Including systematic faults into fault-tree analysis

Israel BARRAGAN SANTIAGO*

Jean-Marc FAURE

Yiannis PAPADOPOULOS**

*** The Mexican Council of Technology CONACYT finances Israel BARRAGAN**

**** University of Hull, UK**

Outline

- Objective of the work
- Extending FTA to address systematic faults
- A vocabulary of gates for systematic faults
- Example
- Conclusion and prospects

Safety analysis of automated systems

Modern automated systems include an increasing number of programmable logic controllers (embedded controllers)

Safety analysis of these systems using for instance Fault tree analysis (FTA), a widespread technique for critical systems, must take into account:

- the physical failures of the components of the process,
- but also the faults caused by the controllers.

SYSTEM SAFETY ANALYSIS =

PROCESS SAFETY ANALYSIS ∧

CONTROLLER(S) SAFETY ANALYSIS

Safety analysis of logic controllers

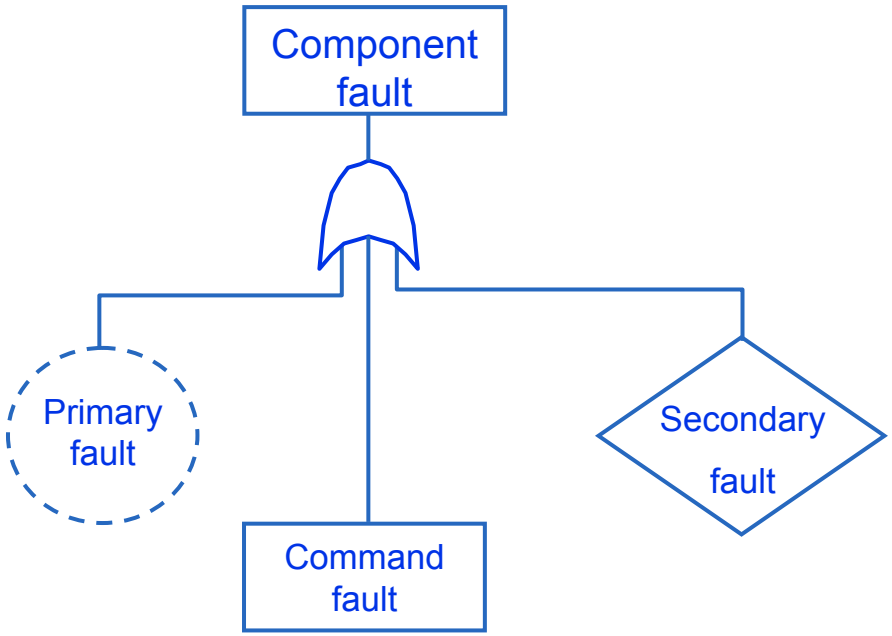
Three categories of controllers faults:

- Hardware failures of the controller components
- Unhanded deviations of controller inputs caused by failures of sensors connected to the controller
- *Design flaws* in the logic (software) of the controller, either a result of coding errors or misinterpretation of control requirements.

The latter ones are *systematic* faults because they can be reproduced every time the conditions that trigger the error in the control logic are present.

But classical FTA relies upon stochastic models …

Classical template for FTA



US NR Commission (1981)

Primary fault: physical failure of the component due to its internal defects

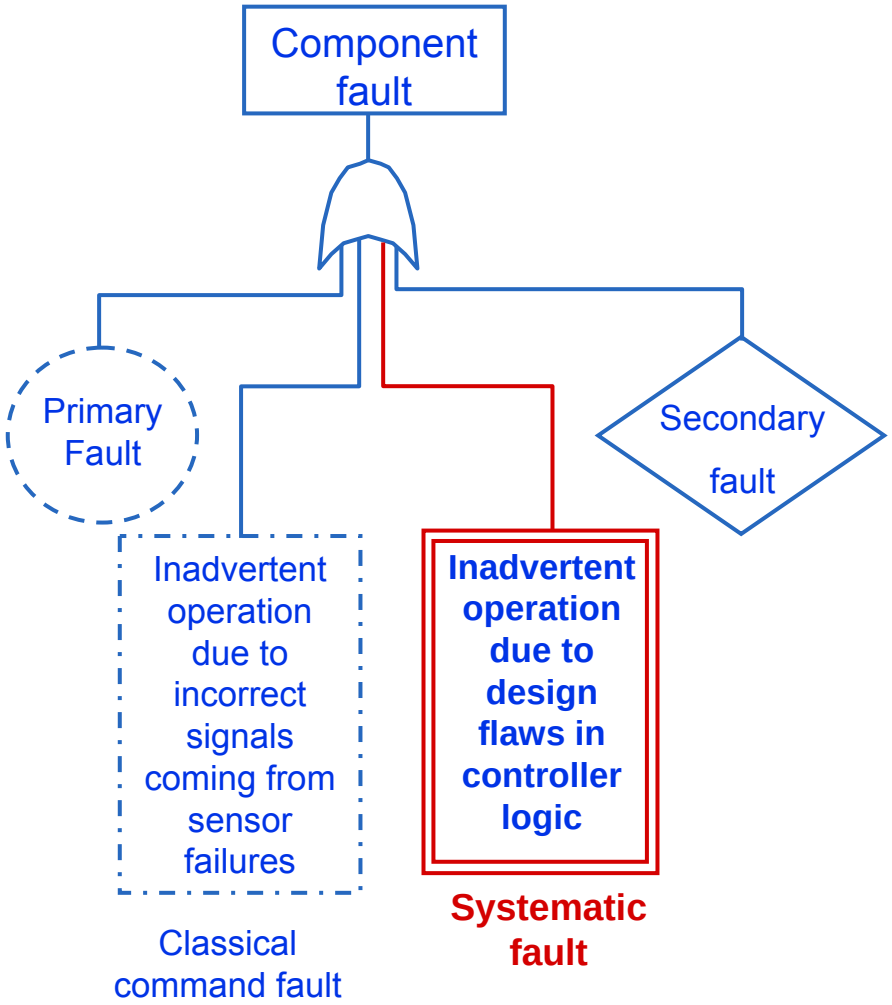
Secondary fault: fault due to excessive environmental or operational stress

Command fault describes a situation in which the component has not physically failed but produces wrong outputs (or no output) in response to inappropriate or misleading inputs received from sensors or controllers that control its operation

Wrong Input → Wrong Output

(WI → WO)

New FT general template



Physical component failure

Classical Command Fault
(WI → WO)

Systematic fault
(RI → WO)

Combination of physical, classical, command and systematic fault (and always faults due to the environment)

Requirements

Systematic faults of logic controllers cannot be described only by combinatory expressions using Boolean connectors (AND, OR, …); they are often featured by erroneous sequences of events or inappropriate delays

- The fault occurs when A is set before B is reset or when signal C is set less than (more than) n seconds.

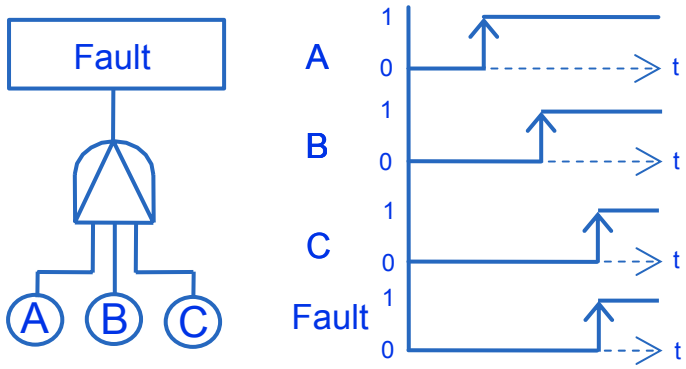
Hence there is a need for gates enabling us to express event ordering and physical time.

These gates shall be formally defined thanks to a formalism of DES (Discrete Event Systems) such as a temporal logic or state automata.

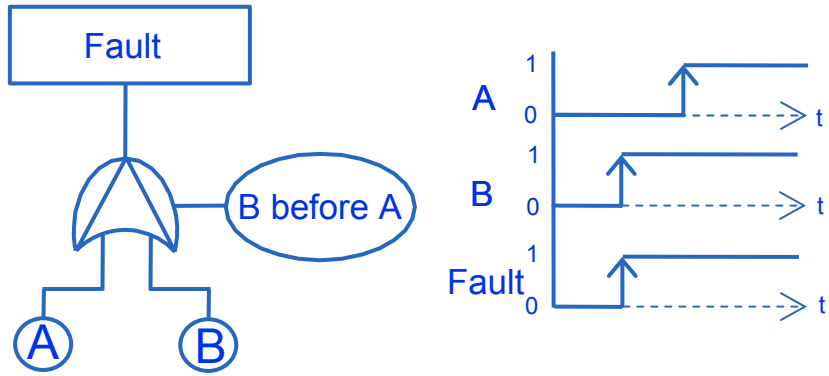
Temporal and timed gates

Temporal gates [FT Handbook, 1981] enable to express event ordering

Priority AND

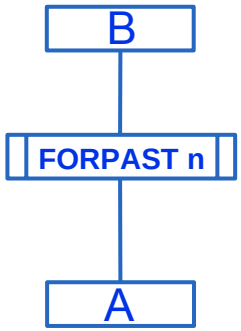


Priority OR



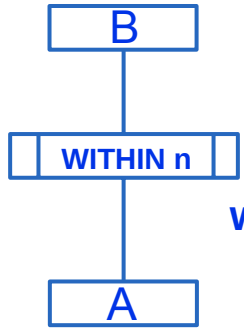
Timed gates [Palshikar, Information and software technology, 2003] enable to express physical time

FORPAST n



A stays TRUE during n time units

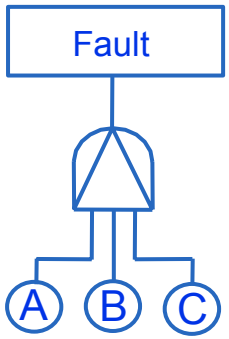
WITHIN n



A stays TRUE at least one time unit within a n time units interval

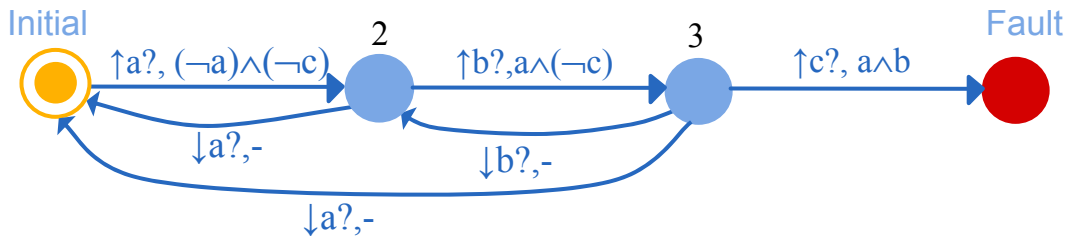
Formalizing temporal gates behaviour

Priority AND



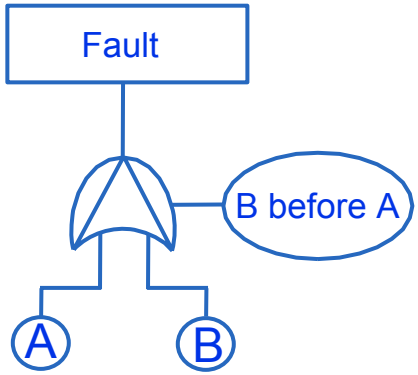
AG ¬ (A => EF (A.B => EF ABC))

In CTL



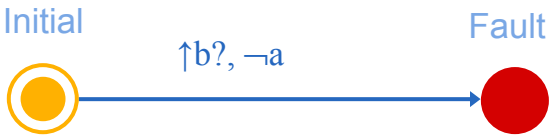
Observer automaton

Priority OR



A (¬ B W A)

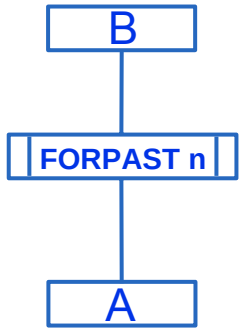
In CTL



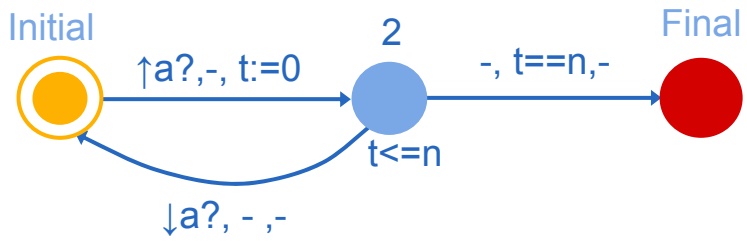
Observer automaton

Formalizing timed gates behaviour

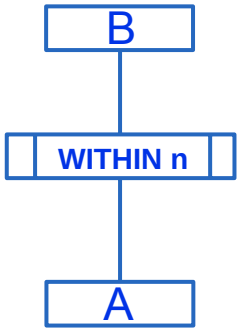
FORPAST n

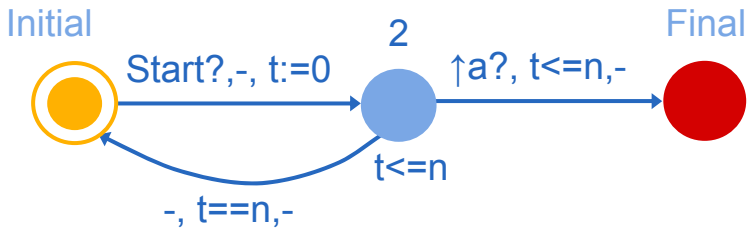


Timed Automata:

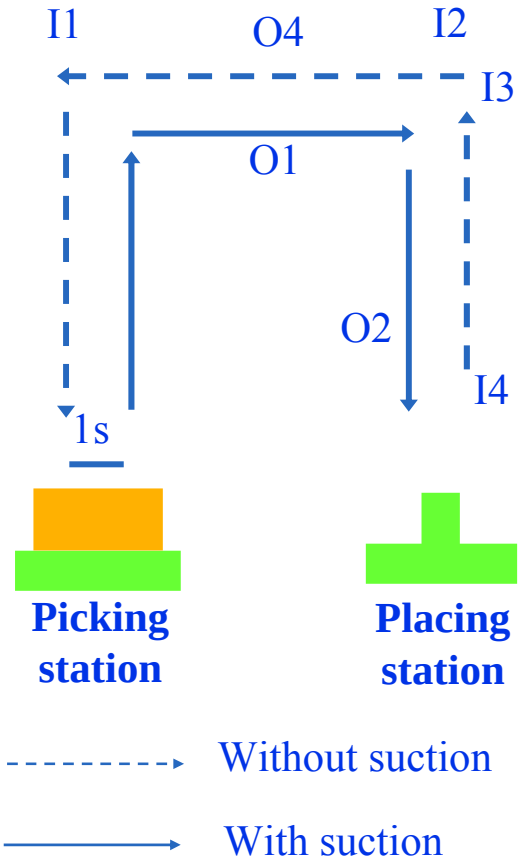


WITHIN n





Example: safety analysis of a pick and place manipulator



CONTROLLER INPUTS

Leftmost Position	I1
Rightmost Position	I2
Upper Position	I3
Lower Position	I4

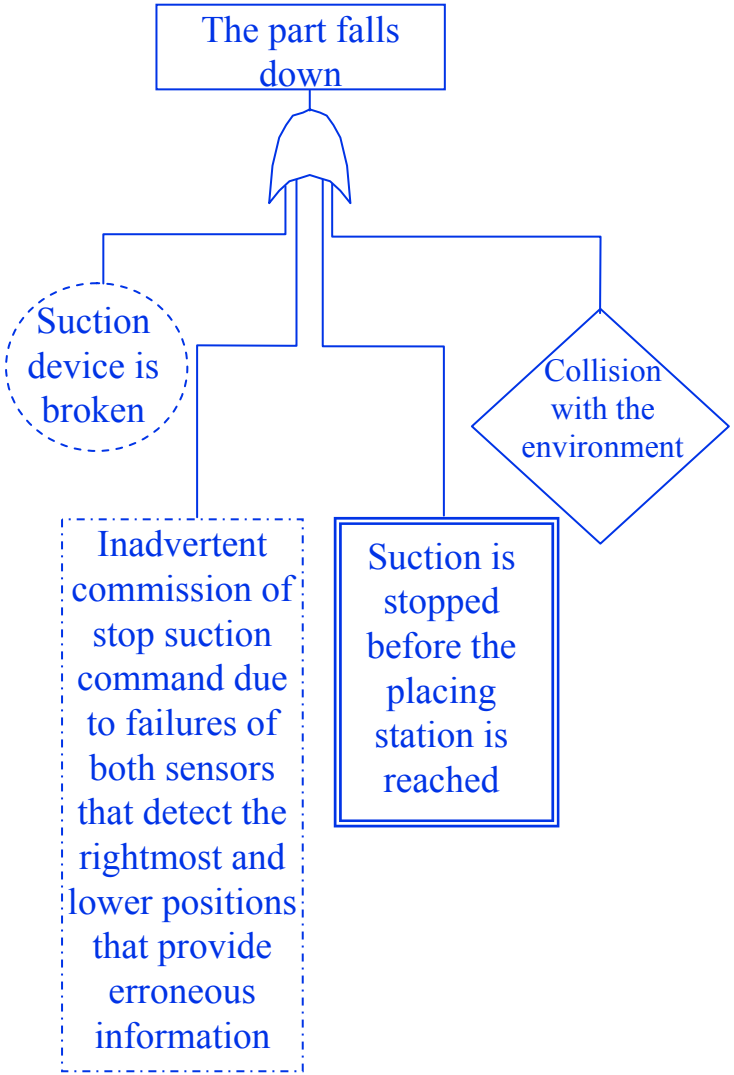
CONTROLLER OUTPUTS

Move to the Right	O1
Move Down	O2
Suction	O3
Move to the left	O4

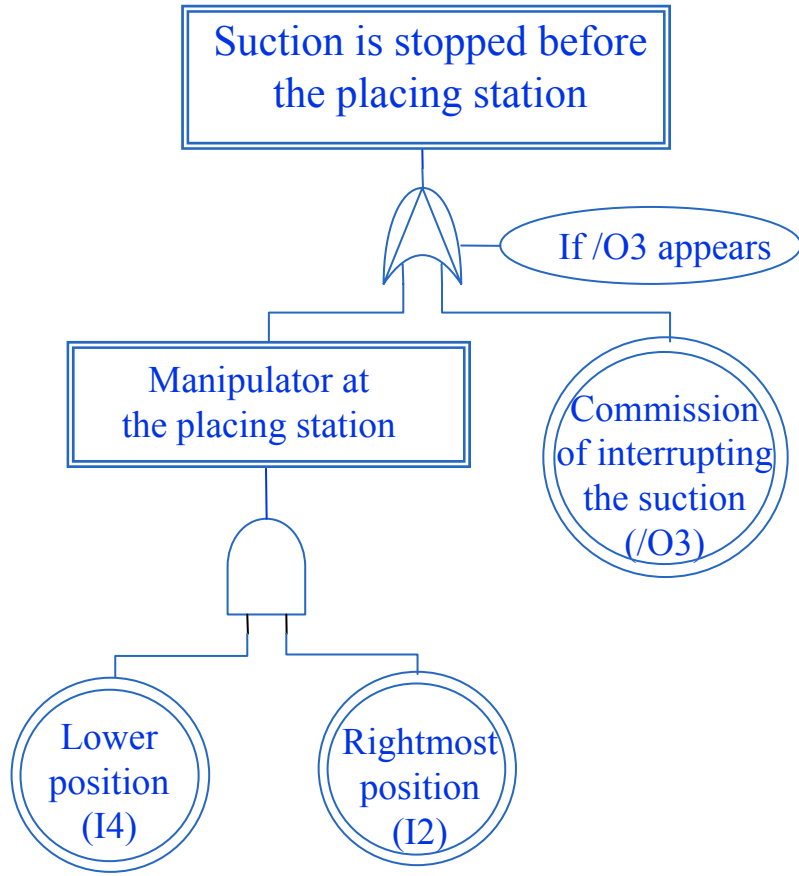
FT analysis: part falling down during the transfer

Example

SYSTEM SAFETY ANALYSIS

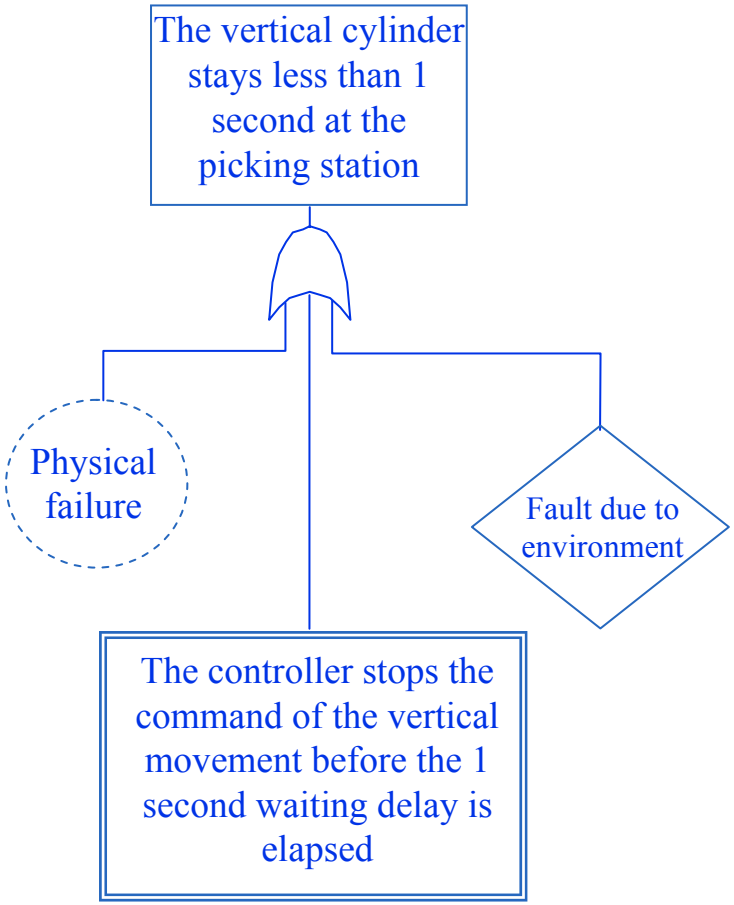


SYSTEMATIC FAULT ANALYSIS

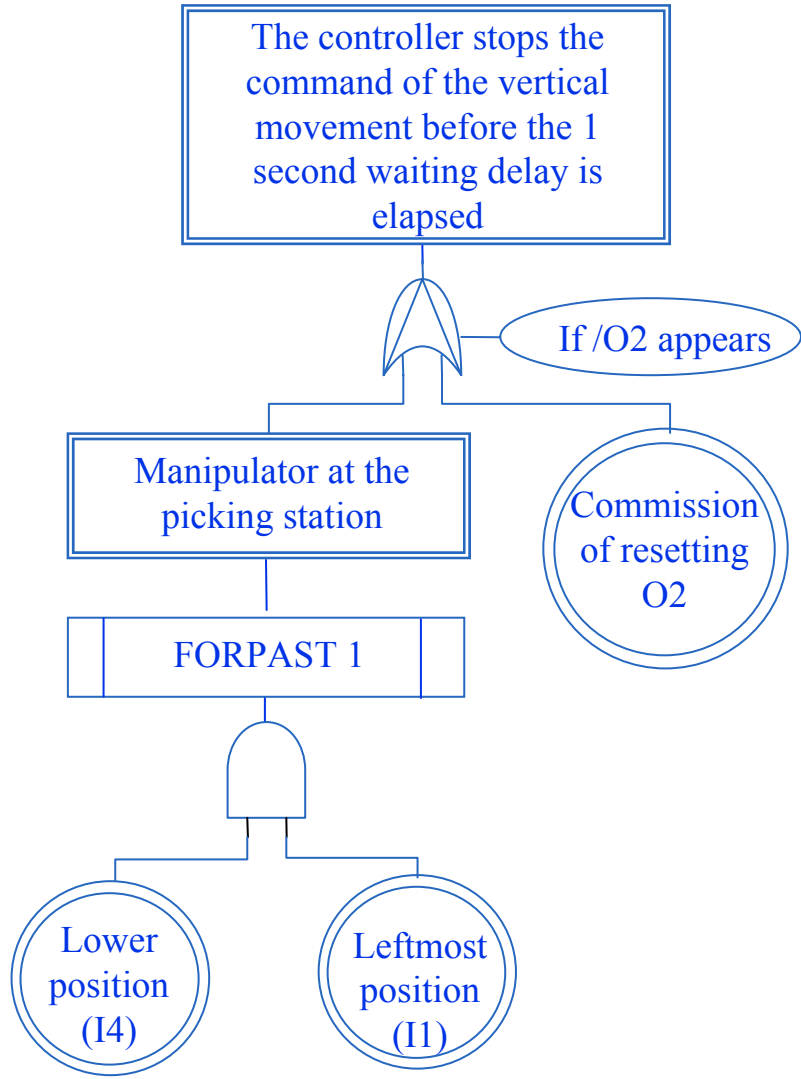


FT analysis: part not picked up

SYSTEM SAFETY ANALYSIS



SYSTEMATIC FAULT ANALYSIS



Conclusions

- To avoid dangerous and/or costly failures, fault tree analysis of complex automated systems must include systematic faults of controllers
- Temporal and timed gates are to be used; the operational semantics of these gates has been formally defined
- Coupling fault forecasting and systematic fault removal, by using for instance model-checking techniques, has been achieved (Barragan and Faure, IFAC WC 2005; Barragan et al, IFAC INCOM 2006)

On-going works and prospects

- Consistency checking and simplification of FT containing temporal and timed gates

  Rules to combine gates have been developed

  Automatic generation of minimal sequences sets described in the form of untimed or timed automata

- Integration of these results (New FT template and temporal and timed gates) within a tool for automatic generation of FT (Hip-Hops, Papadopoulos, Y. and M. Maruhn, 2001).

Including systematic faults

into fault-tree analysis

Thank you for attention

Any questions ?