Improving Availability and Safety of Control Systems by Cooperation between Intelligent Transmitters

F. Brissaud, A. Barros, C. Bérenguer

French National Institute for Industrial Environment and Risk

& Troyes University of Technology

10th International PSAM Conference

7-11 June 2010, Seattle, Washington, USA

utt université de technologie Troyes

INERIS

Overview

I. Introduction

- "intelligent" transmitters
- distributed and networked control systems

II. Control system with cooperating transmitters

- control system of redundant transmitters
- cooperation between transmitters

III. Modelling and evaluating cooperating transmitters

- modelling by stochastic and coloured Petri nets
- availability and safety analyses by Monte Carlo simulations

IV. Discussion & conclusion

I. Introduction

Sensor systems are now able to

- collect data from the physical world
- perform internal data processing
- transmit an elaborate signal

… and to perform advanced functionalities such as

- error measurement corrections
- self-adjustment
- self-diagnoses
- online reconfigurations
- digital and bidirectional communication

making *"Intelligent" transmitters*.

I. Introduction

The use of “embedded intelligence” in control systems

- allows the spatial relocation of some operations,
- forming a ***distributed control system*** (DCS).

In addition, the use of a real-time communication network

- allows interconnecting the elements of a DCS,
- forming a ***networked control system*** (NCS).

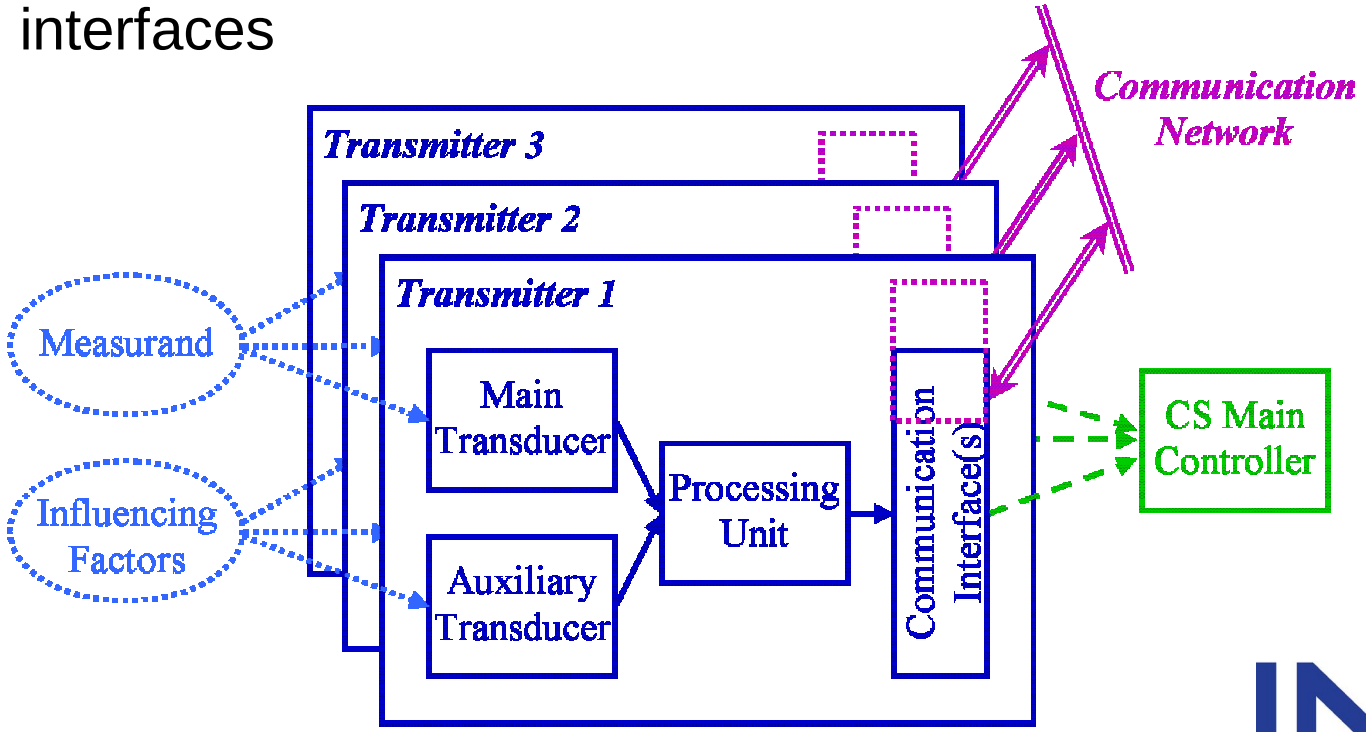
Features of an NCS:

- wiring reduction, lower costs, ease of maintenance, flexibility, …
- system elements such as ***transmitters may implement their operations by exchanging information***
- ***what about dependability?***

II. CS with cooperating transmitters

Three redundant transmitters

- monitor measurand and influencing factors by transducers
- perform measurement results by processing units
- transmit signals to a main controller by communication interfaces



II. CS with cooperating transmitters

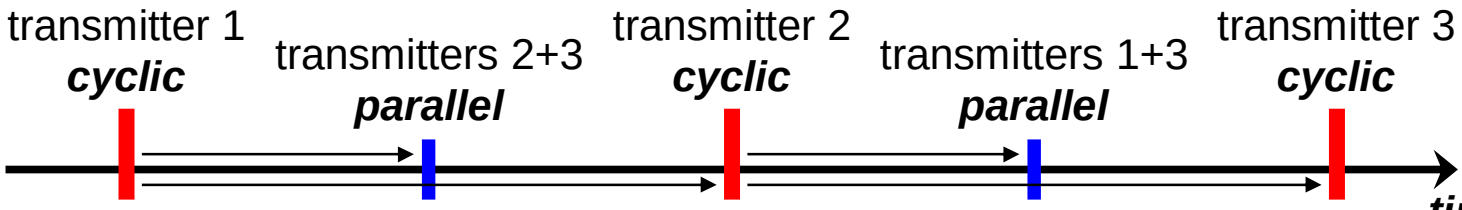
Functional states

- ***self-diagnoses*** are available ***for each transmitter element*** (main transducer, auxiliary transducer, processing unit)
- each transmitter has three functional states
  - ***operating***: transmits right measurement results
  - ***detected failure***: transmits an error signal
  - ***undetected failure***: transmits wrong measurement results
- the control system has then three functional states, according to the three signals received from the transmitters
  - ***operating*** if the majority of non-error signals provide right results
  - ***detected failure*** if all the received signals are error signals
  - ***undetected failure*** in the other cases

II. CS with cooperating transmitters

Communication network between the three transmitters

- the transmitters are allowed to exchange
  - transmitter ***identification***
  - ***values*** of measurand, influencing factors, and measurement results
  - ***self-diagnoses*** for transducers, and processing unit
  - ***diagnosis compilation*** result (to command error signals)
- the transmitters are triggered according to two demand types
  - ***cyclic*** demand type: triggered alone, transmits information to the other transmitters and measurement results to the main controller
  - ***parallel*** demand type: triggered by pair, transmits measurement results to the main controller



II. CS with cooperating transmitters

Procedures to improve CS dependability by cooperation

- ***backup algorithm***
  - only when the transmitter is triggered in a parallel demand type
  - if a transducer (main or auxiliary) is self-diagnosed as failed, then replaces the monitoring value (measurand or influencing factors) by the corresponding value from another transmitter
- ***contrast algorithm***
  - only when the transmitter is triggered in a parallel demand type
  - replaces the diagnosis compilation by a result obtained by comparisons of the measurement results (if two consecutive measurement results from different transmitters are equal, then the corresponding transmitters are diagnosed as “operating”)

III. Modelling and evaluating CS

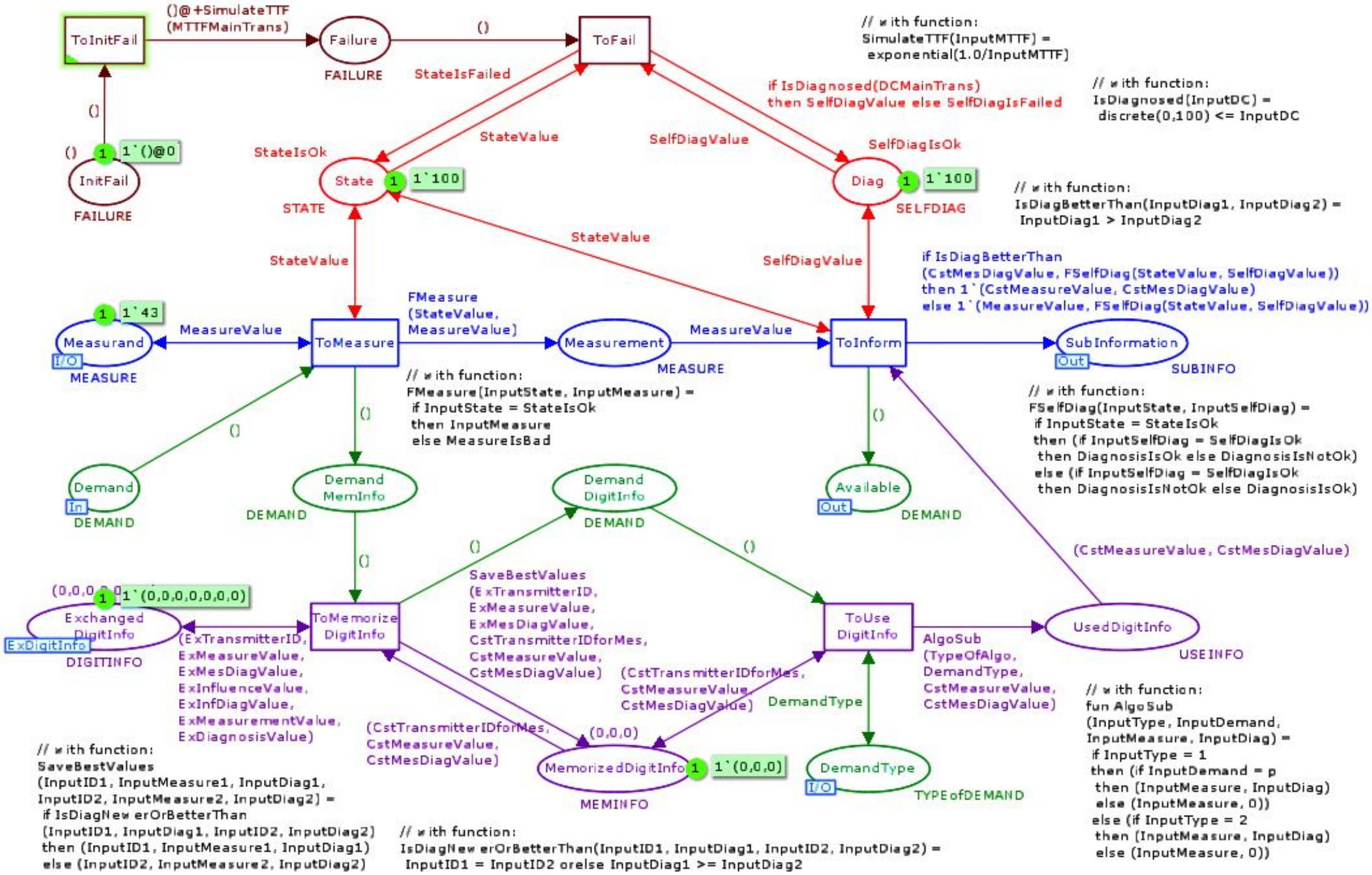
Stochastic and coloured Petri nets

- interesting tool for describing and studying dynamic systems
  - places represent objects and conditions
  - tokens specify the values of these objects and conditions
  - transitions model system activities
- stochastic properties
  - transitions are enabled after random delays
  - to model random failures and imperfect self-diagnoses
- coloured properties
  - values are assigned to tokens, and may be changed
  - to model information (measurements, diagnoses) and properties (states of elements, accuracy of information)
- analyses are performed by simulations

III. Modelling and evaluating CS

CPN tools

- free computer tool developed by the University of Arhus
- designed for coloured Petri nets
- expressive semantics for variable and expression definition, including stochastic properties
- ability to model hierarchical Petri nets
- supports analyses by Monte Carlo simulations


ToInitFail
()@+SimulateTTF (MTTFMainTrans)
Failure
FAILURE
()
ToFail
// with function: SimulateTTF(InputMTTF) = exponential(1.0/InputMTTF)
()
1`()@0
InitFail
FAILURE
StateIsFailed
StateIsOk
StateValue
SelfDiagValue
if IsDiagnosed(DCMainTrans) then SelfDiagValue else SelfDiagIsFailed
SelfDiagIsOk
// with function: IsDiagnosed(InputDC) = discrete(0,100) <= InputDC
State
1`100
STATE
Diag
1`100
SELFDIAG
// with function: IsDiagBetterThan(InputDiag1, InputDiag2) = InputDiag1 > InputDiag2
StateValue
StateValue
SelfDiagValue
if IsDiagBetterThan (CstMesDiagValue, FSelfDiag(StateValue, SelfDiagValue)) then 1`(CstMeasureValue, CstMesDiagValue) else 1`(MeasureValue, FSelfDiag(StateValue, SelfDiagValue))
1`43
Measurand
I/O
MEASURE
MeasureValue
ToMeasure
FMeasure (StateValue, MeasureValue)
Measurement
MEASURE
MeasureValue
ToInform
SubInformation
Out
SUBINFO
// with function: FMeasure(InputState, InputMeasure) = if InputState = StateIsOk then InputMeasure else MeasureIsBad
// with function: FSelfDiag(InputState, InputSelfDiag) = if InputState = StateIsOk then (if InputSelfDiag = SelfDiagIsOk then DiagnosisIsOk else DiagnosisIsNotOk) else (if InputSelfDiag = SelfDiagIsOk then DiagnosisIsNotOk else DiagnosisIsOk)
()
()
()
Demand
In
DEMAND
Demand MemInfo
DEMAND
Demand DigitInfo
DEMAND
Available
Out
DEMAND
(CstMeasureValue, CstMesDiagValue)
1`(0,0,0,0,0,0,0)
Exchanged DigitInfo
ExDigitInfo
DIGITINFO
ToMemorize DigitInfo
SaveBestValues (ExTransmitterID, ExMeasureValue, ExMesDiagValue, CstTransmitterIDforMes, CstMeasureValue, CstMesDiagValue)
ToUse DigitInfo
AlgoSub (TypeOfAlgo, DemandType, CstMeasureValue, CstMesDiagValue)
UsedDigitInfo
USEINFO
(ExTransmitterID, ExMeasureValue, ExMesDiagValue, ExInfluenceValue, ExInfDiagValue, ExMeasurementValue, ExDiagnosisValue)
(CstTransmitterIDforMes, CstMeasureValue, CstMesDiagValue)
(CstTransmitterIDforMes, CstMeasureValue, CstMesDiagValue)
DemandType
(0,0,0)
MemorizedDigitInfo
1`(0,0,0)
MEMINFO
DemandType
I/O
TYPEofDEMAND
// with function: fun AlgoSub (InputType, InputDemand, InputMeasure, InputDiag) = if InputType = 1 then (if InputDemand = p then (InputMeasure, InputDiag) else (InputMeasure, 0)) else (if InputType = 2 then (InputMeasure, InputDiag) else (InputMeasure, 0))
// with function: SaveBestValues (InputID1, InputMeasure1, InputDiag1, InputID2, InputMeasure2, InputDiag2) = if IsDiagNewerOrBetterThan (InputID1, InputDiag1, InputID2, InputDiag2) then (InputID1, InputMeasure1, InputDiag1) else (InputID2, InputMeasure2, InputDiag2)
// with function: IsDiagNewerOrBetterThan(InputID1, InputDiag1, InputID2, InputDiag2) = InputID1 = InputID2 orelse InputDiag1 >= InputDiag2

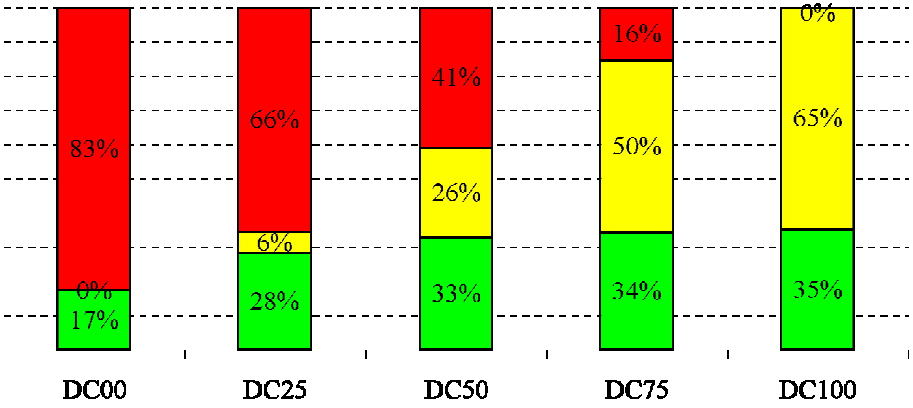
III. Modelling and evaluating CS

Availability and safety analyses

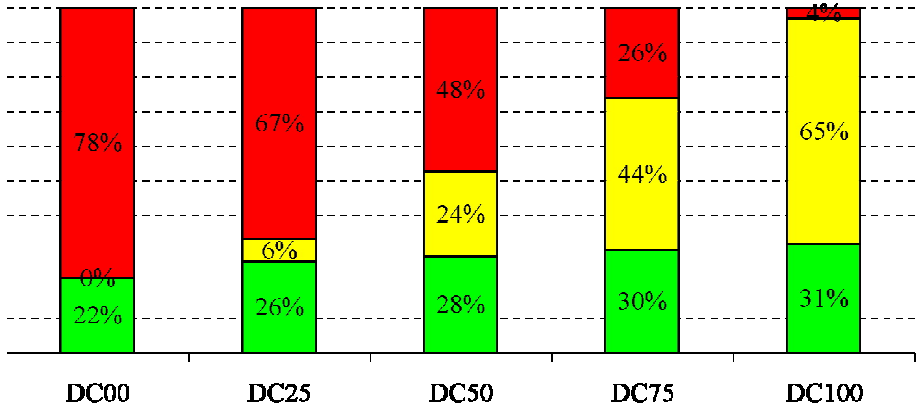
- the times to failure of transmitter elements follow exponential distributions with means equal to *5,000* time units
- when a failure occurs, it is detected by self-diagnoses according to a constant probability equal to the ***diagnostic coverage*** (DC)
- no maintenance action is assumed
- ***availability***
  - average percentage of time that the system is ***in operating state*** during the first *10,000* time units
- ***safety***
  - average percentage of time that the system is ***not in undetected failure state*** during the first *10,000* time units

III. Modelling and evaluating CS

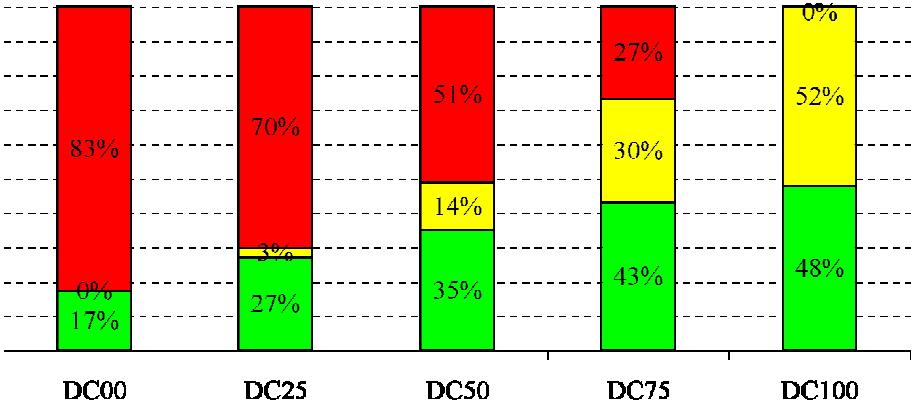
Without *Backup Algorithm* and *Contrast Algorithm*



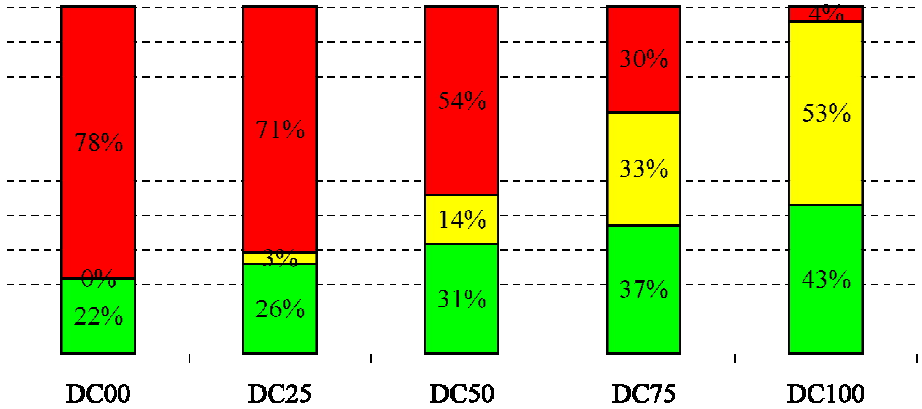
With *Contrast Algorithm* alone



With *Backup Algorithm* alone



With *Backup Algorithm* and *Contrast Algorithm*



Operating | Detected Failure | Undetected Failure

IV. Discussion & conclusion

Effects of the algorithms

- ***backup algorithm*** increases availability but decreases safety, and this effect is increasing according to diagnostic coverage
  → balance between availability and safety
- ***contrast algorithm*** increases both availability and safety when diagnostic coverage is low, but decreases both of them otherwise
  → depends on the diagnostic coverage

Conclusion

- stochastic and coloured Petri nets provide an intuitive tool to model control systems with “intelligent” transmitters
- analyses can show the effects of the “intelligent” functionalities of transmitters on control system dependability
- new technologies have to be used appropriately to meet availability and safety requirements

Thanks for your attention

Questions & comments are welcome

florent.brissaud@ineris.fr

10th International PSAM Conference
7-11 June 2010, Seattle, Washington, USA

utt université de technologie Troyes

INERIS