[FTL R-94-8]

Human-Centered Automation of Air Traffic Control Operations in the Terminal Area

Proposal for the Interdepartmental Doctoral Program in Human Factors and Automation

by

Husni Idris
Flight Transportation Laboratory
MIT
November 2, 1994

Contents

Chapter 1

Introduction

1.1 Brief background on terminal area operations

Air Traffic Control operations are described extensively in the ATC manuals such as the Airman's Information Manual [1] and the ATC Controller's Handbook [2]. Mathematical analysis has also been conducted for the ATC operations as evident in the many theses that have been published in ATC research [3, 4, 5]. A brief description is due here however in order to provide a background for the following document. There are six major ATC functions in the terminal area and a summary of their description in Sadoune's thesis [5] follows:

Flow Management: The flow management purpose is to provide efficient transition between the en-route corridors and the terminal area through the metering fixes. The en-route corridors are the airways connecting the airports, the terminal area is the designated space around the airport, and the metering fixes are the points at which aircraft enter the terminal area under the flow control process called metering. The flow management system is capable of delivering the aircraft to the metering fix at predetermined time, altitude, and speed, minimizing fuel consumption and flight time. Beyond the metering fix however the concern in no longer fuel and cost, it is the separation between the aircraft and the landing schedule. Ground-based flight path generation is needed at that point.

Runway Scheduling: The runway capacity is the limiting factor of the flow of traffic at congested airports. There are many reasons why runways are not used efficiently in the current tactical practice. These include the independent scheduling of landings and takeoffs, the ad hoc fashion in which takeoffs are inserted between landings, and the common use of the first-come-first-serve approach which is fair but not optimal. Runway scheduling is a queuing process and can be optimized for maximum throughput, long term service, and minimum delays of aircraft, taking into account fuel consumption, duration of flight, and other factors. The difficulty is in the dynamic nature of the schedule where modifications are needed as new entrants arrive or as environmental conditions change. The determination of the runway capacity and its improvement through the use of advanced technologies are discussed in [6].

Flow Control: Through traffic redistribution the flow control process helps smooth the demand fluctuations leading to a controlled number of aircraft simultaneously present in the terminal area. Two processes accomplish flow control: metering and holding. Metering divides the approach to the airport into successive stages between metering fixes. The flow management system delivers the aircraft to the metering fixes at the predetermined time, altitude, and speed. Holding points are assigned where holding aircraft are stacked and isolated from traffic. Holding aircraft circle in holding patterns awaiting landing clearance. Therefore, while metering moves the delays resulting from the runway capacity upstream, holding extends the flight path in time to accommodate arrival delays. These practices however can result in idle runway time in favor of more flow control leading to less efficient use of the runway.

Flight Path Generation: There are standard routes both from the terminal area entry points to the runway for approach and from the runway to the en-route corridors for departure. These predefined routes can be used at low traffic flow rates, and add to the precision since automatic flight control systems are capable of flying along them automatically. However they are not optimal in using the space, or in exploiting the aircraft capabilities, or in maximizing the runway capacity. Automated flight path generation allows the incorporation of the space organization, the ATC separation criteria, the landing and takeoff schedule, the aircraft dynamics and performance limitations, and the maneuvering characteristics of the pilot in generating more optimal and flexible paths. This subject will be emphasized further in this document.

Path Conformance Monitoring: In order to supervise the execution of the flight path plan, the radar surveillance system provides vague and non-precise measurement of the position of the aircraft. The controllers base their estimates of the conformance on 2-dimensional radar displays, and have to wait few intervals to estimate the direction of the aircraft. To adjust for the path conformance error the controllers issue heading, altitude, and speed clearances (vectors) to the pilots. Communication between controllers and pilots is done via radio transmission. Errors result from misunderstanding between the pilot and the controller, pilot response, as well as wind and unexpected atmospheric disturbances. Again new technologies and more automation are expected to improve the path conformance capabilities. These include better surveillance using satellites, digital data links for communication between the controller and the pilot, and display of the path to the pilot on board the aircraft. Questions of resolution and threshold of the conformance error become critical to the automation of the monitoring function.

Hazard Monitoring: This includes detecting possible collisions between aircraft and with the ground. There is a trade off between false alarms and missed alarms in setting the threshold for the hazard alarm. Namely the more conservative the alarm threshold is set, the less is the risk of collision due to a missed alarm. But the disturbance to the traffic flow caused by the large number of false alarms is higher.

1.2 Congestion and automation

Unless the infrastructure of the ATC system in enlarged by building more airports and runways fast enough to face the growth in the aviation activities, the air traffic congestion problem will persist. Building airports and runways however is an expensive, long term investment. Therefore the existing ATC system needs to be improved in order to use its full capacity.

The preceding description of the ATC functions in the terminal area highlights the importance of the runway system at congested airports as a bottleneck in the air traffic system. The air traffic controllers are busy with the heavy task of trying to achieve efficient use of these runways. However the rules they have to follow, the tools available to them, and the practices they adopt leave them far from accomplishing this goal. This is evident in all the functions described above, including the ad hoc scheduling of landings and takeoffs, using standard routes for approach and departure, and the reliance on vague and inaccurate surveillance systems.

With the introduction of new technologies the ATC system has improved and keeps improving drastically. Recent examples include the global positioning system (GPS) using satellites and the digital data links which promise great improvement in surveillance and communication capabilities. Advancements in control systems and artificial intelligence are allowing computers to replace humans at many levels. Flight management systems and automatic flight systems as mentioned above allow aircraft to fly along their assigned path and reach fix points at predetermined times automatically in the face varying winds. These automation capabilities are available to replace the human controllers as well as the pilots, especially when GPS and data links provide accurate measurements and fast and reliable communication.

Technological improvements should increase the efficiency of the ATC system when they are properly utilized. Originally automation was always thought to be more reliable than humans; but recent developments in technological capabilities are allowing the achievement of excessive degrees of automation. Experience in the aircraft industry has demonstrated cases where fatal accidents have been blamed on excessive automation [7]. This is leading researchers and engineers to adopt a human-centered approach where automation is designed to assist rather than to replace the human operator or supervisor. The main question in designing the automation becomes what should the human do and what should the machine or the computer do? Therefore, from experience, a human-centered approach should be adopted as automation is introduced into the ATC system. This affects the problem of designing the automation.

1.3 ASLOTS: a human-centered automation system

As mentioned above the terminal areas are bottlenecks in the air traffic system and therefore improving their performance is critical to easing the congestion problem. However, introducing automation into the ATC operations in the terminal area has been a difficult task because of the complexity of these operations. The most successful automation tool in ATC is metering which tries to control the traffic flow upstream of the terminal area. The attempts that have been made since the 1960's to introduce automation tools into the terminal area have been underrated by the controllers. These attempts however increased the workload of the controllers because of the contemporary technological limitations [8]. The new advancements in computer technology offer some powerful tools, especially in terms of human-system interface, that promise to make new automation attempts feasible and beneficial. Some of these new developments at NASA are receiving positive reaction from controllers [9]. Early developments used algorithms that were originally developed for flight management and flight control systems on board of aircraft, and by adding powerful computer graphical tools were made available to the ATC system [8, 10].

ASLOTS [11, 12] is a concept for human-centered automation of the ATC operations in the terminal area, including scheduling landings and takeoffs, generating flight paths, and monitoring conformance and hazards. It assumes that a metering process is in effect leading to a controlled flow of aircraft into the terminal area. The human-centered automation concept lies in automating the tasks of the human controller that are algorithmic, repetitive, and rule-based and therefore can be easily handled by a computer. The human controller is then freed from the load of the tasks that do not require his creativity and is able to use his knowledge and experience in planning a more efficient use of the runway system. The interaction between the computer and the human controller becomes an important issue, and ASLOTS needs to provide tools to the human controller that allow for the easy execution of the planning. The following describes how the ASLOTS concept approaches automating the ATC operations of scheduling, path generation, conformance monitoring, and hazard monitoring.

A schedule for the runway system operations including landings and takeoffs is automatically generated and updated based on the supply of landing aircraft from the metering process and takeoff aircraft from the ground. This schedule tries to maximize the throughput of the runway system. However the actual arrival times of the aircraft will differ from those estimated and assigned by the optimal schedule due to the stochastic nature of the flow and the disturbances in the system. The human controller is provided with a limited ability to modify the sequence and spacing between landing aircraft and to insert takeoffs and missed approach aircraft in the landing flow. With this ability the human controller can close some of the gaps that result in the original schedule and use the runway more efficiently. There is then a new generation of the schedule.

The interface that ASLOTS provides to the human controller to modify the sequence and spacing between aircraft is illustrated in Figure 1.1. Slot markers that correspond to the aircraft scheduled to land and takeoff are displayed along extensions of the runway centerlines. The symbol for the takeoffs differs from that of the landings. The slot markers are positioned and sequenced according to the expected arrival times and takeoff times of the aircraft. They move along the centerline towards the runway at the approach speed declared by the pilots such as to arrive at the runway threshold at their scheduled time. They are a graphical representation to the controller of the current landing and takeoff schedule. As the aircraft intercept the centerline of the runway they meet their corresponding slot markers, and the slot marker at that point disappears. Using the mouse the human controller can move the slot markers along the centerline within a certain range which represents the maximum allowable modification of the aircraft landing schedule. The allowed range is determined by the space limitations and the minimum separation requirement between the aircraft. By moving the slot marker the controller effectively changes the landing time of the aircraft and changes the point where the aircraft intercepts the slot marker on the centerline. By changing the sequence of the slot markers, the controller can change the sequence of the landings. Using the takeoff slot markers the controller can change the takeoff times and insert takeoffs between any pair of landings.

A flight path generator generates a planned path for each aircraft which leads it to intercept its corresponding slot marker at the centerline. In order to compute the path, the ASLOTS path generator uses mainly a dynamic model of the capabilities of the aircraft in terms of its speed, acceleration, and turn rate; a model of the environment in terms of wind speed; and a stan-

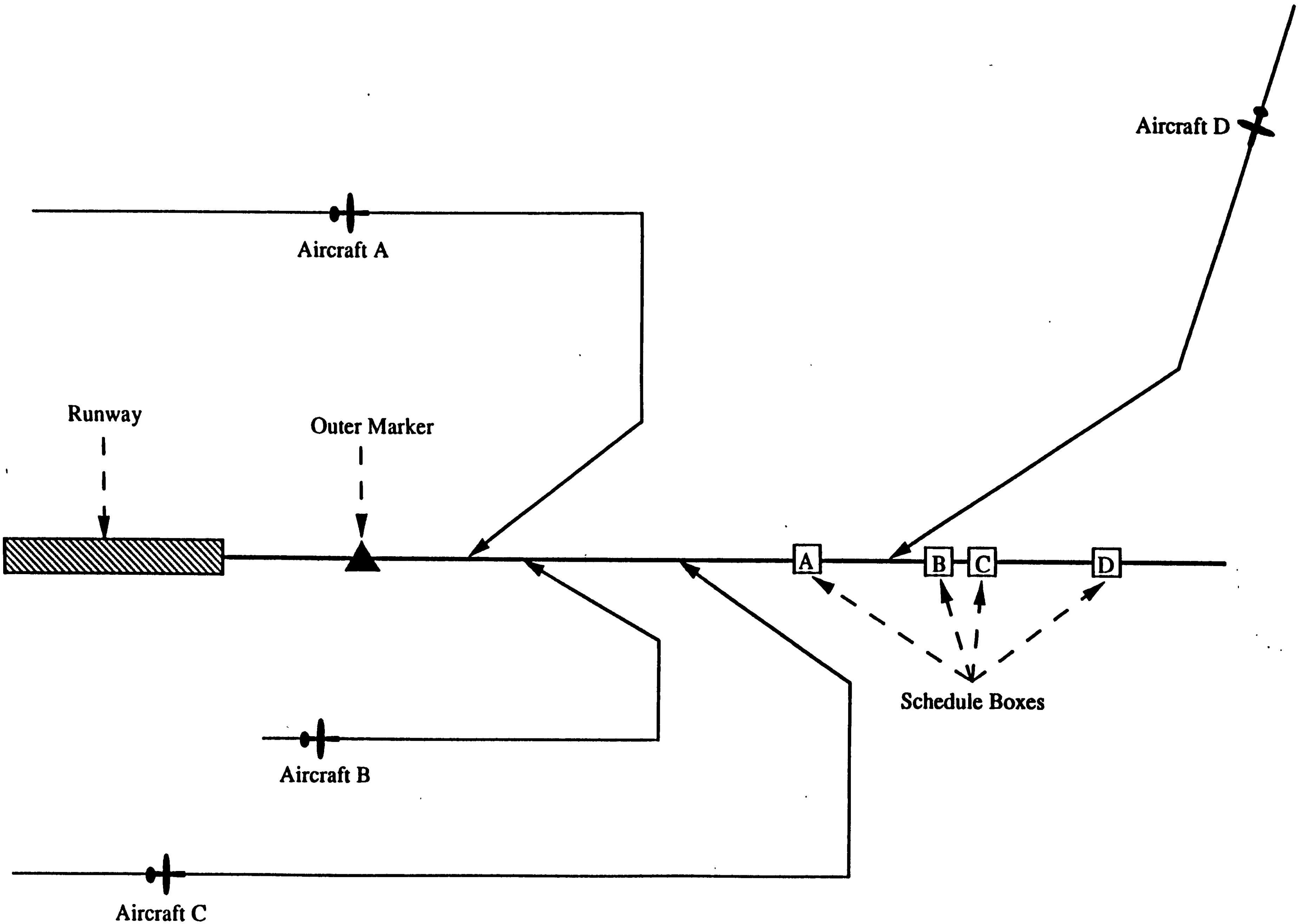

Figure 1.1: The Aslots concept

dard pattern for the arrival path determined on its arrival fix. Two standard patterns: the trombone and the direct patterns are shown in Figure 1.2; they specify the geometry of the path and the speed profile. Given these information and the initial states of the aircraft and the corresponding slot marker, the path generator determines the times when the aircraft should turn from one leg of the pattern to the next and when a speed change is required in order to meet the slot marker at the centerline. When the controller changes the location of a slot marker, the path generator automatically generates a new path for the corresponding aircraft such that it meets the slot marker in the new location. Having computed the times for the aircraft turns and speed changes, ASLOTS provides cues to the controller advising him when to issue the appropriate clearances to the pilots at the appropriate times. This is done graphically using colors, blinking, and appropriate tags on the screen. If data links are available, ASLOTS would send the clearances to the aircraft and pilot automatically.

Path conformance errors will result due to pilot response errors, wind and other atmospheric disturbances. ASLOTS uses the surveillance and tracking data from the radar or the satellite to monitor such conformance errors. Adaptively, ASLOTS generates new paths for the erroneous aircraft such that they will stay on schedule and meet their slot markers. When the errors result in the infeasibility of a path or cause a hazard such as a collision or a missed approach, ASLOTS generates the appropriate cues to the controller. In this way any conformance error has immediate compensation.

Automation is essential to improving the ATC operations especially in the terminal area which is at the heart of the congestion problem. ASLOTS was introduced in this chapter as a human-centered automation system introducing automation to all the operations in the terminal area including scheduling, path generation, conformance monitoring, and hazard monitoring. ASLOTS allows a spectrum of levels of automation and of possibilities of interaction between the human controller and the computer. The design and the development of the ASLOTS automation system is the subject of the rest of this proposal. Chapter 2 models the ATC system in the terminal area as a control block diagram. The automation is introduced into the block. The problem of the automation of such a system is then stated with the goal that its solution tries to achieve. Going over the complexities, the nonlinearities, and the uncertainties that are characteristic of the components of the system, the case is made for a human-centered approach and for the use of a

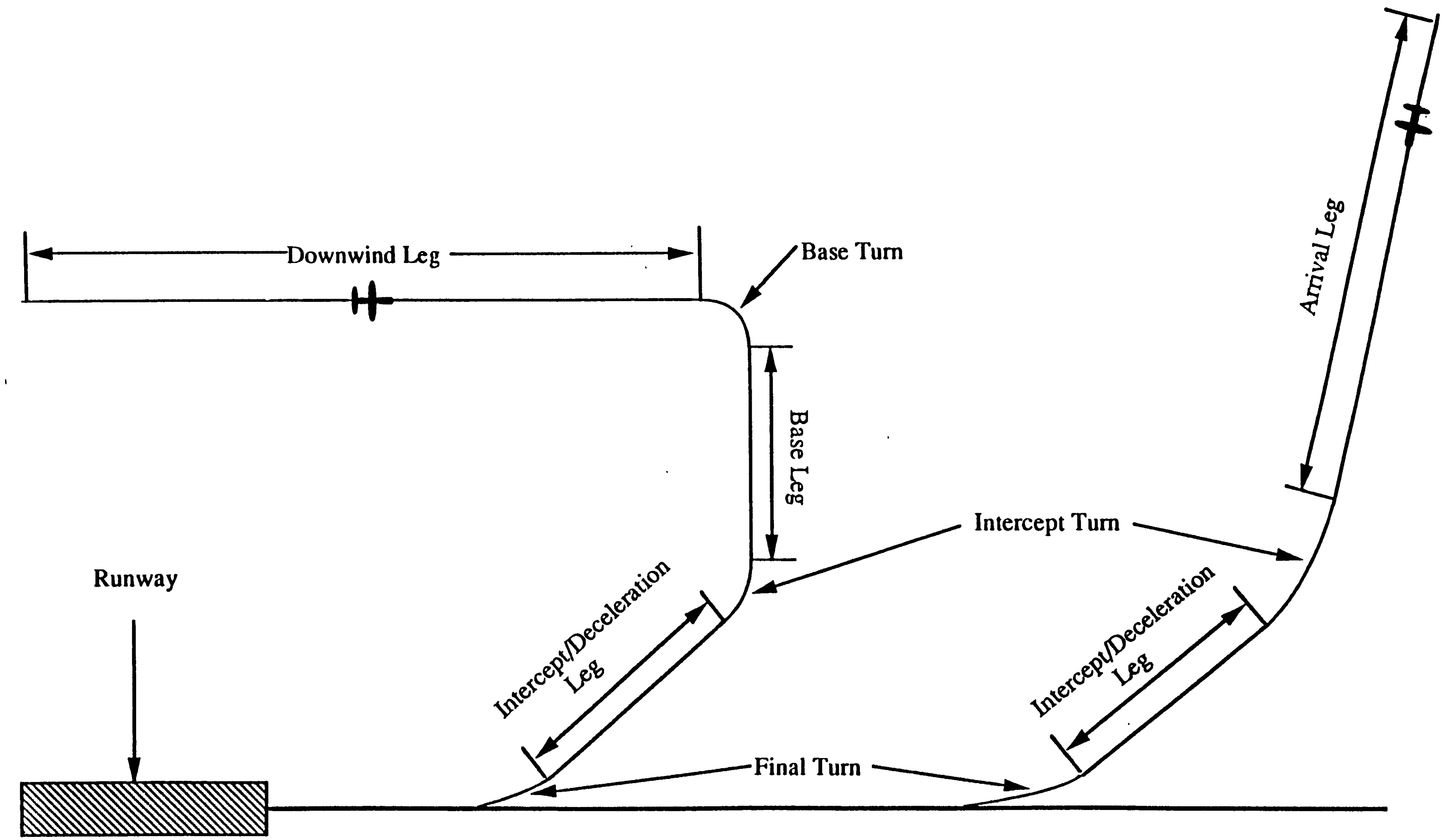

Figure 1.2: The trombone and direct patterns

non-conventional intelligent control structure to implement the automation. In Chapter 3 the intelligent control approach is investigated by outlining the different available tools and their relevance to the problem at hand. A hybrid intelligent control structure is proposed with a plan to implement it in progression. The human-centered approach is discussed in Chapter 4 where the automation problem is reduced to an allocation of tasks and responsibilities between the human and the computer. The different approaches to this problem are also outlined and an approach is proposed to model the human-machine interaction and to choose a suitable level of automation. In the last chapter a scenario is proposed to experiment with and test the tools suggested in this proposal.

Chapter 2

Problem Statement

2.1 The system in block diagram

Figure 2.1 models the task of the air traffic controller in a simple control block diagram. The plant (the controlled component) is the aircraft and the pilot flying it. The task of the controller is to produce and deliver the appropriate control clearances to the plant to keep it on a desired track. The plant is remote from the controller and they communicate through either radio or digital data links. The radar surveillance measures some parameters of the current state of the aircraft and displays them to the controller on the radar screen. The controller is concerned with the conformance error between the current state of the aircraft and the desired state on the reference track. He produces the clearances that would eliminate any such errors hoping to deliver the aircraft to the runway at the scheduled time. In doing so, he follows the procedures of the ATC system which impose separation constraints in space and time between aircraft.

Figure 2.1 shows that the controller is often in charge of more than one aircraft. The controller then has to deliver clearances to all the aircraft keeping each on its track. He tries to make them all meet their scheduled times and also keep them from violating the separation criteria established by the ATC procedures. Overall his main concern is to keep the throughput of the runway as high as possible in terms of the number of operations (landings and takeoffs) per hour without violating the ATC rules.

The global system therefore consists of all aircraft, taking off and landing.

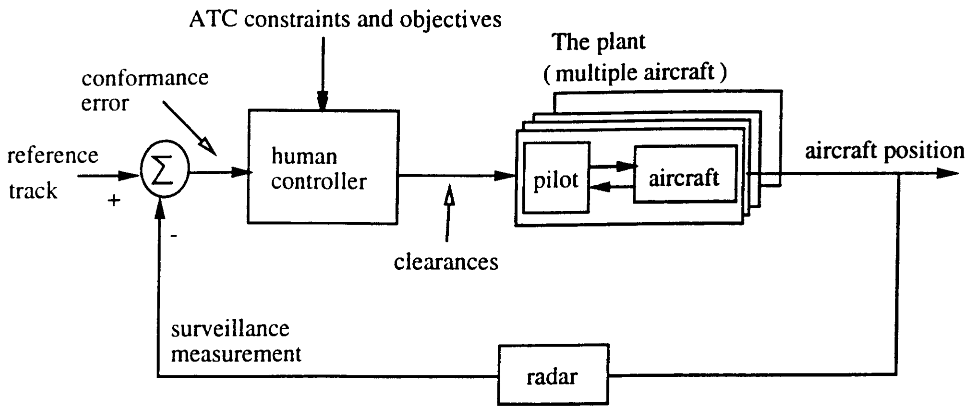

Figure 2.1: The ATC system in a simple control block diagram

Each one-aircraft system is a subsystem of the global one, in the sense that when the controller deals with one aircraft he cannot separat it from the others. This is made evident in the following statement of the problem.

2.2 The ASLOTS problem statement

ASLOTS was introduced as a computer-based human-centered automation system that produces the appropriate clearances to the human controller. The choice of the aircraft's desired path and scheduled time (slot marker location on the runway centerline) are determined interactively by the human controller and the computer. The automation system is included in the control block diagram in Figure 2.2.

When ASLOTS computes a path (in terms of planned clearances) for a single aircraft to meet its slot marker, one is dealing with a subsystem. However, the objectives and constraints in solving this subsystem problem are tied together in the global traffic system. For example the location of the slot marker for an aircraft is chosen based on a schedule of all aircraft. The space management which constrains the space allowed for each aircraft is also determined considering all aircraft. The minimum separation criteria involves both the aircraft for which a path is generated and the neighboring aircraft. The objective for a single aircraft is to meet its own slot marker, but the location of this slot is determined such that the throughput of the runway and the safety of the global traffic system are improved. Therefore

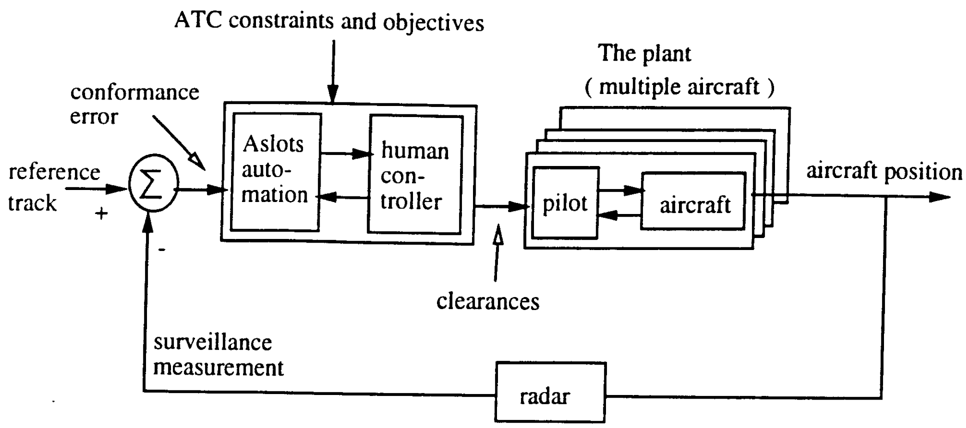

Figure 2.2: The ATC system with the ASLOTS automation

although one is instantaneously dealing with a one-aircraft subsystem, one is concerned with the constraints and the objectives of the global traffic system.

The following statement formalizes this idea emphasizing that in order for the automation to be justified and applicable, one has to be able to rely on it and benefit from it:

The sub-problem: Can one reliably, and optimally, find a solution that delivers an aircraft to its slot marker at its scheduled time, within the constraints and objectives of the global traffic system ?

The global traffic problem: Can one improve the performance of the global traffic system by ensuring the solution to the sub-problem ?

Where reliably and optimally refer to:

- Reliably: The control system should always find a feasible way to deliver the aircraft to the runway to avoid catastrophic consequences.

- Optimally: When a number of solutions exist the control system should choose the one that optimizes the objectives in the global traffic system for a high flow rate.

2.3 A 3-step approach

Given the statement above, the following approach is suggested.

1. Determine the constraints and objectives of the global traffic problem.

2. Design a control system for the sub-problem which takes into account the constraints and objectives of the global traffic problem and solves the sub-problem.

3. Test the control system design in terms of the performance of the global traffic system.

The second item is further divided into 3 steps to design the control system:

1. Obtain a model of the system to be controlled.

2. Determine the uncertainties and nonlinearities in the model.

3. Design a controller that will cause the system to behave as desired incorporating knowledge of the model and uncertainties.

In the following sections this approach is followed through resulting in a proposal for a control design structure.

2.4 Global traffic system constraints and objectives

2.4.1 Global traffic system constraints

- Space management: The flight paths should lie within the allowable space.

- Minimum approach separation: The aircraft should be separated according to ATC rules on the centerline of the runway. This results in setting the capacity of the runway system.

- Collision and conflict avoidance: The aircraft should be separated according to ATC rules throughout the approach and departure.

- Time constraints: The aircraft should meet the centerline at the scheduled time according to a global traffic schedule for all the aircraft.

- Workload smoothing: The maximum peak workload should be less than the controllers can handle in directing all the aircraft.

2.4.2 Global traffic system objectives

- Reduce the minimum ATC separation criteria applicable to use of the automation. This results in the increase of the throughput of the system, by increasing the number of operations on the runway in a given time, to achieve the theoretical capacity.

- Develop a better and more flexible scheduling system.

- Reduce the workload on the controllers and allow for the use of their creativity by automating their manual tasks and involving them in the planning of the arrival and departure flows.

- Introduce a variable level of automation appropriate to varying circumstances.

- Improve the performance of the system in conditions of bad weather and low visibility.

2.5 The characteristics of the single aircraft subsystem

2.5.1 A human-centered model for the subsystem

In the statement of the problem mentioned above it is suggested that the automation is justified when it is both reliable and optimal. The optimality is needed so that the solution to the problem is most beneficial in terms of the global objectives. This is achieved by choosing the optimal solution in the context of some objective whenever a number of feasible solutions exist. A number of objectives were stated above. Some of these can be quantified like the runway throughput but most are very qualitative and subjective, like the notions of safety, workload and flexibility. Choosing the objective function is therefore a complexity that makes the problem more challenging. For example the problem of computing the path that leads an aircraft to its slot marker on the runway centerline is formulated as a linear program in Sadoune [5]. The solution is to be chosen from a feasible polyhedral set. In Chi's solution, the selected objective is to minimize the time that the aircraft spends on the downwind leg of the path pattern [13]. The chosen solution then lies on a corner of the feasible set. Sadoune on the other hand suggests that such a solution is very inflexible since any slight disturbance causes the solution to become infeasible. He suggests that the objective should be more flexibility and therefore the solution should be chosen far from the boundaries of the feasible set. In the current development of ASLOTS, Sadoune's argument is selected and the solution is chosen in the middle of the feasible set (the feasible set in this case is a line segment).

The reliability of the automation system refers to whether a feasible solution that delivers the aircraft to the runway is always guaranteed. The degree of automation is important in answering this question. Is it possible to design an automatic controller that can find a solution in all possible situationss (including emergencies and missed approaches for example)? Or, Does the human controller have to intervene to fill the gap in the situations where the automatic controller is unsuccessful? Even if one was able to list all the situations that one can think of, and automate the procedures needed to handle these situations, the stochastic nature of the air traffic system could always bring obout unexpected situations and emergencies that the automatic controller could not handle at all or in a beneficial manner. As mentioned in Chapter 1, although recent development in the computer technology, both in hardware and software, allows excessive degrees of automation, experience shows that more automation is not always beneficial. This led many researchers and designers in human-machine systems to call for the adoption of a human-centered approach [14, 15].

Following this argument it is assumed unlikely that ASLOTS would be able to find a solution for an aircraft in all possible situations. A human-centered approach is adopted where the human controller is left in the loop. This is shown in the model of the system where both the human controller and the automation (ASLOTS) are included in the controller block (Figure 2.2). The design of such a human-machine system becomes a question of allocation of tasks and responsibilities between the human and the machine. This problem could be either static where a fixed optimal allocation is selected once and for all, or dynamic where the allocation of the tasks is allowed to change as the conditions change. This task allocation problem is essential to the design of the automatic controller (ASLOTS) and the interaction between it and the human controller. Therefore it is addressed in more details in Chapter 4 with a proposal for further research.

The model of the system is therefore a human-centered one which keeps the human controller and the automatic controller in the loop. In order to choose a design structure for the automatic controller it is important as suggested in the design steps above to point out the characteristics of the system components. This is done in the next sections with an emphasis on the nonlinearities and uncertainties in these components.

2.5.2 The plant

The plant in this system consists of the pilot and the aircraft. They are remote from the controller and have to communicate through radio or data links.

A detailed model of the aircraft is implemented in a simulation which is based on the model developed in [16]. This model includes the following nonlinearities and uncertainties:

- Nonlinear equations of motion which integrate the acceleration of the aircraft into speed, altitude, position and heading.

- The control loops of the aircraft guidance are closed so that the aircraft will reach its desired state in terms of speed, heading and altitude commands. However, delays are incorporated in reaching the final state.

- Instrument errors are introduced so that the aircraft would not actually reach the commanded state, although the pilot might think it did by reading the instrument indication.

- Wind also introduces errors since the aircraft is commanded an airspeed. The wind causes the ground track and speed to deviate from the desired values.

Some of the errors above might simulate minor failures. However, more serious failures are not simulated, and the control system needs to be tested against such more catastrophic cases.

The pilot is also not modeled in the simulation. In modern aircraft the pilot is simply setting the commands (heading, speed, and altitude). Therefore, he could be modeled as a delay.

The more important factor though is the uncertainty introduced by the pilot. The human error can go as far as forgetting the command completely.

Therefore, it would be needed to introduce a probability of failing to respond to a command or to respond to it in a wrong fashion.

2.5.3 The surveillance

A radar surveillance system is simulated and it incorporates several sources of errors. These include quantization, random noise, and probability of dropping a measurement.

The radar provides information about the position of the aircraft. A tracking system helps estimate a better position, the speed, and the heading of the aircraft. The tracking system is not implemented in the simulation currently but is important for the control system design.

Another important factor in the surveillance is the sweep rate. The sweep rate might dictate the sampling rate of the control system if it is slower than the control system loop. That is the control system cannot look at data any faster than provided by the radar.

In the future, the radar surveillence could be substituted by satellite surveillance and data links for communication. This allows much higher rates and much more state information on the aircraft. It would also be possible to estimate the wind with the knowledge of the aircraft airspeed and heading via data links. These new development should influence the simulation of the surveillance system.

2.5.4 The human controller

The human controller is a critical component of the system because it constitutes an essential part of the controller being designed. (Namely the tasks of the controller are allocated in cooperation between ASLOTS and the human controller as mentioned earlier). Using the Rasmussen model [17] the tasks of the controller can be simplistically divided into skill-based, rule-based, and knowledge-based tasks as follows:

- Skill: Manual delivery of the vectors to the pilots.
- Rule: Manual-cognitive decision on the vectors needed and their timing based on given rules. The rules are set by the trajectories that have to be followed.

- Knowledge: Cognitive planning including scheduling and choice of trajectories in order to optimize the goals of the system (like safety and throughput of the runway).

[18] offers a task analysis of the controller based on a division of tasks between perception, cognition, and manual response. It models the human controller as an event-driven processor with multiple levels of interrupts.

The ASLOTS automation system offers to assist the human controller in all the task levels. For example using data links the computer can send the clearances to the pilot substituting for the skill-base task of the human controller. Automatic scheduling and path generation assist mainly in the rule-base task by computing the appropriate clearances. Using such assistance, the current manual and cognitive workload of the human controller is reduced, providing free time for the human controller to engage in more creative planning tasks such as schedule modification. Such creative participation from the human controller is needed because of the importance of the knowledge base that he acquires from experience. Such knowledge base is currently wasted because the human controller may be overloaded with the manual generation and delivery of clearances.

2.6 The automatic controller

The design of the automatic controller depends on the allocation of tasks between the human controller and the computer. As mentioned earlier such an allocation may need to be dynamic and flexible in accordance with the changing conditions. Therefore the design of the automation should also be flexible in order to accommodate such dynamics. For example, the automation should be able to accomodate different geometries, and different traffic mix and level. The flexibility of the automation system is also needed to accommodate changes in the technology and/or needs of the ATC environment. ATC needs may require an allocation of more tasks to the computer as the fast advance in technology makes the computers more intelligent. Another reason for flexibility is the progressive introduction of the automation in the terminal area operations. It is more likely that the automation would be introduced at a low level and increased gradually in order for the transformation to be smooth and effective. A flexible design would allow this gradual change to be achieved with minimal changes in the automation structure.

The automatic controller should also be able to handle the complex nature of the system. This complexity results from the nonlinearities and the uncertainties that are present in the components of the system. The dynamics of the aircraft, the wind disturbances, the radar surveillance errors, and the human errors of the pilots and the human controller are sources of such complexities as described earlier. It is very hard to model and predict the behavior of such a human-machine system, especially when considering the global traffic system with all the aircraft included.

With a human-centered automation approach the automatic controller in cooperation with the human controller should be both flexible and able to deal with this complex system. In a simple scenario and assuming that an initial runway operations schedule exists, let's assume that the automated system has the following tasks:

- Find the path for the aircraft to meet the slot marker.
- Move the slot marker to a new location within its range.
- Generate a clearance for the aircraft to move to a new position.
- Monitor conformance errors.
- Deliver the aircraft clearance.

These tasks are essentially the path generation, scheduling, and conformance monitoring functions in the terminal area. It is assumed that all these tasks are shared according to a flexible automation structure by the automatic and the human controller.

With these tasks the automatic controller can perform the following simple logic: Given the position of the aircraft and the slot marker, find the path within a pattern that leads the aircraft to the slot. If no feasible path exists move the slot marker to a new location and/or command the aircraft to a new position where a feasible path exists. If a feasible path is found deliver the clearances at the appropriate times. If all steps have failed to find a feasible path, the human controller then has to intervene and use his creativity to deliver the aircraft to the runway perhaps taking more risks than the automatic controller is allowed. For example the human controller could change the landing sequence of the aircraft or declare a missed approach. At

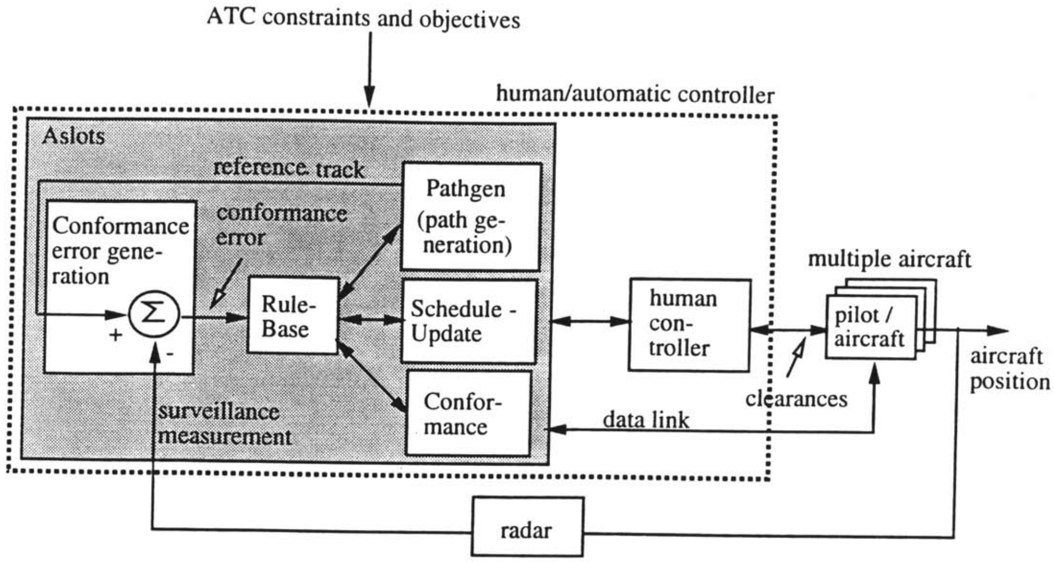

Figure 2.3: The human/automatic controller in the ATC block diagram

any point if the conformance error becomes large according to some criteria, the automatic controller can generate a new path.

In order for the automatic controller to switch between one task and another, it needs a higher level decision making mechanism. This is provided by a Rule-Base that coordinates between the tasks by deciding which task (or tasks) is to be performed at a certain time.

The suggested control system format is included in the block diagram of the system in Figure 2.3. A link connects the human controller with the ASLOTS block showing his interaction both with the Rule-Base and with the different tasks. The human controller is able to influence the Rule-Base suggesting that he can teach it or modify it. He can also move the slot markers, choose a path or a path pattern, and maintain conformance by monitoring the error, and correcting it.

Only five tasks are included in the scenario above; a path generation task, a scheduling task, a clearance generation task, a conformance monitoring task, and a clearance delivery task. The scenario described however is a very simple one. A real situation is more likely to involve a larger number of tasks (or subtasks) and to require handling abnormal conditions such as a sudden runway closure, change of runway, an emergency arrival (because of fire onboard for example), and missed approaches (see [18] for a detailed task analysis). The multi-task structure that is suggested is flexible and can be extended to include new conditions and situations as deemed necessary. For example an algorithm that handles a missed approach can be easily added as an additional task and the Rule-Base can be modified to accommodate the new task. In a multi-processing environment the multiple tasks can be run on separate processors adding to the computational power and making the structure more flexible and extensible.

Whenever the Rule-Base switches to a certain task the corresponding algorithm is executed. The algorithm associated with the path generation task (Pathgen) is the pattern-based one that is currently developed and described in the next section. The scheduling task associated with moving the slot marker is currently assumed to be manual, but it can be automated with a corresponding scheduling algorithm. The same is true for the other tasks that can be programmed and added to the automatic system as the level of automation is increased. Also the suggested structure of the automation allows any such programmed task to be enabled or disabled without affecting the rest of the system. This allows the level of automation and the task allocation between the human controller and the computer to be chosen dynamically as desired.

2.7 Pathgen - the path generator

The structure of the system when the rule base is switched to the path generation algorithm is shown in the block diagram in Figure 2.4. The components are described in the following items:

- The Pathgen algorithm is an optimization algorithm that computes an optimal path for the aircraft given the constraints and the objectives provided by the global system (which includes all aircraft). For example, the schedule provides the time that the aircraft needs to meet the centerline by assigning the slot marker, the space management puts limits on the space allowed for the path, and collision avoidance may require aircraft to fly at different altitudes. The problem is setup as a linear program and the optimal solution is chosen in the middle of the feasible set for optimal flexibility [5, 13].

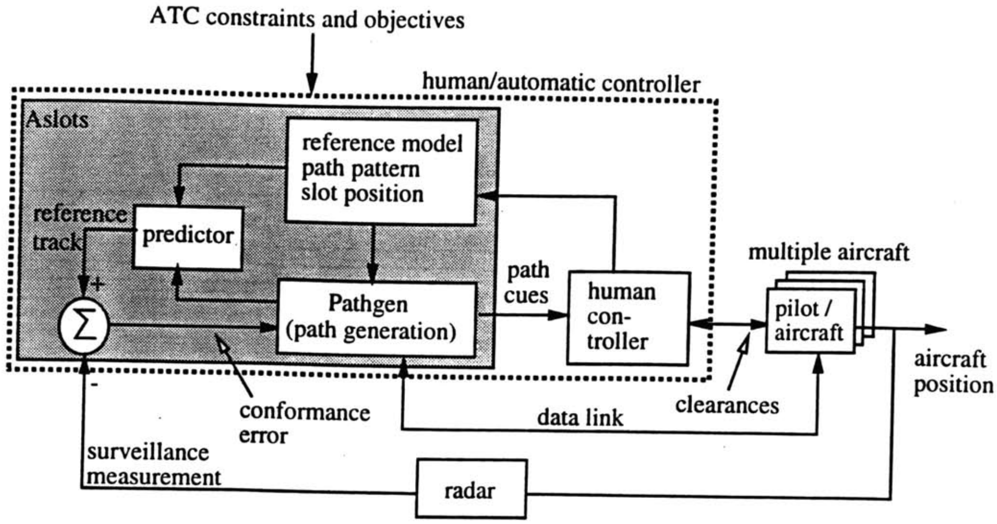

Figure 2.4: The path generation automated algorithm: Pathgen

- To compute the path, Pathgen uses a prescribed geometric pattern such as the trombone and the direct patterns described in Chapter 1 (Figure 1.2). Both the human controller and the Rule-Base are able to choose the pattern.

- The slot assigned to the aircraft dictates to Pathgen a set of points and times where the aircraft can intercept the runway centerline. As described in the next section, both the human controller and the Rule-Base can change the slot position (currently only the human controller). By doing so they may affect the landing schedule of all subsequent aircraft.

- Pathgen also uses a reference model which includes the aircraft dynamics, the wind, and can also compensate for pilot delays. The reference model is shown to be adapted on-line to changes in the wind, the dynamics, or the pilot behavior.

- A predictor uses the reference model, the computed path, and the pattern to compute the predicted desired state of the aircraft at the current time. This state is compared with the measured state of the aircraft to generate a conformance error which is the input to the closed loop control system.

- By closing the loop, Pathgen computes a new path whenever the error is larger than a certain threshold.

The conformance error generation can be part of a separate automatic conformance monitoring task as suggested in the scenario above.

2.8 Shedule-Update algorithm

The slot marker can be moved by either the human controller or any automatic controller. The reason for the movement could be the infeasibility of the path in the old position, or that a new position produces a more optimal schedule, or to keep the separation between the slot markers on the centerline within the minimum requirement.

The decision about how to move a slot involves scheduling. Scheduling is a problem in the global traffic system since all aircraft are included. When the Rule-Base is switched to the Schedule-Update algorithm, a block diagram similar to that of Pathgen would show the interactions and the flow of the algorithm. The same is also true for any other algorithm that is added as more tasks are automated.

2.9 Two main design questions

In the proposed ASLOTS automation structure more and more tasks can be programmed and assigned to the computer. The computer technology today allows this automation to be pursued to high levels. Two questions arise, one is a question of implementation of the automation and the other is of deciding whether a task should be automated or not.

On the implementation side, any task that has a known algorithm can be programmed. If the task is not well defined there are techniques, such as neural networks, by which the computer can learn the task either off-line or on-line and eventually be able to perform it. However, how practical is this learning ability and can the computer acquire the adaptive and creative nature of the human? The interaction between the human controller and the automation is another major question that affects its implementation. How does the computer with its numeric and objective nature interact with the human with his linguistic and subjective nature?

It is clear that the complexity of the problem at hand requires non-conventional techniques. The questions that are raised in this chapter and the structure that is proposed for the automation system fall under the field of "Intelligent Control." The different techniques that intelligent control offers are presented in Chapter 3 where the proposed structure for the ASLOTS automation is expanded into a hybrid intelligent control system.

The question of whether a task should be automated or not is the question of the allocation of tasks and responsibilities between the human and the computer. As pointed out in this chapter a human centered approach is adopted in the proposed automation system due to the belief that the human controller intervention will always be needed regardless of the level of intelligence of the automation. It was also pointed out that a dynamic task allocation would be more flexible allowing a choice of the level of automation as desired and the proposed automation structure will accommodate this feature. The question of task allocation is discussed in Chapter 4 where a method to model the interaction between the human and the machine is proposed to help determine an optimal or suitable level of automation.

Chapter 3

Intelligent Control

In the previous chapter two major questions concerning the design of the automation system were raised: The question of the task allocation between the human controller and the computer, and the question of the implementation of the automation. The task allocation problem is deferred to the next chapter while this chapter deals with the implementation.

It was pointed out that the automation design should be able to deal with the complex human-machine nature of the system. Some of the complexities and their sources were also described. The case was made for the introduction of the field of "Intelligent Control" which offers many tools that are capable of dealing with such complex systems. It is hard to define the term "Intelligent Control", but essentially it includes hybrid control architectures which combine concepts from control theory and artificial intelligence in order to deal with complex control problems. In the following some of the Intelligent Control approaches are summarized indicating their relevance to the problem at hand.

3.1 Knowledge-Based Control

3.1.1 Concept

This control system includes a knowledge-base or a rule-base which is essentially software that employ different AI techniques. Usually the knowledge-base is an expert system in the form of "if..then" statements. This is best

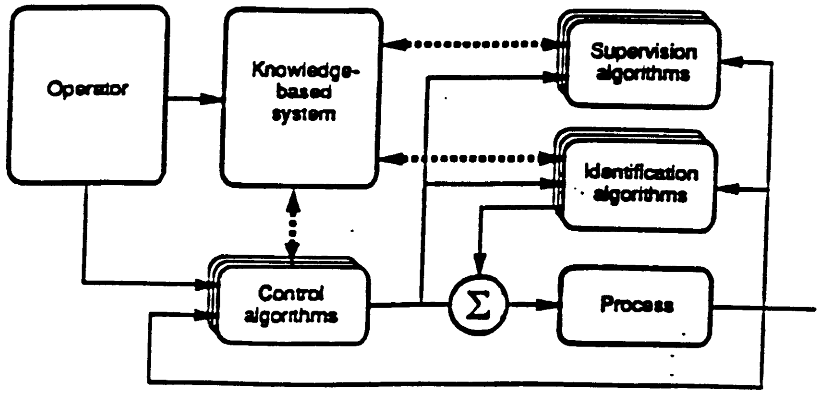

Figure 3.1: An expert control system (from [19])

presented in the expert control system proposed by Astrom [19]. In this system, shown in Figure 3.1, the knowledge-base interacts with the operator as well as the different algorithms (control, supervision and identification) that constitute the rest of the system. One form of interaction is deciding which algorithms to use in a given situation. See [19] for a more detailed description.

3.1.2 Relevance

The control architecture suggested above for the ASLOTS automation system resembles the knowledge-base control architecture shown in Figure 3.1. The Rule-Base is a collection of rules that decide when to switch between a number of algorithms such as Pathgen for generating a path and Schedule-Update for moving the slot marker. The Rule-Base also interacts with the human controller indicating, for example, inquiries and modifications that the human controller can apply to the Rule-Base. Further research into the form and implementation of the Rule-Base and its interaction with the different algorithms and with the human controller is needed. More will be said about this under the Fuzzy Logic Control approach below.

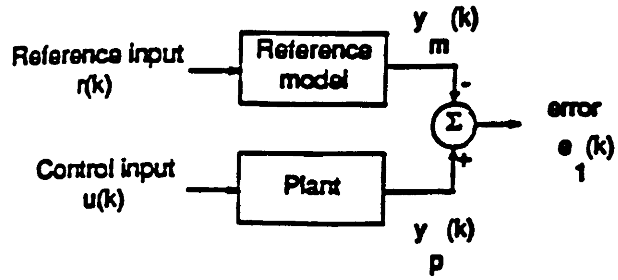

Figure 3.2: The adaptive control problem (from [20])

3.2 Adaptive Control

3.2.1 Concept

Adaptive control deals with the problem of controlling an output in the presence of parametric or structural uncertainty [20]. Traditionally there have been two approaches to adapt to such uncertainties: In the direct approach the controller parameters are adjusted based on the observed error (self tuning). In the indirect approach, a reference model of the plant is used. The plant parameters are estimated, and the controller parameters are adjusted based on the plant estimate. The general problem is to adjust the parameters such that the output of the plant behaves as the output of the reference model. That is, the error in Figure 3.2 is driven to zero. Adaptive control therefore uses system identification techniques to generate a model of the plant. See [20] for a more detailed description.

3.2.2 Relevance

As shown in Figure 3.3 the ASLOTS path generation problem could be modeled as an adaptive control problem where a reference model is used to generate a desired path for the aircraft. Parameters of the model include dynamic parameters of the aircraft such as ground speed and turn rate and environment parameters such as the wind. The pilot can also be modeled by adding estimated delays. There are uncertainties associated with these parameters and therefore estimation and identification techniques could be used to adjust them adaptively. This is shown in Figure 3.3 by feeding back the radar measurement and the error into the reference model block. The control action,

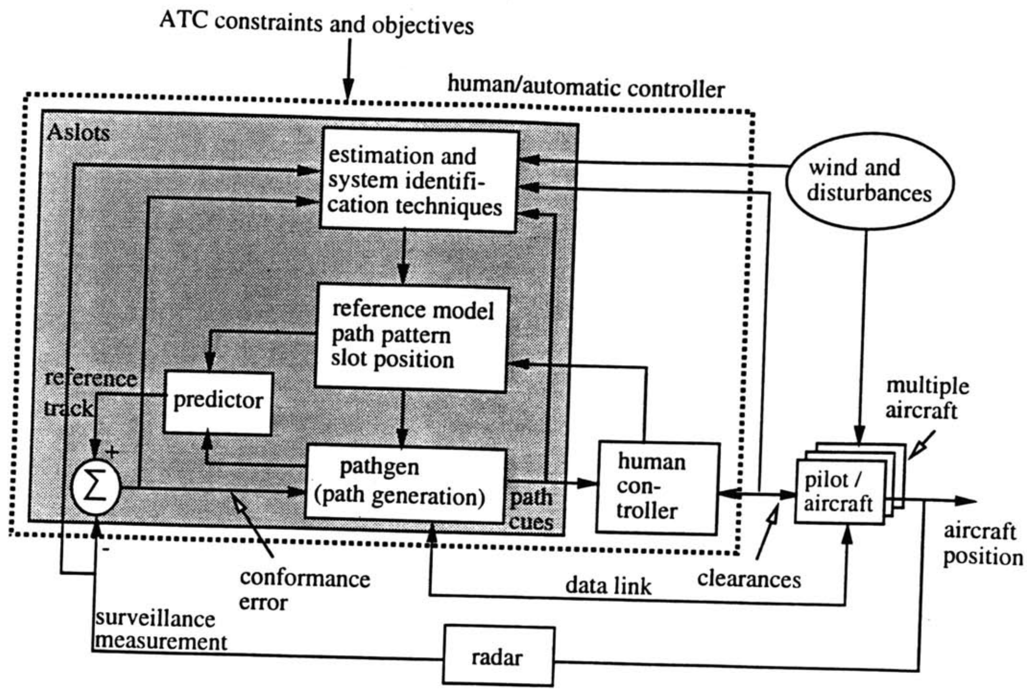

Figure 3.3: The adaptive control concept applied to the ASLOTS path generation task

which is the computed path here, would also be adjusted accordingly. More will be said about the need for the adaptive approach when it is compared with the learning approach in the next section.

3.3 Learning Control

3.3.1 Concept

A learning control system has the ability to improve performance in the future based on experimental information gained in the past, through closed-loop interactions with the plant and the environment [21]. It has the following attributes:

- It is autonomous since it improves its own performance.
- It is dynamic since it varies over time.
- It has memory to exploit past experience.
- It has an objective function to improve performance in its context.
- It has performance feedback to characterize the current behavior.

Learning should be used when the a priori information is limited so that it is impractical to design in advance a system with a required level of performance. Learning could be applied to any part of the system where there is such a lack of a priori information; this includes the model function or parameters and the controller function or parameters.

There are alternatives to the learning approach that deal with the uncertainties in the a priori information. These include robust control where the uncertainties are modeled as best as possible a priori but not on-line. This approach is therefore a fixed approach; and an example of it is "gain scheduling" which uses different parameters of the controller in different regions of the control space depending on the a priori modeled characteristics of that region.

Another approach is to manually execute learning through iterations on control design, testing, redesign and tuning until a satisfactory design is reached. This procedure is usually followed today for flight control systems by industry.

The third approach is the adaptive approach which includes on-line means to accommodate new situations, even if the new situation has been encountered before. The main difference between adaptive and learning control is that an adaptive system treats every distinct operating situation as a novel one, whereas a learning system correlates all past experiences with the current situation and can recall and exploit those past experiences. A learning system would then react more rapidly to changes in the dynamics once it has learned. An adaptive system on the other hand would be slower in reacting to the changes even if the changes are stationary and can be learned and anticipated. Another distinction is that a learning system, by accumulating experience, develops a control law that is suitable throughout the total

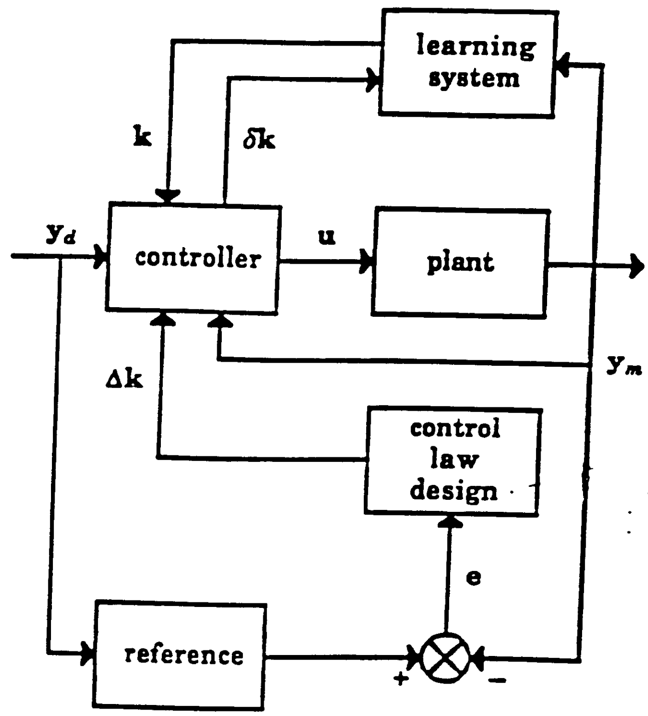

Figure 3.4: Direct adaptive/learning approach (from [21])

operational envelope of the plant. This means that the control law of a learning system is more global, while that of an adaptive system is more local, reacting to the current operational conditions only.

Adaptive and learning systems are therefore complementary. An adaptive system is needed to accommodate novel, time-varying situations which have not been experienced before, and a learning system is needed to accommodate poorly modeled dynamics and stationary behavior. These two approaches are suggested in combination, possibly with an a priori designed controller as well. Figures 3.4 and 3.5 show two architectures for hybrid control systems employing adaptive and learning techniques. See [21] for more details.

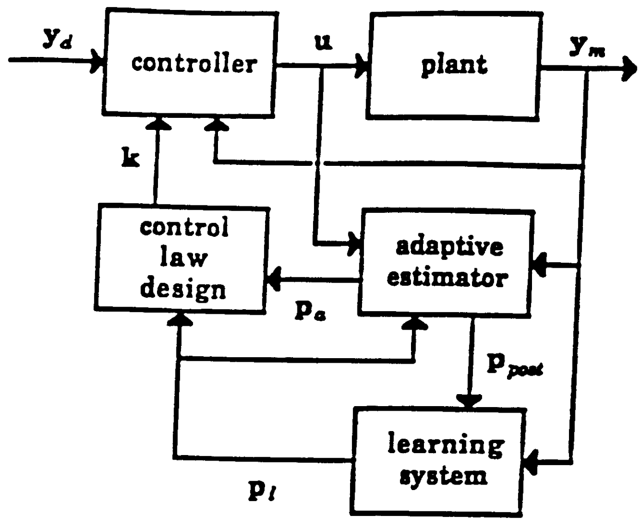

Figure 3.5: Indirect adaptive/learning approach (from [21])

3.3.2 Relevance

As stated above the adaptive approach is needed for the ASLOTS automation system to accommodate the uncertainty and time-variation which result in novel situations that have not been experienced before. This adaptive approach should be complemented with a learning approaach in order to accommodate the slower or stationary dynamics. Learning techniques such as neural networks described in the next section can be used to learn any stationary or recurrent behavior of the plant and the environment. Once this behavior is learned from the past experience its recurrence can be anticipated in the future. Such anticipation would certainly help the automation system to possess faster reaction to a rare event.

3.4 Neuro-Control

3.4.1 Concept

Neural networks are a generic and robust way to estimate functions or mappings. These functions in a control system could be the model of the plant, the inverse of the model or the controller. Figure 3.6 shows a three layer neural network with u as an input and y as an output (v and z are intermediate variables). The main idea is to learn the weights w and b such that the network mappings from the input to the output correspond to the function to be learned.

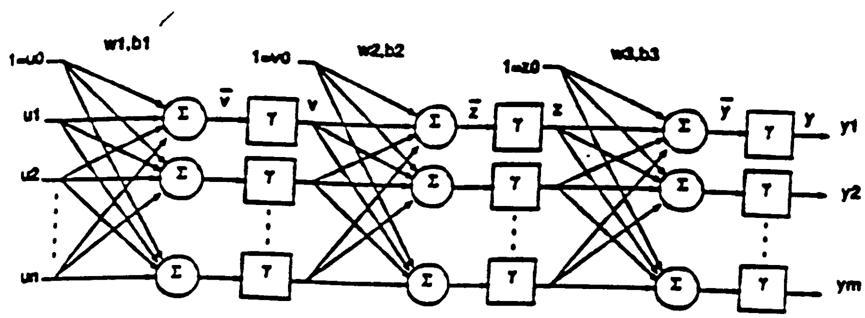

Figure 3.6: A three-layer neural network (from [20])

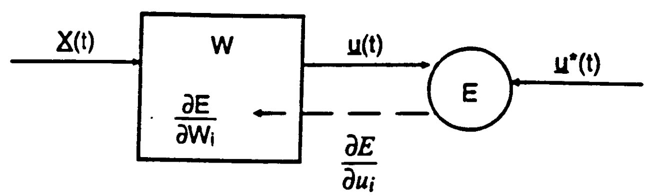

Figure 3.7: Neural network learning through backpropagation (from [22])

Learning is achieved using a data base (examples) of inputs and outputs. This data base is built up from the actual known function or the human expert whose knowledge is to be transferred to the neural network. Figure 3.7 shows the basic idea of learning through backpropagation. The weights W are adjusted by backpropagating the changes in the error E until the error becomes zero.

Narendra [20] classifies the complexities in a control system into three broad categories: Computational complexity, nonlinearity and uncertainty. Because of their parallel nature neural networks are efficient. Because they are generic they can represent nonlinear functions to any degree of accuracy. And because their parameters can be adjusted using input-output data they can be used as adaptive and learning systems under different conditions of uncertainty.

Neural networks are preferred over other approximation techniques like polynomials, orthogonal functions, trigonometric series, splines, and others, because of their implementation in hardware, their robustness (since noise in one node carries very little information in the network), and their suitability to real-time applications. Narendra also reports that it is possible to prove the stability of neural nets where the output depends linearly on the parameters. See [22, 20] for more details about neuro-control.

3.4.2 Relevance

Neural networks can be used with the adaptive approach to add learning capability in real-time. This real-time learning is critical for responding to structural changes which are not anticipated in the design a priori, such as the uncertainties associated with failures and the environment. Neural networks are attractive for real-time learning because of their efficient implementation in hardware. Their generic and robustness described above makes them also attractive since they can be used to approximate any mapping including a human behavior. In theory, then, it is possible to imagine a neural network replacing the pilot or the human controller. Further research is needed to investigate the practical use of neural networks as a learning technique for the ASLOTS automation system.

3.5 Reinforcement-Learning and Adaptive Critics

3.5.1 Concept

These techniques need a model of physical reality (possibly stochastic) and a utility function, and use heuristic approximation to dynamic programming to calculate a critic function. The critic is in the form of a reward or a punishment to the control action, as shown in Figure 3.8. The reward or punishment means that the action is good or bad in the context of the utility function. See [22] for more details.

These techniques are reported to show serious promise for duplicating critical aspects of human intelligence: namely the ability to cope with a large number of variables, in parallel, in real-time and in noisy nonlinear environment. Action dependent adaptive critic (ADAC) is used to account

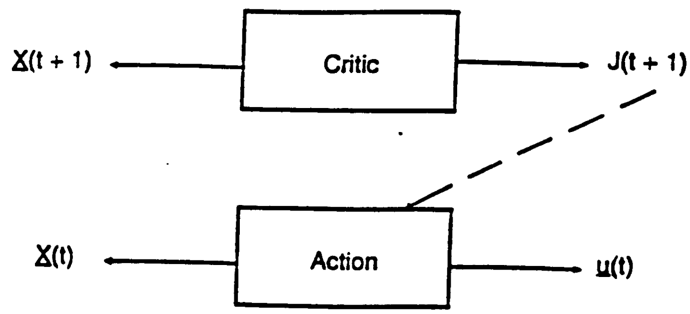

Figure 3.8: The adaptive critic concept: The action is adapted based on the critic function J (from [22])

for errors in model-dependent designs. This leads to more resistance to model errors and to unexpected events.

3.5.2 Relevance

It was pointed out that the control strategy selected for ASLOTS should perform well with respect to the global system objectives. These objectives however are mostly subjective and rarely available in a simple form. Also a simple model of reality is not easily identifiable as indicated in Section 2.5. The need for a model and for utility functions in order to critic the control action in their context seems to be a hard problem. If, however, such a model and a utility function can be identified, the reinforcement learning techniques seem to be worth further investigation.

3.6 Fuzzy Control

3.6.1 Concept

According to [23] Fuzzy linguistic control (FLC) is a knowledge-based control strategy used when:

- Sufficiently accurate and not unreasonably complex model of the plant is unavailable.
- Single precise measure of performance is not meaningful or practical.

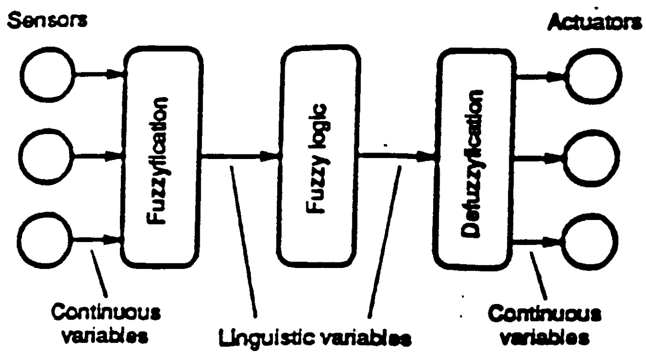

Figure 3.9: A fuzzy controller (from [23])

The concept is to use empirical knowledge about the controlled process in the form of linguistic rules. Figure 3.9 shows the architecture of an FLC.

Crisp numbers are converted to linguistic symbols through the fuzzification process, and symbols are converted to crisp numbers through the defuzzification process. The graphs in Figure 3.10 show the difference between the crisp and the fuzzy definition of the term high, and the partitioning of a variable into three fuzzy subsets: Low, medium and high. In the crisp definition a number has a membership in a set of either 1 or 0 (a number larger than 800 belongs to high with a membership 1, while a number less than 800 has membership 0 in the set high). In the fuzzy definition every number has a membership between 0 and 1 in the set high, representing the degree of belief or certainty with which the number belongs to the set. The fuzzy sets, therefore, have fuzzy boundaries with a gradual increase in the membership (degree of belonging to the set) between the numbers with membership zero (definately not belonging to the set) and the numbers with membership 1 (definately belonging to the set). One result of the fuzzy boundaries of the subsets defined on the variable, is smooth transition between the subsets, rather than ubrupt transition through crisp thresholds.

Once the variables are in a linguistic form, a set of rules (in the form if..then) are applied to them. A control action is determined using one of a number of techniques to aggregate the results of the rules.

The rationale for using fuzzy control is that the symbolic representation of the control algorithm can embody the knowledge of a human operator or expert. The intention is to make the controller behave as if it were the

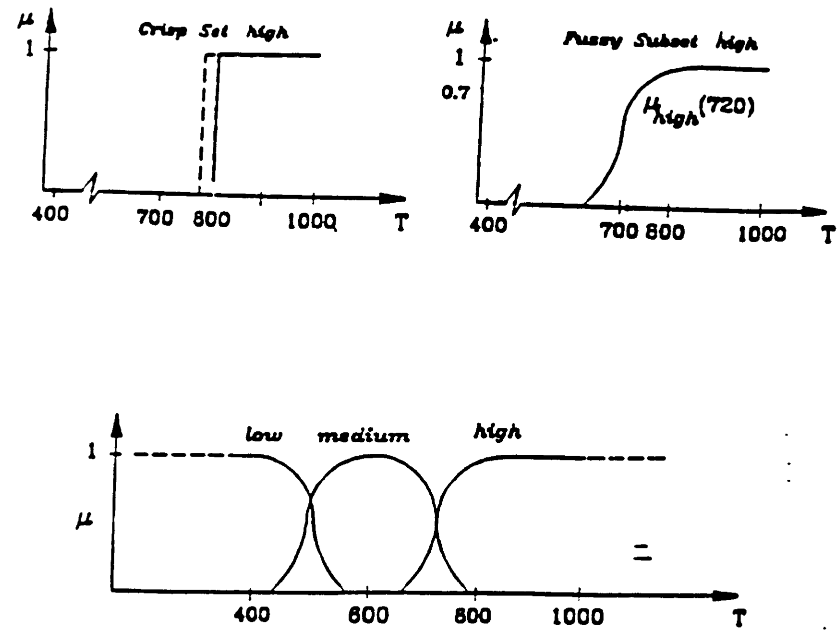

Figure 3.10: Crisp versus fuzzy variables (from [23])

human operator. The fuzzy control approach models the operator rather than the plant eliminating the need for an explicit plant model, and eliminating the need to explicitly translate the external performance specifications into control design objectives. The fuzzy rules include both the plant model and the performance specifications implicitly. [23].

The knowledge or rule base is developed by one or more of the following methods:

- Extracting the expert's experience and knowledge through interviewing (the most common way).
- Modeling the operator's control actions.
- Modeling the process in a fuzzy representation.
- Self organization using learning in a similar fashion to neural networks.

Hybrid architectures of fuzzy linguistic control and other approaches are suggested for the different levels of actions. Saridis [20] suggests three levels of intelligent control: An organization level involving more knowledge and intelligence, an execution level which implements the sequence of actions and involves more precision, and a coordination level which assigns probabilities between the other two levels. A fuzzy control approach is suggested for the high intelligence organization and planning, while conventional proportional-integral-derivative (PID), fuzzy and neural net hybrid controls are suggested for the high precision execution level.

3.6.2 Relevance

The suggested structure for the ASLOTS automation system is hierarchical where a Rule-Base performs a higher level decision making (for example switching between the different control algorithms), while the control algorithms perform lower level control of the plant. The fuzzy linguistic control approach is suitable for the higher level control where the Rule-Base constitutes the fuzzy logic (or inference engine). This Rule-Base is linguistic and consists mainly of rules made by the expert human controller and the procedures of the air traffic control system. The Rule-Base therefore models the human controller and his actions rather than the plant (the pilot and aircraft) in the system. Teaching the Rule-Base about the actions of the human controller would follow the methods described above. Since the Rule-Base is linguistic, fuzzification is needed to convert numerical inputs to the Rule-Base into symbols, and defuzzification is needed to convert symbols out of the Rule-Base into numbers.

The fuzzy linguistic control approach could also be applied at the lower levels in the control algorithms. This is convenient especially when a defined model is not available and the control algorithm simply follows a number of rules or a logic that the human controller usually performs. For example, in the path generation or the scheduling tasks, there are well defined optimization algorithms, but in moving the aircraft to a new location where a path would be found a clear logic is not defined, and fuzzy heuristic rules may be suitable. Also the fuzzy control approach is useful in smoothing the boundaries between different control regions or actions since the thresholds in between would be defined as fuzzy functions rather than crisp or sharp transitions. This makes the control actions and the output of the system smoother and more comfortable.

3.7 Optimal Control

3.7.1 Concept

In this approach the control problem is formulated as an optimization problem with an objective function. The constraints are the dynamics of the system. Dynamic programming techniques are used to solve the resulting problem in the complex cases involving nonlinearities and uncertainties. See [24] for more details about optimal control.

3.7.2 Relevance

Similarly to the adaptive critics approach the relevance of optimal control depends on finding an objective function in the context of which the performance of the control system is to be optimized. This would be relevant where the control strategy is to optimize the performance of the global system, but identifying such objectives in a simple form is a hard problem.

3.8 Expanding the automation system control structure

In this chapter, attention was brought to the relevance of the field of "Intelligent Control" to the ASLOTS automation system. The relevance of each of the approaches mentioned above are summarized as follows:

- Adaptive control can be used to react to fast, time-varying dynamics and events.
- Learning control can be used to increase the flexibility of the system in anticipating events based on past experience.
- Fuzzy control can be used for the higher level, rule-based, linguistic decision making.
- Reinforcement critics or optimal control can be used to adapt the action in the context of the global system performance.

A hybrid control architecture as shown in Figure 3.11 is suggested. It includes the fuzzy linguistic control for the higher level Rule-Base decision making. Fuzzification and defuzzification are needed at the input and output of the Rule-Base. The structure expands one of the lower level algorithms, namely the Pathgen task. It shows the adaptive features of Pathgen where the reference model adapts by monitoring the output of the system and the error. The other tasks are not expanded since the current development of the ASLOTS automation has been concentrated on Pathgen. Learning techniques are shown to affect the Rule-Base as well as the different control algorithms. Also the Rule-Base, the learning, and the control algorithms all behave in the context of the objectives and constraints imposed by the global system. This architecture should be implemented in steps, starting for example with the adaptive features and the fuzzy Rule-Base components. Learning features can be added as needed or deemed necessary.

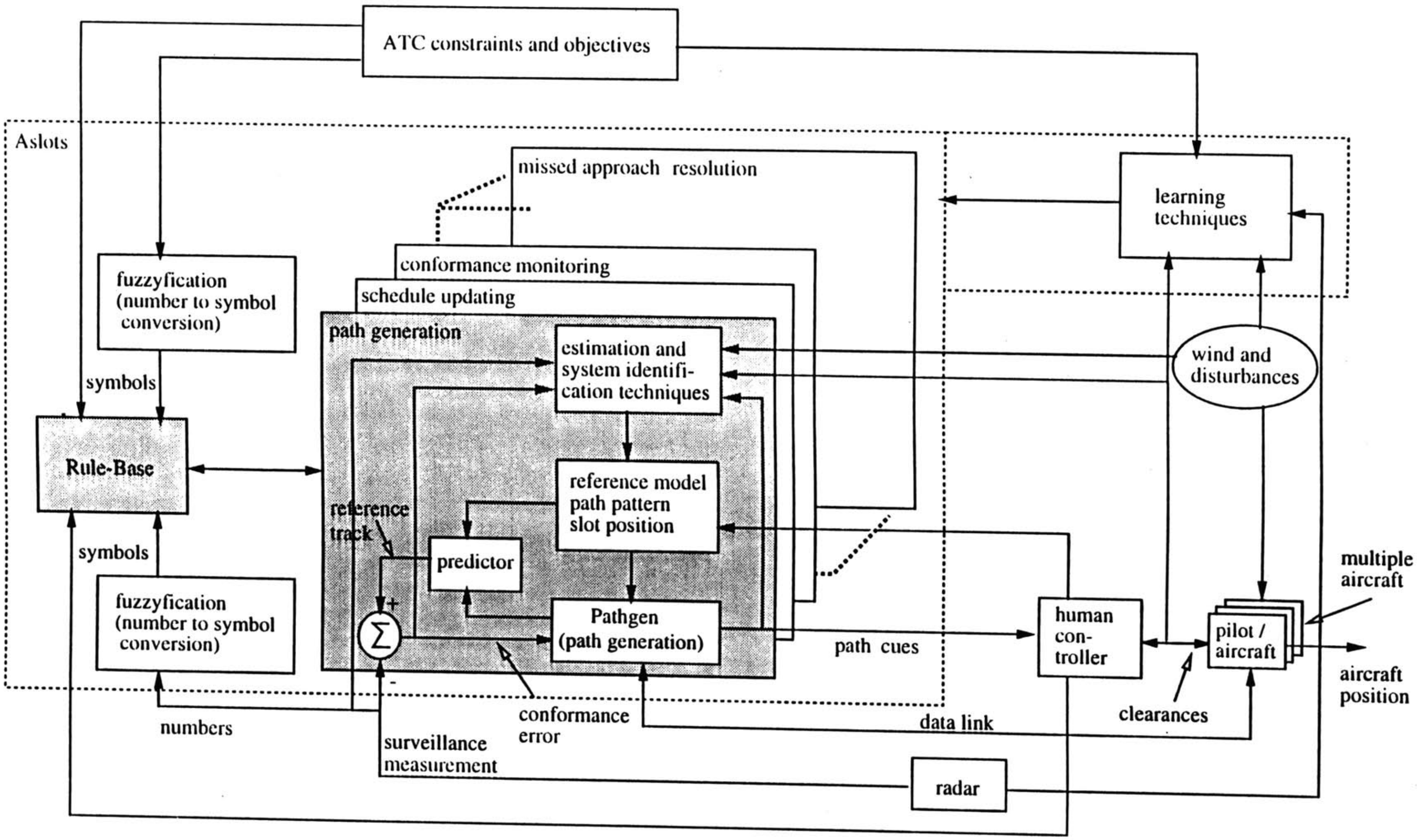

Figure 3.11: A hybrid, intelligent control, multi-tasking structure for ASLOTS

Chapter 4

Task Allocation and The Automation Level

4.1 Introduction

In the design of a human-machine system the human factors engineer is often faced with the problem of a responsible task allocation between the human and the automation. Whether the design is of an automatic machine, a man-machine interface, or a decision aid, the underlying problem is: "What does the human do and what does the machine do"? [14].

Human-machine systems therefore create a real challenge to the old problem of task allocation. Task allocation is a much simpler problem in engineering systems where all the agents executing the tasks are machines. For example parallel processing algorithms allocate multiple tasks to a number of processors whose behavior and interaction are known. Task allocation is also a well understood problem in organizations where all the agents are humans. An example of this is labor division as a resource allocation problem. Task allocation becomes a real challenge when some of the agents are humans and some are machines and humans interact with machines [15]. Human-machine interactions are harder to model and less understood.

Another challenge that human-machine systems introduce is their complex and dynamic nature. The uncertainties in the behavior of the human operator. the differences between human operators, and the uncertainties and time-variation in the process operated are examples of what contribute to the dynamic nature of such systems. This fact explains why choosing the automation level of human-machine systems a priori often failed in practice. Human-machine systems should therefore be flexible and adaptive to the circumstances. This implies that, rather than choosing a fixed task allocation between the human and the machine, task allocation should be dynamic and changes depending on the nature of the task and the agents involved. [14, 15]

In this chapter the task allocation problem between a human and a machine is first presented in the context of human-machine systems with a review of some of the recent approaches to the problem. Then the task allocation problem is formulated simplistically as an optimization problem in task space. This provides insights but proves very simple and hypothetical. Fuzzy logic is then suggested to make the problem more realistic and to be able to use the subjective and linguistic scales available for automation.

4.2 Task allocation in human machine systems

In the context of supervisory control, Sheridan divides a human-machine system into a human operator (or supervisor), a human interactive subsystem (HIS) (or computer), a task interactive subsystem (TIS) (or computer), and a task (or many tasks). The interactive subsystems interact through multiplexed signal transmission (Figure 4.1). This model results in many possibilities for interactions between the human and the task through the interactive channels as depicted in Figure 4.2. Clearly using different combinations of the loops in Figure 4.2 leads to different task allocations between the human and the machine (or machines). The designer is faced with the problem of choosing the "best" of the combinations in some context.

From Figure 4.2 it is clear that the degree of automation of the task varies among the different loops used in the execution. The human on one extreme can perform the task completely manually ignoring the interactive computers. On the other extreme the task could be performed completely automatically through the computers ignoring the human operator. In between there are many levels of interaction corresponding to levels of automation. Sheridan developed a 10-level scale that describes the possible degrees of automation of a task in three phases: analysis, decision making, and execution (Figure

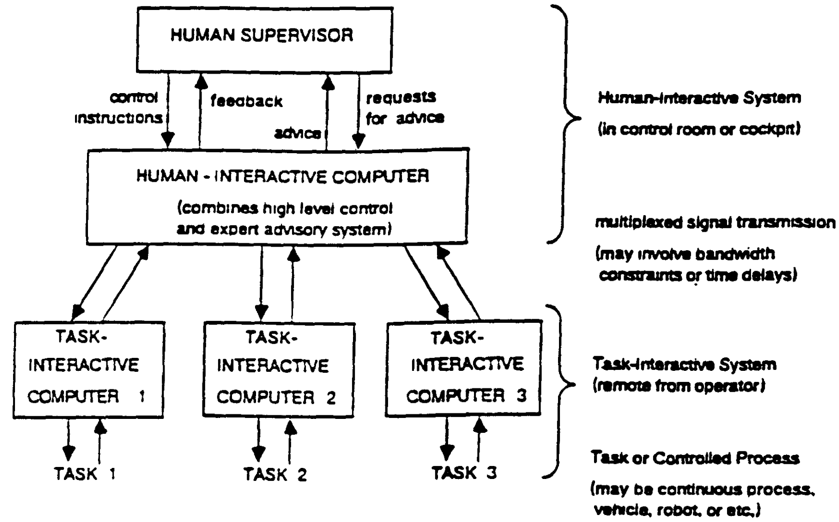

Figure 4.1: Supervision of multiple computers and tasks (from [17])

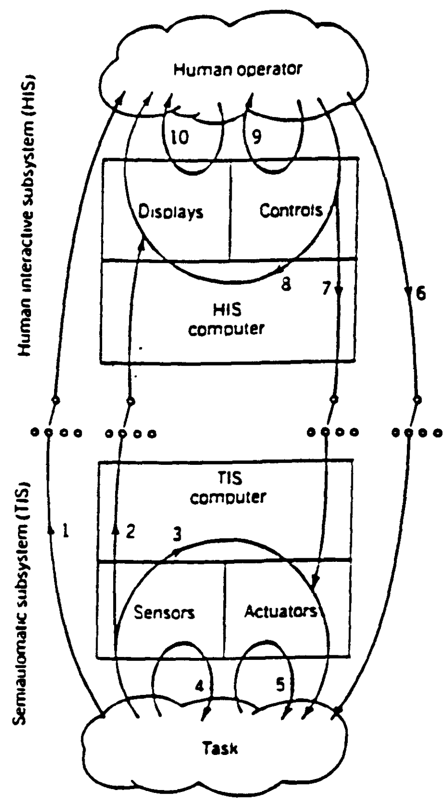

Figure 4.2: Supervisory control as multiple and mirrored loops through the physical system (from [17])

computer —— HUMAN

Increase in Automation Intelligence

Analysis	Decision Process	Execution
HUMAN SUPERVISOR GENERATES ALTERNATIVES	HUMAN SUPERVISOR TAKES DECISION	HUMAN SUPERVISOR EXECUTES
computer generates alternatives	HUMAN SUPERVISOR TAKES DECISION	HUMAN SUPERVISOR EXECUTES
computer generates and selects alternatives	HUMAN SUPERVISOR TAKES DECISION	HUMAN SUPERVISOR EXECUTES
computer generates and adv. best alternatives	HUMAN SUPERVISOR TAKES DECISION	HUMAN SUPERVISOR EXECUTES
computer generates and adv. best alternatives	HUMAN SUPERVISOR TAKES DECISION	computer executes, if HUMAN SUPERVISOR OK
computer generates alternatives	computer takes decision	computer executes, if HUMAN SUPERVISOR GENERATES NO VETO
computer generates alternatives	computer takes decision	computer executes, but must inform HUMAN SUPERVISOR
computer generates alternatives	computer takes decision	computer executes, informs HUMAN SUPERVISOR IF HUMAN SUPERVISOR ASKS
computer generates alternatives	computer takes decision	computer executes, informs HUMAN SUPERVISOR, if computer agrees
computer generates alternatives	computer takes decision	computer executes

Figure 4.3: Level of automation between human (in capitals) and computer (from [14])

4.3). The scale can be applied to any task and can be expanded or reduced as deemed necessary by the actual situation. Choosing a level of automation for a task on the scale means the allocation of the activities of the task between the human and the machine. Both the human and the machine coordinate in the execution of the task according to the description of the automation level.

ASLOTS as described earlier offers the possibility for different levels of automation and interaction between the human controller and the computer. Using the model above ten such levels can be identified in the different ATC operations that ASLOTS tries to automate. The alternatives that are generated by the human controller or by the computer would be alternative paths in the flight path generation task, alternative schedules for the scheduling task, and alternative scenarios for reacting to a conformance error or to hazard in the monitoring tasks. Figure 4.4 shows the ten levels for the path generation task using the paradigm of Figure 4.3.

Path Generation	Path Choice	Sending Clearances
Human controller generates alternative paths	Human controller chooses path	Human controller sends clearances
Computer generates alternative paths	Human controller chooses path	Human controller sends clearances
Computer generates and selects alternative paths	Human controller chooses path	Human controller sends clearances
Computer generates and advises best paths	Human controller chooses path	Human controller sends clearances
Computer generates and advises best paths	Human controller chooses path	Computer sends clearances if human controller ok
Computer generates alternative paths	Computer chooses path	computer sends clearances, if human controller generates no veto
Computer generates alternative paths	Computer chooses path	computer sends clearances, but must inform human controller
Computer generates alternative paths	Computer chooses path	computer sends clearances, informs human controller if human controller asks
Computer generates alternative paths	Computer chooses path	computer sends clearances, informs human controller if computer agrees
Computer generates alternative paths	Computer chooses path	computer sends clearances

Figure 4.4: Level of automation between the human controller and the computer in path generation

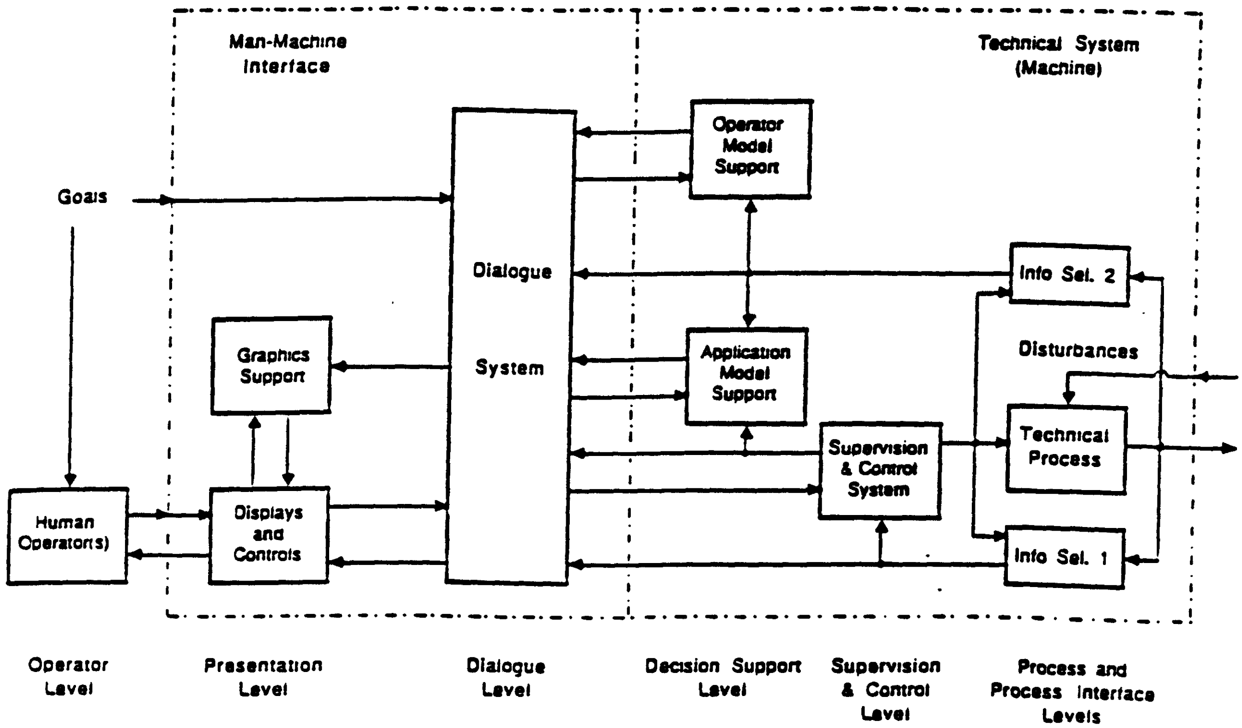

Figure 4.5: Extended operator interface management system structure for dynamical technical systems (from [25])

Johannsen [25], in a similar fashion, divides the human-machine system into six levels: the human operator level, the presentation level providing the displays and the controls to the operator, the dialogue level providing the multiplexed transmission between the operator side and the technical system side. the decision support level including both an operator model and an application model, the supervision and control level providing computer control of the technical process, and finally the process and process interface level (Figure 4.5). This architecture highlights the role of the decision support level which is embodied in both the human and the task interactive computers in Sheridan's model.

Extending this model in Figure 4.6 Johannsen shows several types of coordination between the human and the machine with different degrees of machine subordination [25, 14]. For example, he makes the distinction between the application-oriented decision support for fault diagnosis and the operator-oriented decision support for procedural support.

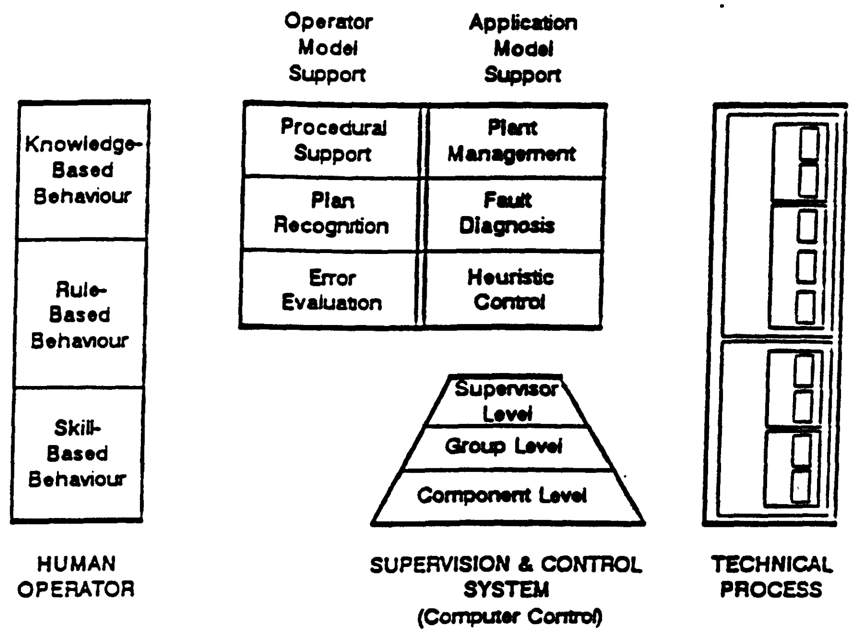

Figure 4.6: Relationship between human operator, decision support system, supervision and control system, and technical process (from [25])

Using the Johannsen model for the ATC system the ASLOTS automation system would represent mainly the decision support level. Some lower level functions, such as sending clearances to the pilots, may be included in the supervision and control system level. As a human-centered concept, however, ASLOTS uses models of the plant and the human controller to generate cues for the controller. The Rule-Base component of ASLOTS, for example, provides operator-model support, since it models the decision making behavior of the human controller. The path generation algorithm on the other hand clearly provides application model support since it uses models of the aircraft and the environment.

In these models for the human-machine system, the task allocation problem is presented in a descriptive manner. Levis [15] describes a method to handle the task allocation problem quantitatively using Petri nets to model the human-machine system as a discrete event system. The task needs to be decomposed to a fine level where each subtask is executed by a single agent (a

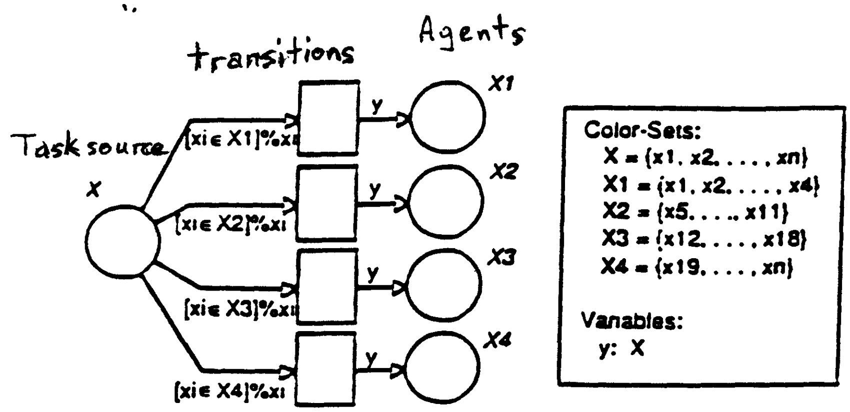

Figure 4.7: Colored Petri net model of task allocation (from [15])

human or a machine). Colored Petri nets are used to combine the tasks with the executing agents into a discrete event network. The task decomposition is then converted to the Petri net diagram that describes the operational execution of the task by the different agents. Figure 4.7 shows a task source x which delivers tasks to four agents x1..x4. Each agent can execute the tasks within a certain color set. If a task exists in the source node and the condition along one of the transition arcs is satisfied (that is the task color belongs to the color set that can be executed by the corresponding agent) then the transition is enabled (fired) and the task is allocated to the agent.

Two concepts are defined to characterize the task allocation network: The complexity and the redundancy. The system has higher degree of complexity when more transitions lead to the same agent (that is the agent needs to process more information). The system has higher degree of redundancy when more agents receive transitions from a source (that is more agents can process the same information). Figure 4.8 shows a Petri net representation of an organization where six tasks can be allocated to two agents, both able to handle all six tasks. The redundancy in this situation is 2 and the complexity is 6. Different task allocation structures can be considered corresponding to different degrees of redundancy and complexity. Figure 4.8 also shows an interaction between the two agents during the processing of the tasks.

Levis, however, criticizes this approach or any other approaches which are based purely on algorithmic and mathematical methods [15]. Such approaches are inherently inferior because they ignore the cognitive, ergonomic and psychological aspects of the problem. For example Muir, Moray and Lee

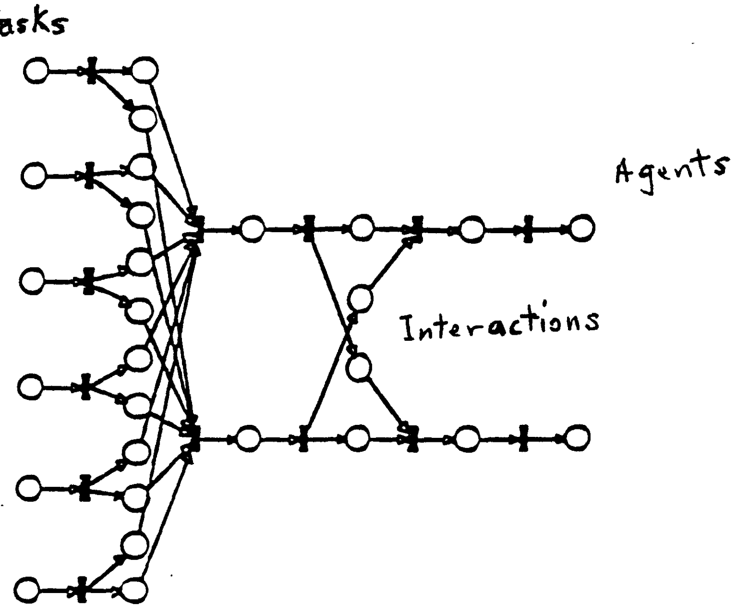

Figure 4.8: Redundancy and complexity in task allocation (from [15])

have shown experimentally that when the operator can choose between performing a task manually or with automatic control, his choice is a function of the difference between his trust in the machine and his self confidence in manual control [15, 26]. The human and the machine should not therefore be separated in a simplistic way and in designing human-machine systems the cooperation and the interaction between the hybrid should be emphasized.

This hybrid relationship however is not completely understood, and the scale developed by Sheridan remains the clearest statement of the distribution of activities of any subtask between the human and the machine at different levels of automation [15, 14]. This scale is attractive, since it describes the cognitive nature of the interaction between the human and the machine and suggests that the level of automation should be selected based on the capabilities of both agents and the coupling between them. It is also attractive because it is descriptive in linguistic form and therefore easily interpreted by the humans involved in the design process. A problem remains, however, since it is not clear how to choose a level of automation from the scale.

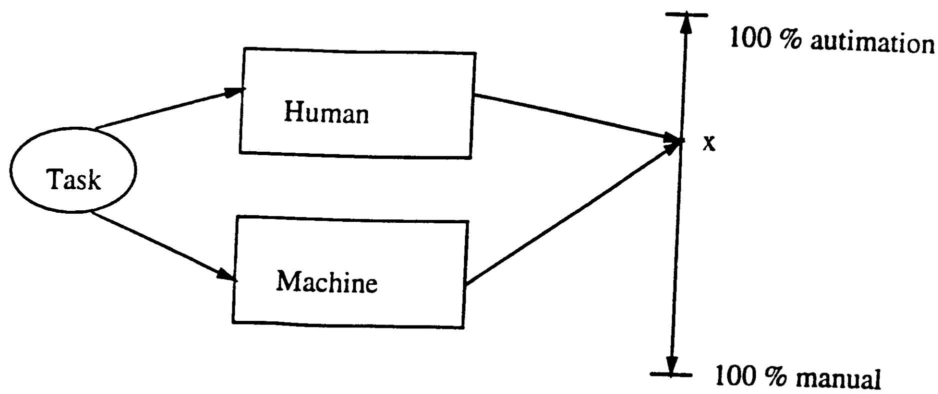

Figure 4.9: Task allocation using the automation level scale

4.3 Task allocation as an optimization problem

In the previous section, a case was made to use the Sheridan scale for choosing an automation level of a certain task. In this section the problem is formulated hypothetically as an optimization problem to choose the best automation level in the context of an objective.

The problem is defined in task space. After task decomposition, n subtasks are to be automated on a scale from 0 to 100. The 0-level of automation corresponds to the completely manual execution of the task, while the 100-level of automation corresponds to the highest automation on the scale where the task is executed completely automatically and the human is ignored. The scale could equivalently be from 0 to 1 or 0 to 10 since the measure of the automation level is relative. Each subtask then forms one dimension resulting in an n-dimensional task space. On each dimension a variable x_i $(0 \leq x_i \leq 100)$ measures the automation level of subtask i and locates it on the automation scale. Figure 4.9 shows the concept while Figure 4.10 shows an example with 2 tasks.

One example for calibrating the measure x could be using information theory. If the tasks are simple so that the information content of each task could be determined, x would be the percentage of the information content of the task that is processed by the machine. A value of 30 for x_i would means that 30 percent of the information content of the task i is processed by the

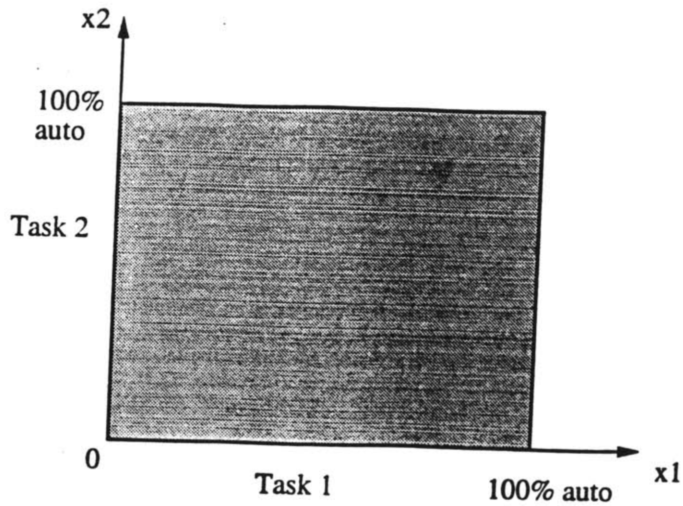

Figure 4.10: The domain for x1 and x2 in a 2-task space

machine and 70 percent by the human. This is a simplistic assumption, however, since one cannot assume that introducing more automation into a task simply subtracts a given portion of the information processed by the human and adds it to the machine. The nature of the task and the interaction between the human and the machine might change considerably as more automation is added. The human, for example, might have to deal with more options and automation modes that he did not have to worry about before the automation. It is assumed, however, that such a measure x of automation can be determined at least subjectively through comparison with the descriptions in the automation scale.

For simplicity, it is also assumed that x is a continuous variable. This is not realistic, since the introduction of automation in a task is more likely to take a discrete manner as suggested in the wording of the automation scale.

Choosing the best values x_i for each task is done in the context of an objective function. The objective is dependent on the situation. For example, the interest might be in increasing the performance of the system, reducing the cost, reducing the time to perform the task, or reducing the workload imposed on the human. The objective could also be one or a combination of these factors. Hypothetically, it is assumed that such an objective exists, and that it can be described as a function f of the automation levels x_i of the tasks involved: $f(x_1, .., x_n)$.

The optimization is also performed in the context of a number of constraints. There are two types of constraints. The first type is of the form $x_i \leq b_i$ or $x_i \geq b_i$. These constraints impose a lower or an upper limit on the level of automation possible or available for the task i. Such constraints may result for example from technological limitations or from laws. The second type is of the form $g_j(x_1, .., x_n) \leq d_j$. These constraints reflect the interaction between the tasks. For example, in order to keep the workload at a certain level increasing the automation level of one task would require reducing the automation level of other tasks, assuming increased automation reduces workload.

The complete optimization problem of the task allocation takes the following form:

$$\begin{gathered} max f(x_1, .., x_n) \\ st \\ 0 \leq x_i \leq 100 \\ x_i \leq b_i \\ g_j(x_1, .., x_n) \leq d_j \\ i = 1...n, j = 1...m \end{gathered}$$

4.4 A linear, 2 task example

If the functions f and g are linear the resulting optimization problem is a linear program. The following is an illustrative numerical example for a 2 task situation. The interpretation of the objective function and constraints can be related to the explanation above.

$$\begin{gathered} max f = x1 + x2 \\ st \\ x_1 \leq 80 \\ x_2 \geq 10 \\ x_2 \leq 90 \\ x_1 + 2x_2 \leq 200 \end{gathered}$$

As shown in Figure 4.11 the optimal solution corresponds to the highest possible value of f within the feasible domain. This optimal value occurs at point A with $x_1 = 80$ and $x_2 = 60$. For an optimal outcome, therefore, 80 percent of task 1 and only 60 percent of task 2 should be automated.

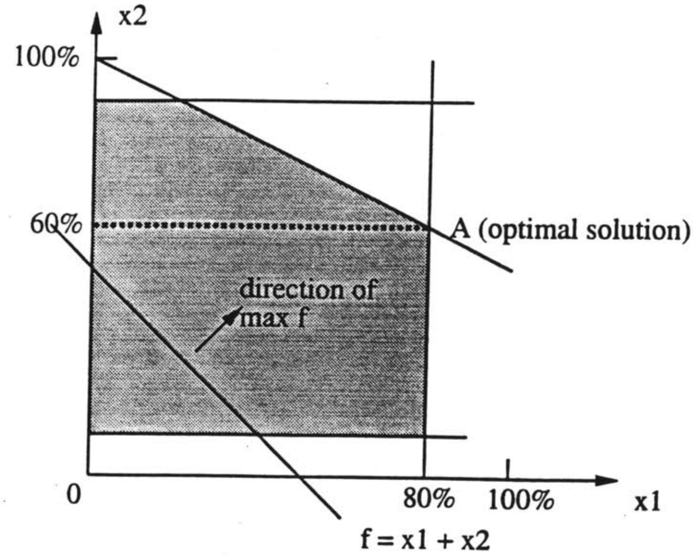

Figure 4.11: A linear program example

For a large number of tasks the Simplex algorithm can be used for an efficient solution to the linear program. If the functions are nonlinear, nonlinear mathematical programming techniques can be employed to determine an optimal solution.

4.5 Fuzzy logic approach

The optimization problem formulated in the previous section is simplistic, namely in its assumption that the measure x of the automation level, the objective function and the constraints imposed on the tasks exist and can be identified. These measures and relationships are subjective as evident from the wording of the automation level scale and from some of the objectives such as performance, safety and workload. Although some objective measures exist, the most reliable measures are subjective and best described in words.

The variable x which measures the automation level of a task represents a cooperation between the human and the machine. Both agents coordinate their actions as described by the corresponding level of the scale. This is an important departure from the usual approach of assigning a task solely to a human or to a machine as described in Section 2 above. It has been emphasized that such a cooperation should be highlighted [14, 15, 17] and this introduces the difficulty of dealing with the interaction between the human and the machine. This interaction is vague and fuzzy in nature, and it is best described in the automation scale using words.

Although the Sheridan automation level scale creates 10 levels, the boundaries between these levels are hardly clear. In fact one can think of a spectrum of possibilities of human-machine interactions which merge the different levels. These boundaries therefore have a fuzzy nature as well and one can also make similar arguments about the fuzzy nature of the limitations and interactions which define the constraints of the problem.

Rather than hiding the important subjective and fuzzy nature of the problem, and making the simplistic assumptions in the optimization problem above, one should emphasize and model such characteristics. Fuzzy logic is a tool that is useful in converting subjective linguistic descriptions into numbers that can be then used in the analytical problem. Also fuzzy mathematical programming is a well established field that deals with optimization problem in a fuzzy environment.

The variable x is defined as a fuzzy variable and its domain, the automatic level scale, is divided into fuzzy sets. Rather than being a crisp number, x is identified by its membership in the fuzzy sets of the scale. The membership in each set is a number between 0 and 1 with 0 as the minimum and 1 as the maximum membership in the set. Figure 4.12 shows an example where the scale is divided into 3 fuzzy automation levels: low, medium and high. The scale could be divided into 10 sets corresponding to the original levels or more or less levels depending on the task involved. The example displays that the automation level of the task is 0 low, 0.3 medium and 0.7 high. If the maximum membership is considered the dominant one, then it can be said that the task is highly automated which has more intuitive interpretation than saying that the task has an automation level of say 80 percent (which is a meaningless number since there may not be a specific design or automation mode that corresponds to exactly 80 percent automation).

In a similar fashion one can identify fuzzy sets for the objective function representing its fuzzy and subjective nature. For example, if safety is the objective one could identify three levels of safety: low, medium and high. Also the same can be done for the constraints where the degree to which a constraint is satisfied by a solution is given by a membership in the constraints fuzzy set. It should be mentioned that not all objectives and constraints need to be fuzzy, since a crisp definition is just a special case of the fuzzy definition where the membership is always 1.

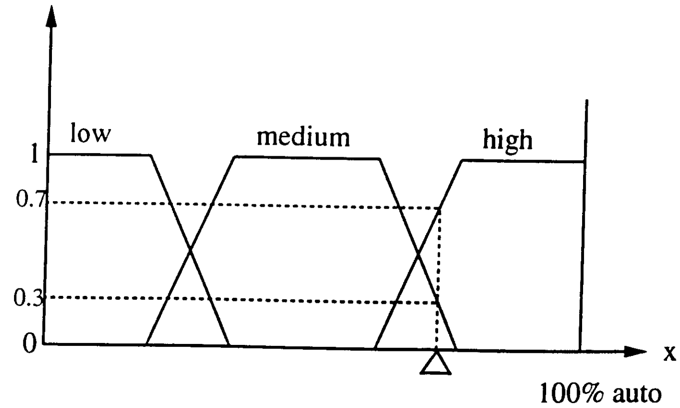

Figure 4.12: A 3-level fuzzy definition of the automation level x

In the fuzzy optimization problem then, the objective and the constraints are defined as fuzzy sets in the domain of the fuzzy variable x. Rather than writing the equations (see [27, 28]) the solution to the fuzzy optimization problem is described qualitatively and graphically for the case of one task. For simplicity it is assumed that there is one objective and one constraint, both defined as fuzzy sets on the fuzzy variable x as shown in Figure 4.13. The membership in the objective set is 0 at very low automation levels, then it increases gradually to 1, and finally it drops to 0 at very high automation levels. Similarly, the membership in the constraint set increases from 0 to 1, then drops back to 0 but over a different range of automation. The two sets intersect as shown. The solution to the problem should satisfy both the constraint and the objective. The solution set is therefore determined by the intersection of the two fuzzy sets corresponding to the objective and the constraint. In the fuzzy logic an intersection between two sets is a fuzzy set the membership in which is the minimum of the memberships in the two original sets. The dashed line in Figure 4.13 shows the minimum of the memberships in the constraint and the objective sets over the domain of the variable x. This dashed line then defines the membership in the solution set, which is also a fuzzy set.

The solution to the problem is therefore a fuzzy set. One may be satisfied with such an outcome with a range of values for the automation level with different membership in the solution set. If a crisp solution is desired, one may choose for example the automation level with the highest membership in the solution set (point A). Such a crisp value, however, would probably

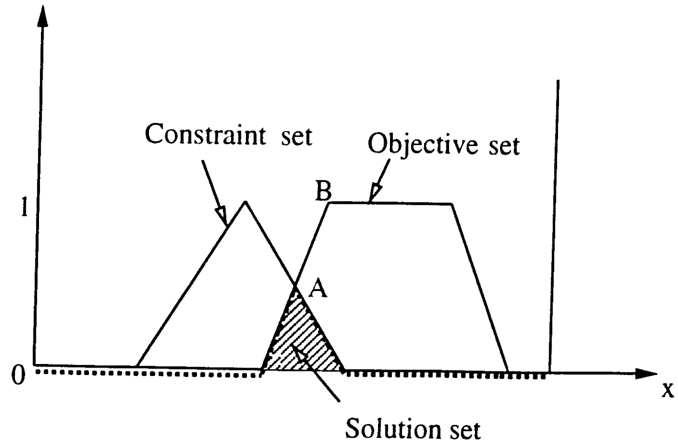

Figure 4.13: The fuzzy optimization solution

be meaningless as pointed out earlier. It is interesting to note that such a choice would have a higher membership in the solution set but a lower membership in the objective set. Point B on the other hand would have a higher membership in the objective set at the expense of a lower one in the solution set. This kind of trade off is introduced by the nature of the fuzzy optimization problem and is not available in the crisp problem. Which choice should be made is again a subjective matter and depends on whether one is willing to give up some assurance of optimality for more assurance of the objective.

4.6 Task allocation: summary and suggestions

In this chapter the problem of task allocation between the human an the machine is investigated. First, a brief history of the qualitative and quantitative approaches to the problem is reviewed and the case is made for the need to highlight the cognitive and ergonomic aspect of the problem. Such aspects, however, are only understood qualitatively and dealt with in a subjective manner, best presented by the automation level scale developed by Sheridan.

An attempt is then made to choose the best automation level on such a scale by formulating the task allocation problem as a simple optimization problem in task space. This approach is criticized as simplistic, since it makes the assumption that the nature of the cooperation and interaction between the human and the machine can be objectively identified and that a measure of the automation level can be determined. The case is made for the necessity to model the subjective and fuzzy nature of such measures and interactions and a fuzzy optimization approach is suggested to deal with this nature.

One of the limitations of the approach is its assumption that the task is divided between a single human and a single machine. Network formulations, such as the Petri net presented above has an advantage in this respect. The network formulation allows the modeling of large scale systems with many tasks and many executing agents (humans and machines), as well as the interaction between the agents.

The cooperation between the human and the machine in the execution of a task should be introduced into such models. Perhaps by using continuous rather than discrete versions of the Petri net [29], a task could be assigned to a human and a machine in proportion. Since it is also important to highlight the fuzzy nature of such cooperation, it is suggested that it might be useful to combine fuzzy logic with Petri nets. This would allow the use of subjective and linguistic automation measures (like the automation scale) and the use of the analytical tools available for Petri nets.

Chapter 5

Plan for Further Research and Development

In the previous chapters the ASLOTS automation problem was discussed in two contexts: Intelligent Control and Human-machine Systems. Intelligent control provides tools by which artificial intelligence is introduced into the automation system and makes it able to deal with the complexities of a human-machine system such as the ATC system. The human-machine system approach reduces the automation problem into a problem of allocation of tasks, responsibilities and intelligence between the human and the machine. The two approaches are therefore complementary where task allocation decides on a suitable level of automation and intelligent control provides the tools to automate the corresponding tasks.

In the first stage of the research the automation problem was pursued with a qualitative and generic approach. This helped in putting ASLOTS within the broader and more general task of automating the ATC activities in the terminal area. As a result, the structure proposed for the ASLOTS automation system (Figure 3.11) is chosen on the basis of flexibility and extensibility. Such a structure emphasizes a multi-tasking approach where each task can be kept manual or automated to any degree independently from the other tasks. A Rule-Base which represents a higher decision making level coordinates the interaction between the different tasks and with the human controller. This approach could be extended further into the multiple tasks where each task is divided into subtasks. The task allocation problem and the tools of intelligent control could therefore be applied at the higher system level and at the lower subtask level deciding to what degree and how to implement the automation of each subtask.

5.1 Intelligent control implementation

The next stage of the research will focus on the elements of the general structure proposed for the ASLOTS automation system. Each element that is found relevant should be investigated in detail with emphasis on practical implementation. In Chapter 3 it was proposed to extend the research, especially in using the following Intelligent Control tools: Adaptive Control, Rule-Base Control, Fuzzy Linguistic Control and Learning Control. In the following list a more concrete plan is proposed to pursue this research progressively:

1. Implement the automation of the path generation task. This is the Pathgen algorithm described in Section 2.7 which computes the path that leads the aircraft to meet its slot marker on the runway centerline. This task is currently under development.

2. Implement the interface to move the slot markers within their feasible range manually. Also implement the automatic schedule updating for all aircraft after the movement of a slot marker (automatic rearward shifting), and after an aircraft intercepts the centerline (centerline adaptation). These procedures ensure the minimum separation between the aircraft on the centerline. This task is also currently under development.

3. Implement the automation of conformance monitoring. This includes measuring the conformance errors between the aircraft's actual and desired states, designing the thresholds for these errors, and generating the appropriate clearances to establish the conformance. This task is currently partially under development as part of the path generation task as indicated in Section 2.7. Namely, the path generated by Pathgen is used by a predictor to estimate the desired state of the aircraft. This state is compared with the actual one to generate the error. The response to the error currently consists only of generating a new path. Conformance clearances that lead the aircraft to conform to an old path

is not implemented. Separate attention, therefore, should be paid to this task especially in designing the error threshold which is candidate for fuzzy modeling.

4. Investigate making the reference model for the ASLOTS path generation adaptive to changes in the dynamics of the aircraft and the environment and test the improved performance (for example, in terms of the error between the aircraft and its slot when they meet). As mentioned in Section 3.2 estimation techniques need to be applied in order to adapt the parameters of the reference model, such as the aircraft speed, the wind, and the pilot delay. For example the use of tracking helps estimate the ground speed of the aircraft.

5. Design the Rule-Base when the path generation and conformance monitoring are automated. This involves the conditions and outcomes of a decision making process (that is if..then statements). At this stage the Rule-Base cannot move the slot marker or the aircraft automatically. However, it can suggest to the human controller to take such an action if that is the outcome of the conditions.

6. Some conditions and outcomes of the rules might be fuzzy in nature (for example, the conformance errors mentioned above). Develop a fuzzy model for these variables (i.e. the fuzzy sets) and the fuzzyfication and defuzzyfication relationships.

7. Test the adaptive, fuzzy-logic, rule-base system. The elements developed to this point should form a functioning system for initial testing.

8. Add to the automation by allowing the Rule-Base to interact automatically with the slots markers. This is the Schedule-Update algorithm mentioned in Section 2.8. This involves adding an algorithm to decide on the optimal movement of the slot and the corresponding schedule, as well as modifying the Rule-Base to accommodate the new automated task. It also involves investigating the human factors issues in human interaction with automatic scheduling systems (see [30]).

9. The automation level can be progressively increased by programming more tasks that deal with more situations, such as missed approaches,

emergency arrivals, runway closures, and changes in the runway assignment or the runway configuration in the multiple runway case. The Rule-Base always have to be augmented accordingly.

10. Learning Control tools should be investigated because they add to the ability of the system in facing rare events such as emergencies. Such situations can be better anticipated if more about the behavior of the system and the environment is learned and stored. For example, recurrent behavior of pilots or aircraft can be classified by type or airline, and such behavior could be anticipated at later times. The same is true for stationary behavior in the wind and the environment. Learning techniques are not as essential, however, to the functionality of the automation system, and therefore their investigation can be postponed to a later stage of the development.

5.2 Task allocation design

Figure 4.4 shows that even when only the path generation task is automated there is a full scale of automation levels possible. It is clear then that one major variable in the system is the level of automation and the corresponding task allocation between the human controller and the computer. The task allocation problem, therefore, is very important and it is proposed that it should be investigated in parallel with the implementation described above. The result of such an investigation is general and could be applied to any automation of a human-machine system. It should result in an algorithm for the dynamic allocation of tasks and a scenario for the progressive introduction of automation in the terminal area operations.

The system should be tested under several modes of automation where the human controller is able to select the mode of automation (see [9]). If the task allocation problem resulted in an algorithm for dynamic task allocation then this algorithm can be included where it suggests the best level or mode of automation under the circumstances. As suggested in Section 4.6 the Petri Net approach to the task allocation problem shows some promise in terms of its ability to model the human machine interaction in a discrete event manner. Combined with the fuzzy modeling of the scale of automation, analytical tools may be available to develop such a dynamic task allocation algorithm. Further research is needed to investigate these ideas.

5.3 Testing and experimentation

Finally in order to test the system with the different modes of automation and the different intelligent control tools mentioned, performance measures need to be established as well as an experimental scenario. Several performance measures are listed below.

- Minimum separation between the aircraft on the centerline as a measure of safety.
- Number of runway operations in a unit of time as a measure of throughput.
- Idle time of runway as a measure of efficiency and throughput.
- Number of occurrences of a conflict or a hazard as a measure of safety and workload.
- Number of human controller and pilot errors as a measure of workload.
- Number of clearances issued by the human controller in a unit of time as a measure of workload.
- Amount of the adaptation movement of the slot at the acquisition of the centerline as a measure of controller performance.
- Secondary task measures for workload.
- Subjective measures for ease of use, clarity, and other human factors.

The experiments will be conducted using the full simulation of the ATC system available in the Flight Transportation Lab. The simulation was described briefly in Section 2.5. It offers graphical interface for several human controllers and pseudo-pilots as shown in Figure 5.1, and is very flexible such that it should allow different modes of automation to be simulated. Figure 5.2 shows a screen dump of the ASLOTS system graphical interface at its current level of development.

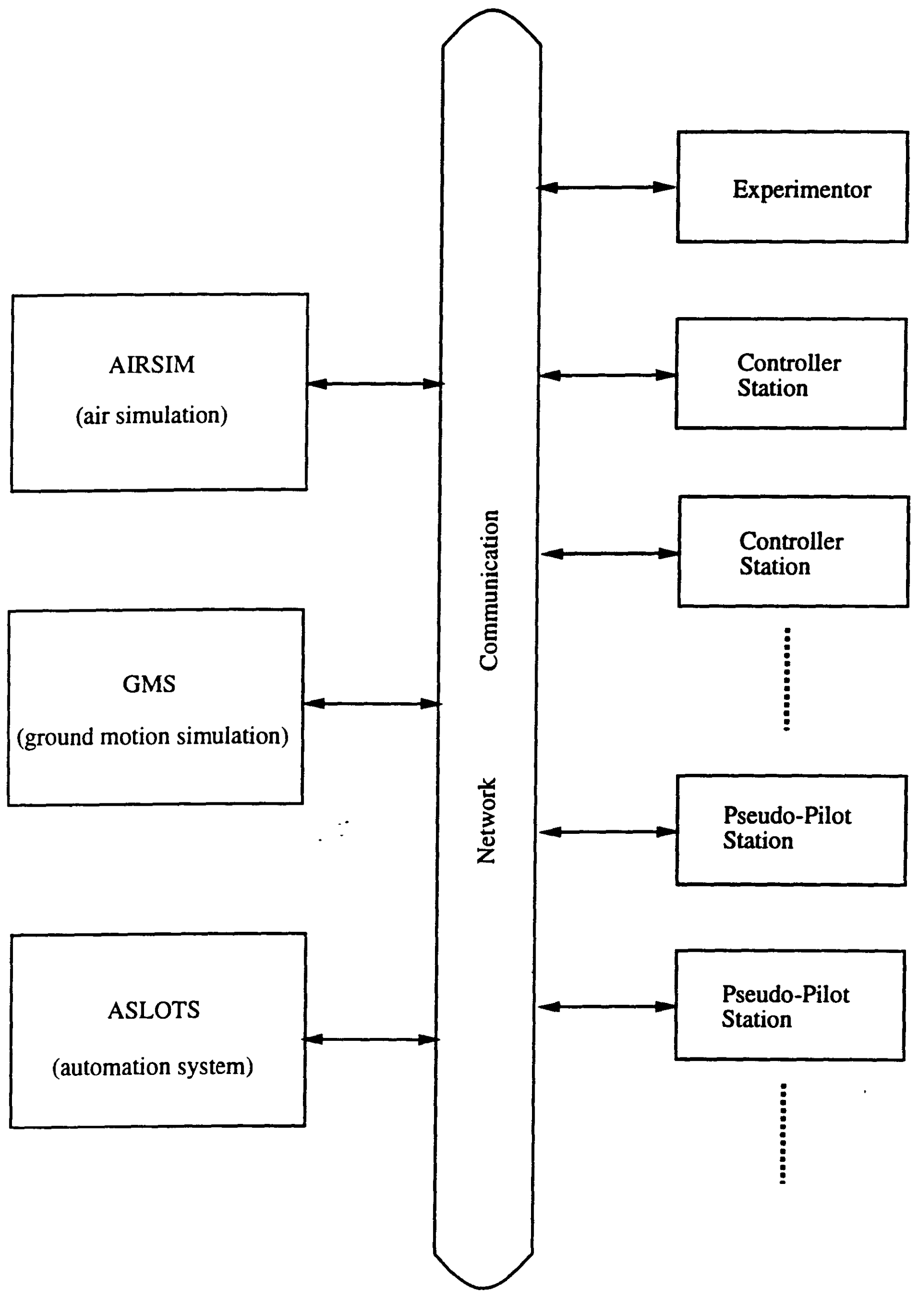

Figure 5.1: The ATC simulation environment

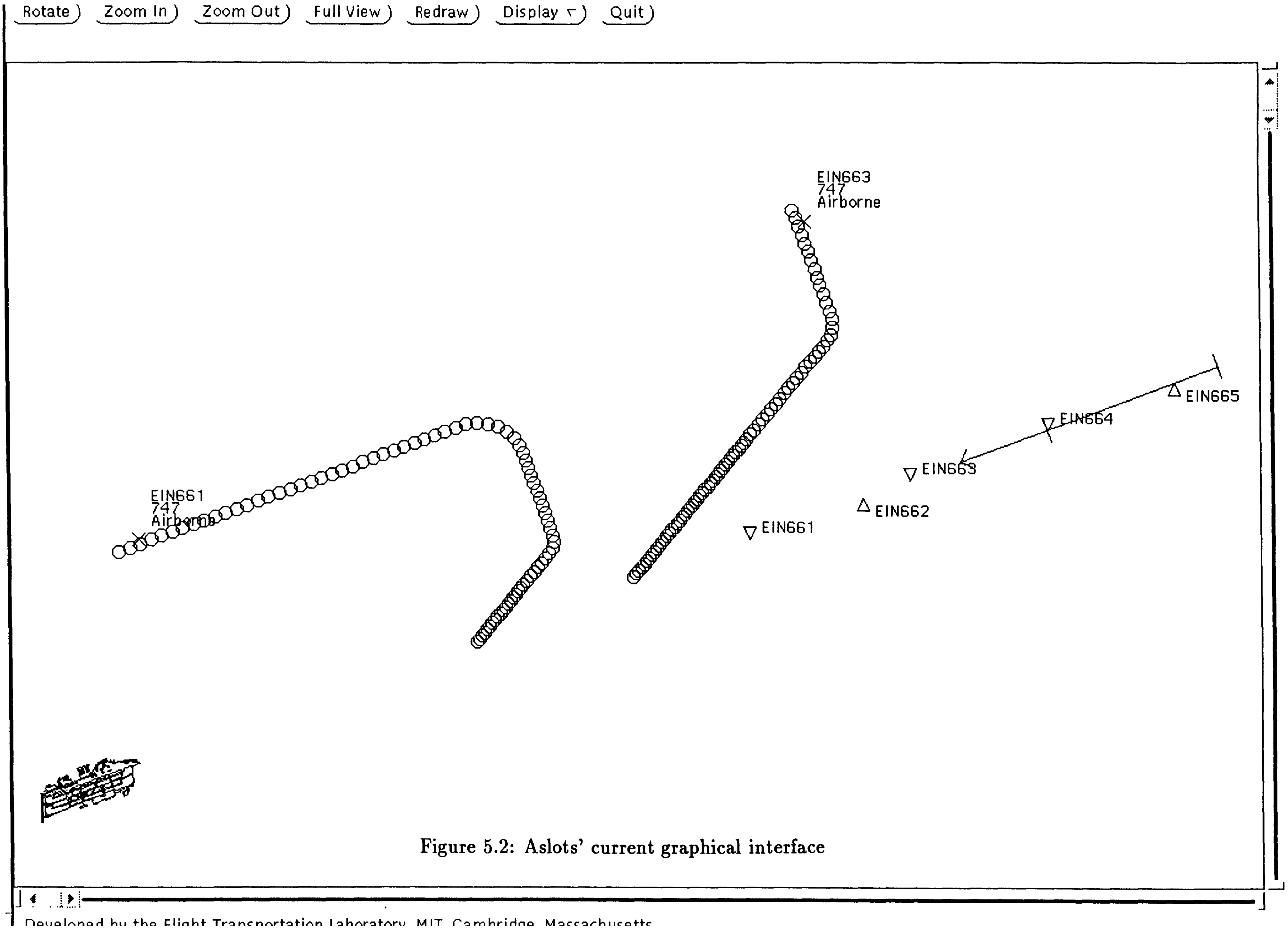

Figure 5.2: Aslots' current graphical interface

The experimental scenario will involve a multiple runway system and only landings at first. Takeoffs will be added to the scenario at a later stage. The experimental design is an important issue involving human subjects, real scenarios if available or real-looking scenarios, statistical analysis, and both objective and subjective measures of the performance of the system. These experiments should prove, it is hoped, that the ASLOTS automation system is reliable and beneficial to be applicable to the ATC terminal area operations.

References

[1] FAA, Washington, D.C. *Airman's Information Manual - Official Guide to Basic Flight Information and ATC Procedures.*

[2] Air Traffic Operations Service, FAA, Washington, D.C. *Air Traffic Control - FAA Order 7110.65.*

[3] R. W. Simpson. Engineering of air traffic control systems. ATC class notes, 1992.

[4] D. A. Travizas. Parallel parametric combinatorial search - its application to runway scheduling. Technical Report R 87-4, Flight Transportation Laboratory MIT, Cambridge, MA, February 1987.

[5] M. M. Sadoune. Terminal area flight path generation using parallel constraint propagation. Technical Report R 89-1, Flight Transportation Laboratory MIT, Cambridge, MA, May 1989.

[6] R. W. Simpson, A. R. Odoni, and F. Salas-Roche. Potential impacts of advanced technologies on the ATC capacity of high-density terminal areas. Technical Report R 86-10, Flight Transportation Laboratory MIT, Cambridge, MA, June 1986.

[7] E. L. Wiener and R. E. Curry. Flight-deck automation: Promises and problems. *Ergonomics*, 23:995–1011, October 1980.

[8] T. J. Davis, H. Erzberger, and H. Bergeron. Design of a final approach spacing tool for tracon air traffic control. Technical Report 102229, NASA, September 1989.

[9] L. Tobias, U. Volckers, and H. Erzberger. Controller evaluations of the descent advisor automation aid. In *AIAA Guidance and control Conference*, Boston, MA, August 1989.

[10] H. Erzberger and W. Nedell. Design of automated system for management of arrival traffic. Technical Report 102201, NASA, June 1989.

[11] R. W. Simpson. An advanced cueing system for final approach spacing. Working Paper in the Flight Transportation Laboratory, MIT, 1990.

[12] R. W. Simpson. The Aslots concept - an interactive, adaptive dicision support concept for final approach spacing of aircraft. Joint University Program Report, Flight Transportation Laboratory, MIT, 1990.

[13] Z. Chi. An adaptive final approach spacing advisory system: Modeling, analysis and simulation. Technical Report R 91-3, Flight Transportation Laboratory MIT, Cambridge, MA, May 1991.

[14] G. Johannsen, A. H. Levis, and H. G. H. G. Stassen. Theoretical problems in man-machine systems and their experimental validation. *Automatica*, 30:217–231, 1994.

[15] A. H. Levis, N. Moray, and B. Hu. Task decomposition and allocation problems and discrete event systems. *Automatica*, 30:203–216, 1994.

[16] W. C. Hoffman, W. M. Hollister, and R. W. Simpson. Functional error analysis and modeling for ATC system concepts evaluation. Technical Report AST-TR-72-9, Aerospace Systems, Inc., Burlington, MA, May 1972.

[17] T. B. Sheridan. *Telerobotics, Automation and Human Supervisory Control.* MIT Press, Cambridge, MA, 1992.

[18] H. L. Ammerman et al. NAS ATC en-route and terminal controller/user operations and information processing tasks. Technical Report CDRL A001, Computer Technology Associates, Inc., Englewood, Colorado, March 1983.

[19] k. J. Astrom and T.J. McAvoy. Intelligent control: An overview and evaluation. In D. A. White and D. A. Sofge, editors, *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches.* Multiscience Press Inc, 1992.

[20] K. S. Narendra. Adaptive control of dynamical systems using neural networks. In D. A. White and D. A. Sofge, editors, *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches.* Multiscience Press Inc, 1992.

[21] W. L. Baker and J. A. Farrell. An introduction to connectionist learning control systems. In D. A. White and D. A. Sofge, editors, *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches.* Multiscience Press Inc, 1992.

[22] P. J. Werbos. Neurocontrol and supervised learning: An overview and evaluation. In D. A. White and D. A. Sofge, editors, *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches.* Multiscience Press Inc, 1992.

[23] R. Langari and H. R. Berenji. Fuzzy logic in control engineering. In D. A. White and D. A. Sofge, editors, *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches.* Multiscience Press Inc, 1992.

[24] D. A. White and M. I. Jordan. Optimal control: A foundation for intelligent control. In D. A. White and D. A. Sofge, editors, *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches.* Multiscience Press Inc, 1992.

[25] G. Johannsen. Towords a new quality of automation in complex man-machine systems. *Automatica*, 28:355–373, 1992.

[26] J. D. Lee and N. Moray. Trust, control strategies and allocation of functions in human-machine systems. *Ergonomics*, 1992.

[27] H. J. Zimmermann. Fuzzy set theory and mathematical programming. In *Fuzzy Sets Theory and Applications*, pages 99–114. Dordrecht: D. Reidel Pub. Co., Boston, MA, 1986.

[28] J. J. Saade. Maximization of a function over a fuzzy domain. *Fuzzy sets and systems*, 62:55–70, 1994.

[29] David R. and H. Alla. Petri nets for modeling of dynamic systems - a survey. *Automatica*, 30:175–202, 1994.

[30] Y. Liu, R. Fuld, and C. Wickens. Monitoring behavior in manual and automated scheduling systems. *International Journal of Man Machine Studies*, 39:1015–1029, 1993.