

Programmation synchrone

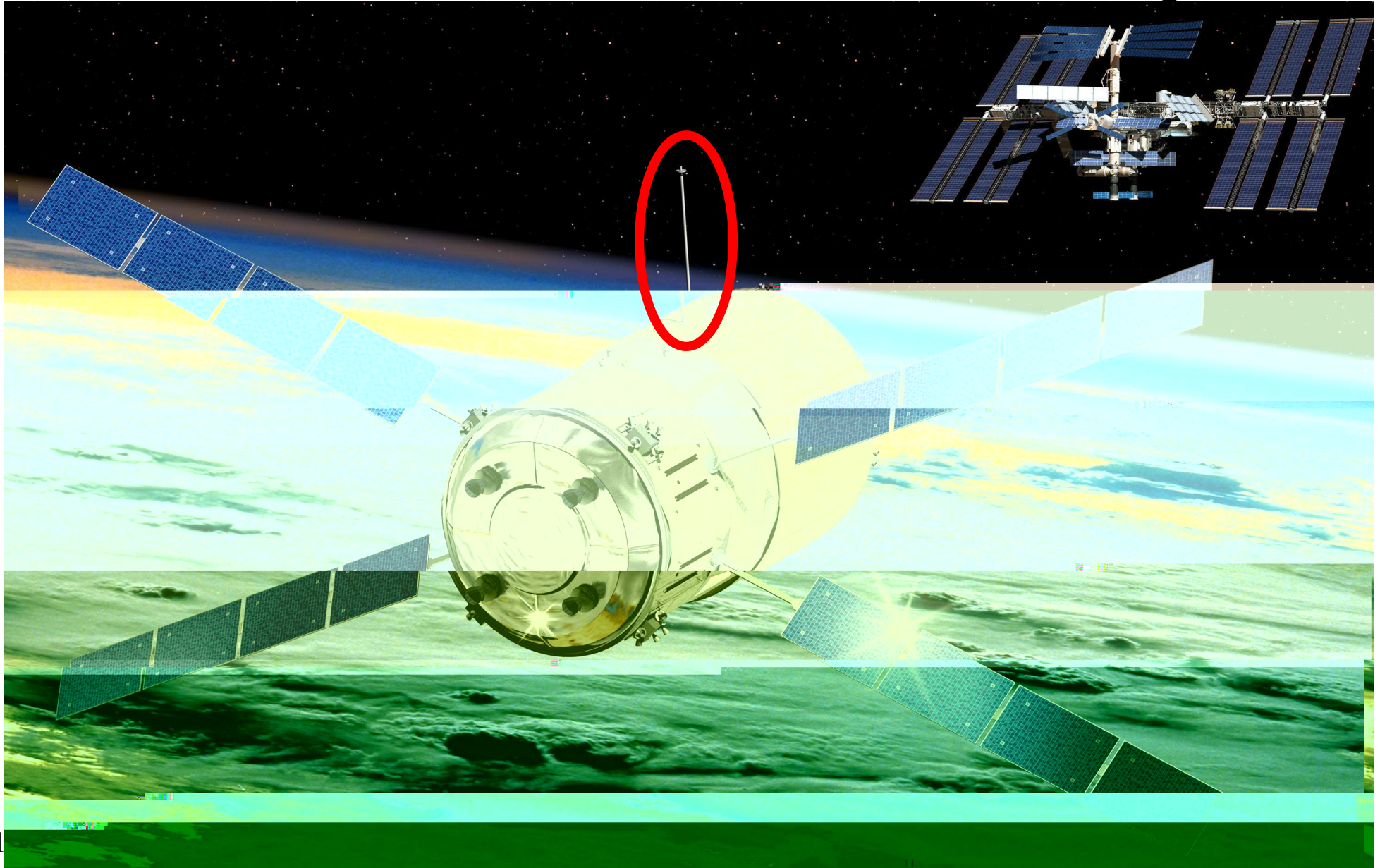
Projet 2010

Déploiement du mât de communication

Date limite: 17 décembre 2010

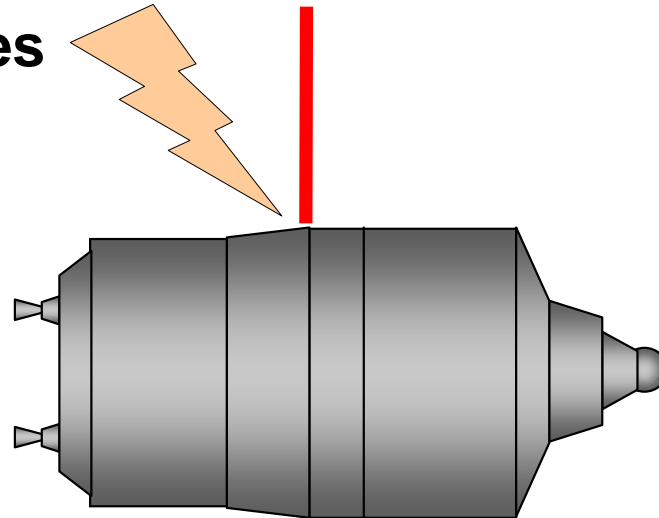
david.lesens@orange.fr

Déploiement du mât de communication d'un véhicule spatial



Déploiement du mât de communication

Couteaux thermiques



**Le logiciel de bord contrôle
l'alimentation des couteaux thermiques
afin de déployer le mât**

Gestion du mât

- Le mât déployable permet la communication avec la station spatiale internationale. Au largage, le mât est replié. Le logiciel de bord est responsable de son déploiement.

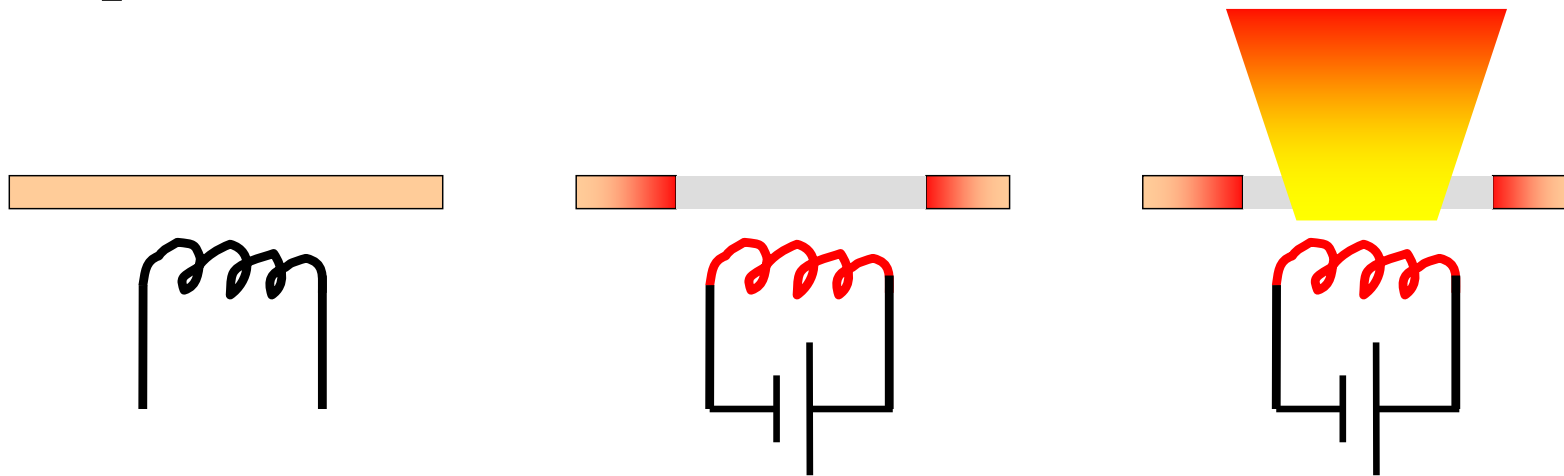
Système:

- ✓ Le mât
- ✓ Des liens et des couteaux thermiques
- ✓ Des batteries
- ✓ Un contrôleur

➔ Objectif: **Développer et valider le contrôleur**

Couteau thermique (1/3)

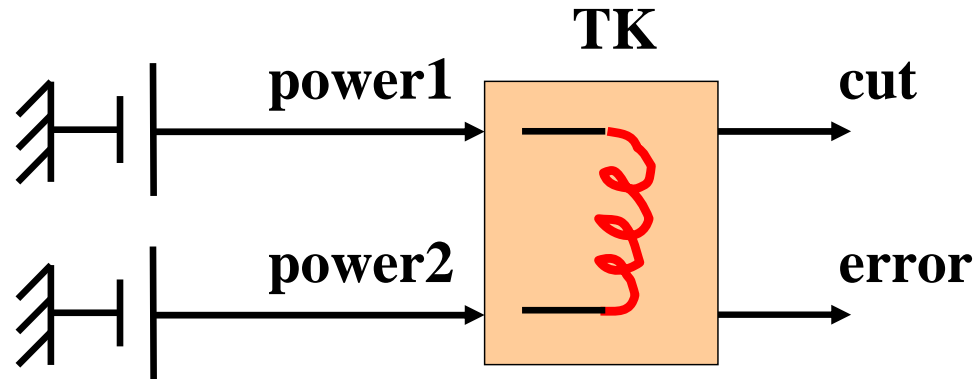
- Un couteau thermique consiste en une « *résistance* » associée à un « *lien* »
- S'il est alimenté par une « *source de puissance* » pendant 2 cycles de suite, le « *lien* » est coupé
- S'il est alimenté pendant 4 cycles (ou plus) de suite, il peut prendre feu



Couteau thermique (2/3)

- Un couteau thermique prend en entrée
 - ✓ Deux sources redondantes de puissance « power1 » et « power2 » de type booléen (valant vraie si active)
- Et en sortie
 - ✓ « cut » valant vraie si et seulement si le lien est coupé
 - ✓ « error » en cas d'erreur:
 - ❖ Les deux sources de puissance sont actives en même temps
 - ❖ Le lien prend feu
 - ❖ La couteau est alimenté alors que le lien est déjà coupé

Couteau thermique (3/3)



➤ Développer l'opérateur « TK » (thermal knife)

01

✓ Sans automate et

02

✓ Avec uniquement des automates (sans FBY ni PRE)

03

➤ Le valider par simulation

04

➤ Prouver formellement que les deux versions (avec et sans automates) sont équivalentes

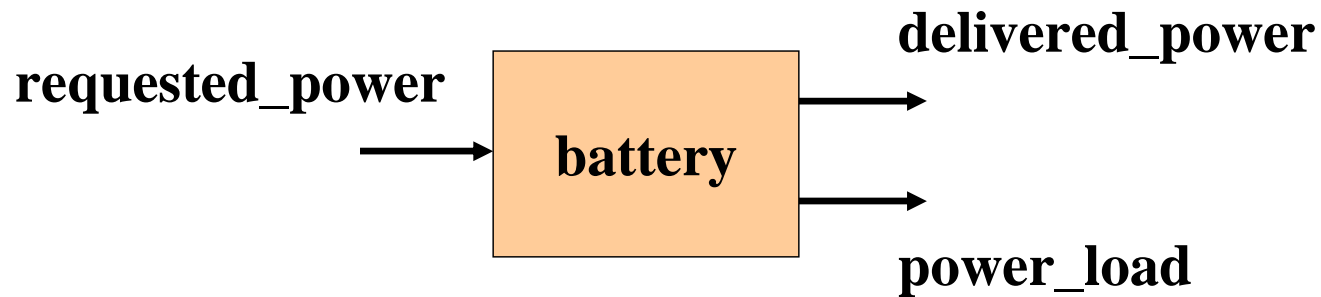
Batterie (1/3)

- Une batterie peut délivrer à la demande une certaine puissance

Batterie (2/3)

- A chaque cycle, les panneaux solaires chargent la batterie d'une unité ($\text{power_load} = \text{power_load} + 1$)
- Si la puissance requise est disponible ($\text{requested_power} \leq \text{power_load}$), la batterie délivre la puissance requise ($\text{delivered_power} = \text{requested_power}$) et la batterie se décharge d'autant ($\text{power_load} = \text{power_load} - \text{requested_power}$)
- Sinon, la puissance délivrée est nulle
- La puissance maximale de la batterie est 6
- Initialement, la batterie est complètement chargée

Batterie (3/3)



- 05 ➤ Développer l'opérateur « battery »
- 06 ➤ Le valider par simulation

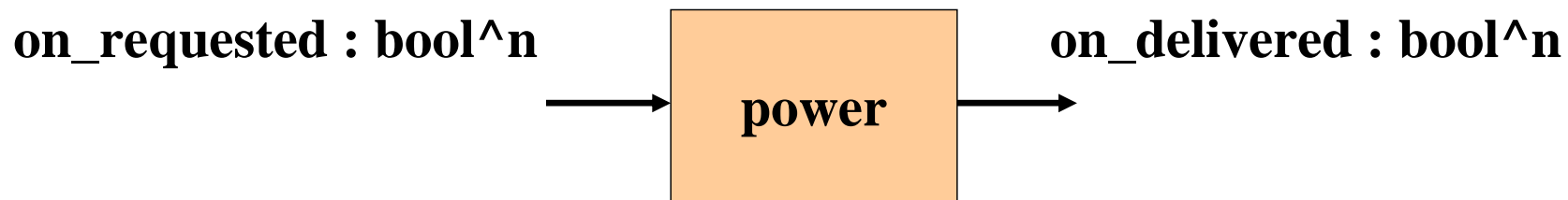
Validation de la batterie

- La charge de la batterie doit être toujours comprise entre 0 et sa charge maximale
- 07 ➤ Ecrire l'observateur de cette propriété
- 08 ➤ Prouver formellement cette propriété sur la batterie

Contrôle de puissance (1/2)

- Une batterie est incorporée dans un équipement de « contrôle de puissance »
- Un contrôle de puissance prend en entrée « n » demandes de puissance (`on_requested` de type `bool^n`)
- Il renvoie en sortie « n » commandes (« `on_realized` » de type `bool^n`)

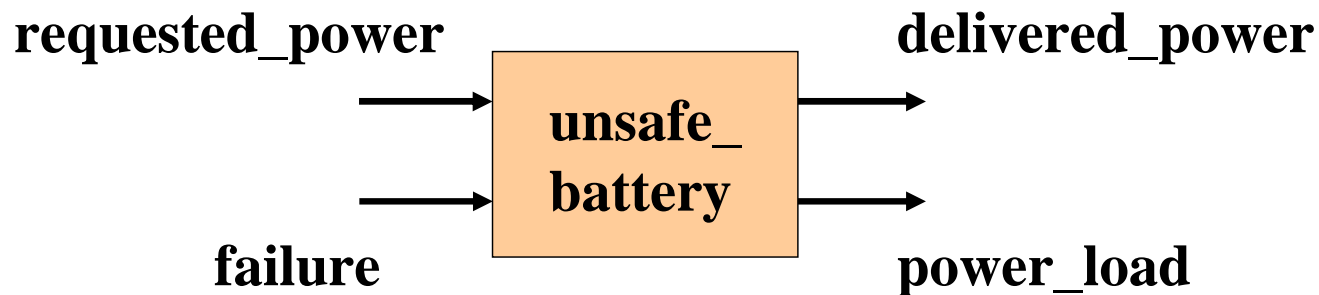
Contrôle de puissance (2/2)



- 09 ➤ Développer l'opérateur générique « power », pour une taille « n » quelconque (« n » étant un paramètre de l'opérateur)
- 10 ➤ Valider par simulation une instance n=4 de l'opérateur « power »

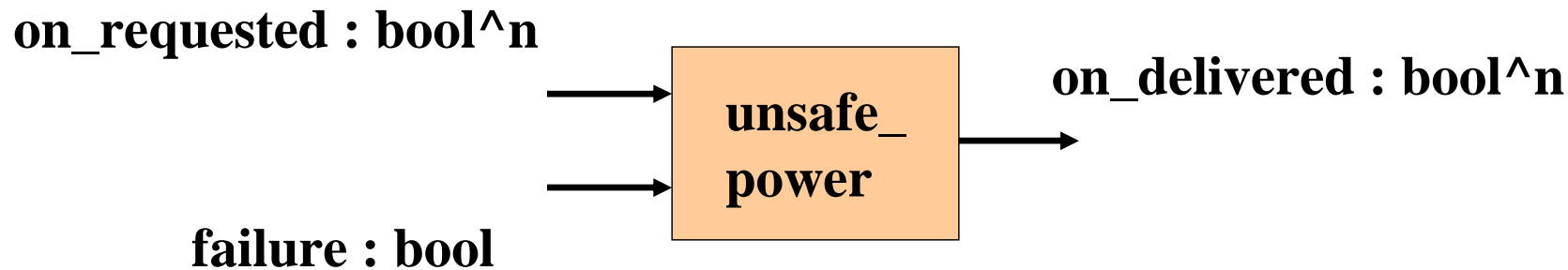
Batterie imparfaite

- Une batterie peut tomber en panne
- Une panne est modélisée par une entrée « failure »
- Lorsqu'une batterie est en panne, elle ne délivre plus aucune puissance



- 11** ➤ Développer l'opérateur « unsafe_battery »,

Contrôle de puissance imparfait

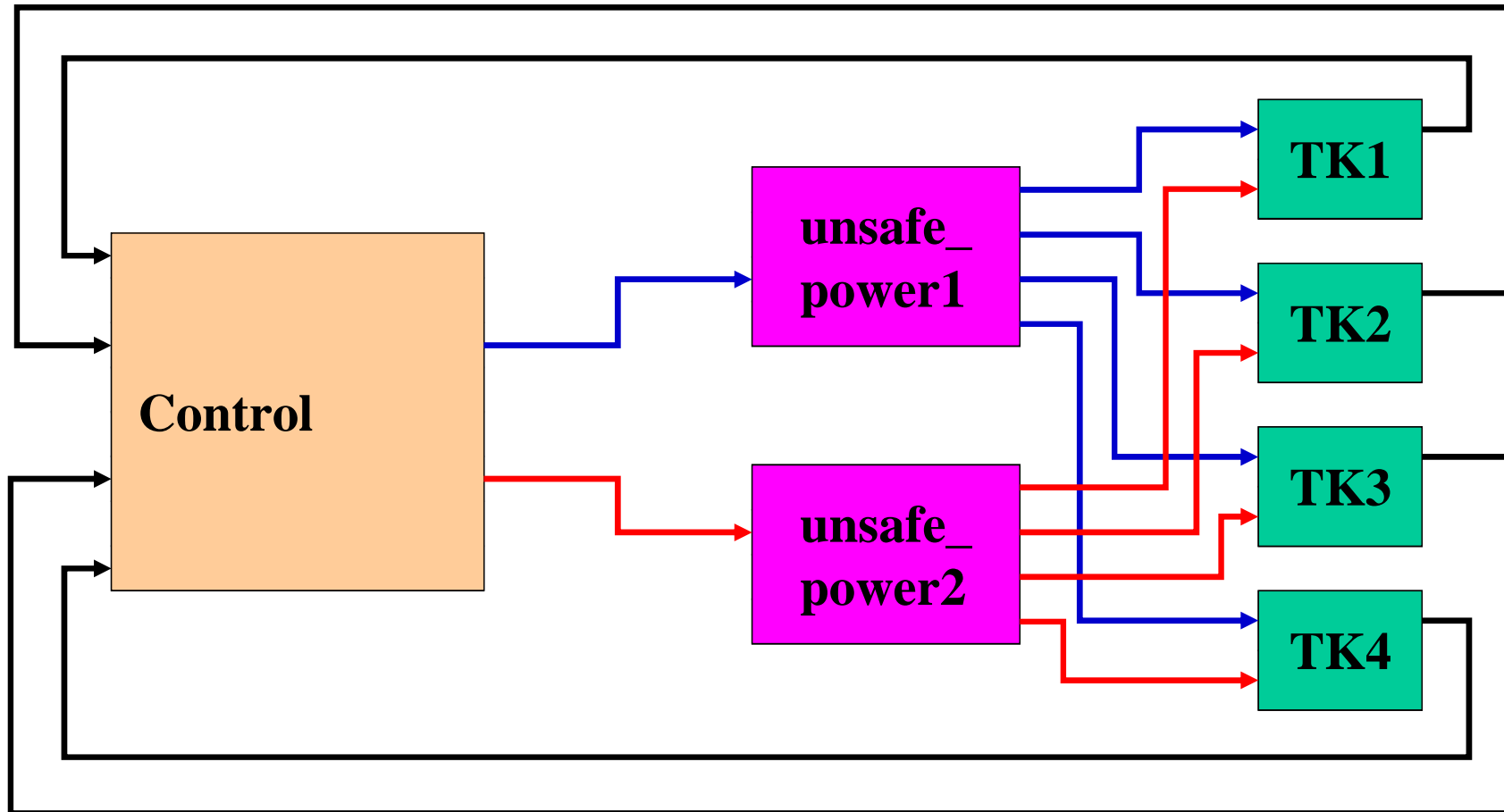


- 12 ➤ Développer l'opérateur générique « `unsafe_power` », pour une taille « `n` » quelconque (« `n` » étant un paramètre de l'opérateur)

Systeme

- Le système est composé:
 - ✓ De 4 couteaux correspondant à 4 liens
 - ✓ De 2 contrôles de puissance de taille 4
(chaque couteau étant donc relié à deux contrôles de puissance)
 - ✓ D'un contrôleur
 - ❖ Entrées: 4 états de liens (TK1_cut, TK2_cut, TK3_cut, TK4_cut: booléen)
 - ❖ Sorties: 2 commandes de 4 couteaux

Le système



- 13 ➤ Développer le contrôleur et le système
- 14 ➤ Valider par simulation
- 15 ➤ Prouver formellement sa correction