution is included as an appendix.

12. KEY WORDS (6 TO 12 ENTRIES; ALPHABETICAL ORDER; CAPITALIZE ONLY PROPER NAMES; AND SEPARATE KEY WORDS BY SEMICOLONS)

control system architecture; robot programming; robot visual sensing; space station servicing; space telerobotics; teleoperation

13. AVAILABILITY

[X] UNLIMITED

[ ] FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS).

[ ] ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402.

[X] ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.

14. NUMBER OF PRINTED PAGES
59

15. PRICE
A04

ELECTRONIC FORM