DISCRETE CONTROL APPLIED TO A ROBOT END-EFFECTOR: COMPARISON OF TWO SOLUTIONS

Luís Fernando Ferreira Furtado
Instituto Tecnológico de Aeronáutica – ITA
furtado@ita.br

Guilherme Kisseloff Coracini
Instituto Tecnológico de Aeronáutica – ITA
guilherme.coracini@gmail.com

Rodrigo Pastl Pontes
Instituto Tecnológico de Aeronáutica – ITA
rpastl@gmail.com

Emilia Villani
Instituto Tecnológico de Aeronáutica – ITA
evillani@ita.br

Luís Gonzaga Trabasso
Instituto Tecnológico de Aeronáutica – ITA
gonzaga@ita.br

Abstract. *Actual researches developed by the aeronautic industries favours the use of COTS (commercial off-the-shelf) industrial robots as solution for the manufacturing process automation. The new robots applications require end-effectors developed to execute several tasks such as searching reference parts, clamping, drilling, inserting rivets etc., thus, prompting the project of multifunctional robot tools.*
*The FARE (**F**uselage **A**ssembly **R**obotic end-**E**ffector) is one example of this new generation of multifunctional robot tools. It was designed and constructed in order to automate the aircraft drilling, sealing and fastening operations using an industrial robot. FARE is already available and operational for experiments at ITA.*
This work compares two different control architectures approaches, created in the LabVIEW™ environment, and used in the FARE Fastening Module as a sample. The first control architecture is created with a Producer/Consumer structure that handles multiple processes at the same time; the second is built with programmable Statechart diagrams, based on finite state machines, that are also are suitable for multitasks.

Keywords: discrete control, producer/consumer, statechart, industrial robots, end-effector

1. INTRODUCTION

Automation by robots has been used in several segments of the manufacturing industry (Holland and Nof, 1999). Although the automotive industry stills keeping the first worldwide position of robot purchases (Minami, 2012), other industrial sectors prompt the development of new robot technologies.

According to Villani et al (2010), the aeronautic industries have interest to automatize some steps of the manufacturing process using COTS industrial robots due the flexibility and the low cost, compared to the automation solutions made of huge and jigged structures. As the drilling process are repeated so many times, the leading aircraft companies started to prompt the development of robot end-effectors to drill fuselage structures, as Dassaut (Costa 1996), Airbus (Cibiel and Prat, 2006), Boeing (Devlieg and Feikert, 2008) and Embraer (Eguti et al, 2012).

A common feature of the end-effectors used in the aeronautic industry is the multifunctionality. Generally, this kind of end-effector performs more than one task and allows the robot to have a number of functions without tool changes, increasing the process flexibility and reducing the process cycle time.

Focused on the development of automation technology with robots for aircraft manufacturing, the Brazilian aeronautic industry and the ITA started a project named ASA (**A**utomation of the **S**tructural **A**ssemby – AME Automação da Montagem de Estruturas) with the support of the Brazilian government, via FINEP.

The goal of the ASA project is: align two cylindrical fuselage sections and then, drill, seal and install fasteners to fix these fuselage sections as one. The FARE (**F**uselage **A**ssembly **R**obotic end-**E**ffector) is a subproject of ASA, aimed to the development of a multifunctional end-effector for this new generation of robot automation solutions designed to the aircraft industry.

The FARE is a mechatronic product composed of a set of modules that require mechanical, electrical and software integration. The objective of this work is compare two software solutions used to control and integrate one of the FARE modules. These software solutions are based on Producer/Consumer and Statechart diagrams, programmed in the LabVIEW^{TM} language, due all the FARE hardware are from National Instruments.

2. FARE

After the alignment of two cylindrical fuselage sections, the robot drives the FARE to a reference point, a vision system measure this reference and calibrates the robot position. Then, the robot moves to the position to drill, and the clamp module is activated in order to keep constant the contact between the robot and the fuselage, avoiding vibration during the drilling process and avoiding the accumulation of chips between two fuselage plates. After the clamp, the robot reads the FARE perpendicularity deviation and calibrates the orientation in order to make FARE orthogonal to the fuselage. Then, while the drilling system makes a countersunk hole, a fastener is sent to FARE and seal is applied on it. When the hole is finished, the fastener is inserted into the fuselage. Figure 1(a) shows the FARE project, Figure 1(b) shows the FARE after the construction and Figure 1(c) presents the FARE flowchart.

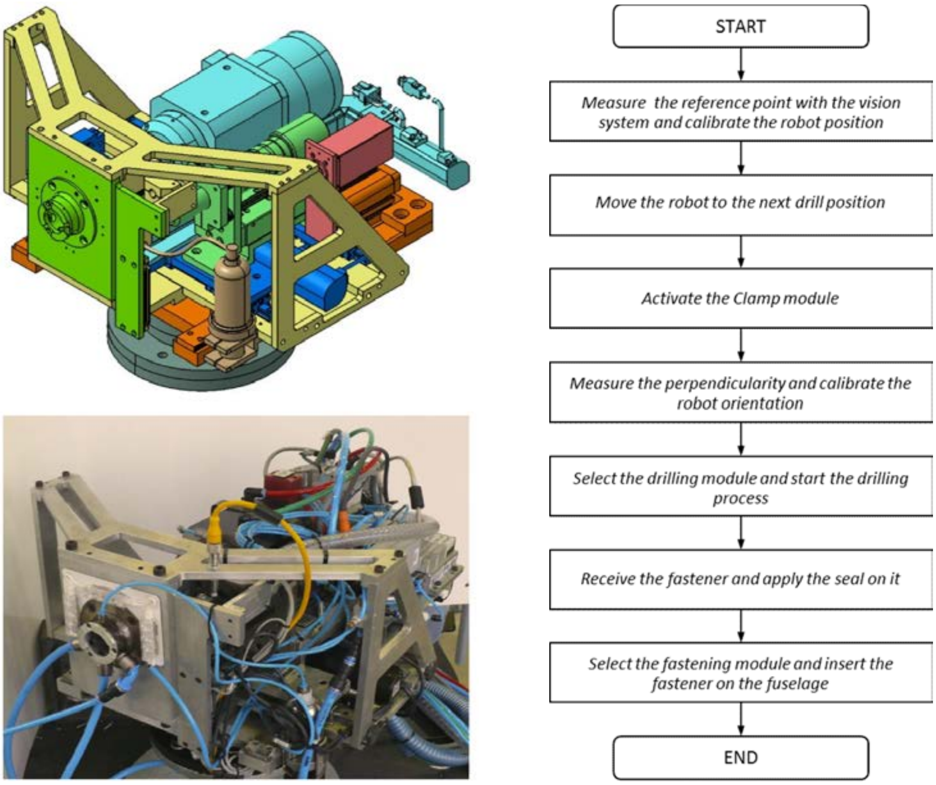

Figure 1. FARE (Fuselage Assembly Robotic end-Effector)

FARE is divided in seven modules: Clamp, Mechanical Platform, Perpendicularity, Vision, Drilling, Fastening and Sealant. However, only the modules Fastening and Sealant will be considered for the comparison proposed in this work, because these modules have a high level of integration between each other and, combined, they uses the majority of the digital inputs and outputs, implying in more complexity to the discrete control software.

2.1 Fastening and Sealant Module

The fastening module comprises a pneumatic fastener inserter (Fig. 6(c)) that is moved in the direction to the fuselage by a pneumatic actuator (Fig. 6(d)). An electric actuator (Fig. 6(g)) moves, transversely, the pneumatic inserter in two distinct positions: loading fastener and inserting fastener. There is also a system to send the fastener from a station in the ground to the FARE, installed on the robot flange. The sealant module is integrated to the fastening loading station on the FARE. The seal is applied and spread, pneumatically, on the fastener holder, that handles the fastener sent from the ground station.

Figure 2(a) presents the capsule to introduce the fastener on the ground station and Fig. 2(b) presents the valve that insert a high air flow to send the fastener from the ground station to the FARE, using a hose (Fig. 3(a)). Figure 3(b) and Fig. 6(e) present the pneumatic actuator that moves the fastener from the reception position (in front of the hose) to the loading position.

Figure 4(a), Fig. 5(a) and Fig. 6(a) present the fastener holder, that is installed on the pneumatic actuator (Figure 3(b) and Fig. 6(e)). The fastener holder has two small holes (Fig. 4(c)) that spread the seal on the contact surface of the fastener. This seal is sent from the sealant tank (Fig. 6(f)) to the fastener holder by a hose (Fig. 4 and Fig. 6(b)). Figure 5(b) presents the sensor that checks if the fastener came from the ground station.

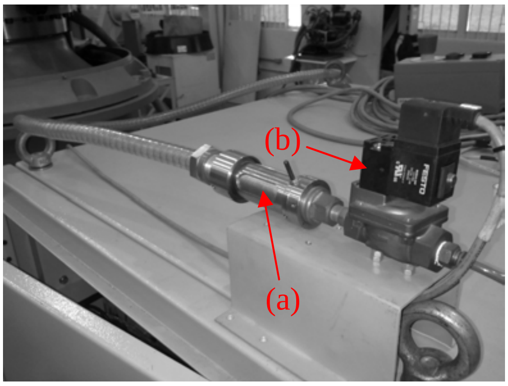

Figure 2. Ground station to send fasteners.

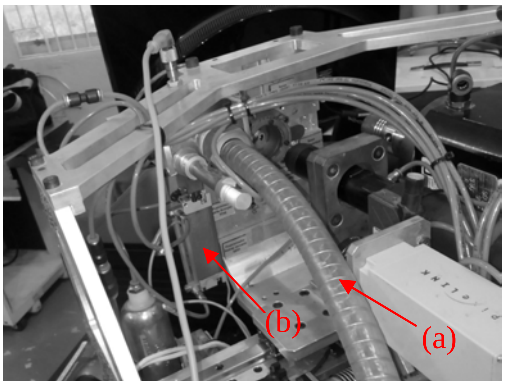

Figure 3. FARE and the hose for fasteners.

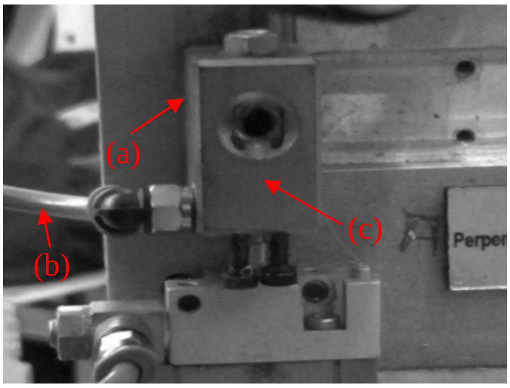

Figure 4. Fastener holder.

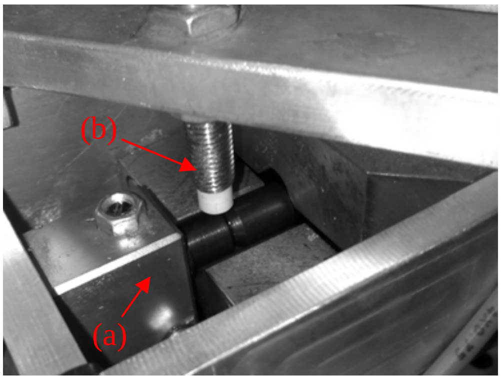

Figure 5. Fastener holder with the fastener.

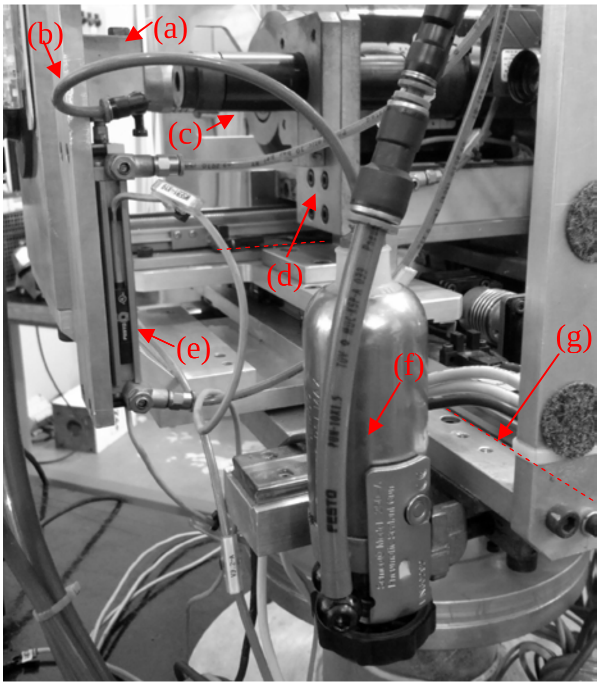

Figure 6. Sealant Module.

3. CONTROL ARCHITECTURES APPROACHES

This work proposes the use of two different architectures approaches, created using a graphic programming language named LabVIEW™, in order to control the FARE Fastening and Sealing modules. The first control architecture is developed using Producer/Consumer structures and the second architecture is built with programmable Statechart diagrams.

Table 1 presents the inputs and outputs lists of the FARE Fastening and Sealing modules, while Fig. 7 to Fig. 12 present the all the functional flowchart of these modules.

Table 1. I/O list of the Fastening and Sealing modules.

INPUTS		OUTPUTS	
Fastener present at the fastener holder	i0	o0	Send the fastener
Fastener holder at the receive position	i1	o1	Move the fastener holder to the receive position
Fastener holder at the delivery position	i2	o2	Move the fastener holder to the delivery position
Tool in the advanced position	i3	o3	Advance tool
Tool in the retracted position	i4	o4	Retreat tool
Tool in the supply position	i5	o5	Move tool to the supply position
Tool in the insertion position	i6	o6	Move tool to the insertion position
Signal from other modules: movements enabled	i7	o7	Switching on the sealant injector
		o8	Switching on the tool

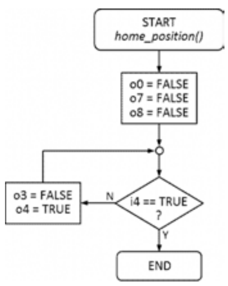

Figure 7. Home position flowchart.

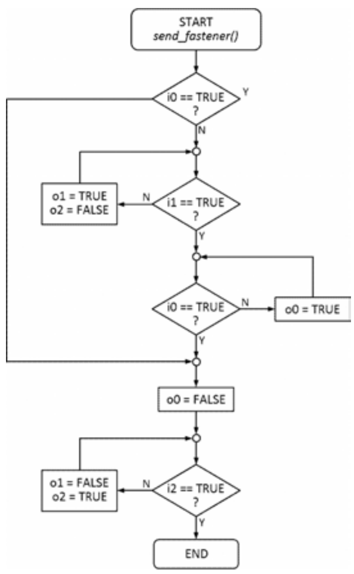

Figure 8. Send fastener flowchart.

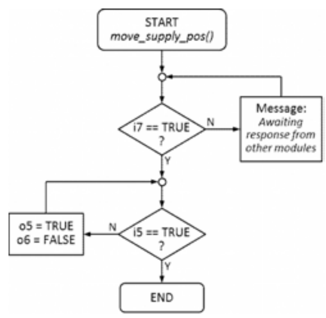

Figure 9. Move fastener inserter to the supply position flowchart.

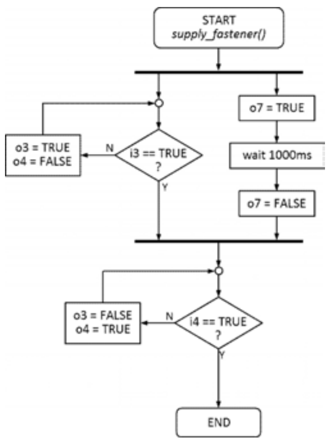

Figure 10. Supply the fastener on the inserter flowchart.

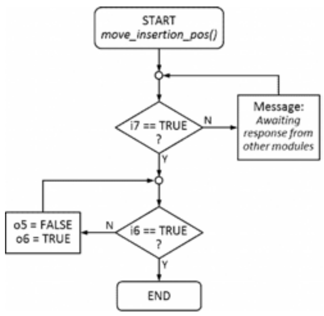

Figure 11. Move fastener inserter to the insertion position flowchart.

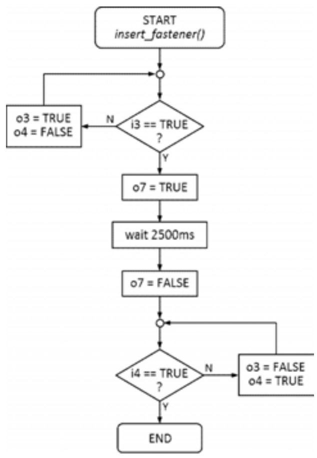

Figure 12. Insert fastener flowchart.

However, to compare two different approaches for all the flowcharts presented so far is too much complex. In order to simplify the comparison, only the main program will be considered. The subprograms (Fig. 7 to Fig. 12) could be created in any kind of programming language, on condition that they can be used by LabVIEW™ on the main program.

Figure 13 present the main program sequence to be created in LabVIEW™. This main program, named main_1(), uses the subprograms (Fig. 7 to Fig. 12) to insert the fastener in the fuselage.

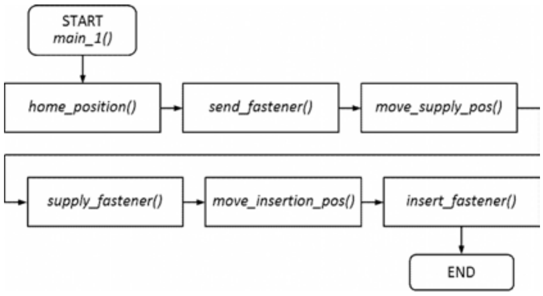

Figure 13. Main programs main_1.

3.1 Producer/Consumer technic

The Producer/Consumer architecture is based on Master/Slave architecture and allows the data sharing between multiple loops running at individual rates (Anjos, 2010). According to National Instruments (2013A), the Produter/Consumer is unique due the buffered communication between loops that makes possible the connection between application processes running at different speeds.

The Produces/Consumer is made of, at least, two parallel loops, divided into two categories: producers and consumers. The communication between producer and consumer loops is made using data queues (National Instruments, 2013A).

Figure 14 presents an example of the Producer/Consumer architecture created in LabVIEW™. The Enqueue/Dequeue Element is the buffer that connects the producer and the consumer.

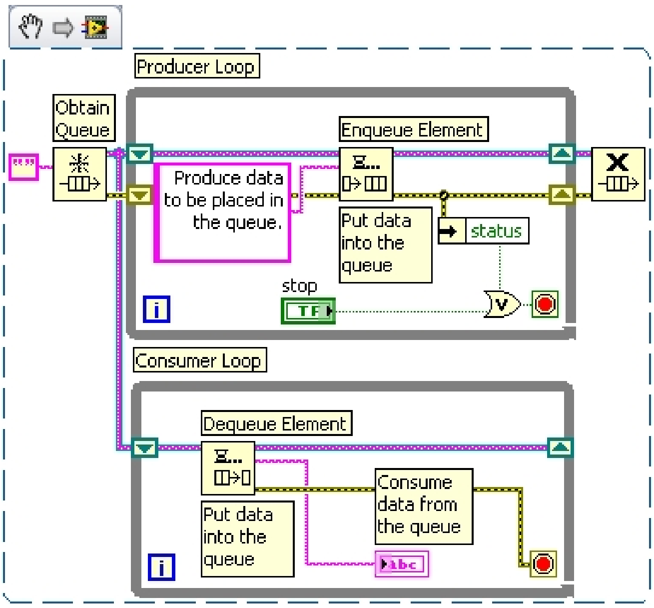

Figure 14. Producer/Consumer architecture example (National Instruments, 2013A).

3.2 Producer/Consumer applied to FARE

Figure 15 presents the Producer/Consumer architecture used to control the Fastening Module and Sealant Module. As shown in Fig. 14, there are two different loops: the producer, on the top, that reads the sensors and refresh the variables to the actuators; and the consumer, on the bottom that control the system sequence.

The consumer is created in a tab control element and each page have controls one step of the cycle.

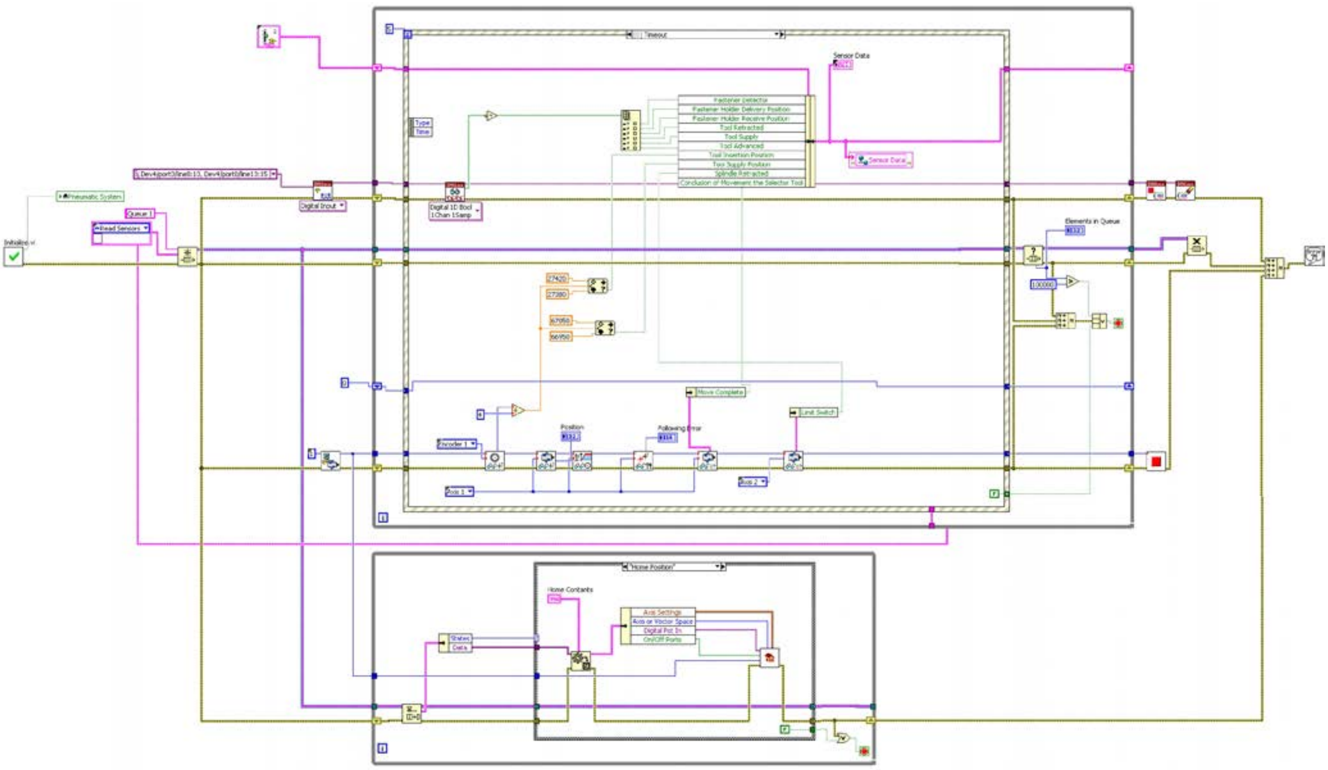

Figure 15. Producer/Consumer architecture used in the FARE control software.

3.3 Statechart technic

The Statechart from National Instruments is an add-on module for LabVIEW™ based on Unifed Modeling Language (UML). According to Alhir (2003), UML is a visual language for modeling and communicating about systems through the use of diagrams and supporting text. The diagrams are suitable for systems that execute some sequential tasks.

A classic state diagram has two main elements: states and transitions (National Instruments, 2013B). Statechart for LabVIEW™ offers tools to create states and transitions fully integrated with the other LabVIEW™ tools. It is possible also create states hierarchy, such as a sequence of events and transitions inside an event.

Figure 16 presents an example with tree main events (Idle, Vending Superstate and Give Change) and two sub-events (Count Coins and Dispense). The lines that connect these events are the transitions. It is possible to trigger functions when an event occurs or creates guards and make the sequences wait some conditions.

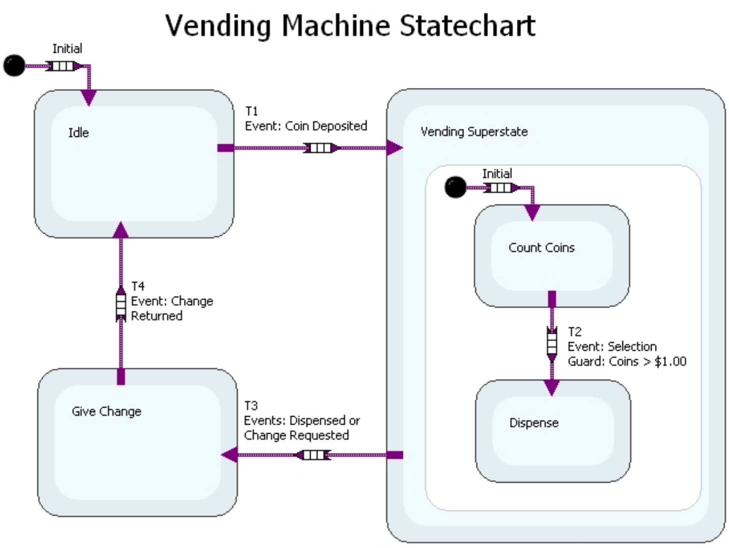

Figure 16. Statechart architecture example (National Instruments, 2013B).

3.4 Statechart applied to FARE

Figure 17 presents the control architecture developed using Statechart to control the Fastening Module and Sealant Module.

Statechart for LabVIEW™ separates the data acquisition and the sequence control into two different visual programming environments. The first one (Figure 17(a)) deals with sensors, actuators and with the information flow to the sequence block made using the Statechart diagrams (Figure 17(b)).

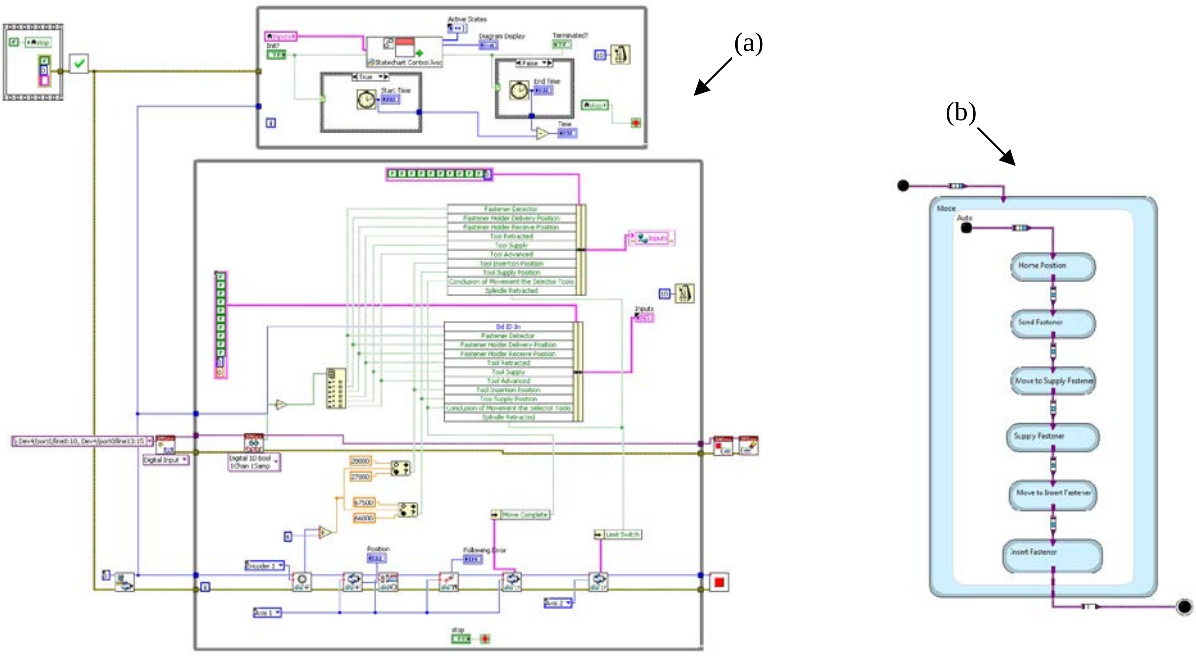

Figure 17. Statechart architecture used in the FARE control software.

4. COMPARING PRODUCER/CONSUMER AND STATECHART

Both control architectures used in this work meet the functional requirements of the Fastener and Sealing modules. Theoretically, the time of execution of both systems is the same. Small deviations of the controller's time should be not considered, because the control triggers the events too much faster than the time elapsed to perform these events. The time is based on the displacement of mechanisms, the travel of the fastener from the ground station to the FARE and the time controlled events, such as the activation time of the seal valve. However, ten sequences will be executed to each kind of control and the average time of execution will be calculated.

The comparison is based on the size of the final programs or on answers of some systematic questions about the programming method.

4.1 Comparison method

The method of comparison is divided into two different lists of questions. Table 2 presents questions to compare the architectures Producer/Consumer and Statechart, pointing what each control architecture can do.

Table 2. Questions to compare the control architectures.

Question number	Question
1	The control architecture used allows the creation of an automatic mode?
2	The control architecture used allows the creation of a manual mode?
3	It is possible to add some features in a finished program or it is necessary to start to programing over again?
4	What is the size, in kbytes, of the final file?
5	How many LabVIEW™ blocks were used?
6	How many LabVIEW™ connection lines were used?

Copyright © 2014 by ABCM

7	How long is the entire cycle execution?

4.2 Results

The results of the experiments are related to the questions shown in Table 2. The answers for these questions are presented side by side, comparing the Producer/Consumer and the Statechart architectures, in Table 3.

Table 3. Answers and comparison of the control architetures.

Question number	Producer/Consumer results	Statechart results
1	Yes	Yes
2	Yes	Yes
3	It is possible to add features in a program already done without start to programming a new program.	It is possible to add features in a program already done without start to programming a new program. After the modifications, the program must be recompiled.
4	79kB	230kB
5	about 45 blocks	about 20 blocks
6	about 300 lines	about 70 lines
7	17971 ± 30 ms	17582 ± 30 ms

Both architectures, Producer/Consumer and Statechart, can handle the control for automatic and manual cycles. In all cases, it is possible to make modifications on the program that is already done. After each modification, the Statechart requires a recompilation that is easy and fast to do.

The Statechart requires less blocks and lines to be created, but the final program, with 230 kB, is bigger than the program created using the Producer/Consumer (79 kB).

The difference around 390 ms between the execution time of the same algorithm for the same plant with Producer/Consumer and Statechart was not expected. The entire control routine was executed 2.2% faster in Statechart than in Producer/Consumer architecture.

5. CONCLUSION

Two discrete control architectures were used in the main program of two modules of a multifunctional end-effector named FARE. These two modules are the Fastening Module and Sealant Module.

Due the complexity of these modules, only the main program was compared. The subprograms were made using other kind of process compatible to LabVIEW^{TM}.

Both modes, manual and automatic, can be created using Producer/Consumer and Statechart. However, even using more blocks and lines, the Producer/Consumer architecture generates files smaller than the Statechart. Statechart uses fewer blocks and lines, because a number of logics and connections are generate automatically after the compilation of the program, what increases the size of the file.

A not expected difference is the execution time of both methods. Statechart was almost 0,4 s faster than the Producer/Consumer. However, to investigate the origin of this difference, more experiments should be done.

At the end, the Producer/Consumer and Statechart architectures presented satisfactory results and the Fastener Module and the Sealing Module of FARE were well controlled.

6. REFERENCES

Alhir, S. S., 2003, "Learning UML", O'Reilly Media, 1st ed. ISBN-13: 978-0596003449

Anjos, J. M. S., 2010, "Proposta de arquitetura de software de controle para efetuador robótico muntifuncional", Dissertation (Master Degree) – Department of Mechanical Engineering, (ITA) Aeronautics Institute of Technology, São José dos Campos, Brazil.

Cibiel, C. and Prat, P., 2006, "Automation for the Assembly of the Bottom Wing Panels on Stringers for the A320", In: *SAE World Congress*. Detroit, USA. SAE2006-01-3143.

Costa, S., 1996, "Dassault adaptive cells", *Industrial Robot*: An International Journal, v. 23, n. 1, 34–40.

Devlieg, R. and Feikert, E., 2008, "One-Up Assembly with Robots", In: *SAE World Congress*. Detroid, USA. SAE2008-01-2297.

Copyright © 2014 by ABCM

Eguti, C., Trabasso, L., Villani, E., Coracini, G. and Furtado, L. F. F., 2012, "Development of a Robotic End-Effector of Drilling and Fasteners Inserter for Aircraft Structures", In: *SAE 2012 Aerospace Manufacturing and Automated Fastening Conference*. Fort Worth, TX, USA. SAE2012-01-1858, doi:10.4271/2012-01-1858.

Holland, S. W. and Nof, S. Y., 1999, "*Handbook of industrial robotics*: emerging trends and industry needs", 2. ed. New York: John Wiley & Sons. 1378 p. (ISBN 0-471-17783-0).

Minami, Y., 2012, "*Executive Summary*: World Robotics 2012 Industrial Robots", Available online on http://www.worldrobotics.org. Accessed in January 2013, pp. 8-18.

National Instruments. Application Design Patterns: Producer/Consumer. Available in: <http://www.ni.com/white-paper/3023/en>. Access in: May 16th 2013A.

National Instruments. Application Design Patterns: Producer/Consumer. Available in: < http://www.ni.com/white-paper/6194/en>. Access in: May 16th 2013B.

Villani, E., Sutério, R., Trabasso, L. G., Furtado, L. F. F., Alvarado, B. H. L. and Amorim, D. Y. K., 2010, "Avaliação metrológica de um robô industrial para montagem estrutural de aeronaves", *Sba Controle & Automação*, v.21, n.6, 634-646.

7. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.