Supernumerary Robotic Limbs: Task Planning, Execution, and Prediction-Based Coordination with the Human Wearer

by

Baldin Adolfo Llorens – Bonilla

B.S. Electrical Engineering
University of Puerto Rico, Mayagüez Campus

SUBMITTED TO THE DEPARTMENT OF MECHANICAL ENGINEERING PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

AT THE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2013

Signature of Author: ______________________________
Department of Mechanical Engineering
August 20, 2013

Certified by: ______________________________
H. Harry Asada
Ford Professor of Mechanical Engineering
Thesis Supervisor

Accepted by: ______________________________
David E. Hardt
Chairman, Department Committee on Graduate Students
Department of Mechanical Engineering

Supernumerary Robotic Limbs: Task Planning, Execution, and Prediction-Based Coordination with the Human Wearer

by

Baldin Adolfo Llorens – Bonilla

Submitted to the Department of Mechanical Engineering
on August 20, 2013 in Partial Fulfillment of the
Requirements for the Degree of

Master of Science in Mechanical Engineering

ABSTRACT

Full automation of repetitive and/or specialized tasks has become a preferred means to meet the needs of manufacturing industries. However, some tasks cannot be fully automated due to their complexity or the nature of the work environment. In such cases, semi-automation through human-robot collaboration is a strong alternative that still maintains a high level of efficiency in task execution. This thesis focused on the control and coordination issues of the Supernumerary Robotic Limbs (SRL); a pair of wearable robotic limbs that are a potential solution to these issues. The first purpose of this study was to adequately model the collaborative aspect of a task that is conventionally performed by two coworkers. This was achieved through the Coloured Petri Nets (CPN) tool, which was able to model the collaboration between two coworkers by using the SRL and its operator instead. The second purpose of this work was to evaluate how to implement a sensor suit to establish reliable communication between the SRL and its operator. Using data-driven methods for detection, we were able to monitor the operator's current state. By combining this data with the CPN task model we were able to relay the operator's intentions to the SRL. This enabled the SRL to follow the CPN process model in a timely and coordinated manner together with its operator. The third and final section of this thesis focused on considering the interchangeability of roles between the SRL and its operator. We used a data-driven approach to model a task where the SRL and its operator had to perform a simultaneous dynamic task. This was performed by using teach by demonstration techniques on process data from two workers. A control algorithm was then extracted from the actions of the supporting worker. Both the process model and the sensor suit, together with the detection algorithms, were implemented and validated using the first prototype of the SRL. Results show that the SRL was successful in autonomously coordinating with its operator and completing an intercostal assembly task.

Thesis Supervisor: H. Harry Asada
Title: Ford Professor of Mechanical Engineering

Acknowledgements

I would like to dedicate this thesis to my family, those who are blood-related and those related to me by heart. I wish to thank my blood relatives for being there for me during every stage of my life. You have offered me your unconditional support in both good and bad times and have always given me an extra push when I need it most. You taught me the value of education, good friendships, and that everything must be kept in balance. I am truly, deeply grateful. To my family by heart, I have met you along my path in academia, sports, and music, and you have proven to be true friends. You have been there for me through thick and thin without expecting anything in return, even if you had to go out of your way to lend me a hand. I am deeply grateful and fortunate to have met you along my journey and you know that you have earned a place in my heart. To my wonderful girlfriend Diamilet, who has been a beacon of light during these years at MIT. There are no words to express how grateful I am for all the unconditional help that you've given me at all times. I could not have been more fortunate to meet you.

I would also like to thank all of my coworkers at the d'Arbeloff lab for their help and support throughout these two years. A very special thanks to Anirban Mazumdar for being an outstanding MSRP mentor before I became an MIT student and for your guidance, help, and support after becoming an MIT student; they were truly exceptional.

I would like to acknowledge the Boeing Corporation for the financial support that they have given me over these past two years. Without your research assistantship I would not have been able to complete this degree.

Last but not least, thank you Professor Asada for your mentorship, continuous support, enthusiasm, and financial help through research and teaching assistantships.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation for New Assistive Technologies in the Aircraft Assembly Industry

The aircraft assembly industry has been experiencing an increase in demand over the past couple of years, prompting the partial or full automation of assembly processes. Full automation has become a preference due to the repetitive and/or specialized nature of most assembly tasks. However, there are particular tasks that cannot be fully automated because they are too complex or operate in a constrained environment, or due to a combination of both of these limitations. Such tasks usually force the human workers to labor in un-ergonomic positions, perform highly physically demanding jobs which can lead to fatigue or injury, be exposed to dangerous environments, and team up to execute certain task stages. These limitations lead to an overall decrease in productivity and safety. One example of such tasks is the installment of intercostal beams in the aircraft fuselage. This task requires the worker to operate in a workspace that covers both low and high grounds. This forces the human to kneel, crouch, or work in the overhead workspace depending on the current stage of the task. In addition, the human worker has to place and hold the intercostal while a second worker clamps it in place before it can be secured to the fuselage. This means that at some stages of the task, a second worker is needed in order for the task to proceed. In order to meet the increase in aircraft demand, a new automation alternative is needed, one that can improve the human's working conditions without the need to alter the aircraft assembly environment. Human – robot collaboration allows the worker to deal with these problems through semi-automation while maintaining a high level of efficiency. When considering robot – human collaborative systems there are two main approaches that can be taken. The first is the use of remote robots, which allows the worker to remain in the control loop. These robots can be either directly controlled by the human or work autonomously in a group that consists of robots and humans. The second option is the use of wearable robots, where the robot's main function is to enhance the human worker's physical attributes or skills. Traditional systems of wearable robots consist of either prostheses or exoskeletons. The first type aids the human by replacing a lost extremity with a robotic counterpart, while the second enhances his or her physical attributes, such as strength and stamina. These alternatives aid the human in very specific tasks. However, they would not help an aircraft assembly worker perform his or her predetermined tasks more efficiently. The prostheses are not suited to this environment because they are not able to match a human's natural limbs in performance, speed, and coordination with the human's thoughts and intentions. The exoskeleton would reduce the wearer's skills, speed, and available workspace. In addition, using this type of robot would increase the difficulty of traversing the aircraft assembly environment. We propose an alternative to the existing methods of semi-automation that addresses the abovementioned manual labor issues. This method, called Supernumerary Robotic Limbs (SRL), is explained in detail in the following section.

1.2 Supernumerary Robotic Limbs

The d'Arbeloff lab has developed a new type of wearable robot called Supernumerary Robotic Limbs (SRL). This semi-automation alternative addresses the manual labor problems described in the previous section while exploiting the repeatability and precision that robots offer.

1.2.1 Concept

The Supernumerary Robotic Limbs (SRL) is a wearable robot consisting of limbs that coordinate with the wearer to complete a desired task in a timely and efficient manner. We classify the potential tasks of interest in two main categories: physically demanding, which require the worker to bear high loads for extended periods of time, and non-physically demanding. This thesis focuses on the non-physically demanding tasks using a prototype of the SRL, called SRLm (Figure 1-1).

These robotic limbs will be used to aid the worker in the completion of a specific task by assuming the role of an additional worker, bearing workpiece loads, and/or performing sub tasks depending on the task's nature. The limbs are attached to the human worker at the iliac crest. This enables the SRL to operate within and beyond the user's workspace as well as transfer the load from the user's upper-body to the legs, just like a hiking backpack. The goal of the SRL is to be perceived as an extension of the human worker. Thus, a high coordination level between the SRL and the human is required. This tool will ultimately increase a worker's productivity and available skillset, both for the purpose of completing manufacturing tasks and increasing the human's safety.

Figure 1-1: Side view of the SRLm prototype, a version of the SRL used for non-physically demanding tasks.

1.2.2 Uses and Benefits

As mentioned in the previous sections, one particular field that can greatly benefit from the SRL is the aircraft assembly industry. This industry is characterized by having a constrained environment (plane fuselage) and a series of specialized tasks that have to be executed repetitively in different places across the fuselage. These tasks cannot be fully automated and some of them even require more than one worker to be successfully completed. Examples of such tasks include:

- Holding a workpiece while a coworker secures it to the airplane's fuselage: The SRL can steadily hold the workpiece while the human uses his or her free hands to secure the piece into the fuselage. These roles can be reversed depending on the human's preferred role and the availability of workspace.

- Wiring of a control box: A person can easily manage the wires while the SRL prepares the environment. The task could be the removal of obstacles or the preparation of the required tools to affix the wires in their respective terminals.
- Having to operate in low or overhead workspaces: The SRL can use a bracing strategy in order to further stabilize the human's posture or it can help with holding the target workpiece in the appropriate position and orientation while the human operates on it.
- Vacuuming carbon composite (highly toxic) drilling residue of a fuselage while a coworker performs the drilling: The SRL can take care of the drilling while the human worker vacuums the drilling residue. The roles can easily be reversed depending on the human's preferred role and the availability of workspace.
- Proactive assistance for tasks that require the use of various different tools: The SRL can be used to manage the tools by handling them to the human in a timely manner. In addition, it could simultaneously perform some of the subtasks by incorporating different end-effectors, which are devices connected to the end of the limbs.

Although this thesis focuses on aircraft assembly tasks, the SRL's concept can be useful to other fields. The SRL's ability to coordinate with the human and the use of different end-effectors can help improve the efficiency in performing tasks related to health care (operation, elderly care, etc.), household (cleaning, organizing, equipment installment, etc.), and space exploration.

Using the SRL to help the human perform a task can greatly benefit the worker. Some of these benefits include an increase in productivity, skills, abilities, and available workspace. By using the SRL to bear loads at the hip we can decrease the worker's fatigue and improve posture, safety, and load bearing capabilities. The SRL, with the proper instrumentation, can also allow monitoring the human worker's actions. This can help record data for error detection correction, determine when a human is exerting a force that could potentially harm him or her, and to continually adapt the SRL to the human worker thus increasing their coordination rate. The later would greatly contribute to the worker's productivity and our goal of making the SRL feel as an extension of the human body.

1.3 Thesis Layout

This work focused specifically on the following aspects of the SRL:

1. Task modeling as a control scheme.
2. Monitoring the human worker's actions in order to coordinate task execution.

The first aspect consisted of creating a model of the task of interest to use it as a plan for the control scheme necessary for the successful completion of the task. This model was focused on the human – robot collaboration aspect of the task. The second point consisted of two main parts. The first was determining what type of data to use to coordinate the SRL and human actions and how to obtain such data. The second part consisted of using teach by demonstration techniques to control the dynamic actions taken by the SRL during the task execution.

The rest of this thesis is divided in the following chapters:

- Chapter 2: Explores previous work in the task modeling, human-robot coordination, and pattern recognition fields.
- Chapter 3: Briefly introduces the design and implementation of the SRLm.
- Chapter 4: Describes the decision-making algorithm used to model the human-robot coordination in the assembly process of interest.

- Chapter 5: Describes how communication between the SRL and its operator is achieved in order to carry out the task plan successfully.
- Chapter 6: Concludes this manuscript and highlights potential future work.

Chapter 2

Prior Work Relating to Wearable and Collaborative Robots and the Objectives of the SRL

2.1 Assistive Wearable Robots

When considering the potential benefits that the SRL contribute to a human-robot collaborative system, we must also consider other wearable robots that are used to aid its operator in completing a particular set of tasks. Over the past 50 years, there has been great progress in the development and implementation of wearable robots. These traditional assistive wearable robot systems are mainly classified as either exoskeletons [1] or active prostheses and orthoses [2,3].

2.1.1 Exoskeletons

An exoskeleton aids the human by enhancing his or her physical attributes, such as strength and stamina [4,5]. These tools are extremely useful when helping the human exert large forces for extended periods of time while keeping fatigue at minimum. Although they enhance the user's physical capabilities, they also have their drawbacks. For example, exoskeletons do not help in coworker coordination. This is due to the fact that exoskeletons merely mimic the actions of its operator, thus leaving all the coordination to the human and the need of an additional coworker for those sets of tasks that require more than one pair of hands. In addition to that, exoskeletons are heavy and not very transportable [6]. This automatically discards this approach when considering tasks that need to be executed in constrained environments and that require the human worker to easily traverse through the assembly facility.

2.1.2 Orthoses and Prostheses

Orthoses are typically used for people with leg problems [4]. These tools focus on increasing the strength capabilities of a damaged leg but not beyond the parameters of a normal human. Therefore they are not relevant to the SRL. Prostheses are used to help people who are missing parts of their limbs. These are designed to physically replace the missing appendix by imitating the human's desired action [2, 3]. Although these represent the concept of having robotic limbs help its wearer, their focus is restoring a human's motor capabilities. The SRL's goal is to increase a human's motor capabilities and skills beyond that of a normal human. Because of this, orthoses and prostheses are an excellent example of hardware implementation of robotic limbs. These can provide design guidance for the SRL. However, their control approach has to be changed accordingly to include a fully functional human worker.

2.2 Robot – Human Collaboration

Human – robot collaboration is another field that has been widely explored as an alternative for semi-automating a particular process. However, all studies in this field prior to the development of the SRL have focused on the use of human – robot teams [7, 8, 9], where the robots are completely independent from the human. These robots are either mounted on a firm base or are able to move throughout the environment without having to rely on the human worker. By sharing a well-defined task model with the main worker, this robot is capable of replacing the human coworker when the main worker is performing a task. However, for this to be true we have to assume that the robot is able to accompany the worker throughout the working environment [8, 9]. This automatically discards robots that are mounted on a firm base. In the case of an aircraft assembly facility, robots that use wheels are also limited due to the complex nature of the workplace. For example, using stairs to traverse through the different levels at a speed that is comparable to the human's would not be possible for this type of robot. Since this robot is independent of the human, it would also decrease the available workspace. In addition, the robot would have to be able to position itself in a way that allows it to perform his helping role. This can easily force the human worker to presume an uncomfortable position in order to accommodate the robot. These problems are easily addressed by the SRL, which is worn at all times by the human worker.

2.3 Task Planning

Numerous approaches have been used to model a task procedure, for example, fuzzy logic [10], a hierarchical architecture [11], hidden Markov models [12], and Petri Nets [13]. All of these methods divide the task in a series of states and transitions. Then, depending on the user's technique of choice, conditions and laws are established to dictate the behavior of the model. The main difference between these techniques lies in how easily we can change one part of our model without affecting the rest. Most of these techniques, although very precise and efficient, cannot easily modify sub-sections of their model in order to accommodate additional states or changes in the definition of the dynamic processes that occur throughout the task's execution. For this reason, we approached our task-planning problem by using a type of Petri Nets called Coloured

Petri Nets [14]. By separating static states from dynamic transitions and linking them using environment conditions or transition outcomes we are able to model the tasks of interest and capture the collaboration between the coworkers while maintaining the dynamic control laws of the system independent from the model. This gives us the ability to modify the control laws of each corresponding transition in order to adapt to variations in the behavior of the SRL's operator.

2.4 Pattern Recognition

Pattern recognition is a field that has been immensely explored. Numerous algorithms, such as Bayesian classifiers and networks, Support Vector Machines, Neural Networks, and Decision Trees have been thoroughly documented [15] and applied to an extremely wide spectrum of applications. Two examples of applications include speech recognition [16] and rehabilitation [17,18]. In this study, we used pattern recognition techniques to successfully classify the actions of the SRL's operator. This enabled us to establish efficient coordination between the SRL and its operator. After recording the data from various experiments, we determined which pattern recognition technique was most efficient when detecting the operator's intent. The complexity of our algorithm depended on the simplicity of the gestures and postures used to determine intent.

2.5 Teach by Demonstration and Task Execution

The last part of this study used teach by demonstration techniques to extract control algorithms from experimental data. Using this data-driven approach to modeling, we were able to increase the simplicity in the control algorithms that are used by the SRL to perform a specific task. The use of motion detection sensors [19, 20] to transfer the skills of a human to a robot have been explored and implemented to the execution of repetitive tasks [21]. Virtual environments have also been used in order to transfer the human's skills and validate the resulting taught behavior [22].

Chapter 3

Brief Design Introduction

Although the design of the SRL is not the focus of this particular thesis, it is important to briefly explain the SRLm, which is the SRL prototype used for non-physically demanding tasks. The design and development of the SRL that is used for the physically demanding tasks is the work of Ph.D. student Federico Parietti and the undergraduate student Kameron Chan at the d'Arbeloff lab.

3.1 Design Concept

As explained in the introduction, the SRL consists of a pair of wearable robotic limbs that are attached to the human and assists the wearer in the completion of particular assembly tasks. The SRL is attached to the wearer at the iliac crest (worn as a hiking backpack). This gives the SRL the capability to operate in the same workspace as the wearer and transfer the weight of held loads from the human's upper-body to the legs.

3.2 Functional Requirements

The SRLm was designed to mimic a human arm considering that:

1. to achieve the goal of perceiving the SRL as an extension of the human body, the SRLm's appearance and dynamic behavior must resemble that of human arms.
2. to collaborate seamlessly with the wearer, the joint torques, velocities, and bandwidth of the SRLm must be adequate to perform the desired task in a manner that resembles the human's arms.
3. since the aircraft assembly tasks and workplaces are well designed for human workers, the SRLm must be highly similar to a human arm to be easily integrated into these facilities.

We also considered the fact that this prototype would have to be worn at all times during the working period of an aircraft assembly employee. This means that the prototype has to be lightweight in order for the human to be able to wear it for extended periods of time. The robot has to be able to collaborate with the human without interfering with the human's actions, that is, the location of the SRLm cannot interfere with the workspace of the wearer's limbs. In addition to that, the motions and forces exerted by the robot have to be kept at a safe level in order to protect the wearer in cases of malfunction.

3.3 Prototype Implementation

The SRLm consists of two robot arms with three degrees of freedom that are attached to a belt worn at the base of the hip. Each arm has two rotational joints at the base, mimicking the human's shoulders, and one at the middle of each arm, mimicking the human elbow. These arms are actuated with Dynamixel's MX-106R DC motors that are able to exert a maximum of 10 Nm. Position is measured using a contactless absolute encoder that is built into these motors. The arms are attached to the front of the operator and torque control is performed through current sensing. By keeping track of the applied torque we are able to have an emergency stop upon detecting contact with the human or after trying to overcome a safe force limit on the environment. In order to keep the SRLm as lightweight as possible we used carbon fiber for the arms and ABS (3D printer) for small custom parts. By directing this load directly on the wearer's iliac crest (base of the hip) the weight is distributed to the legs, further decreasing the amount of force the operator has to exert to wear this. The arms come out from the frontal area of the belt, which keeps in minimal interference with the operator's workspace. Figure 3-1 shows the SRLm. Figure 3-2 shows the SRLm equipped while being worn. We observe that the SRLm can easily access its operator's workspace without interfering with his or her arms.

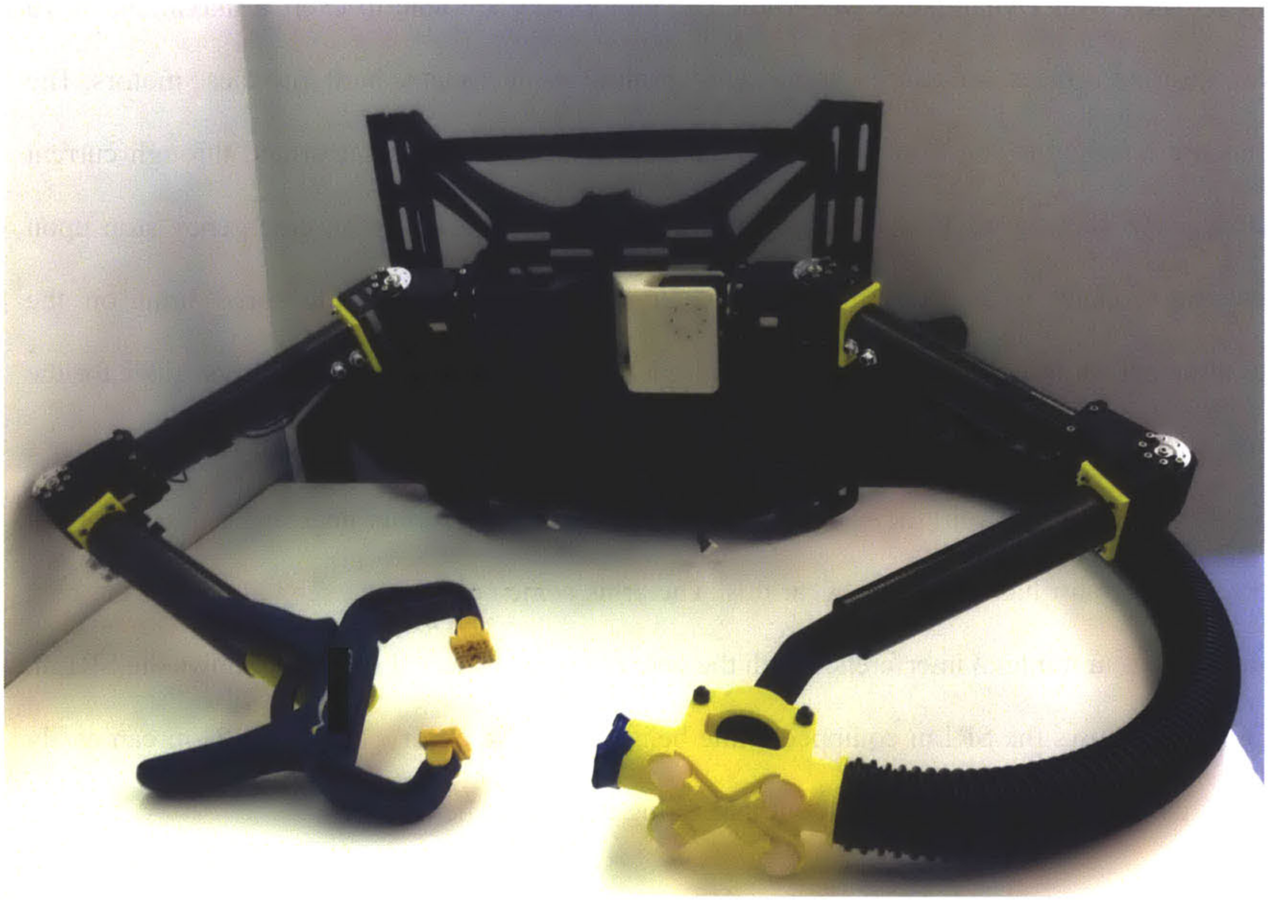

Figure 3-1: SRLm prototype with the vacuum and passive gripper end-effector.

Figure 3-2: SRLm as worn by a human worker.

Chapter 4

Collaboration Modeling for Task Execution

4.1 Human – SRL Coordination

One of the key features that must be present in the SRL is the ability to collaborate efficiently with the human during the execution of a task. In order for this to happen the SRL must be able to identify the task and the time to execute it. If the SRL is not capable of identifying these two aspects on its own, it loses all functionality. This topic has been widely researched in robotics and artificial intelligence. However, just modeling the static states and dynamic transitions that occur during the execution of a task is not enough to capture the collaborative nature that the SRL must exhibit. A bigger structure must be used to model how these states and transitions are connected to each other, define the course of action given certain conditions, and monitor the resources being used during each stage of the task. This allows the SRL to identify what it is supposed to do and when, but also make sure that there is no conflict when deciding which task to perform. This conflict can be represented as an overuse of a system's resources or reaching a point where the robot cannot choose an action to take and arrives at an unintentional dead end in the task model.

Another important property that our model must be able to capture is the concurrent nature of the system we are considering. One of the key advantages of the SRL when compared to other wearable robots is the ability to act independently of the wearer. This allows the SRL to assign different simultaneous tasks to its arms in order to speed up or facilitate the task that the human is performing. In order to model this property, we need a model that can represent static and dynamic actions and also have the ability to include multiple independent actions that need to happen at any given time. This leads to another requirement for our modeling technique: just knowing the order of the tasks that must be executed is not enough; timing is key. In order for the SRL to know when each transition is triggered it must be able to properly assess the operator's current state. By knowing if the operator has successfully completed the task, is in the correct track to completion, or has failed, the robot can reevaluate its course of action to be of optimal use to the human. This means that the modeling technique to be used must be capable of taking into account the current state of each of the resources being used. A flow diagram would be able to model the static stages and dynamic transitions. However, a much stronger technique is required for representing each stage's current state, concurrent event and properly allocating the system's resources

4.2 Task Description

This thesis focuses specifically on one exemplary task: fixing an intercostal on the fuselage. This task was chosen because it exploits all of the advantages of the SRL. During the execution of this task the SRL bears a load in place of the operator and performs a job that would normally be done by an additional worker as the operator performs the main task. We considered the environment of a Boeing assembly facility and focused mainly on the Boeing 787, which has a carbon fuselage. Figure 4-1 shows the environment in which this task is executed.

Figure 4-1: The blue arrow points at the intercostals that need to be fixed and drilled into the airplane's fuselage.

Two trained workers simultaneously carry out this task. The task process goes as follows:

1. The main worker picks up a beam (intercostal) and places it in the respective fuselage place.
2. Once the intercostal is in place, a coworker comes to fix it using a specialized gripper.
3. After the intercostal is secured, the main worker readies the drill to permanently fix it to the fuselage.
4. Once the drill is in place, the coworker then places a vacuum beneath it. This is done to clean the carbon fiber composite that is expelled during the drilling process.

Once the task is completed, the workers recover all the used tools and return to a standby state. Following this standby state, the worker can proceed to other cleanup and assembly tasks as needed. In this study, we focused specifically on the task process up to step 4. Our objective was to equip the lead worker with the SRL, enabling him to complete the task without the need of a coworker. This task, that seems trivial for two human workers, is challenging to model due to the high coordination required between the SRL and the worker. This task's concurrent and non-deterministic nature makes coordination more challenging, because the worker has to deal with parallel tasks, resource allocation, and other task properties such as time delays and queues.

4.3 Petri Nets

In addition to addressing the requirements mentioned in the previous section, we intended to model the task process and resource allocation as similar as possible to a human's. This would allow the SRL to act as a part of the human body and make the worker feel more comfortable when using the wearable robot. We followed the following procedure:

1. The worker identifies the task that needs to be completed.
2. The worker identifies all the tools that are necessary to complete the task.
3. The worker proceeds through the task model. To do this effectively the human must be able to communicate successfully with any aiding workers while keeping track of the environment.

Our approach for this task model was to use Petri Nets (PN). PNs are a graphical method used to model tasks by decomposing them into a set of states called places that are connected through transitions. Tokens are used to show the current state and the conditions needed to proceed through a transition to a new state or place. That being said, it is very similar to the use of flow diagrams or state machines. One fundamental difference between these and PNs is that, by using tokens, PN can activate simultaneous transitions thus enabling us to model tasks that are concurrent in nature. Figure 4-2 shows a rudimentary PN.

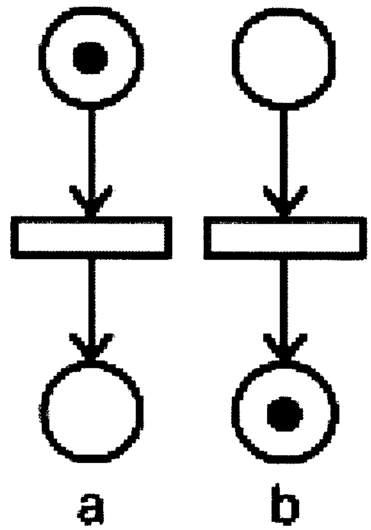

Figure 4-2. In case a. we have the initial conditions where there is one token in the first place. Since the transition only requires one token we move to case b. Here, the tokens used to activate the transition were removed from their original place and moved to the next place.

The circles represent the different places of our process while the box represents a transition. The black circle in the upper place of Figure 4-2a is a token, which notifies us what the current place is by being in the upper circle. In order to move to the bottom place in Figure 4-2a, the transition that is represented by a box has to occur. The arrows that are connected to this transition specify the conditions that must be met in order for this transition to occur. For a transition to occur, there must be a token at each place that has an arrow pointing towards the transition. Given the fact that there is one token at the upper place in figure 4-2a, the transition occurs. When this happens, a token is removed from each of the places that were pointing at the transition and one token is put in each place that is being pointed to by the transition, this is seen in Figure 4-2b. This behavior alone enables us to trigger simultaneous places by having more than one arrow exit the transition, thus covering the concurrent nature of our task of interest. However, this alone is not enough to address all the issues and requirements for our task model.

4.3.2 Coloured Petri Nets

In order for us to fully describe our collaborative task with our model, we turn to a special type of PN called Colour Petri Nets (CPNs) [23]. CPNs take into account two main components: graphics and mathematics. The graphical component of the CPNs is what allows us to separate visually the states from transitions that occur during the execution of a task as done by the PNs. This part also defines the direction followed by the resources/tools when moving through the task process model. For this part, places (or states), transitions, arcs, and tokens (tools/resources) are represented as circles, rectangles, arrows, and dots, respectively. The mathematical part consists of basic primitives with the capabilities of a high level programming language. Combining these two enables us to create a discrete-event model. These properties make CPN ideal for modeling systems where concurrency and communication play a vital role in the task's process. The mathematical properties of CPNs allow us to write statements that describe the behavior of the model. This means that we are able to specify what conditions must be met for a transition to occur, if the token needed for a transition must have a specific property, and how to manage tokens depending on the result of the transition. This allows to keep track of the current system performance and resource allocation based on current transitions through CPN [14].

To properly model the task, we must define a finite number of places P, transitions T, guard functions G, directed arcs A, arc expression functions E, colour sets Σ, coloured set functions C, typed variables V, and an initialization function I. Colour sets and functions, arc expressions, and guard functions will be directly responsible for managing how the resources and tools are used throughout the task process execution. We used the mathematical statements of the CPNs to model our task of interest as follows. The basic CPN that will be used as an example is shown below in Figure 4-3. First of all, we will give each token a colour to identify each particular tool or resource. We assign each token both a number and a string. The number is used in the mathematical expressions that determine which resource to use and the text is used to visualize the location of the tools and resources. Colours are defined as:

colset No = int;

colset Re = string;

*colset NoxRe = product No * Re;*

This allows us to assign each token a number and a string element in that order. For example, we can assign one token with No = 1 and Re = "Human Right Arm", where 1 becomes the token's mathematical identification of the human's right arm. Considering that different tools are used for specific tasks, the CPN must also be able to discern which tools are used for each particular task. Now that we are able to identify the tokens, we proceed to establish the constraint laws for the transitions. This is accomplished through the arc expressions and transition guards. Arc expressions are written right besides an arc and they determine which characteristics of the tokens in the place will be considered for the transition.

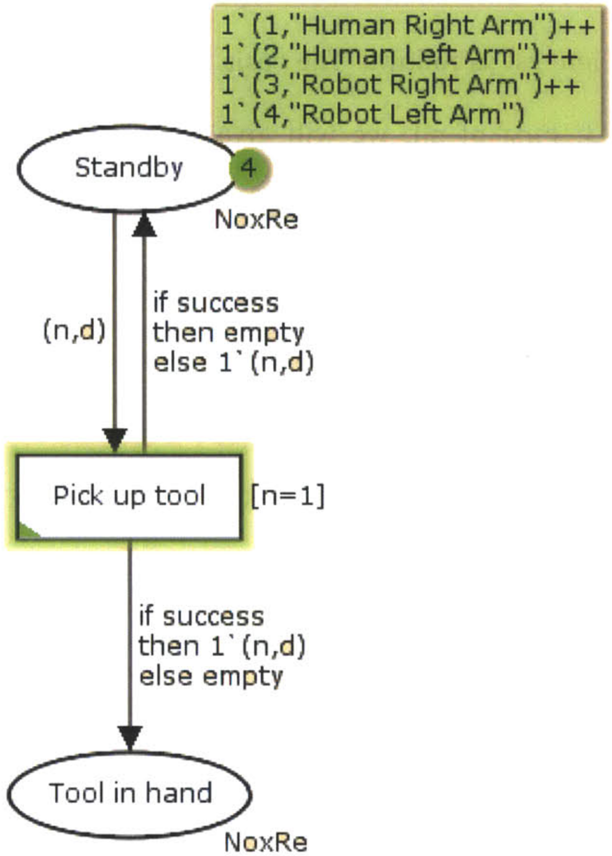

Figure 4-3: A simple example of high-level programming that is achievable through the use of CPNs. The number four inside the green sphere indicates that there are a total of four tokens in that place. These tokens are presented inside the green box to the upper right of the "Standby" place.

These must also be defined as variables. For our example in Figure 4-3 we define these variables as:

colset status = bool;

var n : No;

var d: Re;

var success : status;

Transition guards establish certain criteria that must be true before the transition is able to take place. These are written inside brackets next to the corresponding transition, as shown in Figure 4-3.

In Figure 4-3, the tokens corresponding to all four arms (robot's and human's) are in the "Standby" place. In order to move on to the "Tool in hand" place, the tokens have to go through the "Pick up tool" transition. In order to go through this transition all the conditions in the input arcs must be fulfilled. This occurs when all the variables are bound (all necessary variables specified in the arc expressions can be matched with a respective token from the input place). In this particular case, we are specifying that only the human right hand must be used to pick up the tool. This means that we constrain the variable n to 1, and therefore the only token that is able to fire the "Pick up tool" is {n = 1, d = "Human Right Arm"}. Since we have at least one token that follows the guard's constraint, we are able to fire the transition. Notice than after the transition is made, output arcs determine the token distribution based on the "Success" variable.

The nonhierarchical CPN model has the following places and transitions:

P = {t_and_r, b_in_p, fixed_b, d_in_p, v_in_p, task_end},

T = {p_b, fix_b, p_d, p_v, drill},

where t stands for tools, r for resources, b for beam, p for place, d for drill and v for vacuum. We define the set of colours:

Σ = {No, Re, NoxRe, status, Tools}, where

colset No = int;

colset Re = string;

*colset NoxRe = product No * Re;*

colset status = bool;

colset Tools = list NoxRe;

The Tools colour was chosen to be a list since it would simplify accessing a place's tokens in situations where we would have more than one token. This is due to the fact that in order to describe in detail each step of the task we need to keep track of what resources are at each state and which are needed for each transitions. The initial conditions for the system are the tokens in the "t_and_r" place and are given by:

1\`(1, "HRA")++

1\`(2, "HLA")++

1\`(3, "RRA")++

1\`(4, "RLA")++

1\`(5, "Gripper End effector")++

1\`(6, "Vacuum End effector")++

1\`(7, "Drill")

where HRA, HLA, RRA, and RLA stand for human right and left arm and robot right and left arm, respectively. The resulting structure for the CPN model is shown in Figure 4-4.

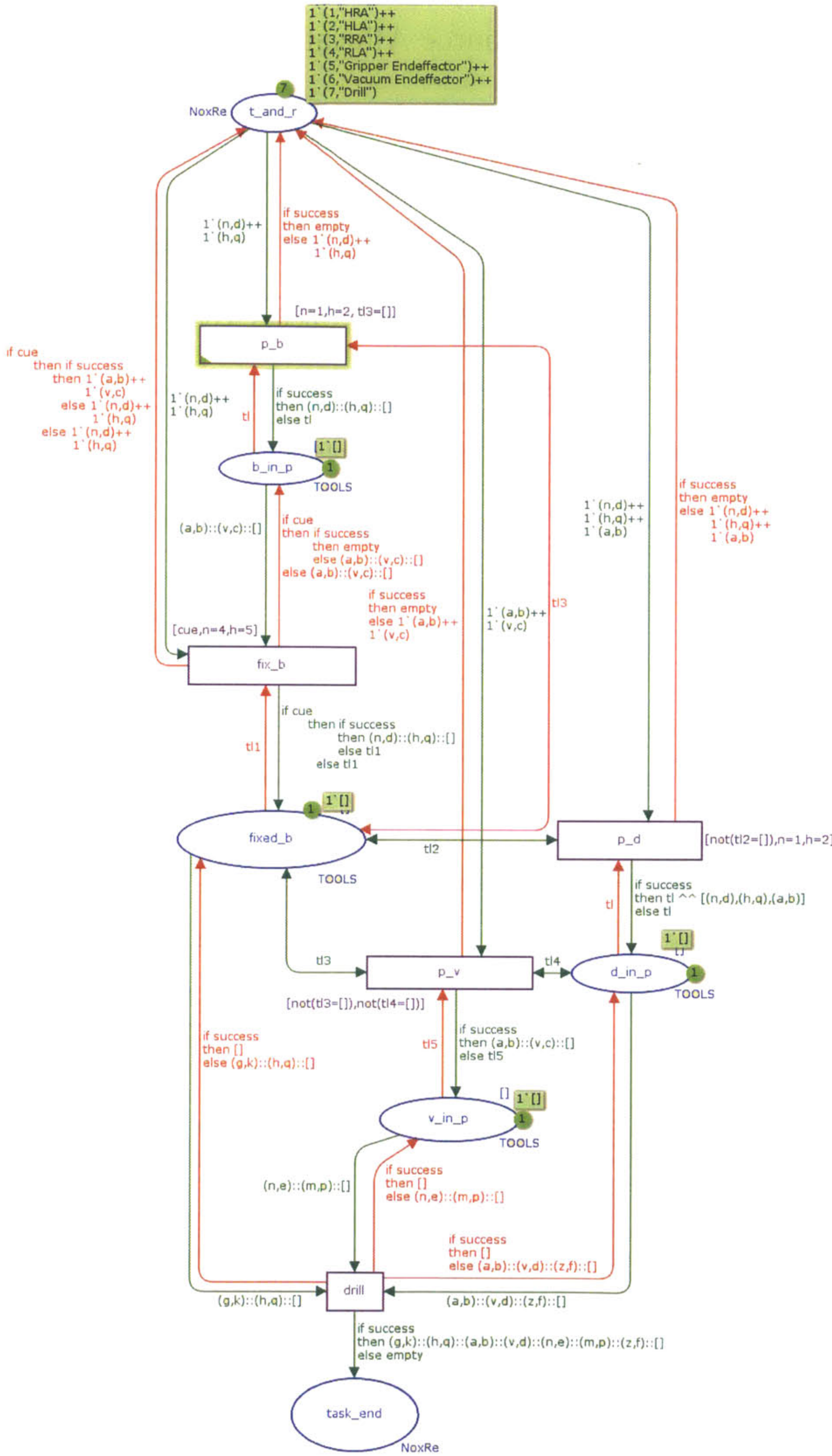

Figure 4-4: CPN model for the intercostal assembly task.

4.3.2 State Space Analysis and Resource Allocation

In order to examine this model we used the state space method. This method consists of exploring all possible transitions and resource allocation that can occur from each place in the CPN. This is done by representing each marking (reachable state by the whole CPN) as a node and then mapping its transition to previous and successive nodes (consecutive reachable states). Each of the arrows that leave a node represents a binding variable that is used to reach that marking. This allows us to check the system for token duplication, misplacement, starvation, and all the possible ending states for our CPN. Using the simulation software CPN tools we are able to map out the state space for our task process model (Figure 4-5).

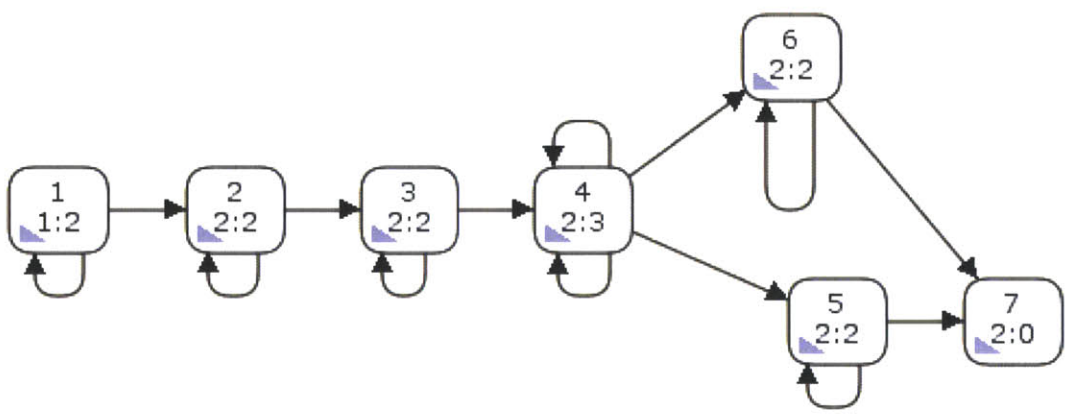

Figure 4-5: This diagram shows that with our task specifications we are able to model the entire task with 7 reachable CPN states and only one final state.

Each node represents a possible state of the entire Coloured Petri Net. The number on the top of each node in Figure 4-5 is the number of the state. The numbers on the bottom of each state represent the previous and successive states from left to right respectively. Node #1, for example, gives the following information: this is the first system overall state of the task model; it can only be predeceased by one state and has two possible outcome states. In this case, the possible previous state is itself and the possible outcomes of this state are to either remain in the same state or proceed to the next state. The later would happen after firing the first transition. This analysis applies to the other nodes as well. Each node's properties and transition's possible outcomes are thoroughly explained in Figures 4-6 – 4-16.

```
1:
Drilling_Task't_and_r 1: 1`(1,"HRA")++
1`(2,"HLA")++
1`(3,"RRA")++
1`(4,"RLA")++
1`(5,"Gripper Endeffector")++
1`(6,"Vacuum Endeffector")++
1`(7,"Drill")
Drilling_Task'b_in_p 1: 1`[]
Drilling_Task'fixed_b 1: 1`[]
Drilling_Task'd_in_p 1: 1`[]
Drilling_Task'task_end 1: empty
Drilling_Task'v_in_p 1: 1`[]
```

Figure 4-6: Node boxes follow the following format: each line begins with the name of task being modeled by the CPN. Following this name, after the apostrophe, are each of the places in the CPN followed by the tokens that are at each respective place. The number at the top left represents the number of the node. Node # 1 represents the initial state of the CPN. All the tokens are located in the t_and_r place. This node represents the initial standby phase before the task execution begins. Note that the other time when we are at this node is when the p_b transition is attempted but not successful.

```
1:1->2 Drilling_Task p_b 1: {tl3=[],n=1,h=2,d="HRA",q="HLA",tl=[],success=true}
2:1->1 Drilling_Task p_b 1: {tl3=[],n=1,h=2,d="HRA",q="HLA",tl=[],success=false}
```

Figure 4-7: In the first box area (red), the first number represents our current place. The second number shows we are transitioning to. The second box (green) shows the name of the task being modeled by the CPN. The third box (blue) indicates the name of the transition we are considering. The fourth and final box (maroon) shows the tokens used and variable values that trigger the transition represented in the first box (red). First transition is "p_b" has two possible outcomes. These are when the "success" variable are true and false. The first one enables the CPN to proceed to node # 2 while the later forces the CPN to remain in node # 1. This behavior models how the human worker will continue to position the intercostal until he is successful.

```
2:
Drilling_Task't_and_r 1: 1`(3,"RRA")++
1`(4,"RLA")++
1`(5,"Gripper Endeffector")++
1`(6,"Vacuum Endeffector")++
1`(7,"Drill")
Drilling_Task'b_in_p 1: 1`[(1,"HRA"),(2,"HLA")]
Drilling_Task'fixed_b 1: 1`[]
Drilling_Task'd_in_p 1: 1`[]
Drilling_Task'task_end 1: empty
Drilling_Task'v_in_p 1: 1`[]
```

Figure 4-8: Node two represents the state following the p_b transition if p_b is successful. Here both human hands are being used to keep the intercostal in place while the rest of the resources remain at standby. Same as the in previous node, the CPN remains at this node if the following transition is attempted but unsuccessful.

```
3:2->3 Drilling_Task'fix_b 1: {n=4,h=5,d="RLA",q="Gripper Endeffector" ...

    ... ,tl1=[],c="HLA",b="HRA",a=1,v=2,cue=true,success=true}

4:2->2 Drilling_Task'fix_b 1: {n=4,h=5,d="RLA",q="Gripper Endeffector" ...

    ... ,tl1=[],c="HLA",b="HRA",a=1,v=2,cue=true,success=false}
```

Figure 4-9: Second possible transition is fixing the intercostal to the fuselage. This is achieved using the SRL with a gripping end-effector. This leads to freeing the human hands. This transition can only be triggered once the cue has been given to the robot and the output of the transition arcs depends on the result of the task (successful or not).

```
3:
Drilling_Task't_and_r 1: 1`(1,"HRA")++
1`(2,"HLA")++
1`(3,"RRA")++
1`(6,"Vacuum Endeffector")++
1`(7,"Drill")
Drilling_Task'b_in_p 1: 1`[]
Drilling_Task'fixed_b 1: 1`[(4,"RLA"),(5,"Gripper Endeffector")]
Drilling_Task'd_in_p 1: 1`[]
Drilling_Task'task_end 1: empty
Drilling_Task'v_in_p 1: 1`[]
```

Figure 4-10: Node three represents the state where the SRL has replaced the human and has fixed the intercostal to the fuselage. When this is done, the operator's hands are relieved from duty and resume their standby state. This allows the human worker to start the next transition: positioning the drill.

```
5:3->4 Drilling_Task'p_d 1: {n=1,h=2,d="HRA",b="Drill",a=7,q="HLA" ...
    ... ,tl=[],tl2=[(4,"RLA"),(5,"Gripper Endeffector")],success=true}

6:3->3 Drilling_Task'p_d 1: {n=1,h=2,d="HRA",b="Drill",a=7,q="HLA" ...
    ... ,tl=[],tl2=[(4,"RLA"),(5,"Gripper Endeffector")],success=false}
```

Figure 4-11: The third transition consists of the SRL operator placing the drill in the correct position. In order to do this, the intercostal must be secured to the airplane fuselage. To ensure this, the transition will not be triggered unless the "RLA" and "Gripper Endeffector" tokens are present in the place that corresponds to the intercostal being secured.

```
4:
Drilling_Task't_and_r 1: 1`(3,"RRA")++
1`(6,"Vacuum Endeffector")
Drilling_Task'b_in_p 1: 1`[]
Drilling_Task'fixed_b 1: 1`[(4,"RLA"),(5,"Gripper Endeffector")]
Drilling_Task'd_in_p 1: 1`[(1,"HRA"),(2,"HLA"),(7,"Drill")]
Drilling_Task'task_end 1: empty
Drilling_Task'v_in_p 1: 1`[]
```

Figure 4-12: Node four represents the state when the operator has positioned the drill in place and is waiting before the SRL starts to position the vacuum. The only free resources are the SRL's right hand and the vacuum end-effector. Previous states leading to this node include node #3 and the other two each represents a failed attempt at reaching node # 5 or 6. Notice that nodes 5 and 6 are in parallel, meaning that only one of them can occur during the execution of the task.

```
8:4->4 Drilling_Task'p_v 1: {tl5=[],tl3=[(4,"RLA"),(5,"Gripper Endeffector")] ···
    ··· ,tl4=[(1,"HRA"),(2,"HLA"),(7,"Drill")],c="RRA", ···
        ··· b="Vacuum Endeffector",a=6,v=3,success=false}
9:4->6 Drilling_Task'p_v 1: {tl5=[],tl3=[(4,"RLA"),(5,"Gripper Endeffector")] ···
    ··· ,tl4=[(1,"HRA"),(2,"HLA"),(7,"Drill")],c="Vacuum Endeffector", ···
        ··· b="RRA",a=3,v=6,success=true}
10:4->4 Drilling_Task'p_v 1: {tl5=[],tl3=[(4,"RLA"),(5,"Gripper Endeffector")] ···
    ··· ,tl4=[(1,"HRA"),(2,"HLA"),(7,"Drill")],c="Vacuum Endeffector", ···
        ··· b="RRA",a=3,v=6,success=false}
7:4->5 Drilling_Task'p_v 1: {tl5=[],tl3=[(4,"RLA"),(5,"Gripper Endeffector")] ···
    ··· ,tl4=[(1,"HRA"),(2,"HLA"),(7,"Drill")],c="RRA", ···
        ··· b="Vacuum Endeffector",a=6,v=3,success=true}
```

Figure 4-13: The fourth transition represents when the SRL positions the vacuum and readies for the drilling task to be carried out. In order for the SRL to undertake this task, the intercostal has to be secured to the fuselage and the drill has to be properly positioned, otherwise it is impossible for this transition to be triggered.

```
5:
Drilling_Task't_and_r 1: empty
Drilling_Task'b_in_p 1: 1`[]
Drilling_Task'fixed_b 1: 1`[(4,"RLA"),(5,"Gripper Endeffector")]
Drilling_Task'd_in_p 1: 1`[(1,"HRA"),(2,"HLA"),(7,"Drill")]
Drilling_Task'task_end 1: empty
Drilling_Task'v_in_p 1: 1`[(6,"Vacuum Endeffector"),(3,"RRA")]
```

```
6:
Drilling_Task't_and_r 1: empty
Drilling_Task'b_in_p 1: 1`[]
Drilling_Task'fixed_b 1: 1`[(4,"RLA"),(5,"Gripper Endeffector")]
Drilling_Task'd_in_p 1: 1`[(1,"HRA"),(2,"HLA"),(7,"Drill")]
Drilling_Task'task_end 1: empty
Drilling_Task'v_in_p 1: 1`[(3,"RRA"),(6,"Vacuum Endeffector")]
```

Figure 4-14 shows node # 5 (top) and # 6 (bottom). Only one of these two nodes can occur during the execution of the CPN. This will be determined by the arc expressions that follow the p_v transition. Notice how the only difference between these two states is the order of the tokens in the "v_in_p" place. This means that the same resources are being used to perform the same task. This results in these two nodes being the same (for any practical means). The fact that they are represented in different nodes means that I could have been more detailed in the arc expressions following the previous transition.

```
11:5->7 Drilling_Task'drill 1: {n=6,m=3,e="Vacuum Endeffector",p="RRA", ···
    ··· k="RLA",h=5,g=4,q="Gripper Endeffector",f="Drill", ···
        ··· d="HLA",b="HRA",a=1,z=7,v=2,success=true}

12:5->5 Drilling_Task'drill 1: {n=6,m=3,e="Vacuum Endeffector",p="RRA", ···
    ··· k="RLA",h=5,g=4,q="Gripper Endeffector",f="Drill", ···
        ··· d="HLA",b="HRA",a=1,z=7,v=2,success=false}

13:6->7 Drilling_Task'drill 1: {n=3,m=6,e="RRA",p="Vacuum Endeffector", ···
    ··· k="RLA",h=5,g=4,q="Gripper Endeffector",f="Drill", ···
        ··· d="HLA",b="HRA",a=1,z=7,v=2,success=true}

14:6->6 Drilling_Task'drill 1: {n=3,m=6,e="RRA",p="Vacuum Endeffector", ···
    ···k="RLA",h=5,g=4,q="Gripper Endeffector",f="Drill", ···
        ··· d="HLA",b="HRA",a=1,z=7,v=2,success=false}
```

Figure 4-15: The final transition in our CPN model consists of drilling. All conditions required for this task have been fulfilled and now the only thing missing is for the actual drilling to occur.

```
7:
Drilling_Task't_and_r 1: empty
Drilling_Task'b_in_p 1: 1`[]
Drilling_Task'fixed_b 1: 1`[]
Drilling_Task'd_in_p 1: 1`[]
Drilling_Task'task_end 1: 1`(1,"HRA")++
1`(2,"HLA")++
1`(3,"RRA")++
1`(4,"RLA")++
1`(5,"Gripper Endeffector")++
1`(6,"Vacuum Endeffector")++
1`(7,"Drill")
Drilling_Task'v_in_p 1: 1`[]
```

Figure 4-16: Node 7 represents the final stage of the CPN model. Here, the task of interest has been successfully completed.

4.3 Discussion

After carefully checking each of the state space nodes we can conclude that the task is executed without any token duplication, misplacement, or starvation and has only one possible ending state. This CPN is implemented in the SRLm in order to determine which course of action must be taken. Using this nonhierarchical CPN to model a particular task also helps us organize the SRL's task database. By using this framework to model other tasks we are able to incorporate our CPN models into a hierarchical CPN (HCPN). Each model can be portrayed as a module in a HCPN that will be executed depending on the operator's intention. This HCPN would be the SRL's decision - making algorithm. In this HCPN we can assign the tools' place as the starting point. Ultimately the SRL can make the decision of which action to perform by taking into account the indicators that it receives from the environment and the wearer.

Now that we covered how to use CPN's to fully model the collaborative aspects of our task of interest we are able to move on to how to use these tokens, conditions, and general model to achieve an efficient coordination between the human and the SRL.

Chapter 5

Human –Robot Coordination

5.1 Leader/Follower Relationship

As described in Chapter 4, our task of interest requires the collaboration between two coworkers. Coordination and shared knowledge about the task's goals and procedure is essential in order for the SRL to serve its purpose. Using CPNs addresses the latter. It is easy for a human worker to identify a task's current state, what needs to be done next, and which tools to use and how to use it. The worker, when working in a team, is also able to assess the situation and decide together with his partner if switching roles would prove more efficient. This leader/follower relationship is also represented in our CPN model. Although this is trivial for two human workers, the same cannot be said for the SRL. Nevertheless, here is where we exploit the advantages of using CPNs as our decision-making algorithm. The CPN already depicts the flow of a task in a way that closely resembles the human train of thought when executing a particular task. Now that we have our task divided into static states and dynamic transitions, we can focus on the elements that determine the flow of the task: the indicator variables. These indicators that are part of the arc expressions and a transition's guard can be assigned to physical system events. For example, in Chapter 4, when we went over the task's model, one of the guards for the "fix intercostal in place" transition was named cue. If the SRL has the means to monitor it's operator's actions, then it is also able to use pattern recognition techniques to determine whether the cue was given or not. Using this approach we can monitor the leading role that is being carried out by the human and let the SRL perform the follower role.

5.2 Transition Coordination

As mentioned previously, coordination plays a vital role in the SRL's performance. When considering our CPN model, we break down our task of interest into five dynamic transitions and six static states. Each of these transitions has been given a respective guard and arc expressions that describe the conditions that must be met in order for these transitions to occur. In other words, the SRL's efficiency when recognizing if these conditions have been meet will determine the level of coordination between the SRL and the operator.

An example of these conditions can be explained using our first transition: placing the intercostal. Looking at the guard expression we see that this transition is only possible if the operator's arms are not performing any other task and only if the intercostal has not been fixed. The later condition, although it is a trivial observation for the human worker, needs to be an explicit condition for the SRL. If not for this condition, the CPN would not be able to discern if the human is using his free arms to place the drill or to place another intercostal. In this case we would need additional indicators in our CPN model to be able to further identify the operator's intention. We include this condition as a means to maintain the CPN with the minimum number of indicators and variables. Since the human worker carries out this transition, the only variable that we need to assign to the physical world is "success". It is important to make clear that the variable success constantly appears throughout our CPN model. However, the definition, or assignment, of this variable will be different for each transition. This variable needs to have a value of 1 for when the operator has successfully placed the intercostal and 0 for when he has been unsuccessful. Having a value of 1 allows the tokens to move into the next place of our CPN model, whereas a value of 0 will keep the tokens in their current place. This behavior represents the fact that the human worker takes time to complete this task and will not stop until it has been achieved. In order to determine if the operator has been successful or not we need an array of sensors that monitor his actions as well as the environment.

5.3 Wearable Sensors

In order to be able to coordinate the SRL with its operator we need to be able to monitor the operator's actions. It is important to take into consideration the nature of our system and its goal: the SRL is a wearable robot that will be perceived as an extension of one's body. Since the SRL will be worn at all times while the human worker performs various tasks through out the airplane fuselage, we cannot rely on sensors that are fixed to the environment. Thus, we use a suit that is worn together with the SRL at all times. We can attach a wide variety of sensors to this suit. Going back to the analysis in the previous section, we can see that most of our transitions consist of moving an object to a particular location and orientation. Because of this, for our first prototype of the sensor suit, we considered the use of inertial measurements units (IMUs) as a means to monitor the operator's actions. We start with 3 IMUs, each located in the operator's wrists and the back of his or her head. Each IMU unit is equipped with an accelerometer, gyroscope, and compass. Using the readings form these sensors we are also able to obtain each IMUs' Euler angles for our training data sets. Our first prototype of this suit is shown in Figure 5-1.

Figure 5-1: The sensor suit has inertial measurement units (IMUs) in the red locations: one on the back of the head and one in each wrist.
(a red dot is used in the left picture to locate the IMUs).

5.3.1 Experimental Setup

In order for the SRL to determine if the task has been completed successfully, it needs to be able to interpret the sensor suit's recovered data. There are two main approaches that can be taken for this: model driven approach or a data driven approach. In order to determine which approach would be more effective in our case, we considered the fact that our tasks involve a human operator. This means that human uncertainty is involved in the dynamic process of executing the task. A human can determine countless paths that lead to a successfully completed task. Identifying which path is being used is a challenge in itself, and would require extensive knowledge about the different models for the task. Obtaining and analyzing the equations of motion for these possible paths and each sub task can be quite tedious and challenging. Because of this, we discarded a model driven approach. Contrary to the model driven approach, the data driven approach is when we use training data to identify when the task has been completed successfully. This bypasses the necessity of fully understanding the physical laws that govern each action taken by the human and focuses on the final outcome. Once an algorithm has been trained, it can be easily used with real time data to determine whether the task has been completed successfully or not. One of the disadvantages of using a data driven approach is that it becomes difficult to make predictions and that initial conditions can affect the performance of our algorithm. By obtaining multiple training data sets from different trials and outcomes we hope to obtain al algorithm that can effectively determine the state of success of each of the CPN model's transition.

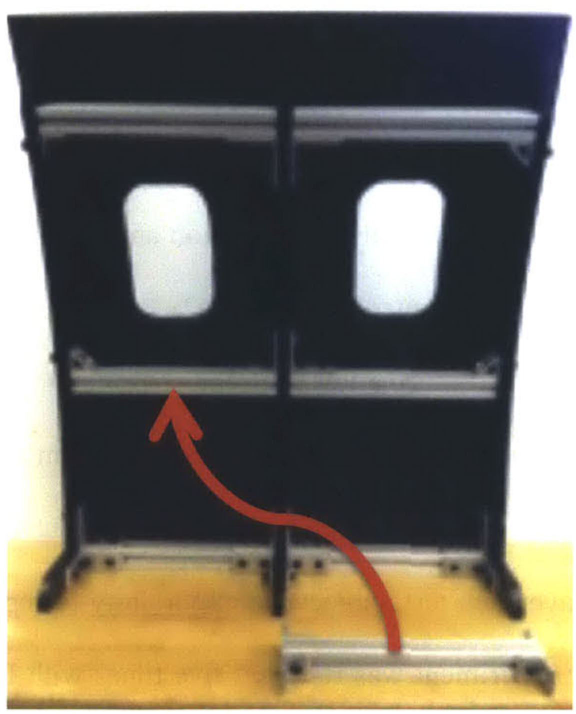

Figure 5-2: Fuselage mockup. The first transition of our CPN model is to place the intercostal in place. The red arrow represents where the intercostal is picked up from and where and how it has to be positioned.

We replicated each of the steps of our CPN model in order for us to obtain our training data sets. For this we prepared a small mockup of a Boeing 787 airplane fuselage, this is shown in Figure 5-2. Using this mockup we performed each transition that is part of the CPN model while taking data. Given that we know when the task is completed, we can look at the data's behavior leading up to that point. Then we proceed to choose which classification technique is more appropriate when determining if the task was successfully completed. The tasks to be measured are:

1. When the intercostal is in place (placed and held by the human).
2. Target location for the SRLm in order to fix the intercostal.
3. When the drill is in place (placed and held by the human).
4. Target location for the SRLm's vacuum end-effector.
5. When the drilling task has been completed.

5.3.2 Success Variable

For the first success variable, we looked at the position and orientation of the intercostal when it is in place. This task is also represented in Figure 5-2. To do this we created a LabView virtual instrument (VI) that records all information from the sensor suit as a function of time while the operator performs the task. In addition, the VI notified the SRLm's operator when he could start the task. This enabled us to control the initial conditions of our experiment. Each experiment lasted 30 seconds and gave each test subject the opportunity to repeat the action of placing the intercostal 4 times. This experiment was repeated five times with five different subjects. It was performed in the same way for when the SRLm's operator has to position the drill after the SRLm has fixed the intercostal and for when the operator uses the drill to secure the intercostal to the fuselage. In the latter experiment, the human started with the drill already in position. This was done to follow the CPN model. Experiments of the same nature were repeated for the remaining tasks. For the second success indicator, which corresponds to the SRL being in the correct position to use the gripper end-effector to fix the intercostal to the fuselage, we took advantage of the SRLm's back-drivability. While the SRLm's operator held the intercostal in place, a second worker would move the SRLm's corresponding hand into position. This was the data used to determine the endpoint position that can be interpreted as the success variable for this particular transition. Once again, this experiment was repeated five times with five different test subjects. This was also the case for when the SRLm had to properly position the vacuum after its operator had successfully positioned the drill.

5.3.3 Data Acquisition and Processing

All the recorded data was obtained through the use of National Instrument® LabView®. The sensor suit's IMUs are Pololu© 's MiniIMU – 9 v2, which contains a gyro, accelerometer, and compass (L3GD20 and LSM303DLHC carrier) and is interfaced through the Arduino Pro Mini to the computer via USB. The sampling rate for the VI used is 0.03s. The front panel for the LabView VI used to record the data from our experiments is shown in Figure 5-3. We define a vector X that contains all the recorded data points such that

$$X = \left\{ \begin{matrix} x_1 & x_2 & \cdots & x_i \end{matrix} \right\}^T$$

where x_i are the data vectors obtained from each IMU. Our success variables that are read from the human for his or her first two transitions depend on orientation and velocity. For this we focus only on the acceleration and velocity readings. The observed trend for both tasks resembles the reaction shown in Figure 5-3. By using the orientation of the arms when they are stabilized and a relatively small velocity we can infer that the intercostal is in place. Using our training data we determined an orientation and maximum velocity threshold that would be used to determine whether the SRLm's operator is successfully holding the intercostal in place. Needles to say, these conditions cannot be evaluated in a single point in time, otherwise our success detection rate would not be accurate. For this we have to consider these conditions throughout a specified period of time to verify if they hold true. Using the training data from each of the test subjects, we observed that the time required for moving the intercostal and carefully orienting it properly is an average of 2.4 ± 0.3s. Evaluating our orientation conditions for only that time window

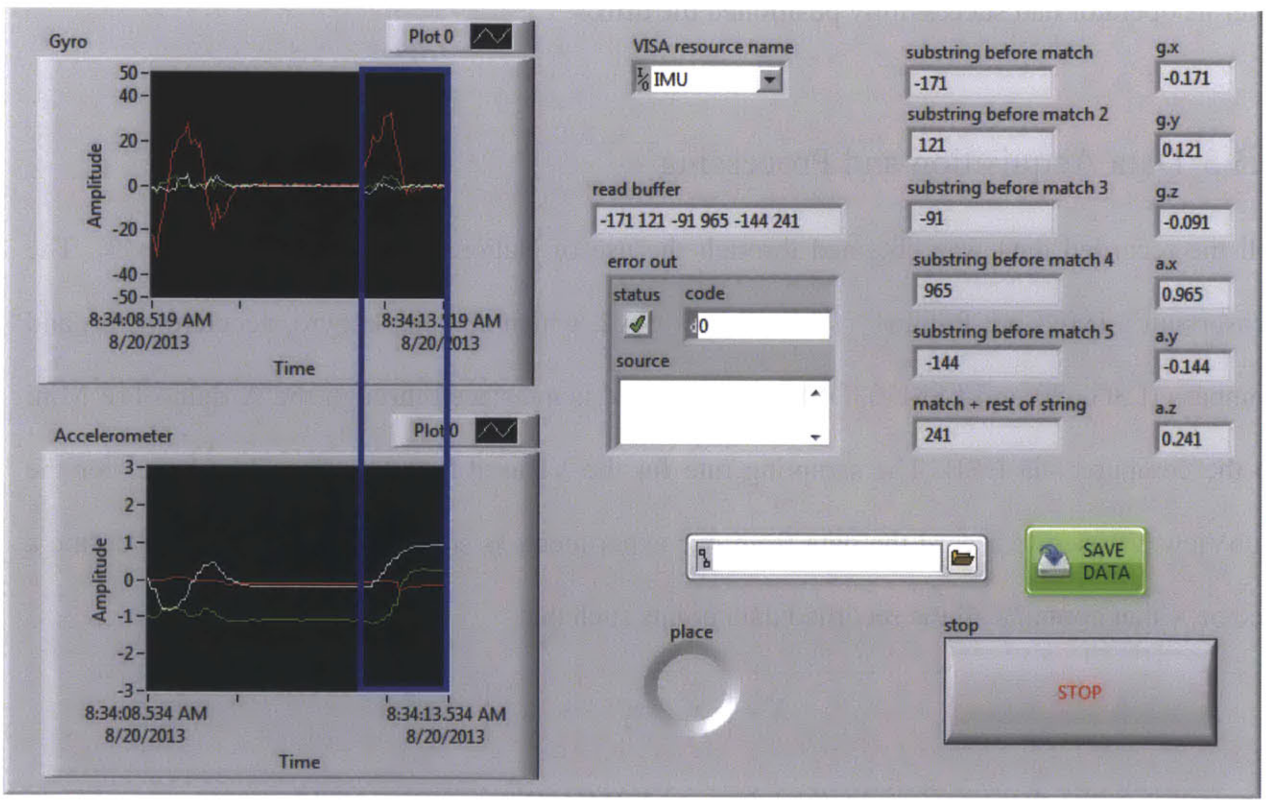

Figure 5-3: Front panel of the VI for acquiring data from the SRLm's Operator while executing the first transition. The images to the left are the raw data obtained form the IMUs. The circle labeled "place" to the left of the stop button turns on yellow when indicating the operator that he can start performing this sub task. The blue square shows the IMU signals of the right hand during the period in which the SRLM's operator proceeded to properly place the intercostal.

depending on the velocity variation can be easily implemented with LabView using case structures, comparison blocks, and elapsed time counters. This same procedure was used for the positioning of the drill. For this second experiment the SRLm had already fixed the intercostal to the fuselage. For this one the average time obtained from the training data was 2.6 ± 0.4s. Using this information obtained from the training data we developed the posture detection VI shown in Figure 5-4.

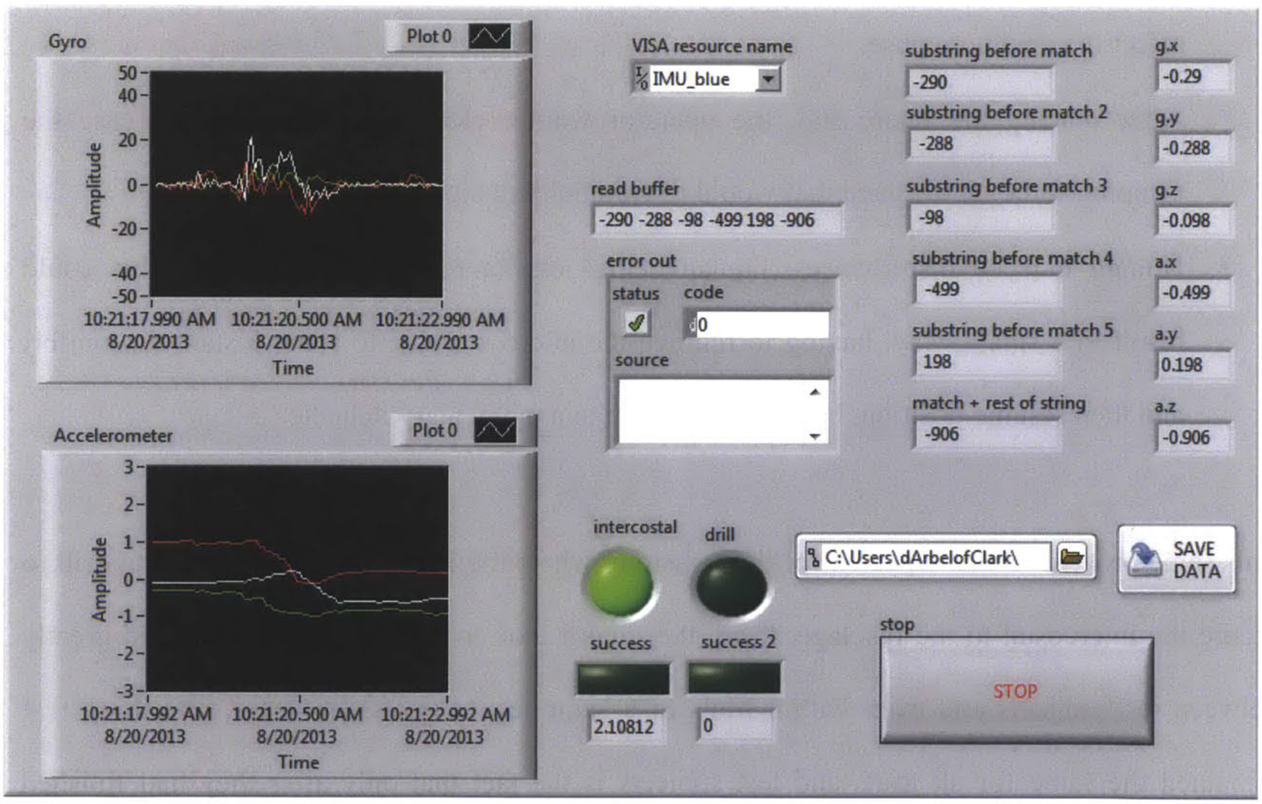

Figure 5-4: Posture detection VI. Using velocities that are within the minimum and the determined orientations we can identify which kind of posture is being assumed by the SRLm's operator. The task starts from rest with the spike in velocity (top graph) and is detected to be in the right position when the orientation (lower graph) matches the expected one.

In Figure 5-4 we can see that although the operator is assuming the posture that corresponds to holding the intercostal in place, it still cannot be classified as success because this posture has only been held for 2.11s and there was a small spike in velocity during the still period. This was repeated for the placing of the drill task. Using this VI we are able to identify when a task has been successfully 87% of the time. This VI discerns between the two postures 99% of the time. Some of the reasons why the posture was not properly classified as successfully completed are:

- After placing the intercostal, the operator would brace to the fuselage while holding the intercostal. This would still keep the intercostal properly fixed to the fuselage but would affect the operators pose.
- After placing the intercostal, the operator would release one hand and increase the gripping force on the hand that would remain holding the intercostal.
- Human induced disturbances: human beings cannot remain perfectly still, this could result in a small sway, having to remove the intercostal due to some instant discomfort and then resume position, having to stop due to a momentary itch, etc.

This same procedure was repeated for the transition where the SRLm's operator uses the drill to secure the intercostal to the fuselage. Here, the time it took to complete the task varied greatly between test subjects and even within trials of a same test subjects. However, one thing that remained the same for all trials and test subjects is the fact that only after they had finished securing the intercostal they exhibited a high acceleration of the hands towards the lower part of the body. This was easily perceived after the relative still state where they operated the drill. This was the criterion used to identify when the drilling task was successfully completed.

Now we turn to determining the endpoint position that corresponds to a successfully positioned SRLm end-effector. This is done in a similar way as the procedure explained previously. Using the SRLm's back-drivability, the second worker places the end-effector in the correct position and orientation. This information is recorded using the contactless absolute encoders that are built in the motor and LabView as the interface. Similarly to the previous case, an average of the trajectories traveled and final endpoint position were calculated. The results present one of the greatest advantages of the SRLm: the fact that its operator is wearing it at all times makes the SRLm's home position to be in the same relative position to the task. When a human is performing a task, he or she will always assume the position that makes it most comfortable for him to execute such task. For our tasks of interest, the operator is already facing forward and at a particular distance from the task's workspace. This automatically gives the SRLm the same initial position and orientation and the human, and since the operator will remain in the same position and orientation while performing his sub tasks, so will the SRLm.

This is seen in the SRLm's recorded data as all the trajectories and endpoint positions were nearly identical. Since the gripping end-effector for the SRLm is passive and the vacuuming end-effector is not in direct contact with the environment, sliding control [24] was used for the SRLm endpoint position control. Another advantage of the SRLm being worn is that in the case that there is a small error in the end-effector position for a particular trial the human can compensate for the error by moving as needed. Since the focus of this thesis is the coordination of the SRLm and its operator when executing a set of tasks we will not address the challenges brought to the control of the SRL due to being mounted on to a dynamic base.

5.3.4 Gesture Recognition

Using the success variable to determine the movement of tokens through the model is good enough for transitions that require the SRL and the operator to perform independent tasks on different objects. However, this is different for when they work on the same object. Let us consider our second transition: fixing the intercostal to the fuselage. For this transition we also have to consider its leading place and what is happening in the physical world. The SRLm's operator just finished positioning the intercostal in the fuselage and we are ready to have the SRLm come in to use the gripper end-effector on the intercostal. Before the SRLm comes in we must make sure that the conditions are right for it to approach the workspace. An additional indicator is needed because, although we can now determine when the intercostal is in place, there may be several facts that can compromise the completion of the task. The operator might also resort to the behavior explained in the previous section, causing the SRLm to revert its decision to that of "the operator has not been successful yet". For example, after positioning the intercostal the human might switch to a posture that is more comfortable for when the SRLm takes action. The SRLm would immediately recognize this and it will not attempt to execute the next transition. This is why the "cue" variable was included in the guard expression return to the previous place in the CPN model. This variable will determine if the necessary environment conditions are met in order for the SRLm to fix the intercostal. Monitoring the entire environment as well as unintentional human induced disturbances is impossible for the SRLm. Just the fact that the SRLm would have to be constantly trying to predict if there will be a human disturbance is not possible with only these 3 sensors.

However, as expressed previously, some tasks that are inherently difficult for the robot can be trivial for the human operator. This is one such case. The human worker can easily determine if all the conditions have been met for the SRLm to initiate his transition. Now that the "cue" variable is based on human decision we can have the same approach as we did for the "success" variables that were dealt with in the previous subsection. Given that an aircraft assembly facility is a noisy environment we discard using verbal communication with the SRLm. This is also due to one of the SRL's main goals: feeling as an extension to the body. Although using explicit gestures to communicate with the SRL is not as natural as moving one's limbs, it is nowhere near as unnatural as talking to your limbs in order for them to act. We consider that the workers prefer to use their heads to communicate with others when both their hands are busy and they are operating within a noisy environment. Also, subconsciously nodding when something has been achieved or when a task has been completed successfully was a trait that was noticed frequently when we performed the tests in the Success Variable subsection of this chapter. For these reasons we decided to use gesture recognition, more specifically a nod, to have the human worker indicate when the SRLm can trigger the transition to fix the intercostal.

5.3.5 Data Acquisition and Processing

We used the same approach to determine which signals were most important when detecting a nod. We proceeded to gather some experimental data before deciding what kind of algorithm should be used for the gesture detection. This was done in this manner because if the nod can be easily determined from just looking at the recorded signals we can chose a simple detection algorithm instead of resorting to a complex one such as Support Vector Machines or Neural Networks. The experiments for this section were similar to the ones in the previous sections of this chapter. We prepared a LabView VI that would give a signal to the operator. The signal consisted of turning on a light on the front panel. Once the light turned on the human would proceed to position the intercostal. Once the intercostal was properly placed and the human was ready to proceed to the next transition he would nod. Each test would last 30 seconds and would give the SRLm's operator 5 indicators to commence his task. These were generated randomly and would be 4 seconds apart at minimum, thus giving the operator enough time to complete the task and nod. This test was performed 5 times with 5 different test subjects. Figure 5-5 shows the front panel of the LabView VI, as well as a usual nod after the indicator turned on.

Figure 5-5: LabView VI for obtaining the training data for nod detection. The sensor shown here is the one attached to the back of the head. We can see that the nod is easily detectable, as the head tends to remain in a static position throughout the execution of the test.

As expected, it was clearly visible from the measured signals when the nodding occurred, therefore a simple threshold relationship would be enough to detect the nod. The signals used to determine the threshold values for the nod detection are the Z and Y velocity readings. These were chosen so that we could detect the nodding action regardless of the posture of the wearer. Accelerometer and Euler Angle readings were discarded as they vary with the head's orientation.

In our case we consider any point within the 20% of our maximum velocities for each nodding action in the training set. This constraint prevents us from giving false positives when the worker performs an unsecure or slow nod, thus taking into account human induced disturbances that could easily affect the SRLm's decision algorithm. The nodding detection results are shown in Figure 5-6. We can successfully detect 90% nods, and ignore feints as long as they don't approach 80% of our maximum velocities from the training data sets. Although these techniques work for the scope of this thesis, we are not able to detect more complex posture – gesture combinations. The complexity gestures and posture relationships used for the transitions increases with the complexity of the task of interest. Therefore, when extending the SRL's database on task models we might need to resort to more powerful pattern recognition techniques, such as Support Vector Machines (SVM) classifiers [15], as mentioned previously. This technique has been widely studied for pattern recognition of one or multiple categories [25, 26] and for time-series data [27].

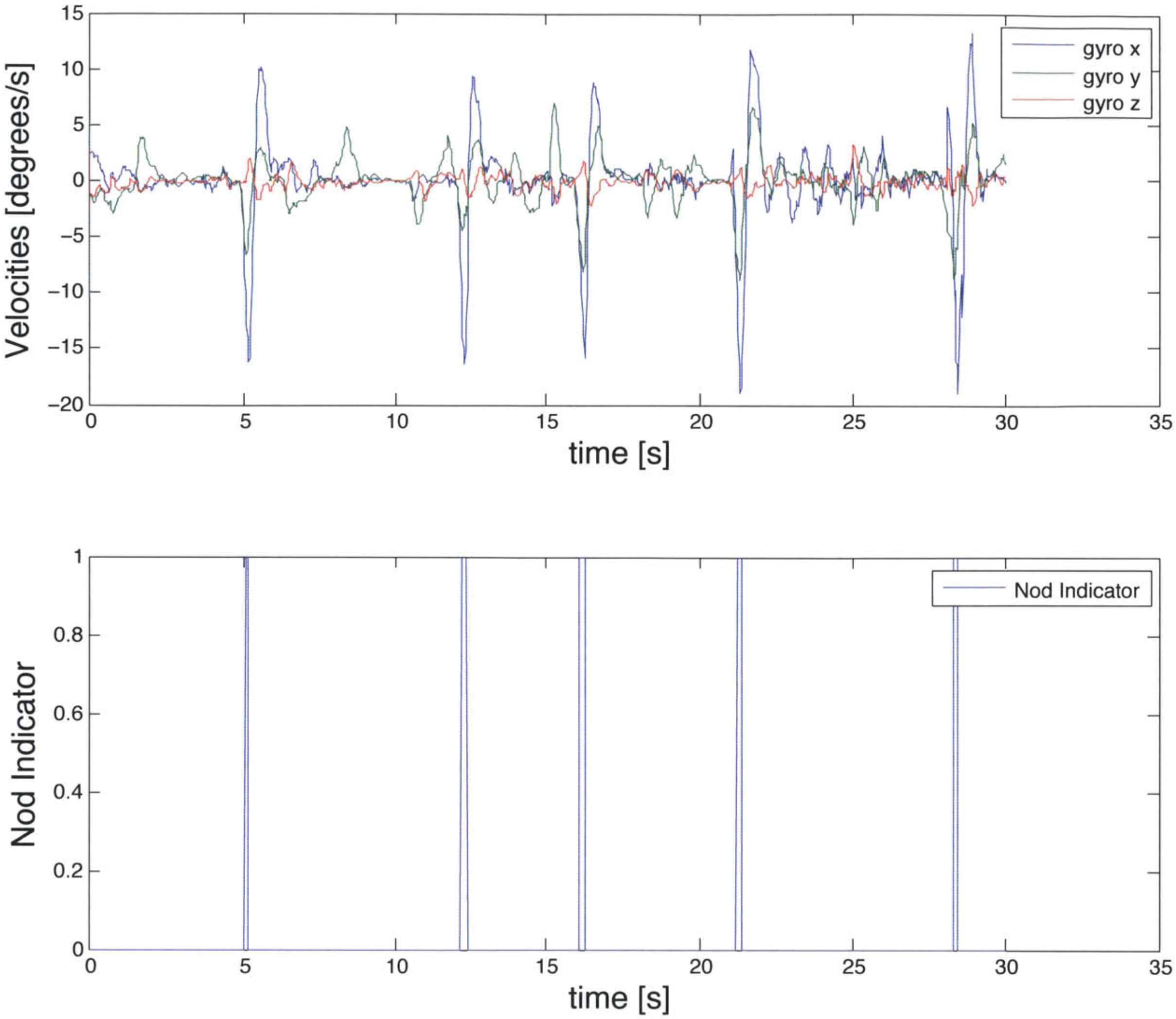

Figure 5-6: Nod detection results obtained applying threshold values to the head sensor.

5.3.6 Implementation

Both the results of the previous sections in posture and gesture recognition as well as the CPN decision algorithm were implemented in a LabView VI and tested on the SRLm. The Front Panel of this VI is shown in Figure 5-7. Some initial experiments were carried out with a modified version of the SRLm. For this full-scale implementation, the SRLm was modified to have 2 degrees of freedom in each arm. Although this limits the SRLm to operate in the horizontal plane, wearing the SRLm in the chest area, as opposed to the hip, can compensate for this. The only change made to the CPN algorithm was due to the fact that the gripper end-effector is passive. For our implementation purposes, the SRLm's operator actuates the gripper. The only time when a human gives an explicit command to the VI is after the grip has been tightened and the intercostal is secured to the fuselage. For the posture and gesture recognition implementation

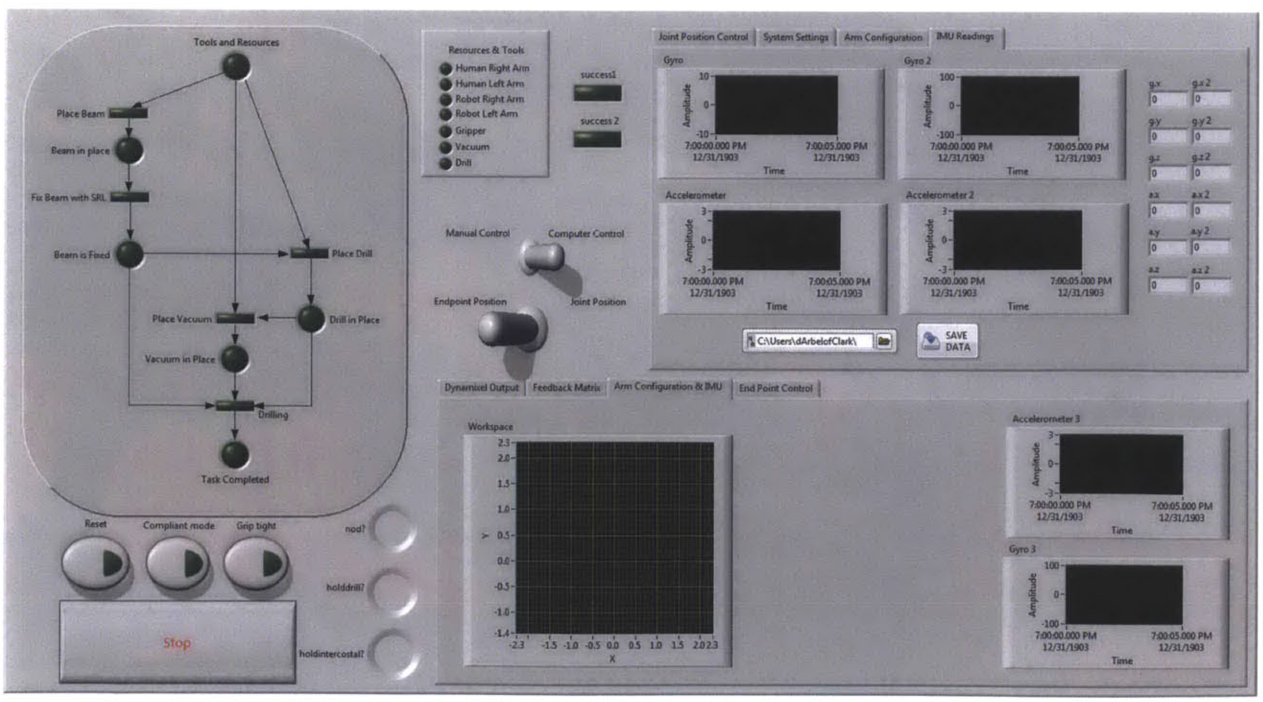

Figure 5-7: Front Panel of CPN model and hardware implementation. On the top left part of the panel we can see the CPN model. The places light up when there are tokens present.

we focused on using the IMUs located in the right hand and the back of the head. This was done because the test subjects were right handed and using the right hand sensor was enough to detect the postures for holding the drill and the intercostal.

Right after the operator has successfully positioned the intercostal and given the cue, the SRLm went directly into the "fix the intercostal" transition. When this decision has been made by the CPN, the control laws for moving to the desired position are assigned to the SRLm. Even though its operator removes the right hand in order to assist the SRLm with the gripping task, the SRLm continues to execute his subtask. This shows that the nod was successfully detected and that all conditions needed for the SRLm to act were matched. This is shown in Figure 5-8.

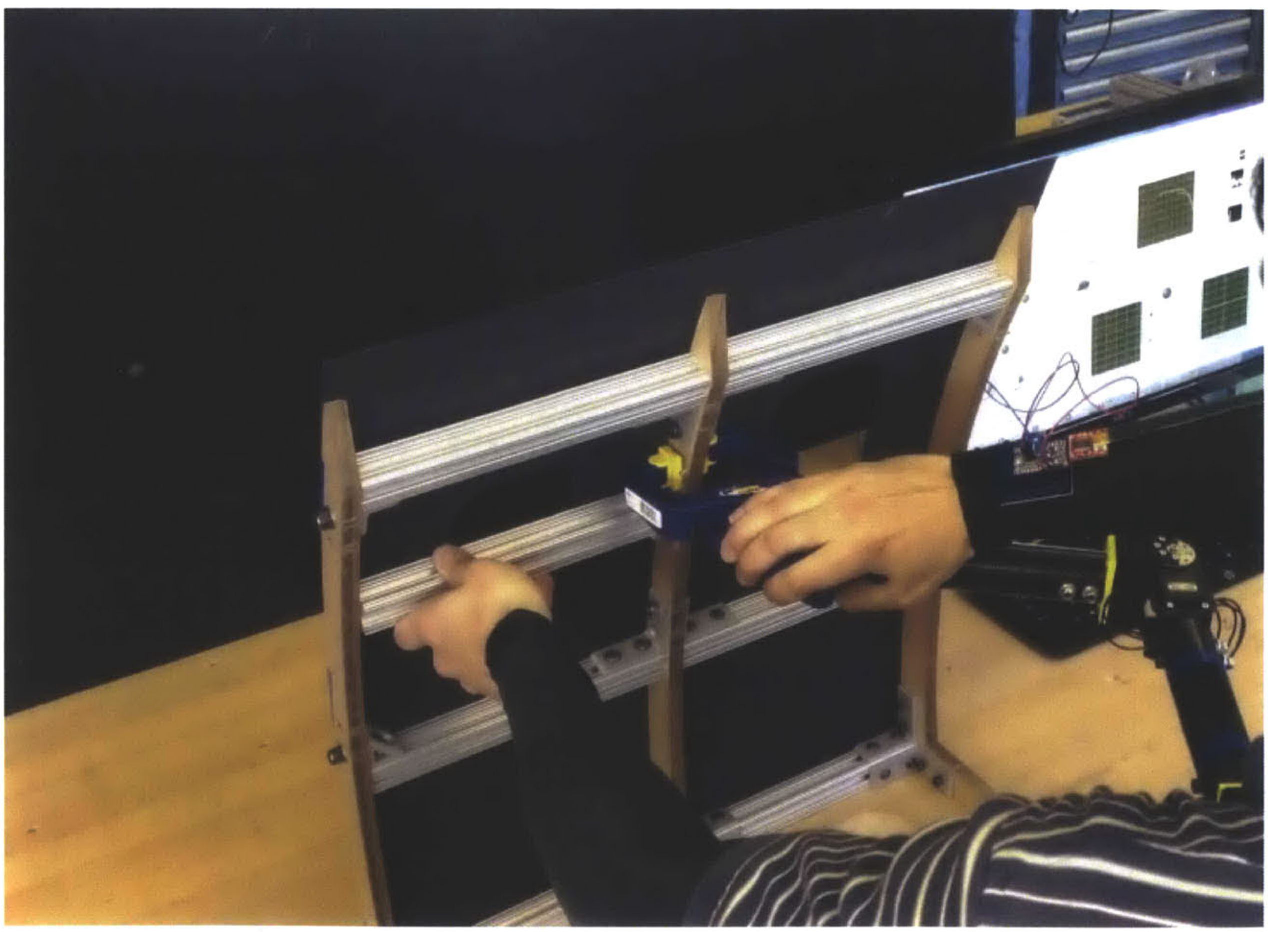

Figure 5-8: Once the operator has positioned the intercostal and given the cue, the SRLm automatically moves in to place the gripper end-effector in its assigned position.

After the SRLm confirms that the intercostal has been fixed to the fuselage, the tokens in the CPN model are transferred to their next places. This means that the CPN is aware that the human's arms are free and that the gripper end-effector and the SRLm's right hand are in the "fixed intercostal" place. This automatically assigns the static conditions that the SRLm's right hand will maintain while on this CPN place. These are to maintain the current position and to keep exerting the force required to hold the intercostal. The later is meant for when the gripping end-effector is automated. This is shown in Figure 5-9.

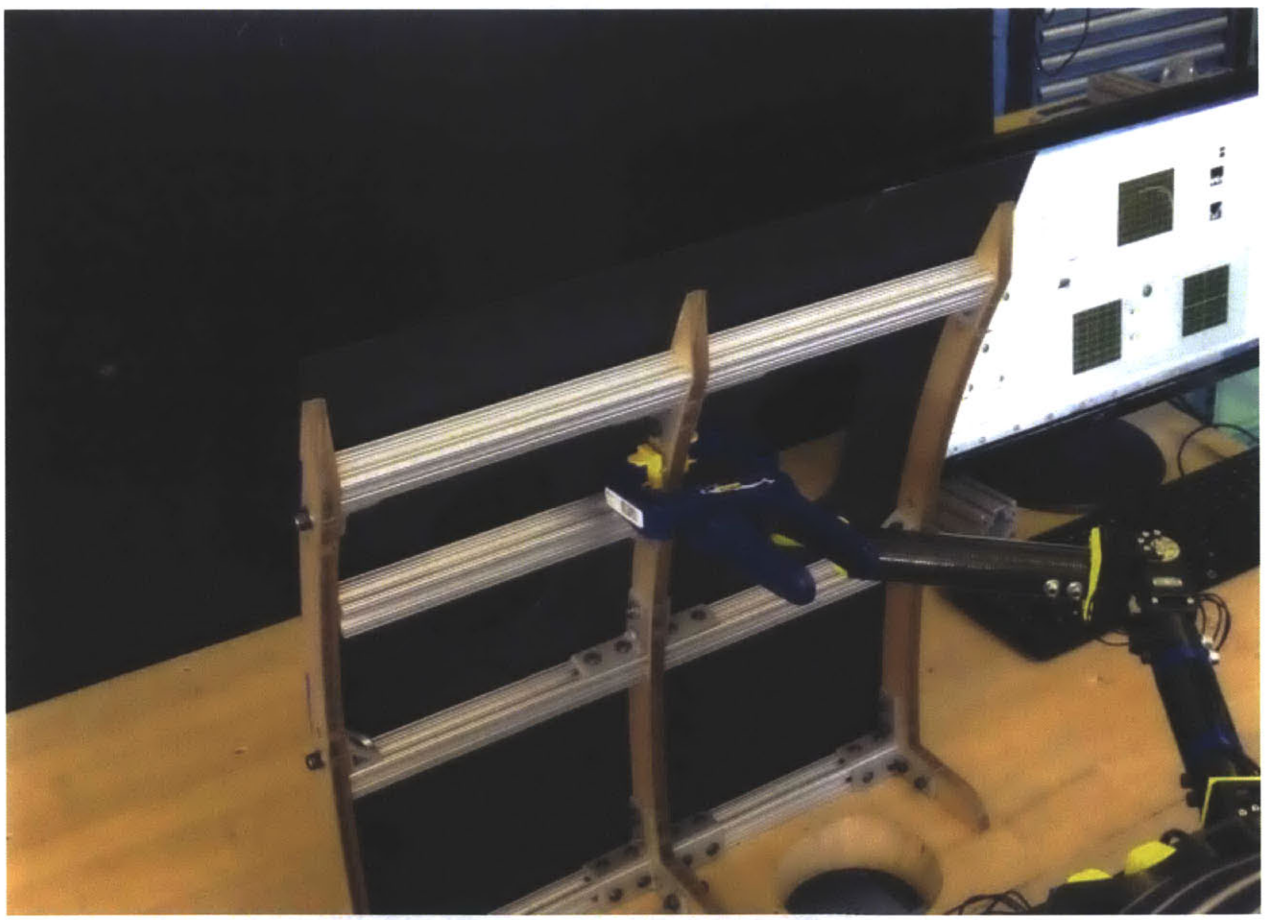

Figure 5-9: SRLm keeps the intercostal fixed to the fuselage after the "fix intercostal" transition occurs. After the intercostal is confirmed to be fixed in place the operator can proceed to pick up and place the drill.

Once the operator places the drill in place, the SRLm automatically proceeds to his transition corresponding to positioning the vacuum. Once again, by triggering this transition, the CPN executes the control laws that move the SRLm's left arm into position. This is shown in Figure 5-10. After the operator is finished securing the intercostal to the fuselage and releasing the gripper end-effector then both the operator and the SRLm resume to their standby position.

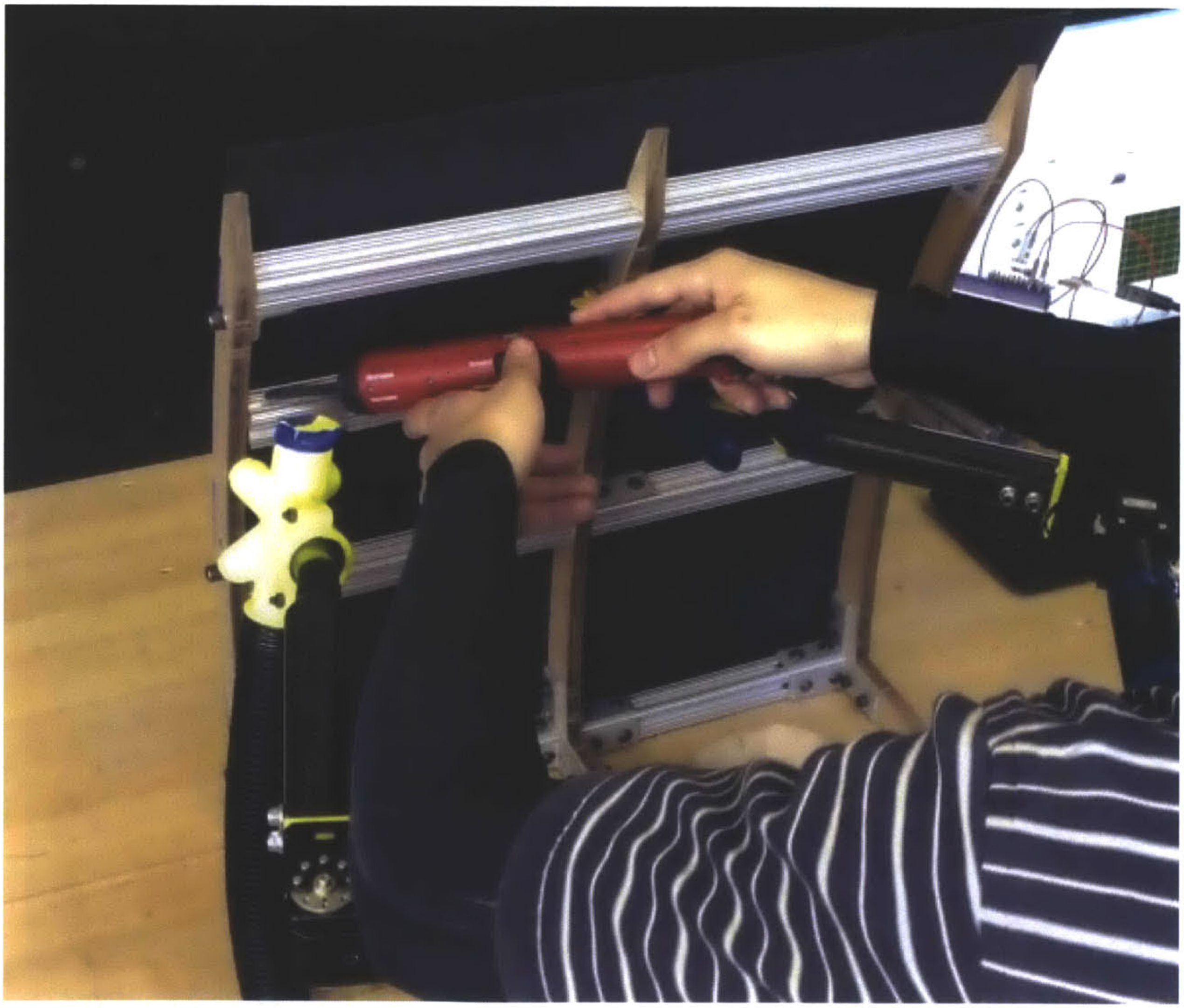

Figure 5-10: SRLm automatically approaches the position of the drill once its operator has finished successfully positioning the drill.

5.3.7 Results and Discussion

Two test subjects executed the complete assembly task repeatedly; each repeated the whole task 10 times. When executing the complete assembly task, both test subjects succeeded 90% of the time. Once an operator was aware of the indicators used by the SRLm, he was more aware of the actions that he had to take and through practice in each of the trials he was able to increase his efficiency while performing the task together with the SRLm. The failed tests were usually during the first trial. By failure we mean that the operator was stuck at a single transition of the CPN model for more than one minute, excluding the final drilling part. Reasons for failure are mainly due to the SRLm not picking up the correct indicators in order to trigger a transition.

Assigning the output of our posture/gesture recognition algorithms we are able to integrate our sensor suit directly with the CPN conditions and guards; which is then used to relay the human worker's intention to the robot. The CPN can then assign the corresponding task dynamic model to the SRLm depending on the places and transitions that the CPN's tokens are currently activating. Our posture/gesture algorithm works for simple gestures that can be easily classified. By focusing on the nodding action we are able to simulate coordination similar to that of working together with another human. To accomplish our goal of making the SRL feel like an extension of the human body we need to make this communication process more intuitive. For this we need to identify other signals and rely more on the wearer's natural gestures while executing the task. A possible approach is to add force sensors in the safety gloves to monitor the worker's grip pattern. Another approach may be the use of gaze tracking to give the SRL a means to share the same eyes as the human. In this scenario, a simple threshold technique to identify these complex indicators would not suffice. A stronger classifier, such as the Support Vector Machines, could be used address this issue.

5.4 TEACH BY DEMONSTRATION

As explicitly expressed in the previous sections of this chapter, when a transition is triggered by the CPN model, a control law is assigned to the SRLm. This control law ensures that the SRLm performed the required action, which could be getting its end-effector to a particular point in space, following a particular trajectory, or exerting a force on a certain point. This was mainly covered in the section 5.3.2 of this chapter. By using the SRLm's back-drivability, we are able to record the joint's trajectories that make the end-effector get to their desired location. When the relationship is simple and the action is independent of the operator's current actions, we use the approach presented in section 5.3.2.

However, this is not the case for situations where the SRLm and the operator have to perform dynamic actions simultaneously or when the task itself cannot be expressed with a simple modeling technique. For this case we consider a situation that is not represented in the current CPN model. Although it is not present in our current model, it is possible to easily incorporate it although it increases the complexity of our decision-making algorithm by adding additional node ruptures (nodes that are parallel in the state space analysis) before the final state. This situation goes as follows:

1. The SRLm has successfully completed fixing the intercostal with the end-effector.
2. The operator's next task is to pick up the drill and get ready to secure the intercostal to the fuselage, but after a long day of work he prefers to presume the role of the SRLm, which is only to vacuum.
3. The SRLm detects the operator's intention and using a drill end-effector positions itself. After doing so, he waits until the operator has successfully positioned the vacuum in place before he stars securing the intercostal.

Another variation of this scenario is that the operator prefers to completely reverse roles with the SRL. This way the SRL could use one limb to brace to the environment and increase its stiffness and then use the other arm to perform the drilling. In fact, there are multiple variations of this scenario where the SRLm then must have available a model for a dynamically complex task such as drilling and be able to coordinate his actions with the operator's simultaneous actions. Once again, due to the complex nature of the task in question we will utilize a data-driven approach. To explore dynamically performing collaborative tasks, such as the ones described above, we will consider another aircraft assembly task that requires two workers.

The majority of aircraft assembly operations are joining operations; workers mate workpieces and fasten them with a tool, or make holes, dispense sealant, place screws, and tighten them. The procedure of each step of operation is well documented in a manual. Joining two workpieces entails two operations; one is to place the parts at a specified location, and the other is to fasten them. Typically a single worker performs the operation in two steps. First, the worker places the workpieces and secures them with a jig and a fixture. Second, the worker holds a tool and joins the parts together. This two-step operation can be streamlined if an assistant can hold the workpieces while the worker fastens them. Figure 5-11 shows a simple example of this kind of manufacturing operation: joining a workpiece to a structure. A drill is used to make a hole in the metal plate. In typical aircraft manufacturing, the tool used is too heavy to carry with a single hand. Both hands must be used to hold the tool while drilling the workpiece. The two most important parts of this task are the beginning of drilling and when the drill goes through the workpiece. The first part is when the most erratic movement is felt. During the second, the force felt on the plate decreases but then increases again when the lead worker attempts to remove the drill from the plate.

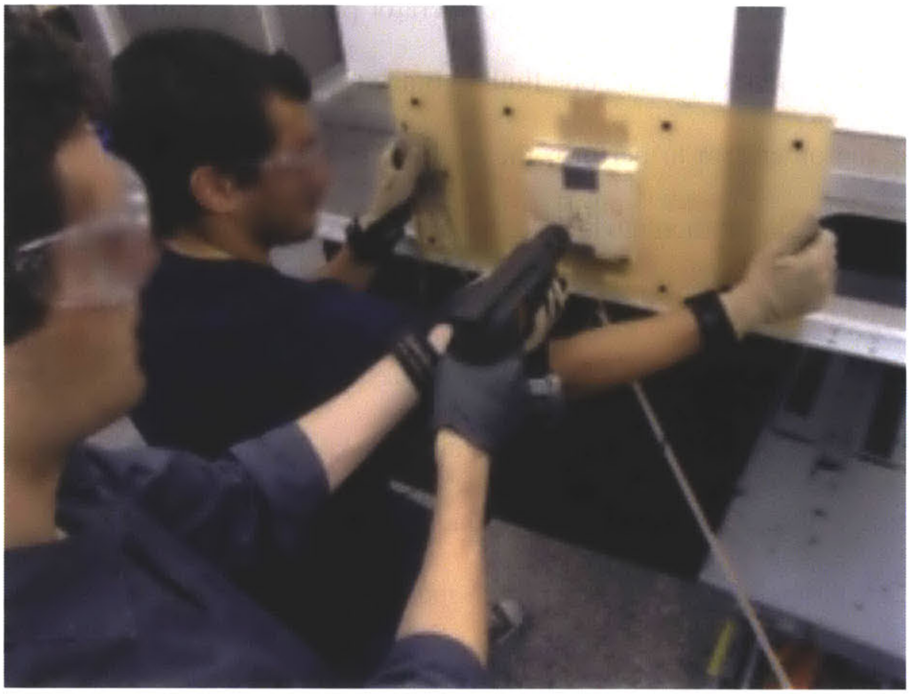

Figure 5-11: Demonstration data acquisition of two workers' cooperative operations: lead worker holds a hand drill while the follower holds a workpiece that needs to be fixed to the plane.

This causes the part to be suddenly pulled towards the drill, causing a jerky motion. To avoid this, the feed rate and the cutting force must be controlled properly and the workpieces must be held firmly to bear the sudden change in the reaction force. Concerted operations are required among the four hands of both workers.

We aim to perform this collaborative task with the SRL. The worker's role is to regulate the drilling operation, while the robotic system bears the weight of the part. This needs a high level of coordination between the human and the SRL. Also, the robot must adapt itself to the state of the human and of the task process. We now consider teaching-by-showing as an intuitive method for transferring intended motion, skills, and strategies from the human to target robots. This section briefly describes a data-driven, intuitive approach to teaching the SRL concerted operations with the human. Once the demonstration data have been collected, the challenge is to extract estimation algorithms for controlling the SRL while it performs the task together with the single lead worker. Our goal is to replace two hands among the four with two robotic limbs. Let us first assume that the lead worker's dominant hand, say, the right hand, takes over the control of the task process. In the drilling example, the right hand of the lead worker determines the feed rate and drill pressure, which are the key variables for executing the task. The behaviors of the follower hands are therefore reactive; they watch the leader hands and decide what to do. This implies that there is a causal relationship between the dominant hand and the other hands. As shown in Figure 5-12, the behaviors of the follower hands may be described as causal dynamic processes receiving signals mostly from the dominant leader hand and reacting to the received signals. The questions are then which specific signals the follower hands use, what model structure is efficient for predicting the behaviors of the follower hands, and how the model can be tuned. We treat this problem as a system identification problem.

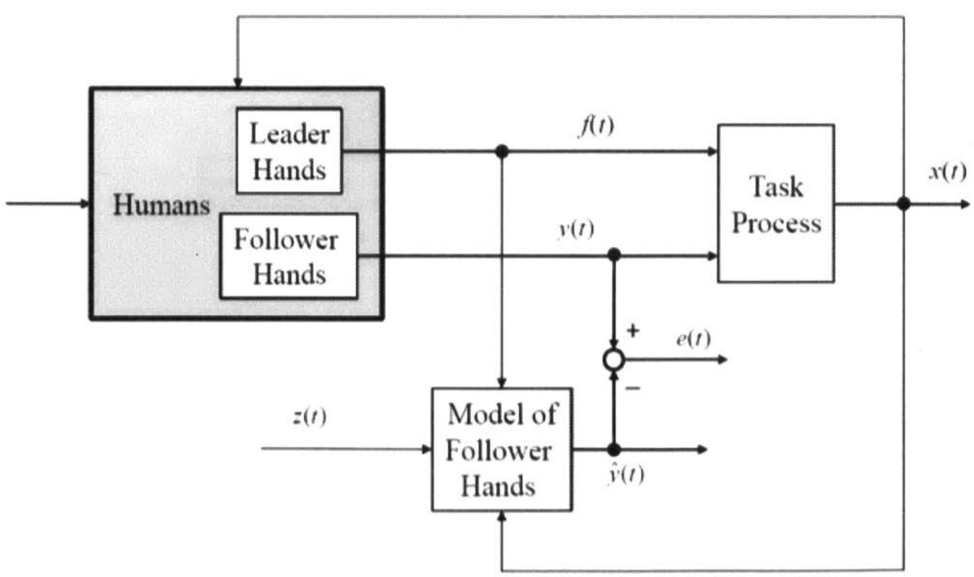

Figure 5-12: Dynamic model of the follower's role.

5.4.1 Experimental Setup

Figure 5-13 shows a scheme of the experimental setup, including thin film force sensors (S1 and S2), a camera (S3), and the aluminum plate. The output signals were chosen such that they could be easily monitored through the use of wearable sensors on the worker's personal protection equipment (such as their helmets and gloves) and mounted sensors on the SRL (such as cameras and proximity sensors). The force sensors used were manufactured by *FlexiForce®* (model A201). The camera measured the distance from the drill to the plate x, while the first and second group of sensors measured the leading (cutting) force f and the following (holding) force y respectively. The plates used for the test had a thickness of 3/8 inches and their center was at a height of 4 feet 8 inches from the ground.

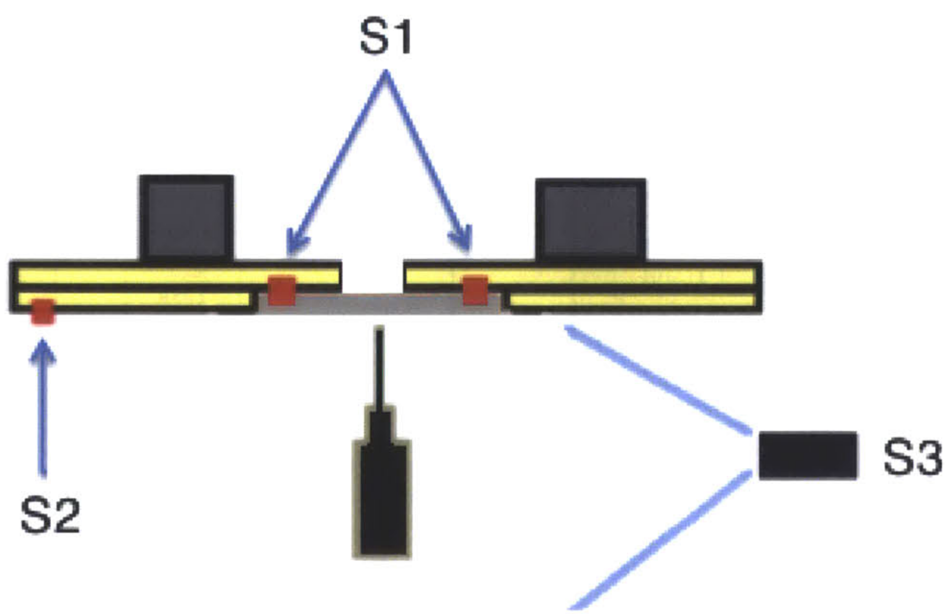

Figure 5-13: Scheme of the experimental setup. S1 is the group of force sensors measuring the leading force f, S2 is the group of force sensors measuring the following force y, and S3 is a camera that records the position x of the drill with respect to the aluminum plate (light grey and yellow).

The test subjects were instructed to keep drilling without stopping (leading worker) and to hold the plate in place with enough force to prevent rattling and erratic motions (assistant worker).

5.4.2 Data Acquisition and Processing

All experimental data was recorded using National Instrument's LabView. For the experiment presented in the previous section three pairs of volunteers participated in the experiment and a total of 26 datasets was collected. The datasets were divided in two groups: 20 were used to determine the parameters and the remaining 6 were used for validation. The sampling rate for the VI was 0.017s. The general trend present throughout all 26 datasets is shown in Figure 5-14. The following hand keeps applying a stable force to the workpiece as the drill comes closer to breaking through the plate. When the drill breaks through the workpiece, the plate holding force increases in order to compensate erratic movements. In this instant, the leading (cutting) force stops increasing and after a small time interval decreases abruptly in order to minimize the vibrations of the plate as it goes through it.

Given this data set of demonstrations, we proceed to identify the dynamic model relating the follower motion to a time sequence of the dominant hand motion and other measurements. Let $f(t)$ and $y(t)$ be leading hands and the follower hands applied force at time t, respectively. The position of the drill is denoted by $x(t)$. The change in $x(t)$ represents the distance the drill has traveled into the aluminum plate. One time sequence of demonstration data is represented as a data set:

$$S^j = \left\{ x^j(t), y^j(t), f^j(t) \mid t = 1, \cdots, N \right\},$$

where the same demonstration task is repeated m times: $1 \leq j \leq m$.

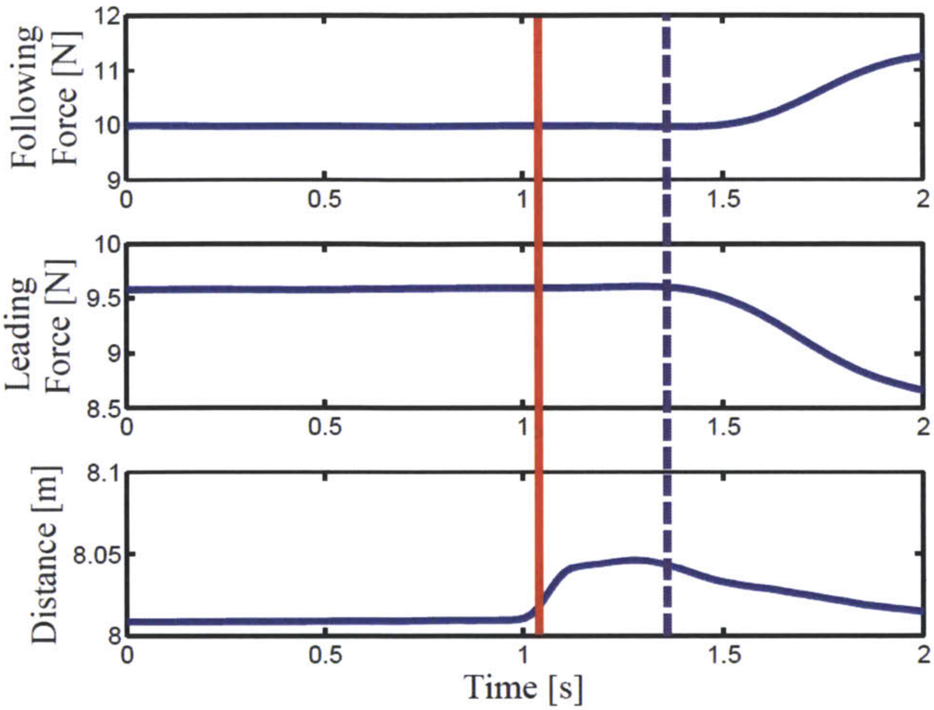

Figure 5-14. General trend for all datasets. The red line indicates the time for which the drill breaks through the workpiece, while the purple dashed line indicates the time when the lead worker starts to remove the drill from the part.

We proceed to identify a dynamic model of the follower hand that can predict its applied force in relation to the dominant hand motion and the drills relative position: $\hat{y}(t) = f(f(t), x(t); t)$. For the purpose of notational simplicity, we deal with scalar quantities for $f(t)$, $y(t)$ and $x(t)$. Let us begin with considering an ARMAX (Auto Regressive Moving Average with eXogenous input) model for the dynamics of the follower hand:

$$y(t) = -\sum_{i=1}^{n_a} a_i y(t-i) + \sum_{i=0}^{n_b} b_i f(t-i) + \sum_{i=1}^{n_c} c_i x(t-i) + z(t) + v(t) \qquad (1)$$

where a_i, b_i, c_i are parameters to identify, $v(t)$ is a zero mean, random variable representing un-modeled dynamics. $E[v(t)] = 0$, and $z(t)$ represents the follower hand motion that is not correlated with other terms:

$$f(t), \dots, f(t-n_b); x(t), \dots, x(t-n_c); y(t-1), \dots, y(t-n_a)$$

The problem is to find parameter values involved in eq.(1) as well as the time function $z(t)$ representing the motion that cannot be predicted from other measurements. Let $\varphi(t)$ and θ, the regressor and parameter vector associated with eq.(1), be defined as

$$\varphi(t) = (-y(t-1), \dots, -y(t-n_a), f(t-1), \dots, f(t-n_b), x(t-1), \dots, x(t-n_c))^T$$

$$\theta = (a_1, \dots, a_{na}, b_0, \dots, b_{nb}, c_0, \dots c_{nc})^T$$

Using estimated parameter values $\hat{\theta}$ and estimated uncorrelated function $\hat{z}(t)$, the follower hand motion is predicted as

$$\hat{y}(t) = \hat{\theta}^T \varphi(t) + \hat{z}(t) \qquad (2)$$

The parameter vector $\hat{\theta}$ and the uncorrelated function $\hat{z}(t), t = 1, \dots, N$ are obtained from the demonstration data sets $S^j, j = 1, \dots, m$ based on least mean square estimate:

$$\min_{\substack{\theta \\ z(1),\dots,z(N)}} \left\{ \sum_{j=1}^{m} \sum_{t=1}^{N} \left(y^j(t) - \hat{y}^j(t; z(t)) \right)^2 \right\} \qquad (3)$$

where the second term in the parenthesis is the prediction of the follower hand motion for the j-th demonstration data: $\hat{y}^j\left(t;\hat{\theta},z(t)\right)=\hat{\theta}^T\varphi^j(t)+\hat{z}(t)$ where $\hat{z}(t)$ is given by:

$$\hat{z}(t)=\frac{1}{m}\sum_{j=1}^{m}\left(y^j(t)-\hat{\theta}^T\varphi^j(t)\right)$$

where $\hat{\theta}$ is an optimal solution to (3). The predicted holding force of the follower, given by (2), provides us with an algorithm determining the gripping force that the SRLm has to exert on the plate in order to prevent it from coming loose. The second term on the right hand side, $\hat{z}(t)$, can be viewed as a feedforward term, while the first term is a type of feedback that modifies the feedforward term based on the measurement of $f(t\text{-}1)$, $y(t\text{-}1)$, $x(t\text{-}1)$ and their preceding values. Each of the time delays for the $\varphi(t)$ terms was set to 340ms.

5.4.3 Results and Discussion

In order to obtain the coefficients for our model we evaluated two cases. For the first one we used a model that only takes into consideration the autoregressive terms of the holding force and the force exerted by the leading hand with the drill (c terms in $\hat{\theta}$ are 0). For the second model we considered the autoregressive terms and the position of the drill (b terms in $\hat{\theta}$ are 0). This way, by comparing the resulting models we can determine which signal is more heavily weighted by the follower when he determines how much force he has to use to hold the plate in place. Table 10-1 shows the values obtained for $\hat{\theta}$'s coefficients for both models. We proceeded to test the performance of our models by predicting the holding force on one of the data sets reserved for validation. The results are shown in Figure 5-15.

Model Parameters	*Input data: cutting force f*	*Input data: drill position x*
a_1	2.2933	-0.505
a_2	0.7811	1.4889
b_1	-2.3519	0
b_2	-2.3325	0
c_1	0	-7.1790
c_2	0	7.4747

Table 5-1: The task dynamic model has been identified using as input data the cutting force f or the drill position x.

Figure 5-15 shows that both of our predictive models are below the required force for holding the plate while the leading hand is still drilling. However, although this could theoretically lead to the task's failure we take into consideration one of the key facts expressed in the beginning of section 5.4: the points of interest are when the drilling starts and when the drill breaks through the plate. Our force prediction between these two points are not crucial because it is during that period that the cutting force is still being applied to the plate and the drill motion has been stabilized. These two facts add an additional holding force that is not perceived by the following hand's force sensors but by the plate in itself. However, this advantage does not exist at the point when the drill breaks through the plate. Thus, the effectiveness of our predictice models can be evaluated by taking into account their performance in this instant. Using this criterion, we observe that the prediction of the model based on the cutting force is either closer or predicts a higher force than the actual force produced by the coworker during these tests. On the other hand, the model based on the plate's position although its shape closely resembles the actual holding force, we can observe that its magnitude is not as accurate.

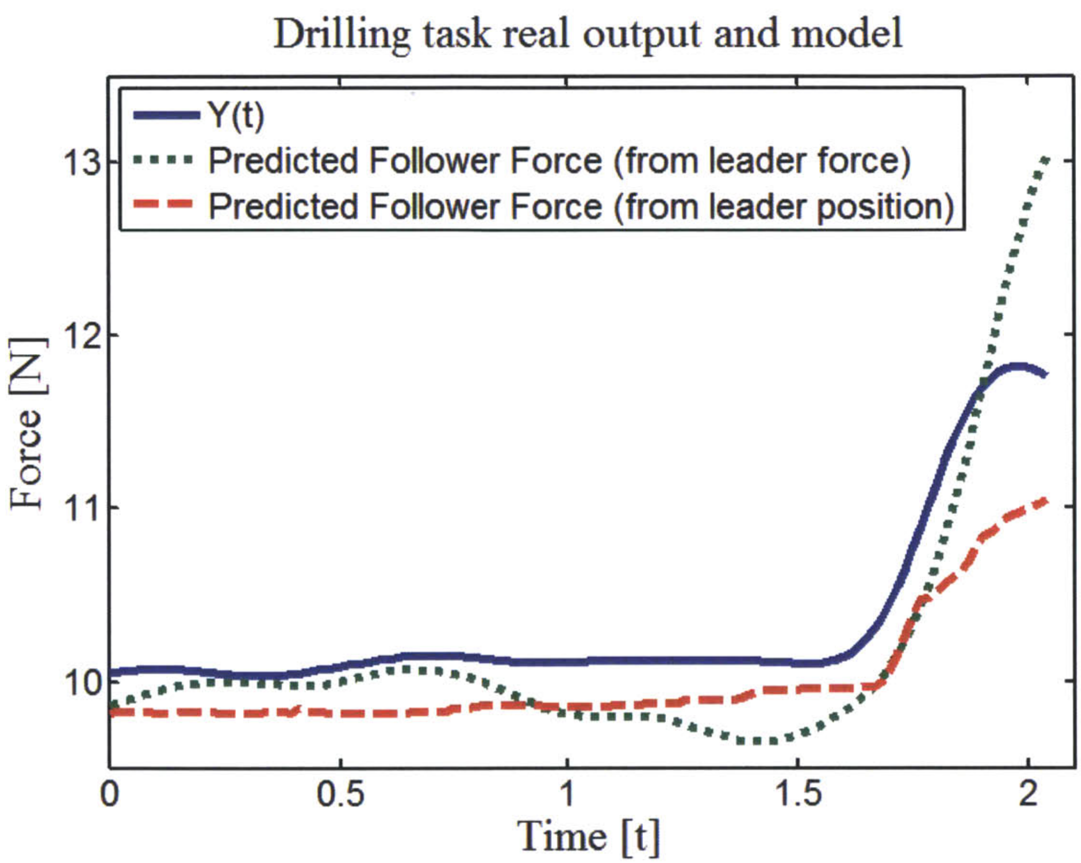

Figure 5-15: We can see how our models are able to predict the holding force that has to be applied by the SRLm. The blue graph represents the actual holding force of our validating data set. The green graph represents our prediction using the autoregressive terms and the cutting force of the drill as inputs to our model. The red graph represents our prediction using the autoregressive terms and the position of the drill as inputs to our model.

Using this analysis we can conclude that although the force based model is not very accurate, it would provide a prediction that is highly likely to succeed in holding the plate in place throughout the drilling process. Another approach that might result in a better prediction is using an average of both model's predictions. This might have the effect of balancing the disproportionate predictions of the force based model with the form accuracy of the position based model This can be used to also conclude that the coworkes uses the perceived cutting force as an indicator to control his applied force (magnitude) and uses the drill's relative position to adjust the pattern used to apply the force.

Chapter 6

Conclusion and Future Works

In this thesis we study various control and coordination aspects of the Supernumerary Robotic Limbs (SRL), a wearable robot whose main goal is to augment the capabilities of the user while being perceived as part of his or her body. We have shown how Coloured Petri Nets (CPNs) can be used to effectively model specialized aircraft assembly tasks for a worker using the SRL. By thoroughly specifying all the conditions that must be met for each transition of the CPN model we obtained a decision-making algorithm that can successfully coordinate the SRL with its operator throughout the execution of the intercostal assembly process. Using the State Space method we were able to analyze the CPN model's resource and tool allocation management, reachable states and end states. After analyzing our results, we conclude that through rigorous definition of each state and transition of the CPN we are able to obtain a model with minimal system states, perfect resource management and only one system end state, which corresponds to a successfully completed task. This CPN model can be used as a module in an overall hierarchical CPN (HCPN). This HCPN, containing multiple CPN modules for several assembly processes, would be in charge of the SRL's decision - making process.

Combining our task model with the sensor suit allows us to use predetermined gestures and detected postures to relay the human's intention to the SRL. These are incorporated into the CPN model, which is in charge of leading the SRL into the system dynamics relevant to the determined next task. This coordination between the SRL and the human worker resembles the coordination between two workers. In order to move towards perceiving the SRL as part of the human body we have to further explore human behavior during aircraft assembly tasks. Future work in this area includes increasing the number and variety of sensors that monitor human behavior. For these complex sensor networks the use of Support Vector Machines could help detect more complicated gesture – posture combinations present in other, more complex, assembly processes. This method can be used to include more intuitive but harder to identify gestures as indicators in our CPN model. Further work in the use of CPN models as a decision-making algorithm includes taking human induced uncertainties into account. These may compromise the task's successful completion or change the CPN model itself.

Finally, a coordination-based control algorithm for collaborative drilling tasks can be implemented based on the system model that has been derived from demonstration data. Further work in this area includes the expansion of sensor variety when determining these control algorithms. Also, these control algorithms should be implemented and tested on the SRLm to evaluate their real-time performance.

Bibliography

[1] Kousidou, S., Tsagarakis, N., Caldwell, D. G., and Smith, C., 2006. "Assistive exoskeleton for task based physiotherapy in 3-dimensional space". *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, February, pp. 266-271.

[2] Bitzer, Sebastian, and Patrick van der Smagt, 2006. "Learning EMG control of a robotic hand: towards active prostheses." IEEE International Conference on Robotics and Automation (ICRA). pp. 2819-2823.

[3] Dhillon, G. S., & Horch, K. W., 2005. *"Direct neural sensory feedback and control of a prosthetic arm"*. Neural Systems and Rehabilitation Engineering, IEEE Transactions on, 13(4), 468-472.

[4] A. Dollar and H. Herr, 2008. "*Lower extremity exoskeletons and active orthoses: challenges and state-of-the-art*" IEEE Transactions on Robotics, vol. 24, no. 1, pp. 144–158.

[5] C. Walsh, K. Pasch, and H. Herr, 2006. "*An autonomous, underactuated exoskeleton for load-carrying augmentation*" Processings of IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), pp. 1410–1415.

[6] Perry, J. C., Rosen, J., & Burns, S. (2007). Upper-limb powered exoskeleton design. Mechatronics, IEEE/ASME Transactions on, 12(4), 408-417.

[7] Hoffman, G., and Breazeal, C., 2004. "Collaboration in human-robot teams". *Proc. of the AIAA 1st Intelligent Systems Technical Conference, Chicago, IL, USA*. September.

[8] Yamamoto, Y., Eda, H., & Yun, X., 1996. "Coordinated task execution of a human and a mobile manipulator". In *IEEE Proceedings in International Conference on Robotics and Automation*, Vol. 2, pp. 1006-1011

[9] Awais, M., and Henrich, D., 2010. "Human-robot collaboration by intention recognition using probabilistic state machines". *2010 IEEE 19th International Workshop In Robotics in Alpe-Adria-Danube Region (RAAD)*, June, pp. 75-80.

[10] Jayasiri, A., Mann, G. K., & Gosine, R. G. (2011, May). Tightly-coupled multi robot coordination using decentralized supervisory control of Fuzzy Discrete Event Systems. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on* (pp. 3358-3363). IEEE.

[11] Pereira, G. A., Pimentel, B. S., Chaimowicz, L., & Campos, M. F. (2002). Coordination of multiple mobile robots in an object carrying task using implicit communication. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on* (Vol. 1, pp. 281-286). IEEE.

[12] Hiratsuka, M., & Asada, H. H. (2000). Detection of human mistakes and misperception for human perceptive augmentation: Behavior monitoring using hybrid hidden markov models. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on* (Vol. 1, pp. 577-582). IEEE.

[13] Desrochers, A. A., & Al-Jaar, R. Y. (1995). *Applications of Petri nets in manufacturing systems: modeling, control, and performance analysis* (Vol. 70). Piscataway^ eNJ NJ: IEEE press.

[14] Jensen, K., & Kristensen, L. M. (2009). "Coloured Petri Nets: modelling and validation of concurrent systems", Springer, Chapter 1-7.

[15] Theodoridis, S., & Koutroumbas, K. (1999). "Pattern Recognition", Fourth Edition, Elsevier Inc.

[16] Jurafsky, D., Martin, J. H., Kehler, A., Vander Linden, K., & Ward, N. (2000).*Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition* (Vol. 2). Upper Saddle River: Prentice Hall.

[17] Li, G., Schultz, A. E., & Kuiken, T. A. (2010). Quantifying pattern recognition—Based myoelectric control of multifunctional transradial prostheses. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, *18*(2), 185-192.

[18] Veraart, C., Wanet-Defalque, M. C., Gérard, B., Vanlierde, A., & Delbeke, J. (2003). Pattern recognition with the optic nerve visual prosthesis. *Artificial organs*, *27*(11), 996-1004.

[19] Calinon, S., & Billard, A. (2007, August). Active teaching in robot programming by demonstration. In *Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium on* (pp. 702-707). IEEE.

[20] Kosuge, K., Fukuda, T., & Asada, H. (1991, August). Acquisition of human skills for robotic systems. In *Intelligent Control, 1991., Proceedings of the 1991 IEEE International Symposium on* (pp. 469-474). IEEE.

[21] Calinon, S. (2008). Robot programming by demonstration. In *Springer Handbook of Robotics* (pp. 1371-1394). Springer Berlin Heidelberg.

[22] Onda, H., Kitagaki, K., & Suehiro, T. (2005, August). Visualization and simulation of sensory events as a representation of states for state-based teaching by demonstration in VR. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on* (pp. 2405-2410). IEEE.

[23] Viswanadham, N., & Narahari, Y. (1987, March). "Coloured Petri net models for automated manufacturing systems". *Proceedings of The 1987 IEEE International Conference in Robotics and Automation,* Vol. 4, pp. 1985-1990.

[24] Slotine, J. J. E., & Li, W. (1991). *Applied nonlinear control* (Vol. 199, No. 1). New Jersey: Prentice hall.

[25] Fung, G. M., and Mangasarian, O. L. (2005). "Multicategory proximal support vector machine classifiers. *Machine Learning"*, *59*(1), 77-97.

[26] Huang, C. L., and Wang, C. J. (2006). "A GA-based feature selection and parameters optimization for support vector machines". *Expert Systems with applications*, 31(2), 231-240.

[27] Köknar-Tezel, S., and Latecki, L. J. (2011). "Improving SVM classification on imbalanced time series data sets with ghost points". *Knowledge and information systems*, *28*(1), 1-23.