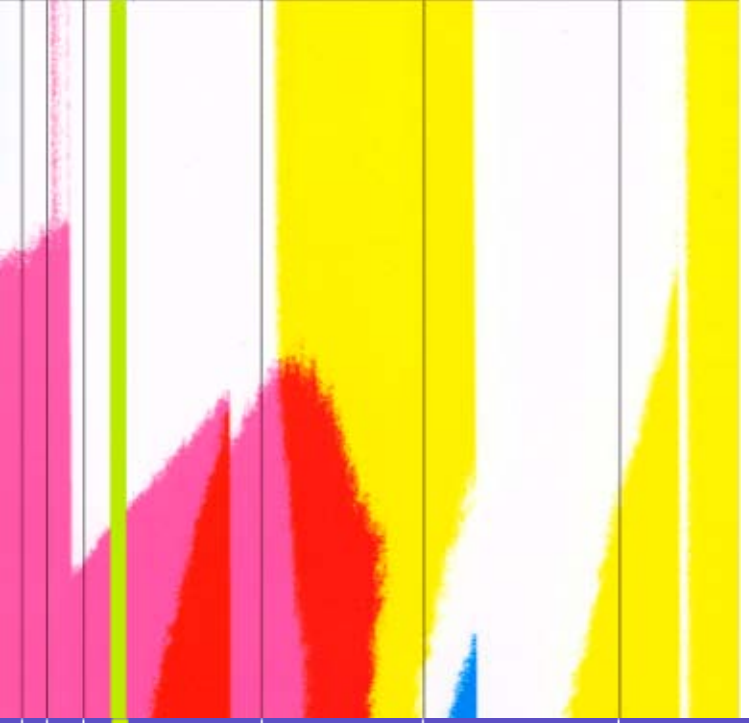


Componentizing and Distributing Feedback

Paths Ahead Symposium
November 2009

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

Albert Benveniste

INRIA

Why are these large and complex systems??

What the manager sees:

+ $$

What a designer sees:

What another designer sees:

Making large systems small: getting a birds eye view while keeping track of details

Opportunities for control in systems engineering

In the late 80's and early 90's there was a clear decline of control w.r.t. computer engineering in industry

Digital computing went embedded, hence software and architectures became the dominant issues

Simulink emerged that allowed physicists to handle modelling

Opportunities for control in systems engineering

In the late 80's and early 90's there was a clear decline of control w.r.t. computer engineering in industry

Digital computing went embedded, hence software and architectures became the dominant issues

Simulink emerged that allowed physicists to handle modelling

The 2000's have reverted the trend

The core of safety critical embedded software is about control

More functions, more functions distributed across the system, and across systems of systems

Computer scientists get interested in control and even express respect

Alas…

What desperately needs control:

What the focus of control engineers often is:

Making large systems small: getting a birds eye view while keeping track of details??

What desperately needs control:

What the focus of control engineers often is:

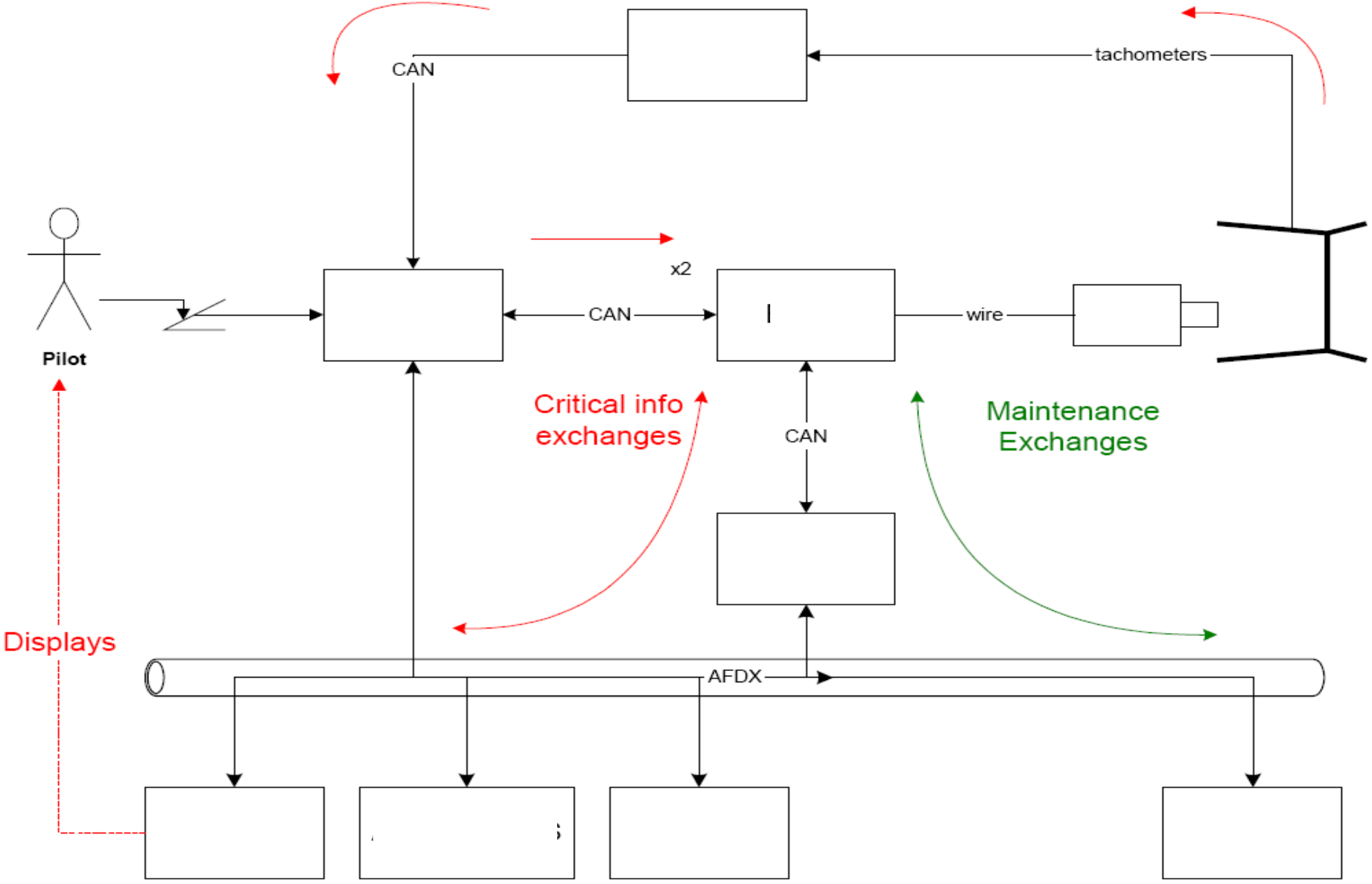
Making large systems small: getting a birds eye view while keeping track of details??

Subcontracting, componentizing
Acknowledging that large systems are distributed

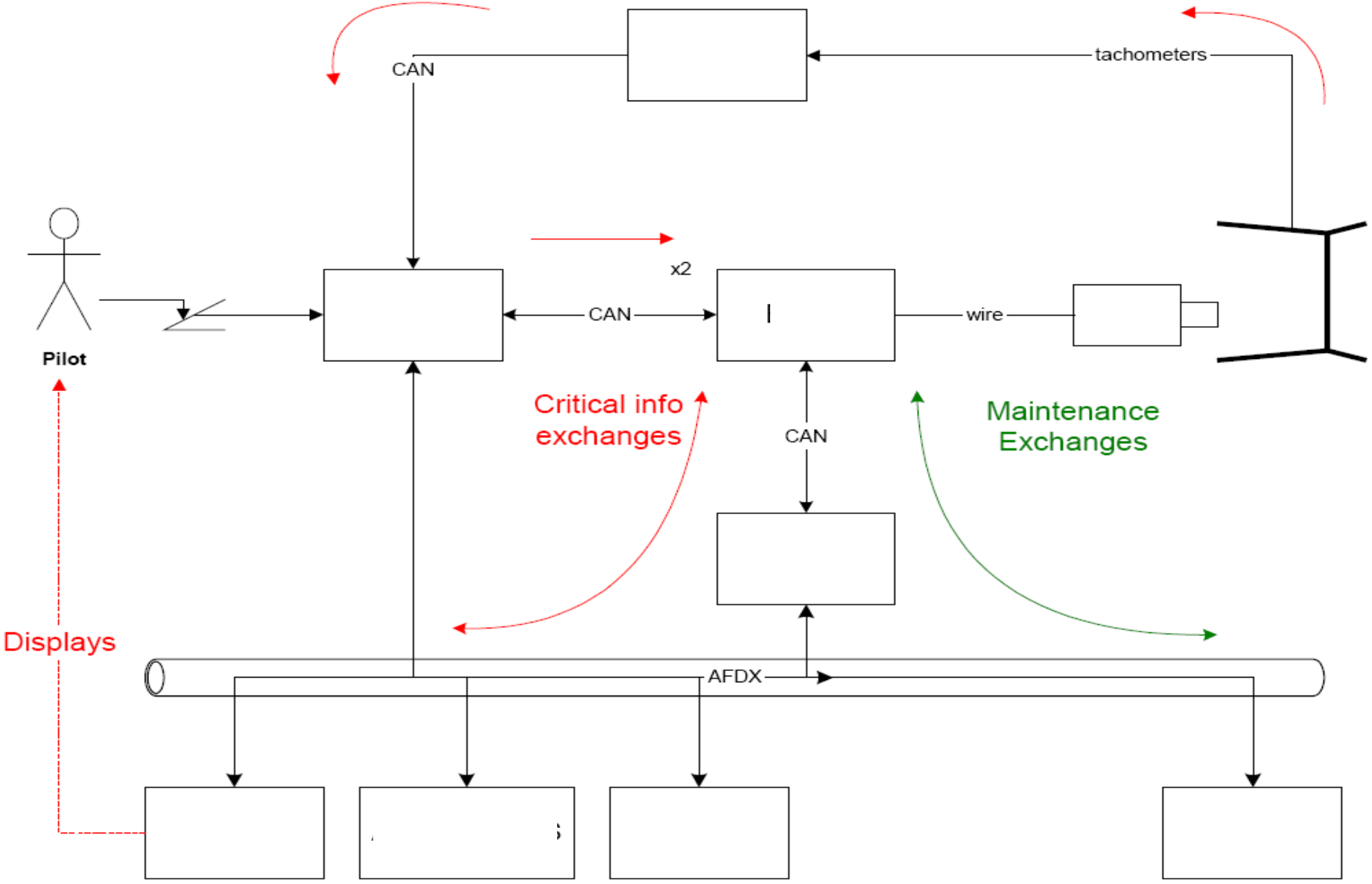
Subcontracting

Componentizing

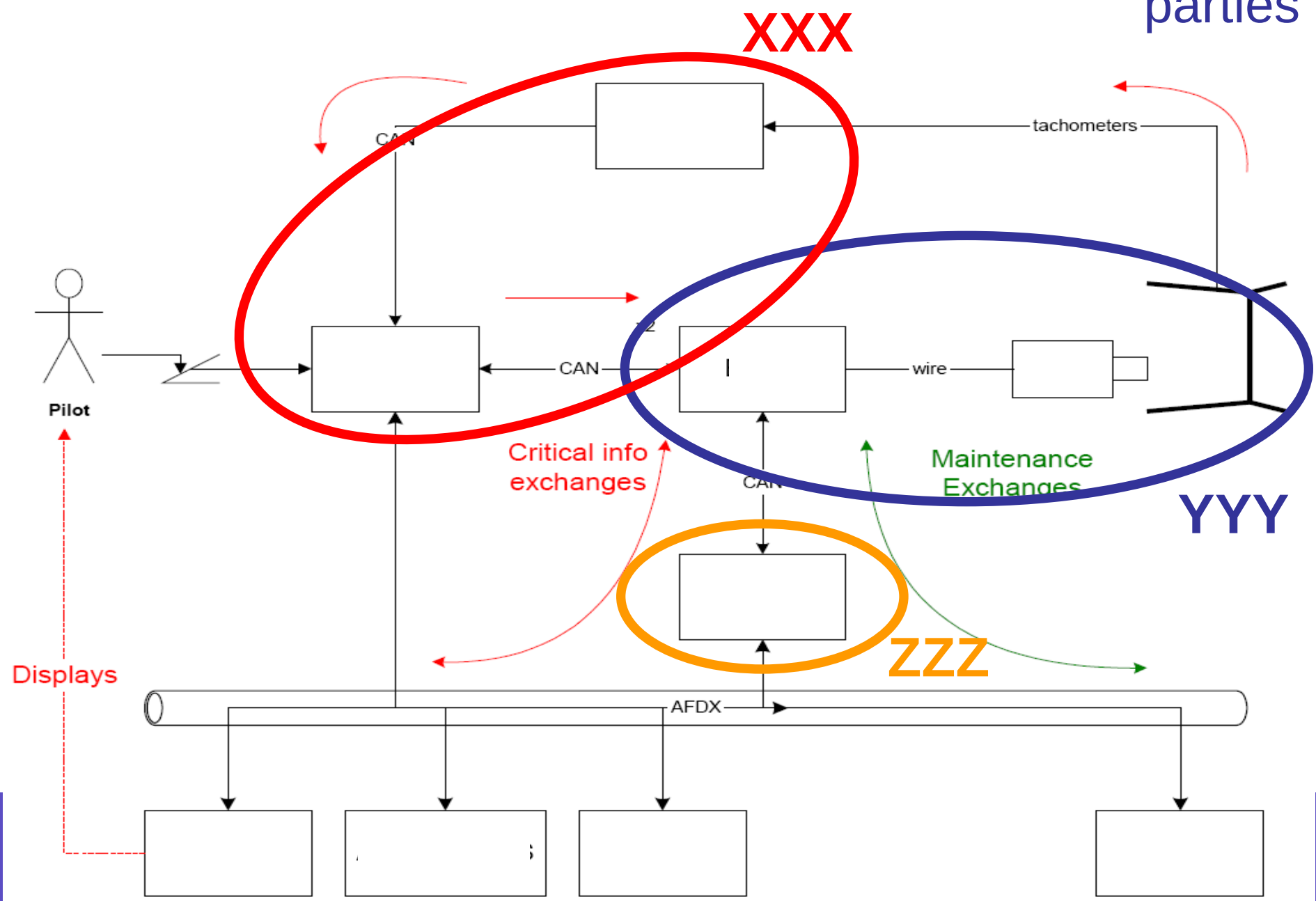
An interesting real industrial case



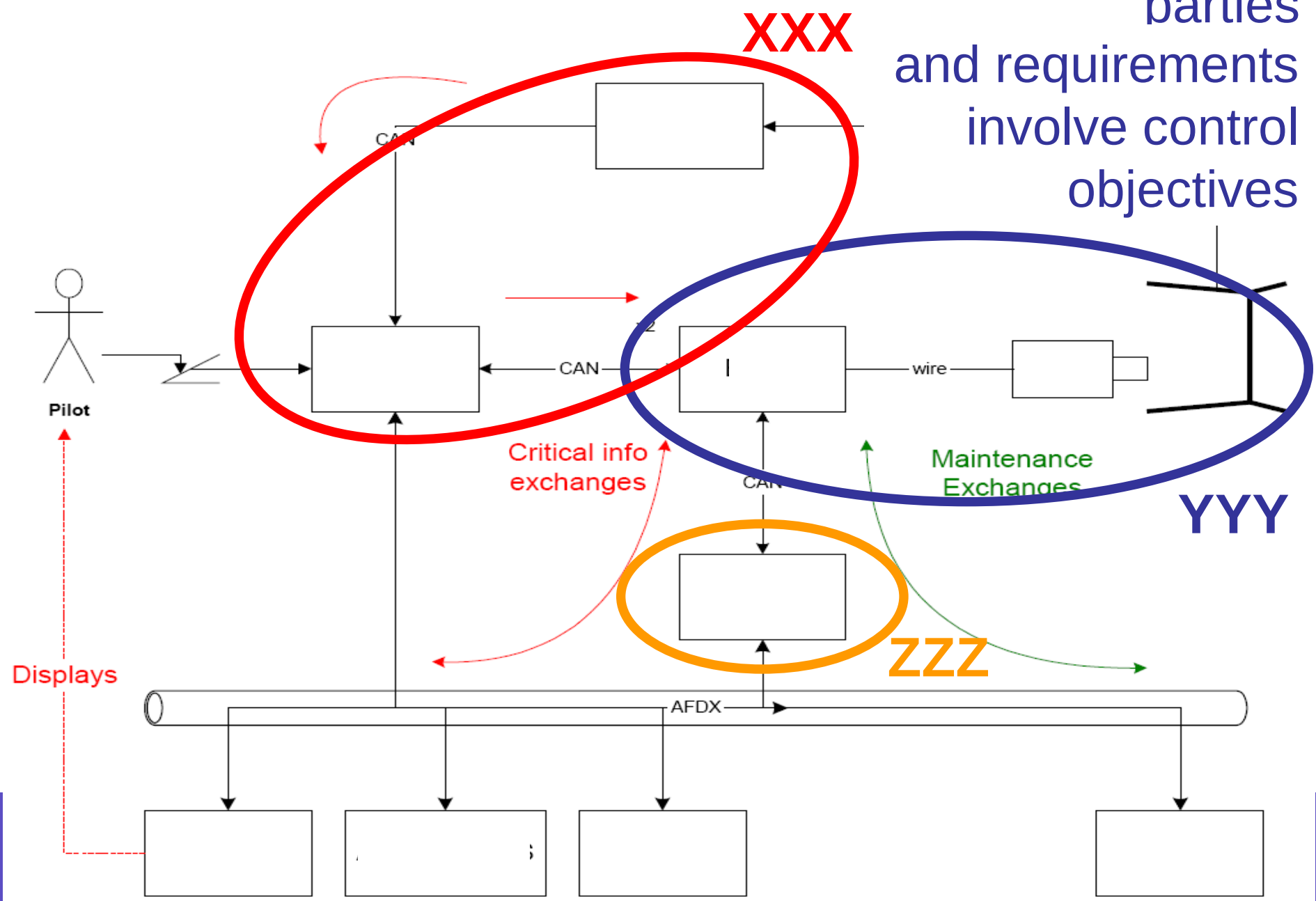
This real system involves a global feedback loop



It has, however, been subcontracted to 3 different parties



It has, however, been subcontracted to 3 different parties and requirements involve control objectives



Componentizing, sub-contracting

What does our community offer?

- For subcontracting feedback
- For having control specs as part of requirements
- For component-based control design
- Should we build on top of MBPC???

Did we devote enough effort?

- To make stability compose
- To develop passivity based design

Distributing Feedback

Large systems are distributed

Computer scientists have devoted a great deal of effort to this

System architectures:
fault tolerance, partitioning
“distributed algorithms”
TTA

With lots of theoretical effort:
theory of concurrency
time-based distribution

And effectiveness
(despite problems remain)

Large systems are distributed

Computer scientists have devoted a great deal of effort to this

System architectures:
fault tolerance, partitioning
“distributed algorithms”
TTA

With lots of theoretical effort:
theory of concurrency
time-based distribution

And effectiveness
(despite problems remain)

Did the control community achieve the same?

Did we develop control architectures?
Did we think of how to blend control with other systems aspects?

Did we develop enough mathematical frameworks for distributed control?

Even though we may have mathematics, did we think of how to expose them to the designer?

Replay Matlab-Simulink success story?

Systems engineering offers more and more opportunities for challenging research in control…

provided control researchers further broaden their vision