Technical Report

Reporting CNR (full name and address):	Industrial Technology and Automation Institute - (ITIA) Viale Lombardia, 20/A I-20131 Milan (MI), Italy Tel.: (+39 02) 7064.3978 Fax: (+39 02) 7064.3915				
Responsible Person:	Emanuele Carpanzano				
Project Name:	Studio, progettazione ed implementazione del sistema di controllo e supervisione del sistema di movimentazione e trasporto ad architettura molecolare. *Parte seconda*.				
Contract:	Contratto di Ricerca tra Torielli rag. Pietro & C. S.p.A. e Consiglio Nazionale delle Ricerche (Istituto di Tecnologie Industriali ed Automazione) stipulato in data 16 Gennaio 2002				
Document Title:	Transport Cell Control System Strategy				
Document Ref.No.:	2MaCS/TR-03/CS100			**Page:**	1
Date of issue:	21.02.2003	**Classification:**	Confidential	**No. of pages:**	29
Client(s):	Torielli				
Author(s):	F. Airoldi, G. Carlino, E.Carpanzano, A. Cataldo, (ITIA)				
Distribution:	Torielli, ITIA-CNR				
Keywords:	Flexible Manufacturing Plant, Modular Control System Architecture, Simulation, IEC 61131-3° Standard				

Summary:

The productive plant laid in Ozzero is characterized by an innovative molecular transport line system based on the integration of six single transport cells which constitute a continuous route along which the different shoes to be machined move from an operation station to another one. Each of these cells is constituted of a rotating table, a rotating island and a manipulator. The table has twelve slots and is used to move the shoes from a cell to another one. The island has twenty-four slots and is used to direct the shoes toward the different operation units where the shoes are machined. The manipulator has three arms and is used to move the shoes among tables and islands.

Objective of this report is to describe a control strategy to be applied to the handling system molecular transport line of the productive plant. In particular the conceived control strategy is restricted to a generic single transport cell. The modularity of the transport line allows using the same control strategy for all transport cells.

Purpose of this report is to document the activities developed by ITIA about the design of the control system strategy.

1 GENERALITIES 3
1.1. INTRODUCTION 3
1.2. OBJECTIVE 3
1.3. PURPOSE 3
2. CONTROL SYSTEM DESIGN 4
2.1. PRODUCTIVE PROCESS DEFINITION 4
2.1.1. Molecular Transport Line and Single Transport Cell 4
2.2. CONTROL SYSTEM SPECIFICATION 6
2.2.1. Model of the Generic Single Transport Cell 6
2.2.2. Model of the Last Transport Cell 8
2.2.3. Model of the Extended Single Cell 8
2.2.4. Extended Cell Pseudo Code 10
2.3. ARCHITECTURAL CONTROL SYSTEM DESIGN 12
2.3.1. Transport line 12
2.4. CONTROL SYSTEM FUNCTIONAL DESIGN 13
3. CONTROL SYSTEM CODE IMPLEMENTATION IN ISAGRAF 16
3.1. SUPERVISOR MODULE SFC CODE IMPLEMENTATION 16
3.2. CONTROLLER MODULE SFC CODE IMPLEMENTATION 18
4. CONTROL SYSTEM SIMULATION IN ISAGRAF 21
5. TRANSPORT LINE TECHNOLOGICAL CONSTRAINTS 23
5.1. MOVEMENTS OF PAIR OF FORMS 23
5.2. SEQUENTIAL MACHINING OPERATIONS ON ISLANDS 23
5.3. TABLES STEP BY STEP ROTATION 24
6. CONCLUSIONS AND FUTURE WORK 25
7. ACRONYMS AND DEFINITIONS 26
7.1. MUST-KNOW TERMS IN CONTROL AUTOMATION 26
7.2. ACRONYMS 27
8. AKNOWLEDGEMENT, REFERENCES AND LINKS 28
8.1. REFERENCES 28
8.2. INTERNET LINKS 28

1 GENERALITIES

1.1. INTRODUCTION

The productive plant laid in Ozzero is characterized by an innovative molecular transport line system based on the integration of six single transport cells which constitute a continuous route along which the different shoes to be machined move from an operation station to another one. Each of these cells is constituted of a rotating table, a rotating island and a manipulator. The table has twelve slots and is used to move the shoes from a cell to another one. The island has twenty-four slots and is used to direct shoes towards the different operation units where shoes are machined. The manipulator has three arms and is used to move the shoes among tables and islands.

1.2. OBJECTIVE

Objective of this report is to describe a control strategy to be applied to the handling system molecular transport line of the productive plant (see Ref 1 for more details). In particular the conceived control strategy is restricted to a generic single transport cell. The modularity of the transport line allows using the same control strategy for the entire transport cells.

1.3. PURPOSE

Purpose of this report is to document the activities developed by ITIA about the design and simulation based verification of the control system strategy.

2. CONTROL SYSTEM DESIGN

2.1. PRODUCTIVE PROCESS DEFINITION

The process to be automated is described, and the activities to be performed and the objectives of the automation system are defined.

2.1.1. Molecular Transport Line and Single Transport Cell

In the Fig. 1 the integrated pilot plant layout is shown. In particular in this report attention is paid to the transport line, which has a molecular architecture. This means that the base element used to build up the integrated transport line is the single transport cell.

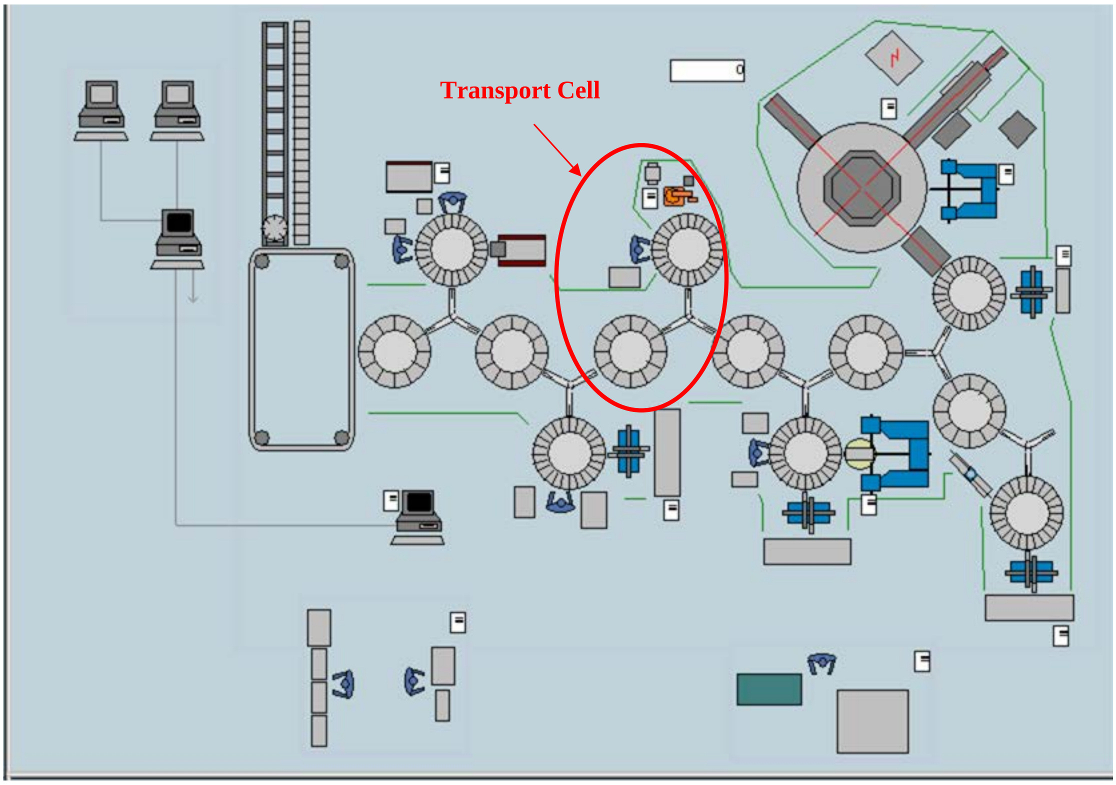


Fig. 1: Integrated Pilot Plant Layout

The single transport cell is built up from a rotating table, a rotating island and a manipulator, see Fig. 2.

Each table houses twelve slots on which the shoe trees can be placed. The stand alone shoe trees are called “bases” while the shoe trees which are going to support the semi-manufactured shoes are called “forms”. The tables are dedicated to the transport of the bases and forms.

Each island instead houses twenty-four slots. The islands are used to move the different forms from an operation station to another one.

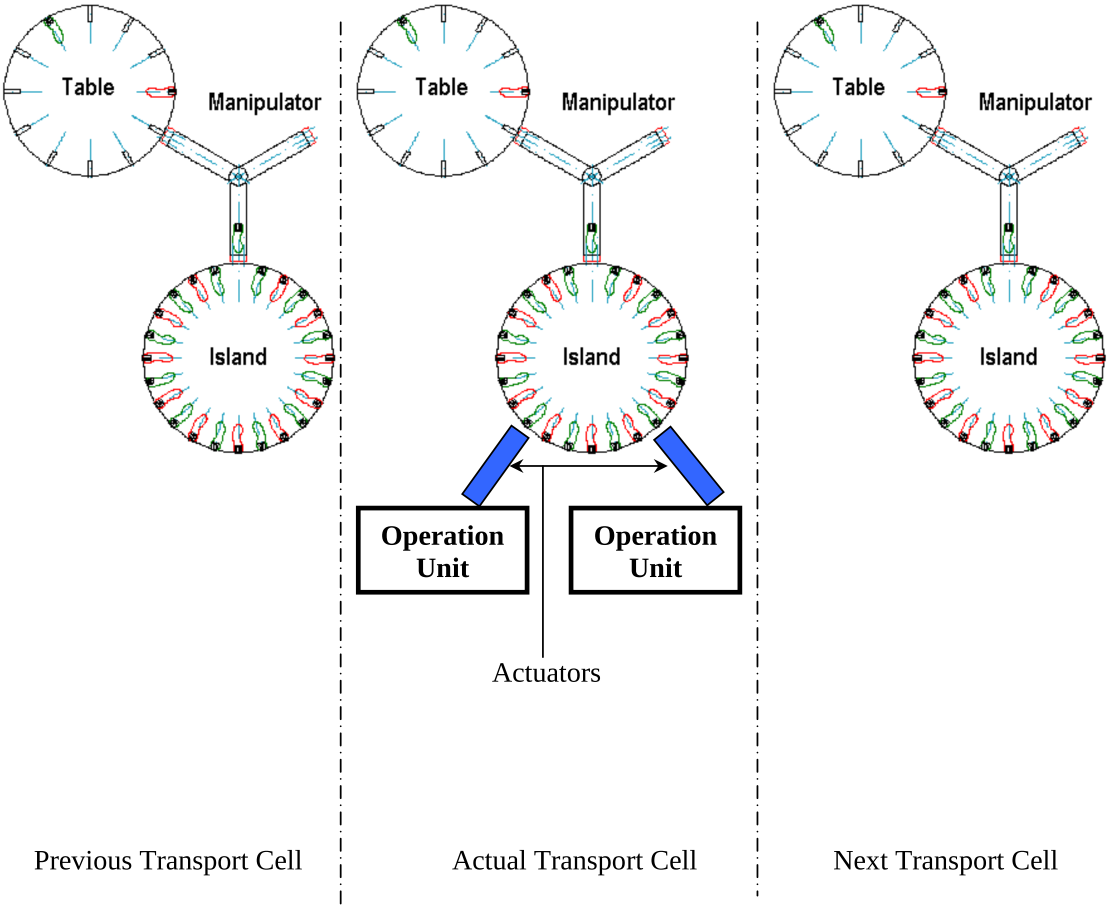


Fig. 2: Transport Cell

The basis and the forms are characterized by different transport philosophies.

The bases flow backward through the transport line because a generic form whose working process is over becomes a base that must be restored in the warehouse located at the beginning of the transport line. Therefore a base moves from the table of a cell to the table of the previous one. Into the last cell the movement occurs between the island (on which the base is laid after the shoe has been removed from the regarding form) to the table.

The forms flow forward along the transport line passing from an operation station to the following one, according to their working operation schedule. The operations to be executed respect a predefined sequence (for example the humidification and toe lasting operation must be brought before the dryer operation) that is obtained by means of an opportune plant layout.

Nevertheless the sequential order of operations doesn't exclude the possibility of overtaking among different forms during their build up process. In fact given that the different pair shoes to be produced can be different one from another (in terms of models, form, used materials, size, etc.) in general they are processed through different operations and characterized by different working operations and time schedules.

This implies that a form is characterized by three possible movements:

- From the table to another table when a form doesn't need to be machined from the operation stations laid in a specific cell

- From the table to the island when a form needs to be machined from the operation stations laid in a specific cell
- From the island to the table when the form machine operations in a specific cell are over.

All the base and form movements among tables and islands are carried out by the manipulators while the movements of the forms among the islands and the operation stations are performed by specific actuators (which are not taken into account in this work).

2.2. CONTROL SYSTEM SPECIFICATION

Starting from the productive process definition and from the customer requirements the control system specifications are formalized. The control system together with the productive process constitutes an automated system able to satisfy the productive process functionalities and requirements.

2.2.1. Model of the Generic Single Transport Cell

As said in the objective of this report the conceived control system philosophy regards a generic single transport cell.

The generic cell is shown in Fig. 3.

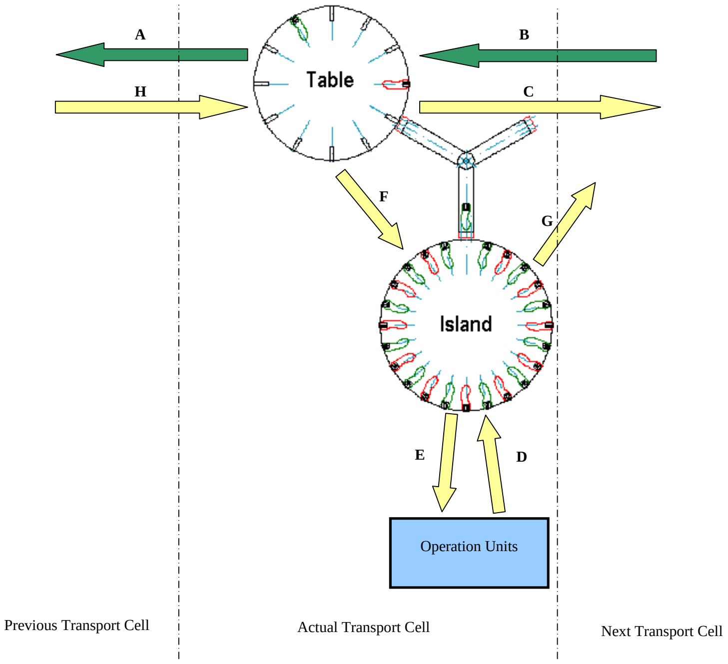


Fig. 3: Single Cell Control System Strategy

The possible movements that can be executed on bases and forms are formalized as “cases” characterized by letters:

A	“Base” from actual cell table to the previous cell table
B	“Base” from previous cell table into the actual cell table
C	“Form” from actual cell table to the next cell table
D	“Form” from Operation Unit to the actual cell island

E	“Form” from actual cell island to the Operation Unit
F	“Form” from actual cell table to the actual cell island
G	“Form” from actual cell island to the next cell table
H	“Form” from previous cell table to actual cell table

Moreover some of the presented movements can be executed simultaneously by co-ordinating the table, island and manipulator working functions. Therefore the following combined movements are formalized:

A+H, B+C, B+F, B+G, B+F+G, F+G, I+F.

Note: for simplicity only one operation station has been considered in the generic cell. This doesn’t modify the control system strategy concept because with more than one operation station it is necessary only to add further cases similar to D and E.

The basic control concept to be applied to each cell is to give priority to the bases backward flow through the transport line so to avoid its saturation, i.e. deadlocks. To do that it has been assumed that the maximum allowable number of forms laid on a table must be less or equal to eleven, keeping the twelfth slot free for the bases. In this way, by giving maximum priority to the movement A, a base finds always a free slot on a table to be placed on.

In practice more bases or forms that must be moved can be present in a cell at the same time. Therefore one of all the possible movements that can be executed must be selected and carried out. This selection is carried out according to the priorities formalized by means of the following Table 1.

In order to interpret in the right way the table, an example is given: case D (see regarding column) can be carried out only if cases BFG, BF and BG have not to be executed or have already been executed.

Table 1: Priority Logic for a Generic Cell

	AH	**A**	**BFG**	**BC**	**BF**	**BG**	**B**	**C**	**D**	**E**	**FG**	**F**	**G**	**H**
AH	-	X	X	X	X	X	X	X	-	-	X	X	-	X
A	-	-	X	X	X	X	X	X	-	-	X	X	-	N.O.
BFG	-	-	-	X	X	X	X	X	X	X	X	X	X	X
BC	-	-	-	-	X	X	X	X	-	-	X	X	X	X
BF	-	-	-	-	-	X	X	X	X	X	X	X	X	X
BG	-	-	-	-	-	-	X	X	X	X	X	X	X	X
B	-	-	-	-	-	-	-	N.O.	-	-	X	N.O.	N.O.	X
C	-	-	-	-	-	-	N.O.	-	-	-	X	X	X	X
D	-	-	-	-	-	-	-	-	-	X	X	X	X	-
E	-	-	-	-	-	-	-	-	-	-	X	X	X	-
FG	-	-	-	-	-	-	-	-	-	-	-	X	X	X
F	-	-	-	-	-	-	N.O.	-	-	-	-	-	N.O.	X
G	-	-	-	-	-	-	N.O.	-	-	-	-	N.O.	-	-
H	-	N.O.	-	-	-	-	-	-	-	-	-	-	-	-

N.O.= Never Occur

Note: the combined movements are characterized by a higher priority respect to the corresponding single ones. In fact if for example case A should have a higher priority than AH then case AH could never be executed. So for the cases BFG, BC, BF, BG with respect to B and for the case FG with respect to F.

In the table some cases have no meaning, for example the case F with respect to case B. In fact the combined movement BF is carried out which has a higher priority then both B and F.

2.2.2. Model of the Last Transport Cell

The last cell working function is slightly different from a generic one because some cases don’t occur (B, C and G) while a new case “I” must be taken into account, see Fig. 4. Such a movement springs from the removing shoe from the shoe tree operation (which can be considered like a specific operation station). The base laid on the table must be moved on the island so to flow backward to the warehouse.

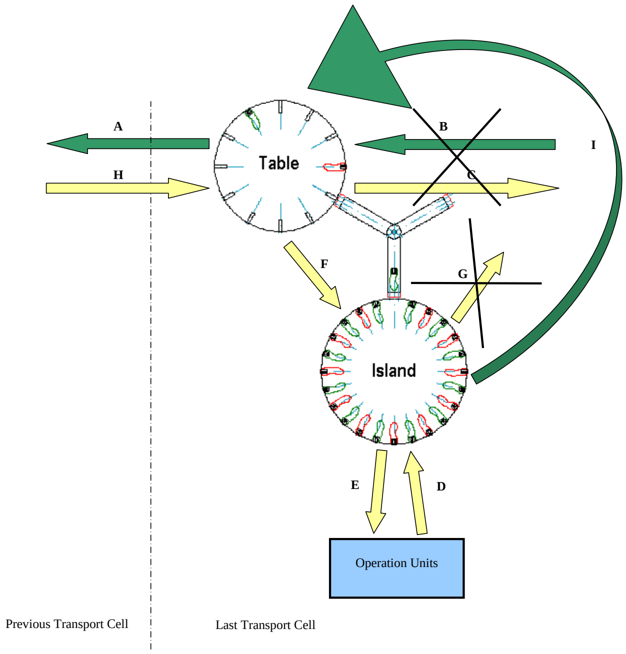


Fig. 4: Last Cell Control System Strategy

This new case “I” can be executed simultaneously with the case F by specifying a new combined movement “IF”.

In Table 2 the priority logic for the last cell is formalized.

Table 2: Priority Logic for the Last Cell

	AH	A	D	E	F	H	IF	I
AH	-	X	-	-	X	X	X	X
A	-	-	-	-	X	N.O.	X	X
D	-	-	-	X	X	-	-	-
E	-	-	-	-	X	-	-	-
F	-	-	-	-	-	X	-	N.O.
H	-	N.O.	-	-	-	-	-	-
IF	-	-	X	X	X	X	-	X
I	-	-	X	X	N.O.	X	-	-

N.O.= Never Occur

2.2.3. Model of the Extended Single Cell

As said above the last cell working function is slightly different from the previous five ones because the cases B, C and G never occur. Nevertheless it is possible to think about an extended cell, whose working function is compatible both with the generic cell one and with the last one, in which all possible cases A, B, C, D, E, F, G, H, I, AH, BFG, BC, BF, BG, FG and IF are represented, see Fig. 5.

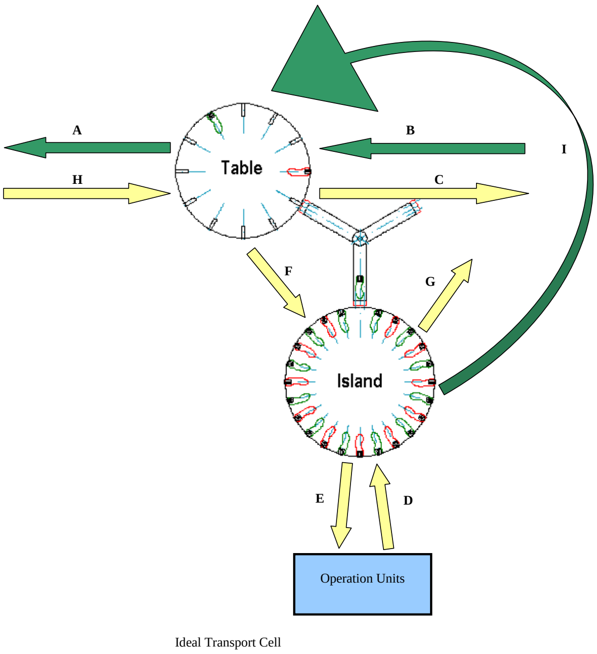


Fig. 5: Ideal Transport Cell Control System Strategy

In particular in a generic cell application the cases I and IF are never carried out. On the contrary in the last cell application the cases B, C, G and all the regarding combined ones are never executed. The main advantage of using the model of an extended cell consists in the possibility to have only one control system strategy for all the cells.

So a new priority logic table is obtained, see Table 3 that consists of the merging of the two tables presented above (see Table 1 and Table 2).

Table 3: Priority Logic for the Extended Cell

	AH	**A**	**BFG**	**BC**	**BF**	**BG**	**B**	**C**	**D**	**E**	**FG**	**F**	**G**	**H**	**IF**	**I**
AH	-	X	X	X	X	X	X	X	-	-	X	X	-	X	X	X
A	-	-	X	X	X	X	X	X	-	-	X	X	-	N.O.	X	X
BFG	-	-	-	X	X	X	X	X	X	X	X	X	X	X	X	X
BC	-	-	-	-	X	X	X	X	-	-	X	X	X	X	X	X
BF	-	-	-	-	-	X	X	X	X	X	X	X	X	X	X	X
BG	-	-	-	-	-	-	X	X	X	X	X	X	X	X	X	X
B	-	-	-	-	-	-	-	N.O.	-	-	X	N.O.	N.O.	X	X	X
C	-	-	-	-	-	-	N.O.	-	-	-	X	X	X	X	-	-
D	-	-	-	-	-	-	-	-	-	X	X	X	X	-	-	-
E	-	-	-	-	-	-	-	-	-	-	X	X	X	-	-	-
FG	-	-	-	-	-	-	-	-	-	-	-	X	X	X	-	-
F	-	-	-	-	-	-	N.O.	-	-	-	-	-	N.O.	X	-	N.O.
G	-	-	-	-	-	-	N.O.	-	-	-	-	N.O.	-	-	-	-
H	-	N.O.	-	-	-	-	-	-	-	-	-	-	-	-	-	-
IF	-	-	-	-	-	-	-	X	X	X	X	X	X	X	-	X
I	-	-	-	-	-	-	-	N.O.	X	X	X	N.O.	X	X	-	-

N.O.= Never Occur

It must be highlighted that in this extended cell a new combination of two single movements I and C could be considered as a combined movement IC. Nevertheless such movement will never be executed neither by the generic cell nor by the last one. In fact the generic cell will never execute the case I while the last cell will never execute the case C. Therefore this possible combined movement is not taken into account in the extended cell control system strategy and so it is not considered in the extended cell priority logic table.

2.2.4. Extended Cell Pseudo Code

In this paragraph the pseudo code which implements the extended cell priority logic is shown. Such a pseudo code represents a further step towards the formalization of the extended cell control system strategy.

```
If ( $AH$ ) {
    Execute AH;
}
If ( $A \& \overline{AH}$ ) {
    Execute A;
}
If ( $BFG \& \overline{AH} \& \overline{A}$ ) {
    Execute BFG;
}
If ( $BC \& \overline{AH} \& \overline{A} \& \overline{BFG}$ ) {
    Execute BC;
}
If ( $BF \& \overline{AH} \& \overline{A} \& \overline{BFG} \& \overline{BC}$ ) {
    Execute BF;
}
If ( $BG \& \overline{AH} \& \overline{A} \& \overline{BFG} \& \overline{BC} \& \overline{BF}$ ) {
    Execute BG;
}
If ( $B \& \overline{AH} \& \overline{A} \& \overline{BFG} \& \overline{BC} \& \overline{BF} \& \overline{BG}$ ) {
    Execute B;
}
If ( $C \& \overline{AH} \& \overline{A} \& \overline{BFG} \& \overline{BC} \& \overline{BF} \& \overline{BG} \& \overline{IF} \& \overline{I}$ ) {
    Execute C;
}
If ( $D \& \overline{BFG} \& \overline{BF} \& \overline{BG} \& \overline{IF} \& \overline{I}$ ) {
    Execute D;
}
If ( $E \& \overline{BFG} \& \overline{BF} \& \overline{BG} \& \overline{D} \& \overline{IF} \& \overline{I}$ ) {
    Execute E;
}
If ( $FG \& \overline{AH} \& \overline{A} \& \overline{BFG} \& \overline{BC} \& \overline{BF} \& \overline{BG} \& \overline{B} \& \overline{C} \& \overline{D} \& \overline{E} \& \overline{IF} \& \overline{I}$ ) {
    Execute FG;
}
If ( $F \& \overline{AH} \& \overline{A} \& \overline{BFG} \& \overline{BC} \& \overline{BF} \& \overline{BG} \& \overline{C} \& \overline{D} \& \overline{E} \& \overline{FG} \& \overline{IF}$ ) {
    Execute F;
}
If ( $G \& \overline{BFG} \& \overline{BC} \& \overline{BF} \& \overline{BG} \& \overline{C} \& \overline{D} \& \overline{E} \& \overline{FG} \& \overline{IF} \& \overline{I}$ ) {
    Execute G;
}
If ( $H \& \overline{AH} \& \overline{BFG} \& \overline{BC} \& \overline{BF} \& \overline{BG} \& \overline{C} \& \overline{FG} \& \overline{F} \& \overline{IF} \& \overline{I}$ ) {
    Execute H;
}
If ( $IF \& \overline{AH} \& \overline{A} \& \overline{BFG} \& \overline{BC} \& \overline{BF} \& \overline{BG} \& \overline{B}$ ) {
    Execute IF;
}
If ( $I \& \overline{AH} \& \overline{A} \& \overline{BFG} \& \overline{BC} \& \overline{BF} \& \overline{BG} \& \overline{B} \& \overline{IF}$ ) {
    Execute I;
}
```

2.3. ARCHITECTURAL CONTROL SYSTEM DESIGN

The different control system functional modules are identified by specifying for each one the regarding input/output signals and the implemented functionalities, as well as the hierarchical relationships among the modules themselves.

2.3.1. Transport line

With reference to the transport line the architectural control system design can be defined as shown in Fig. 6.

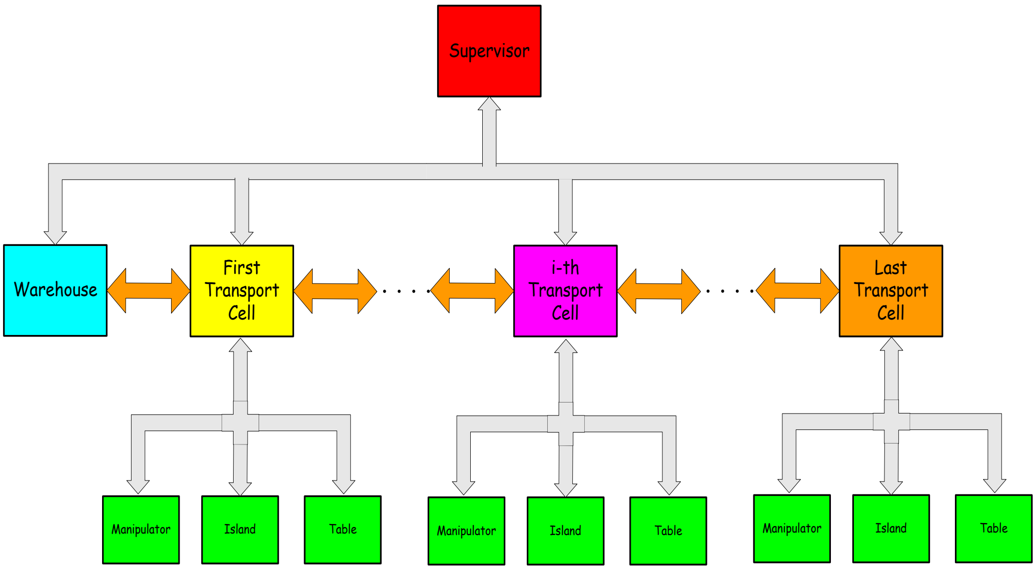


Fig. 6: Transport Line Architectural Control System Design

A supervisor system is placed at the top of the automation system to monitor the whole transport line and, if necessary, to force remote control actions in case the operator needs to drive the plant manually.

In case of plant automatic working function the supervisor, which schedules the operations to be executed on each form, communicates to the first cell all the route that a specific form must follow. Such route is in practice translated into movements to be executed by the different cells. The information associated to the route is assumed to be transmitted by a cell to the other during the form movement.

Each single cell control system, according to the priority logic, implements the requested movements (by means of the regarding table, island and manipulator and by coordinating itself with the control systems of the adjacent cells) and updates the position of the bases and forms laid in the regarding cell, that is available at the supervisor system so that the operator knows the actual situation of the plant.

2.4. CONTROL SYSTEM FUNCTIONAL DESIGN

The internal structures of the cell functional modules are defined in detail, by structuring them in one or more further functional modules, according to their complexity, and by specifying for each module the input/output signals and the associated functions. In this phase for each functional block the control algorithms which realize the different working functions and the data used by the control algorithms are defined. In order to do that the control system logic functions have been described by means of the SFC language included in the IEC 61131 part 3 standard.

The functional block that formally describes the extended cell control system strategy, which can be applied to all transport cells, is shown in Fig. 7. In particular for the generic tern i, a *supervisor* module and a *controller* module can be distinguished. The first one plans the next action to be executed according the values of its input and output signals, while the second one executes the algorithm corresponding to the specific action to be performed.

In Fig. 8 the connections among the control system functional blocks of more adjacent terns are represented. Notice that only the functional blocks of adjacent terns are connected, and that such connections involve only the supervisor modules of adjacent terns.

In Fig. 9 the functional blocks for the control of *table*, *island* and *manipulator* of a generic tern i are introduced as well as their connections with the other modules, in particular they are connected to the tern controller module.

In detail, the *supervisor module* implements the priority logic defined in Table 3. Therefore its input and output signals are evaluated in order to define which cases have to be considered among the ones associated to the rows and columns of Table 3. This is done by means of the following algorithm:

```
if [Base_back(i) = true] then [A = true];
if [Form_forward(i) = true and Form_on_the_table] then [C = true];
if [Form_forward(i) = true and Form_on_the_island] then [G = true];
…
```

Once the above algorithm has been executed, all possible actions to be executed are evaluated in order to establish which ones have to be executed according to the defined priority rules and to the available resources. Notice that more actions can be simultaneously executed if they don't use the same resources (i.e. table(i), island(i) and manipulator(i)).

The supervisor implementation can be effectively performed by defining more parallel SFC sequences, one for the starting of each possible action, whose evolutions are synchronised according to the priority rules given in Table 3 and to the availability of resources.

As for the *controller module*, this contains the algorithms that implement the different actions. Such an implementation can be performed by means of more SFC sequences executed in parallel, whose evolutions are controlled by the supervisor module. Notice that more SFCs may involve in parallel since more actions may be executed simultaneously.

Finally, the *table*, *island* and *manipulator* modules, implement the basic functions to control the rotations and base/form exchanges among such devices. Some more implementation details are given in the next section.

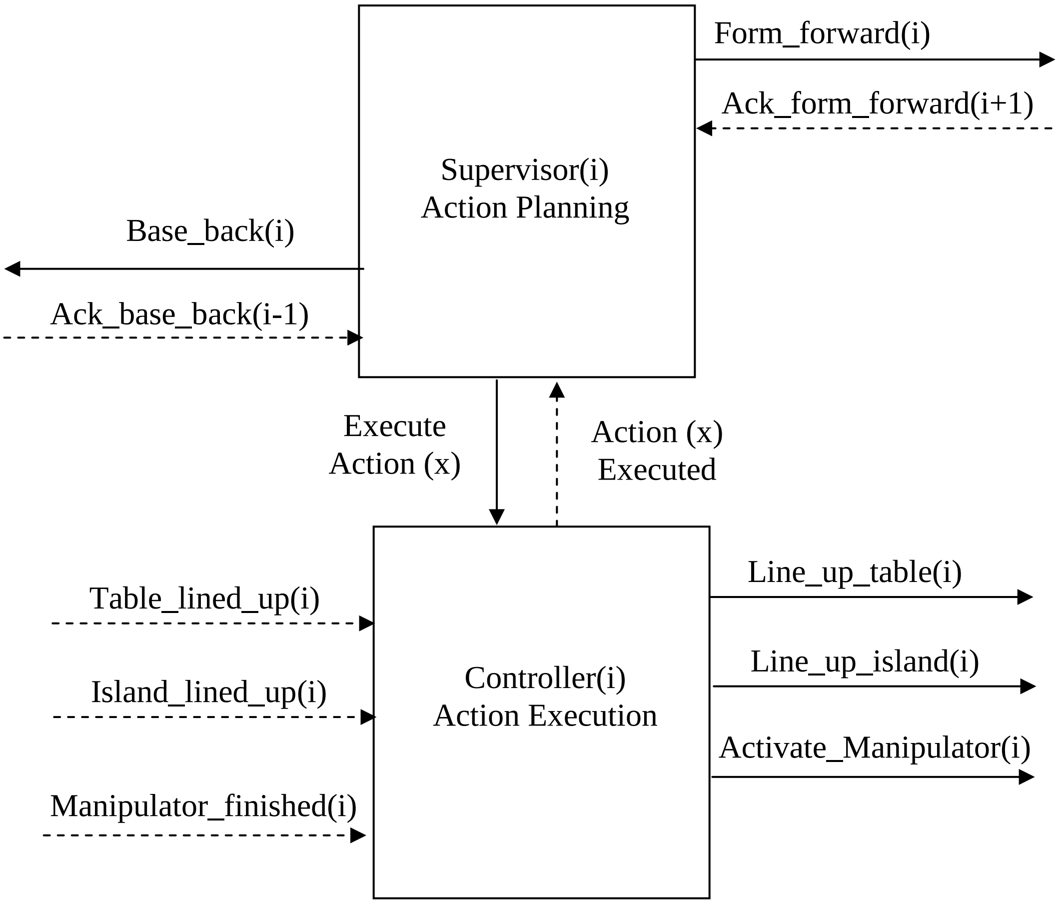


Fig. 7: Extended Cell Control System Functional Architecture

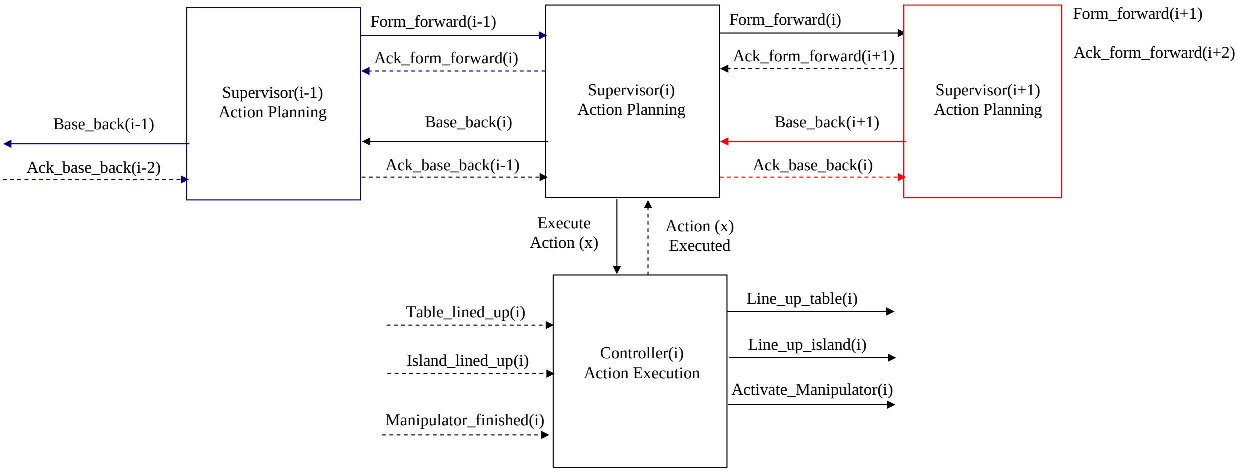


Fig. 8: Connections among Terns Functional Modules

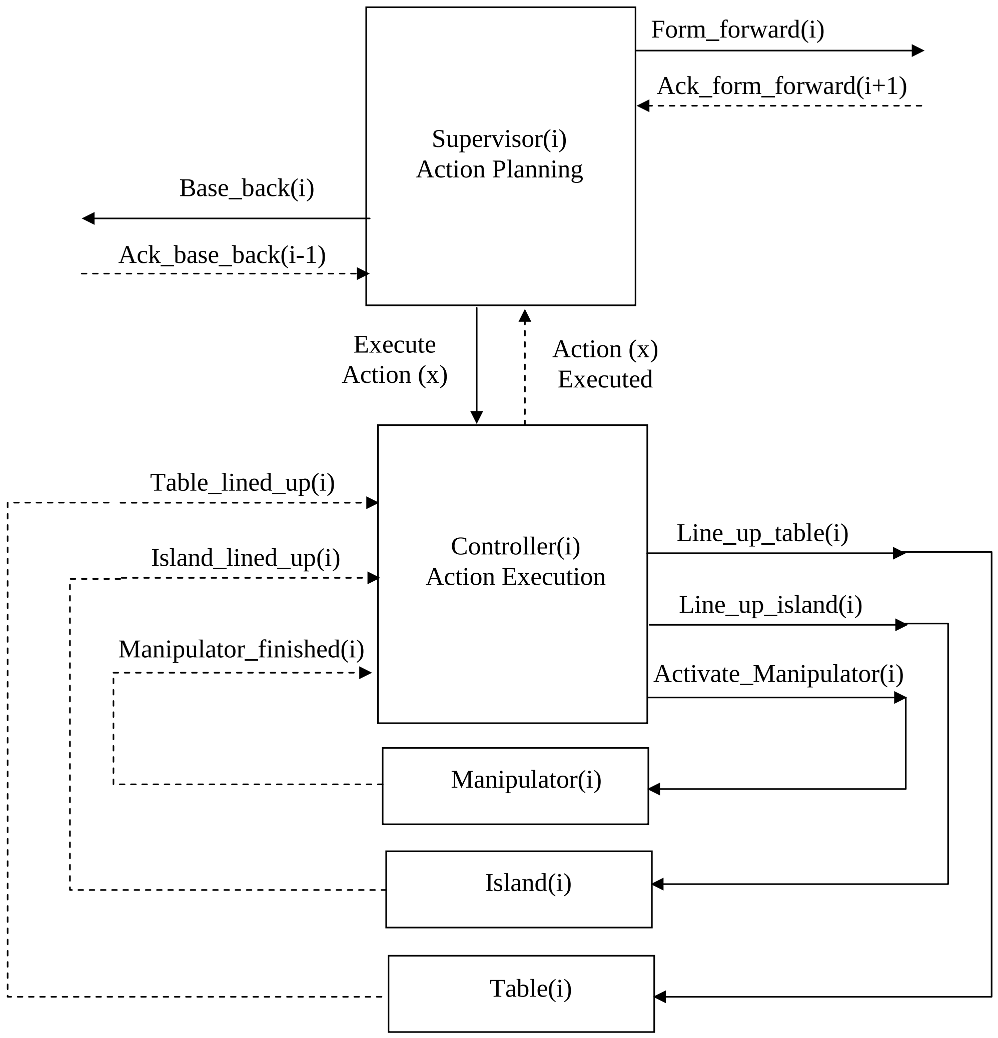


Fig. 9: Table, Island and Manipulator Functional Blocks

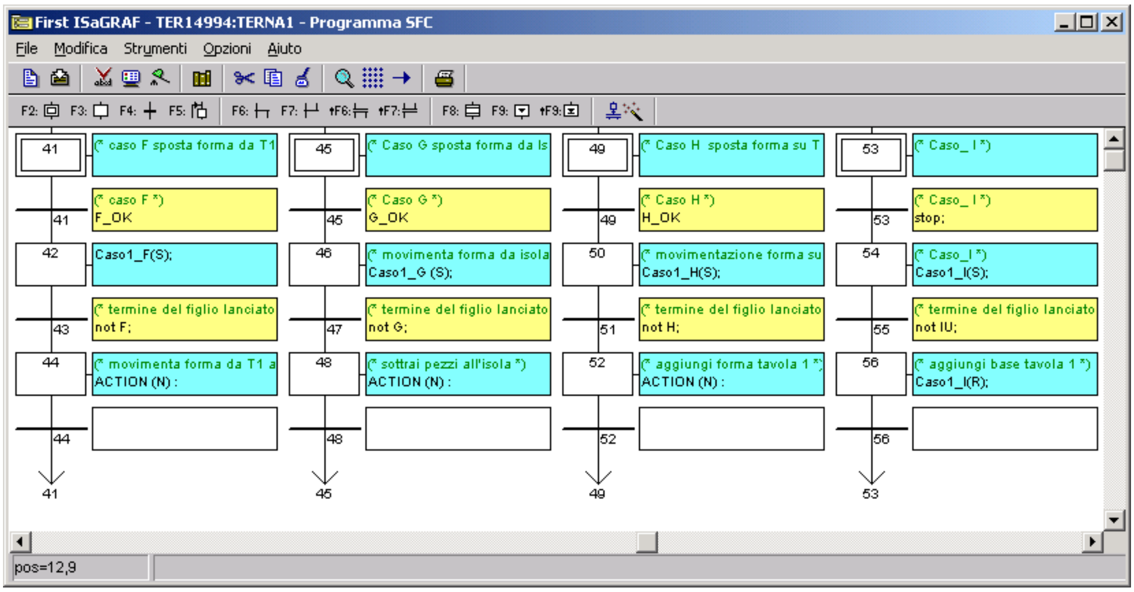
3. CONTROL SYSTEM CODE IMPLEMENTATION IN ISAGRAF

The control software code implementation in IsaGraf regarding the cell control system functional blocks is here discussed. In particular, the software coding is carried out by using the Sequential Functional Chart (SFC) language that is one of the five languages defined in the standard IEC 61131 part 3.

3.1. SUPERVISOR MODULE SFC CODE IMPLEMENTATION

A sketch of the tern supervisor module software implementation is illustrated in Fig. 10 in which the parallel SFC sequences, one for the starting of each possible action, are highlighted.

Each sequence is characterized by an own initial state that make the different sequences independent one from each other. An example of sequences is showed in Fig. 11 and Fig. 12.



```
(* caso F sposta forma da T1
(* caso F *)
F_OK
Caso1_F(S);
(* termine del figlio lanciato
not F;
(* movimenta forma da T1 a
ACTION (N) :
```

```
(* Caso G sposta forma da Is
(* Caso G *)
G_OK
(* movimenta forma da isola
Caso1_G (S);
(* termine del figlio lanciato
not G;
(* sottrai pezzi all'isola *)
ACTION (N) :
```

```
(* Caso H  sposta forma su T
(* Caso H *)
H_OK
(* movimentazione forma su
Caso1_H(S);
(* termine del figlio lanciato
not H;
(* aggiungi forma tavola 1 *)
ACTION (N) :
```

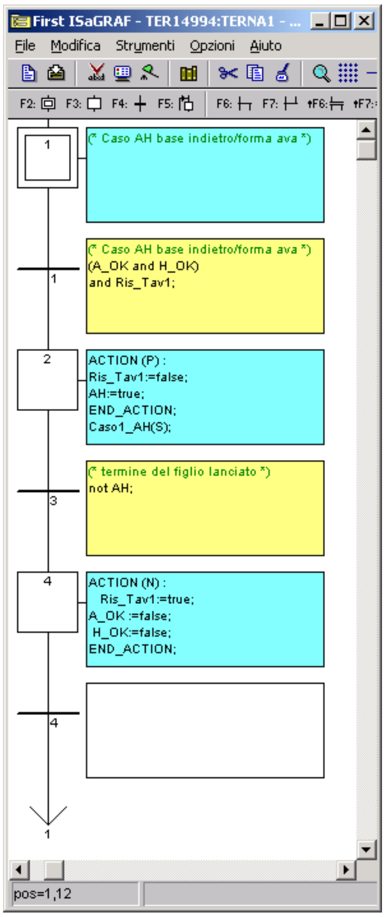
```
(* Caso_ I *)
(* Caso_ I *)
stop;
(* Caso_I *)
Caso1_I(S);
(* termine del figlio lanciato
not IU;
(* aggiungi base tavola 1 *)
Caso1_I(R);
```

Fig. 10: Supervisor Module SFC Code Implementation

Specifically, in Fig. 11 and Fig. 12 the sequences for the cases corresponding to actions AH and A are presented. The two sequences are similar, the only difference regards the name of the used variables.

In the first transition the priority rules given in Table 3 are implemented together with the evaluation of the resources availability (for the sequence AH the used resource is the table).

In the second step the control variables are set (in particular the used resource for this action is set to be not available) and the sequence used to execute all the operations of the action AH is activated. As soon as such sequence ends then the last step is executed in which the control variables are reset (in particular the resource used for this action is set to be available).



```
(* Caso AH base indietro/forma ava *)
```

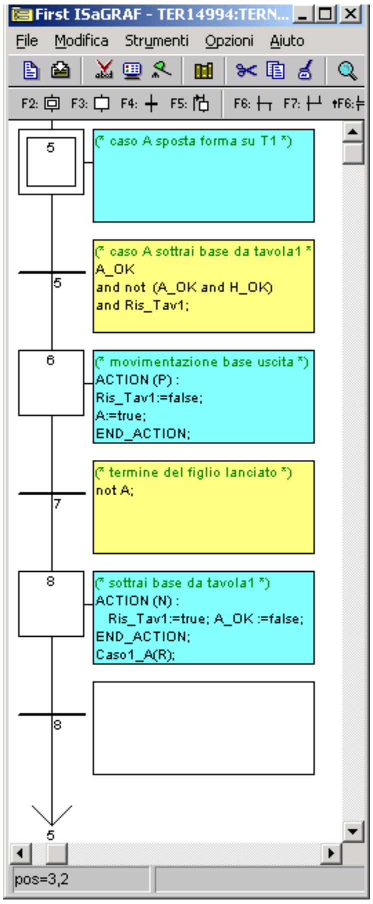
```
(* Caso AH base indietro/forma ava *)
(A_OK and H_OK)
and Ris_Tav1;
```

```
ACTION (P) :
Ris_Tav1:=false;
AH:=true;
END_ACTION;
Caso1_AH(S);
```

```
(* termine del figlio lanciato *)
not AH;
```

```
ACTION (N) :
   Ris_Tav1:=true;
A_OK :=false;
 H_OK:=false;
END_ACTION;
```

Fig. 11: SFC Sequence for case starting of action AH



```
(* caso A sposta forma su T1 *)
```

```
(* caso A sottrai base da tavola1 *
A_OK
and not (A_OK and H_OK)
and Ris_Tav1;
```

```
(* movimentazione base uscita *)
ACTION (P) :
Ris_Tav1:=false;
A:=true;
END_ACTION;
```

```
(* termine del figlio lanciato *)
not A;
```

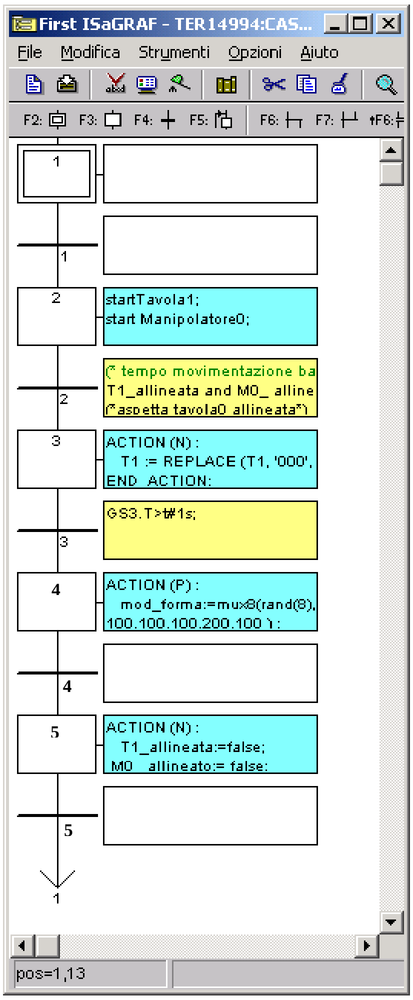
```
(* sottrai base da tavola1 *)
ACTION (N) :
  Ris_Tav1:=true; A_OK :=false;
END_ACTION;
Caso1_A(R);
```

Fig. 12: SFC Sequence for starting of action A

3.2. CONTROLLER MODULE SFC CODE IMPLEMENTATION

The typical structure of the controller module SFC code implementation is shown in Fig. 13 and Fig. 14, where actions AH and A SFC implementations are depicted as examples.

At first the rotation of the resources (table) involved in the operations is required. As soon as the resources are aligned, the manipulator is activated so to execute the exchange of the base/form. The position of the exchanged base/form is updated and the action SFC implementation ends resetting the control variables for a new action.



```
startTavola1;
start Manipolatore0;
```

```
(* tempo movimentazione ba
T1_allineata and M0_ alline
(*aspetta tavola0 allineata*)
```

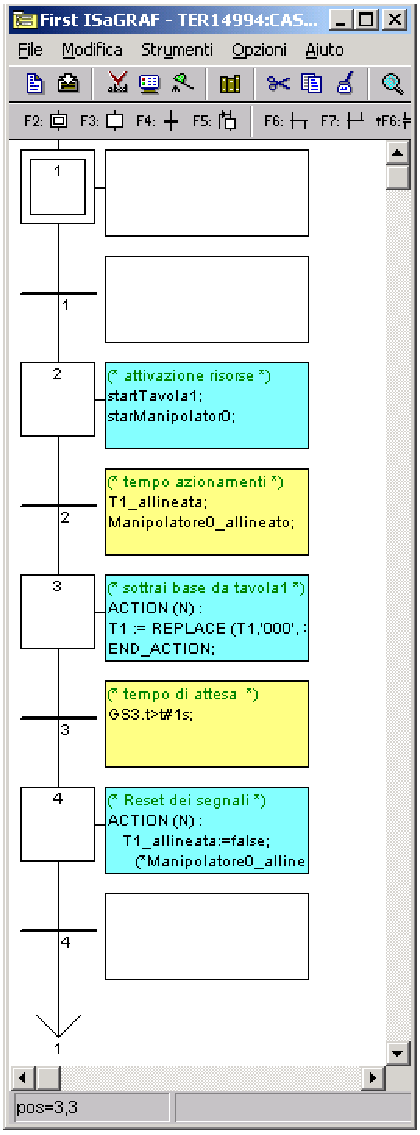
```
ACTION (N) :
   T1 := REPLACE (T1, '000',
END_ACTION;
```

```
GS3.T>t#1s;
```

```
ACTION (P) :
   mod_forma:=mux8(rand(8),
100,100,100,200,100 ) ;
```

```
ACTION (N) :
   T1_allineata:=false;
 M0   allineato:= false;
```

Fig. 13: Action “AH” SFC Implementation



```
(* attivazione risorse *)
startTavola1;
starManipolator0;
```

```
(* tempo azionamenti *)
T1_allineata;
Manipolatore0_allineato;
```

```
(* sottrai base da tavola1 *)
ACTION (N) :
T1 := REPLACE (T1,'000', :
END_ACTION;
```

```
(* tempo di attesa  *)
GS3.t>t#1s;
```

```
(* Reset dei segnali *)
ACTION (N) :
   T1_allineata:=false;
      (*Manipolatore0_alline
```

Fig. 14: Action “A” SFC Implementation

4. CONTROL SYSTEM SIMULATION IN ISAGRAF

The priority rules defined in Table 3 have been validated by means of simulations. Therefore a proper simulation model has been defined and run in ISAGRaF.

In Fig. 15 the tern control system software structure implemented in ISAGRaF is depicted. Each tern control system program is structured by means of a supervisor (terna0, terna1, etc.) and a number of sequences (Caso0_A, Caso0_BFG, etc.).

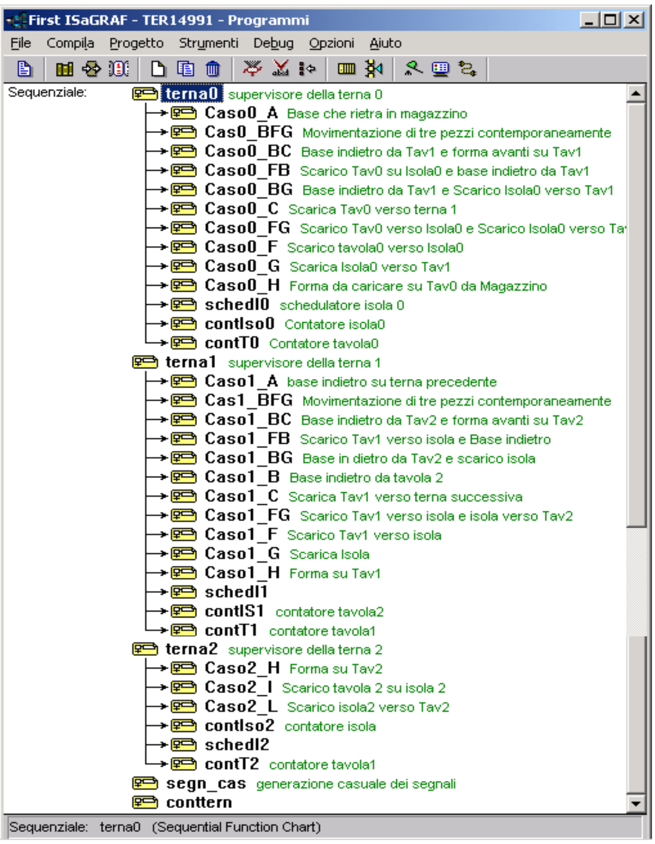
The supervisor implements the priority rules formalized in Table 3 and, once the action to be executed has been selected, starts the regarding sequence.

Each sequence executes all those operations needed to carry out the action selected by the supervisor.

In Fig. 16 the representation of three terns and a warehouse is depicted. At the beginning a form is picked up from the warehouse and left on the table of the first tern and, according to the shoe model to be produced, is addressed on the island of the same tern or to the next tern. The different machining operations to be done on each form are simulated by means of timers which bind the different forms to stay on the regarding island for a given interval time.

For each table and island a counter updates properly the number of forms and bases laid on it.

After the last machining operation has been performed the forms move backward to the warehouse thorough the tables.



Sequenziale:

- terna0 supervisore della terna 0
  - Caso0_A Base che rietra in magazzino
  - Cas0_BFG Movimentazione di tre pezzi contemporaneamente
  - Caso0_BC Base indietro da Tav1 e forma avanti su Tav1
  - Caso0_FB Scarico Tav0 su Isola0 e base indietro da Tav1
  - Caso0_BG Base indietro da Tav1 e Scarico Isola0 verso Tav1
  - Caso0_C Scarica Tav0 verso terna 1
  - Caso0_FG Scarico Tav0 verso Isola0 e Scarico Isola0 verso Ta
  - Caso0_F Scarico tavola0 verso Isola0
  - Caso0_G Scarica Isola0 verso Tav1
  - Caso0_H Forma da caricare su Tav0 da Magazzino
  - schedI0 schedulatore isola 0
  - contIso0 Contatore isola0
  - contT0 Contatore tavola0
- terna1 supervisore della terna 1
  - Caso1_A base indietro su terna precedente
  - Cas1_BFG Movimentazione di tre pezzi contemporaneamente
  - Caso1_BC Base indietro da Tav2 e forma avanti su Tav2
  - Caso1_FB Scarico Tav1 verso isola e Base indietro
  - Caso1_BG Base in dietro da Tav2 e scarico isola
  - Caso1_B Base indietro da tavola 2
  - Caso1_C Scarica Tav1 verso terna successiva
  - Caso1_FG Scarico Tav1 verso isola e isola verso Tav2
  - Caso1_F Scarico Tav1 verso isola
  - Caso1_G Scarica Isola
  - Caso1_H Forma su Tav1
  - schedI1
  - contIS1 contatore tavola2
  - contT1 contatore tavola1
- terna2 supervisore della terna 2
  - Caso2_H Forma su Tav2
  - Caso2_I Scarico tavola 2 su isola 2
  - Caso2_L Scarico isola2 verso Tav2
  - contIso2 contatore isola
  - schedI2
  - contT2 contatore tavola1
- segn_cas generazione casuale dei segnali
- contern

Sequenziale: terna0 (Sequential Function Chart)

Fig. 15: Supervisor Module Program Structure

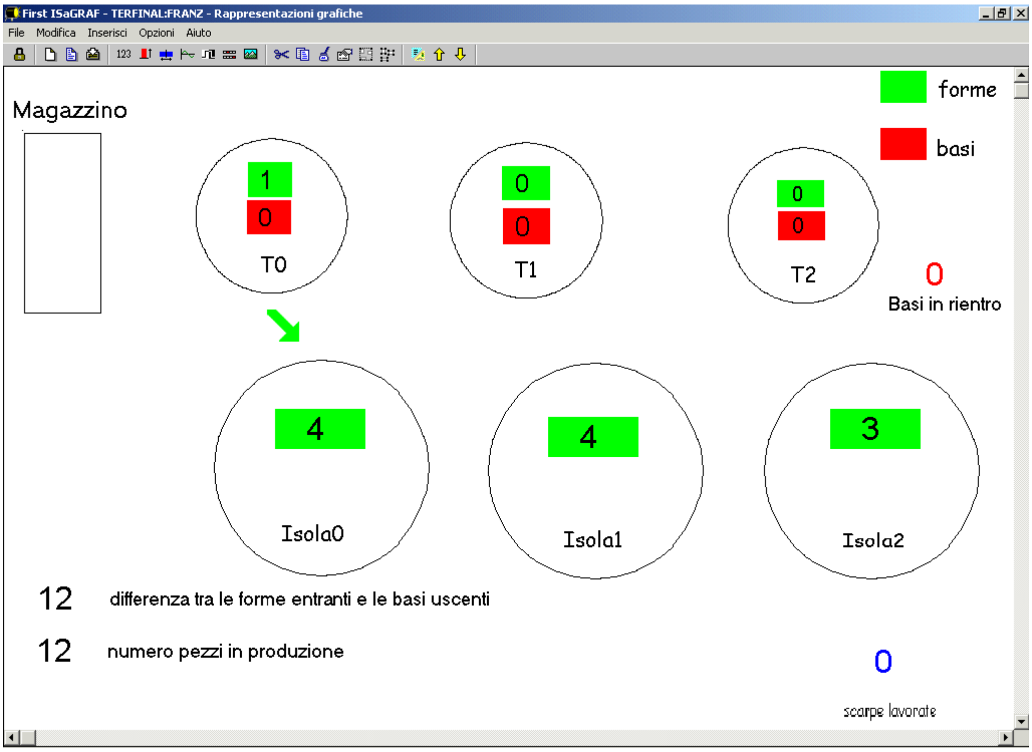


Fig. 16: Simulation Graphical Interface

5. TRANSPORT LINE TECHNOLOGICAL CONSTRAINTS

In this chapter the limitations of the presented general control strategy due to three specific constraints are discussed. In particular, the considered limitations are introduced once at a time so to discuss the possible extensions of the presented control system strategy to deal with these constraints. The first constraint regards the need of moving pair of forms (right and left) and not single ones. The second constraint regards the assumption that the machining operations on the islands are executed sequentially. The third constraint regards the rotation functionality (step by step vs. continuous) of tables and islands.

5.1. MOVEMENTS OF PAIR OF FORMS

This constrain comes from the necessity to machine and to collect at the end of the production process pair of shoes and not single shoes which must be coupled in a following moment.

This constraint has a very slight impact on the cell control system strategy presented above. In fact in order to face this request it is sufficient to repeat twice the same movement required by the priority logic.

Nevertheless, in this way it is necessary to have three slots free on a table, instead of only one, to move a pair of form towards the transport line so that at least one slot free remains available for the backward movement of the bases which, on the contrary, can move individually. In fact the bases can get back to the warehouse individually.

Therefore the cell control system strategy presented above can be used also with this constraint on condition that:

- Each selected movement must be repeated twice;
- For the forward forms movement the receiving table must have at least three slots free.

5.2. SEQUENTIAL MACHINING OPERATIONS ON ISLANDS

If it is addumed that the islands are rotateted step by step and that at every step the form that is aligned with a working station must be machined, then the slowest operation to be done in a cell represents the bottle neck for the execution of another island step rotation.

In this way the cell control system manages the island rotation step by step and at every step evaluates the possible operations that can be carried out. In such a case the only difference with respect to the presented control strategy consists on the necessity to stop the island rotation whenever operations FG or G can’t be executed (for example when the table of the following cell has not enough slots free).

So the modifications to be carried out on the presented cell control system strategy are:

- Island rotation step by step.
- Stop the island rotation if the cases FG or G can’t be executed. Instead the cases D and E can be carried out without any restriction.
- The priorities assigned to the different cases could be the same, but if it happens that the cases F and G are simultaneous then necessarily the case FG must be executed and the case F can’t be taken into account.

The first constraint about the necessity to move pair of forms towards the transport line doesn't have any impact on the modifications required to respect the second constraint. In fact, if the case G can be executed then it can carried out twice, because on the receiving table there must be at least three slots free to allow the movement of pair of forms.

5.3. TABLES STEP BY STEP ROTATION

This third constraint consists on the necessity to rotate step by step both the islands and the tables. In fact, it must be taken into account that according to the present realization:

- The time needed to complete a rotation step (both for a table and an island) is about 2s
- The average time needed to complete a generic machining operation is about 20s.

This means that a complete rotation of an island requires about 24x2=48s. Thus if islands are allowed to freely rotate, in the worst case, the operator is forced to wait for 48s to operate about 20s on a form already laid on the island.

Moreover, it is necessary to force also the tables to work in a sequential way as the islands. Since otherwise a request to align a slot of a table in an exchange position, in the worst case, could take about 24s, which is a significant time consumption.

This last consideration implies that all the pair of forms flowing through the transport line could not overtake each other. This means also that the molecular transport line structure working function is similar to that of a standard conveyor belt one. The only difference between them is that in the molecular structure the pair of forms could avoid to flow through a specific island if specific operations must not be done, while in case of the conveyor belt such forms have to cover the whole route.

The control system strategy presented above is conceptually applicable also if such a constraint holds, anyway the transport system performance may be significantly reduced.

6. CONCLUSIONS AND FUTURE WORK

In the present report a general control strategy for the molecular transport system has been defined, and represented through the SFC formalism. Furthermore, it has been implemented and simulated in ISAGRaF in order to verify its correctness.

Future work will be adaption of the presented control solution to the technological constraints of the transport system and its implementation and testing on the target system.

7. ACRONYMS AND DEFINITIONS

7.1. MUST-KNOW TERMS IN CONTROL AUTOMATION

Advanced control—Process control strategies beyond PID loop control, such as feed forward, dead-time compensation, lead/lag, adaptive gain, neural networks, and fuzzy logic.

Fieldbus architecture—Control architecture that uses digital, serial, multi drop, two-way communications between and among intelligent field devices and control/monitoring systems.

Human-machine interface—Method of displaying machine status, alarms, messages, and diagnostics, often graphical display on a personal computer, providing operator feedback.

IEC 61131—International standard for machine control programming tools. Part Three provides five languages with standard commands and data structure, allowing changes to programming software with less extensive training.

IEC 61499—International standard for industrial-process measurement and control systems. Part One provides functional blocks allowing to describe functional control systems architecture.

Intelligent field devices—Microprocessor-based devices capable of providing multiple process variables, device performance information, diagnostic results, and execution of assigned control functions.

Intelligent I/O modules—I/O module that provides intelligent, on-board processing of input values to control output values, bypassing the PLC or control controller for routine decision making.

Internet—Global collection of industrial, commercial, academic, government, and personal computer networks that exchange information.

Interoperability—When products are replaceable by a similar product from another vendor.

MES—Manufacturing Execution System delivers information-enabling optimisation of production activities from order to goods. It guides, initiates, responds to, and reports on plant activities.

Microsoft Windows Operating Systems—The most widely used operating systems for personal computers. Microsoft NT is a desktop and server package for enterprise-wide applications. Microsoft 95 is a self-contained operating system a built-in and enhanced version of DOS. Microsoft CE is a compact version of Windows for handheld PCs and embedded devices.

Object-oriented software—Software that uses and reuses parcels of code to build applications modelled on object techniques including COM/DCOM, Java, and CORBA standards.

OLE for process control (OPC)—Object linking & embedding (OLE) that treats data as collections of objects to be shared by applications supporting OLE specifications. OPC provides extensions to OLE to support process control data sharing.

Open controller—Controller that looks like a traditional PLC but is a PC operating in a Windows environment with software control.

Open systems—Hardware/software designs in which a degree of interchangeability and connectivity give users choices. Systems complying with the seven layers of the ISO-proposed open-system interconnect, 7-layer model.

PC control—Software-configured control strategy using standard personal computer hardware and software.

PID (Proportional, integral, derivative control)—An intelligent I/O module or program instruction which provides automatic closed-loop operation of process control loops.

Programmable Logic Controller (PLC)—A solid-state control system with user-programmable memory for storage of instructions to implement specific control and automation functions.

S88—An international standard developed by ISA that uses object-oriented concepts to define terminology and models for batch control processes.

Soft logic—Controller is the software, which can run on a variety of personal-computer form factors. Most useful in applications requiring high data collection and processing as well as communications to other networks.

7.2. ACRONYMS

MES - Manufacturing Execution System

OPC – Ole for Process Control

PID – Proportional Integral Derivative

PLC – Programmable Logic Controller

SFC - Sequential Functional Chart

8. AKNOWLEDGEMENT, REFERENCES AND LINKS

8.1. REFERENCES

Ref 1: “Development of the processes and implementation of the management tools for the Extended UseR Oriented Shoe Enterprise”, Version 2, 31.07.2002, Torielli, Siemens, Lirel.

Ref 2: Airoldi F., Ballarino A., Bressanelli M., Carlino G. and Carpanzano E. “Structured design and verification of logic control software for industrial DCS” (in Italian). *Proc. International Conference on Automation within New Global Scenarios*, Milan, Italy, November 19-21, 2002, paper 5.3.

8.2. INTERNET LINKS