

# Culture Scientifique de Base en Sciences pour l'ingénieur

**Commander des procédés**

**L'Automatique**

## Nous voulons commander !!

Pourquoi ?? Un besoin, une nécessité

On veut améliorer :

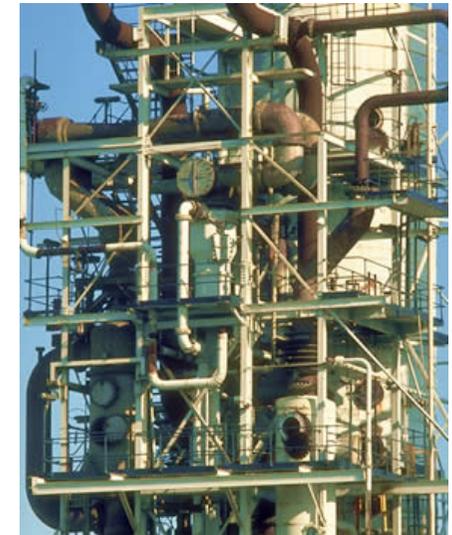
- ✓ Le comportement global nominal
- ✓ Les performances : (coût, vitesse, précision, qualité, etc...)
- ✓ Supprimer (ou atténuer) perturbations (défauts, usures, vieillissements, etc...)

# 1) Généralités

## 2) Les automatismes

## 3) L'automatique

# Commander quoi ?



Un peu de tout.....voire plus

# Commander comment ? Avec quoi ?



Des automates programmables



Des ordinateurs industriels

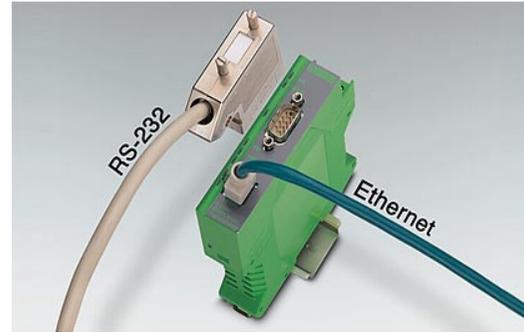
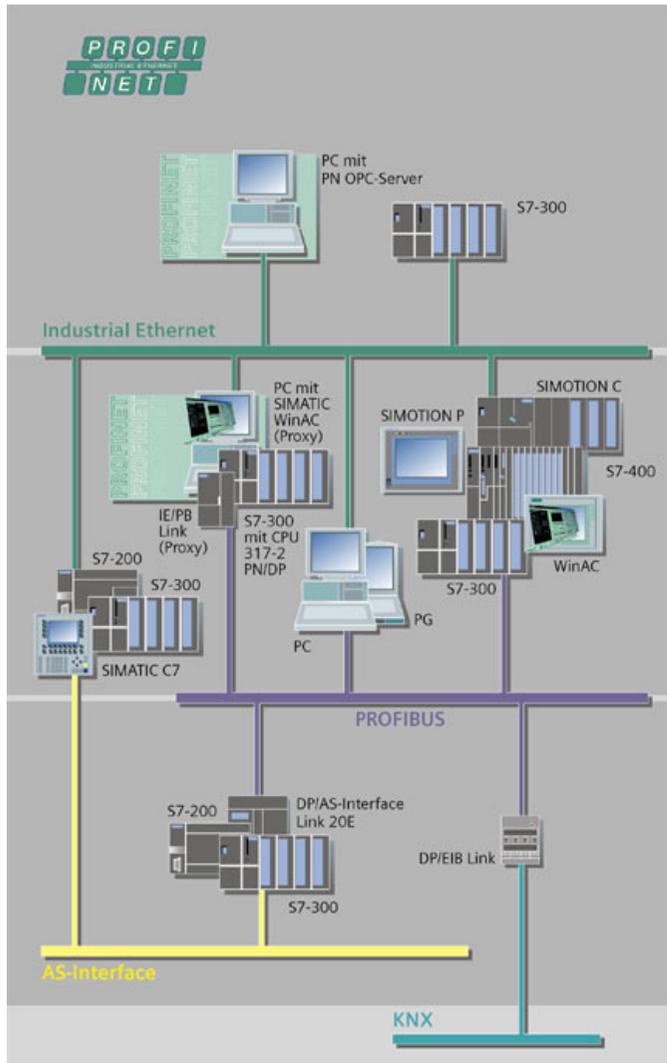


Des outils de l'informatique industrielle

- Il imagine les programmes des automates ou des ordinateurs.
- Il communique avec ceux-ci soit pour charger de nouveaux programmes, soit pour indiquer au programme en cours d'exécution les paramètres qu'il souhaite voir respectés (température...).
- Il y a une interface entre l'homme et les machines (ordinateur, pupitre de commande, matériel spécifique à base d'informatique industrielle).



# Comment tout cela communique-t-il ?

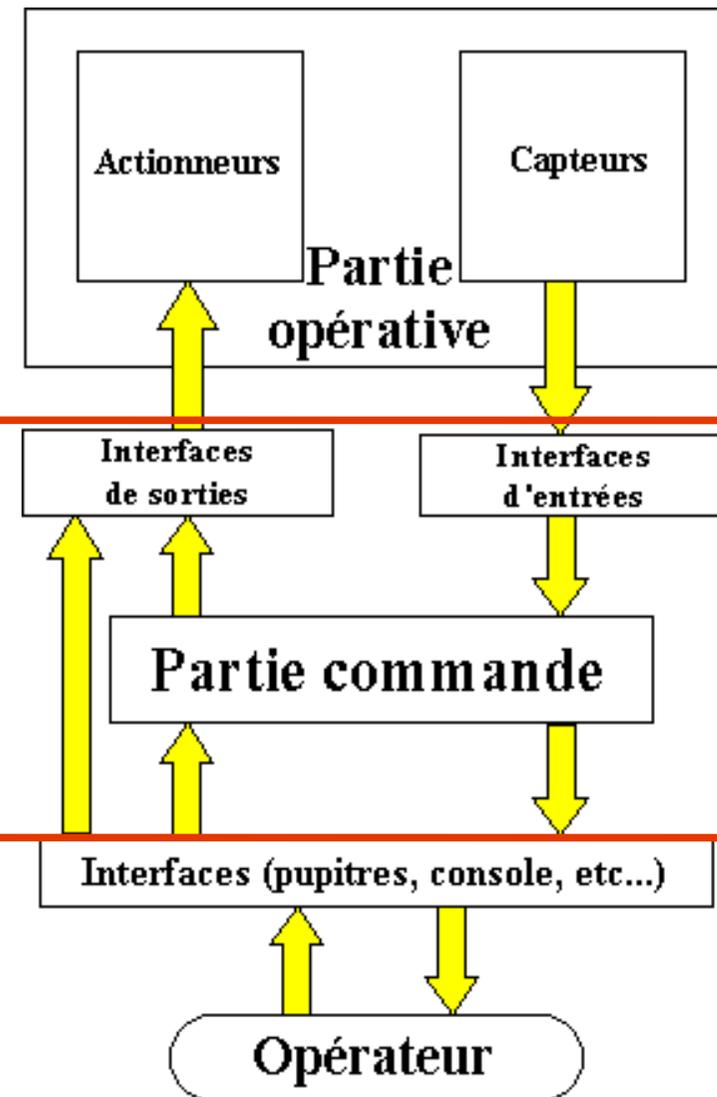


Par des réseaux de terrain ou réseaux industriels

La partie **Opérative** ou système à commander. C'est aussi ce que l'on appelle le procédé ou le processus. Cela peut être une machine à laver ou une chaîne de production...

La partie **Commande** qui doit fournir les ordres au système à commander.

**L'Interface** Homme Machine : IHM.



## Les deux

Le **numérique** pour :

- La programmation
- L'acquisition et le traitement des données
- La communication

.....

L'**analogique** pour :

- Les actionneurs
- Les capteurs
- Fournir l'énergie nécessaire

.....

Elle possède des boutons : marche/arrêt, programmateur qui sépare l'action en plusieurs étapes, etc. Au début d'une étape (événement binaire) une action est déclenchée. A la fin d'une étape (autre événement binaire) une action est arrêtée. C'est un système à événements discrets : **un automatisme**.

La température demandée est de 40°C, un dispositif se met en fonctionnement pour assurer cette température pendant tout le temps désiré. C'est un système à événements continus : **un asservissement (ou régulation)**. Cette partie est souvent traitée en échantillonné par un petit microprocesseur correctement programmé.

Il y a deux grandes classes de problèmes :

Les systèmes déclenchés sur des événements

 **AUTOMATISME**

Les systèmes analogiques

 **AUTOMATIQUE**

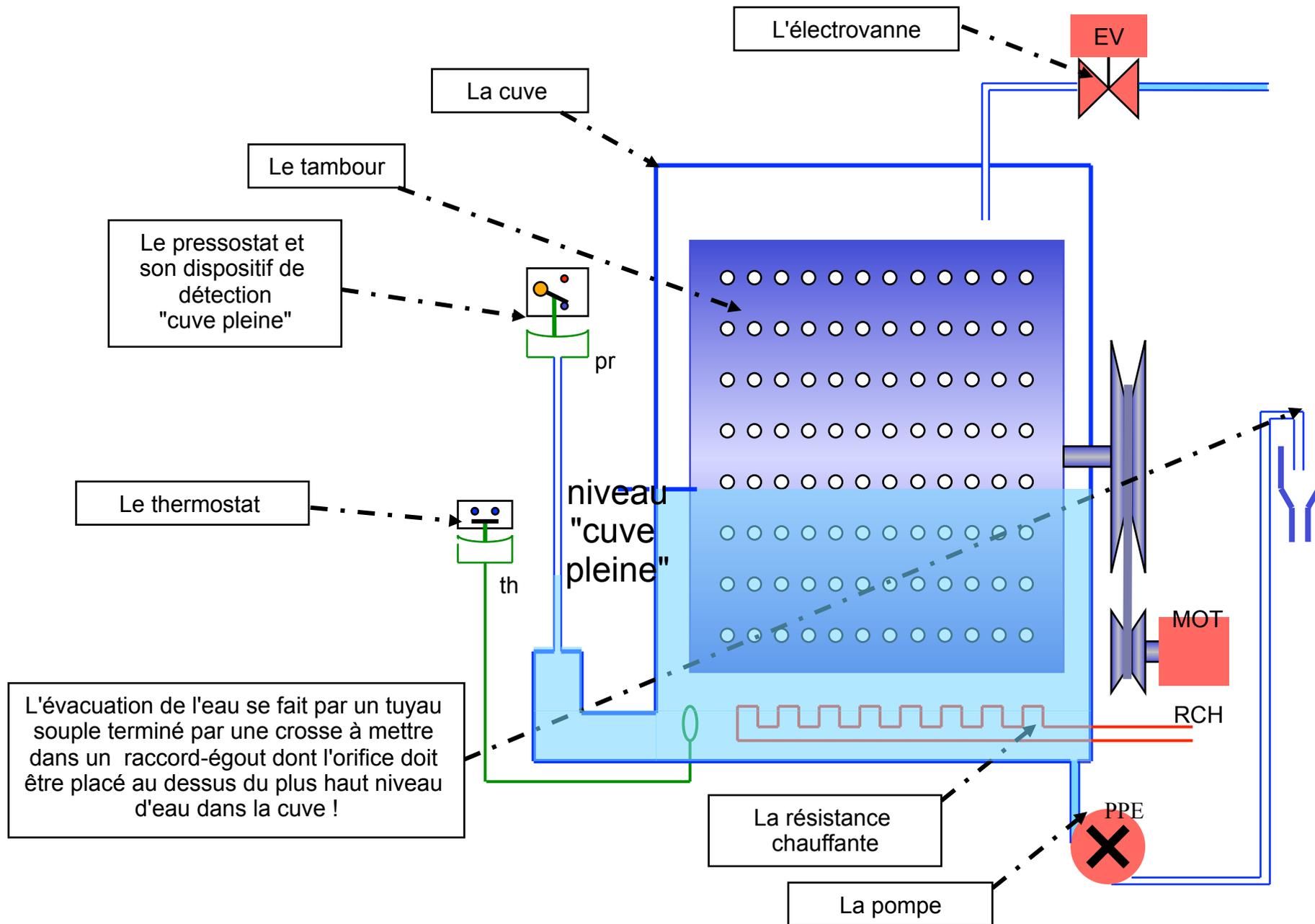
Dans ma machine à laver, il y a des échanges d'informations :

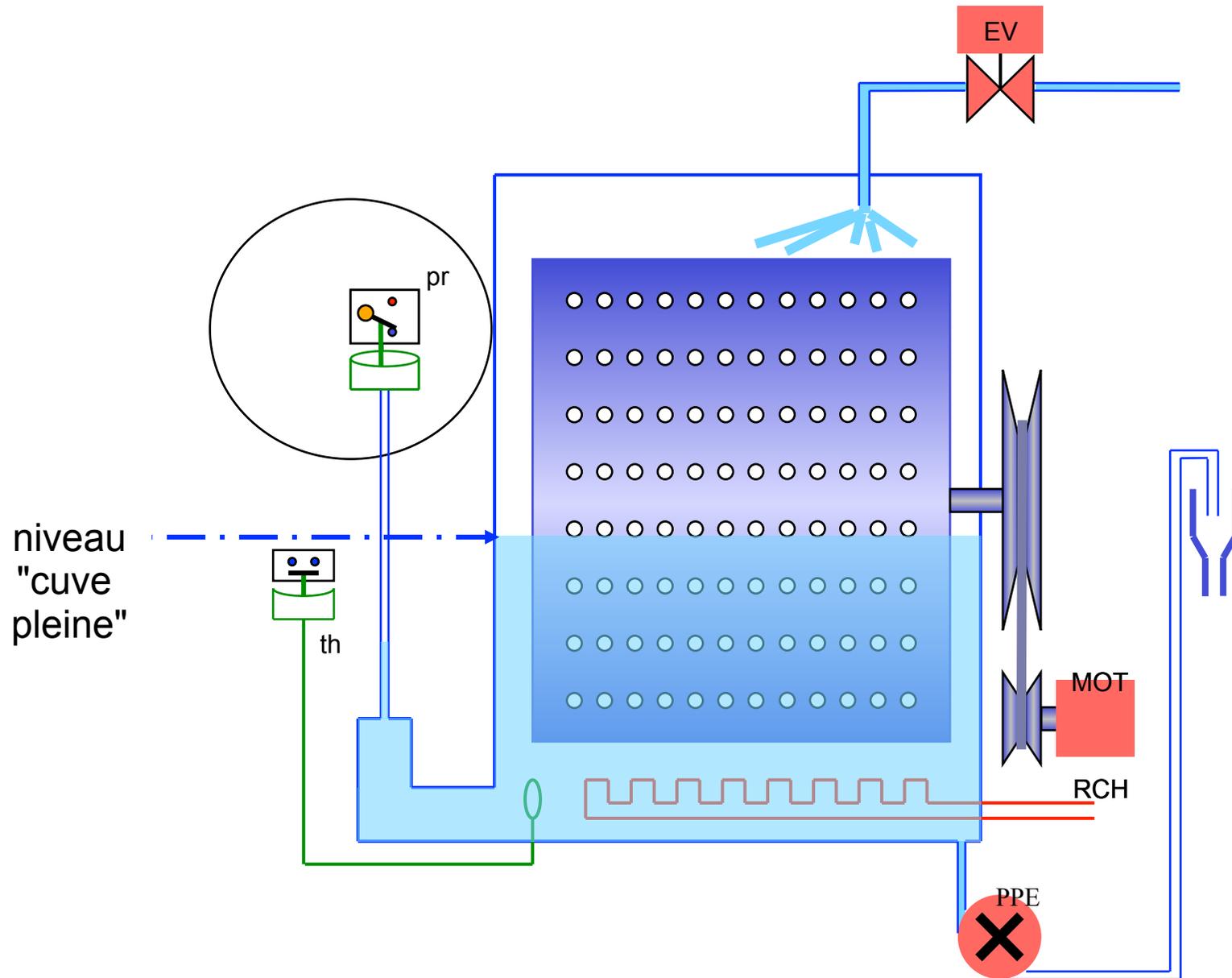
Le programmeur (automatisme) donne l'ordre au régulateur de se mettre en action.

Lorsque la température de 40°C est atteinte, le régulateur le fait savoir au programmeur (indication binaire) pour qu'il puisse continuer son action.

Un système automatisé comprend les deux types de systèmes : automatismes et régulateurs. Le fonctionnement d'un appareil simple ou complexe (avion,...) implique beaucoup de systèmes des deux types qui fonctionnent en coordination.

*Dans le domaine industriel, à l'échelle d'une fabrication (usine) il y a un grand nombre d'automatismes et de régulateurs : nous avons un S.A.P. (Système Automatisé de Production).*





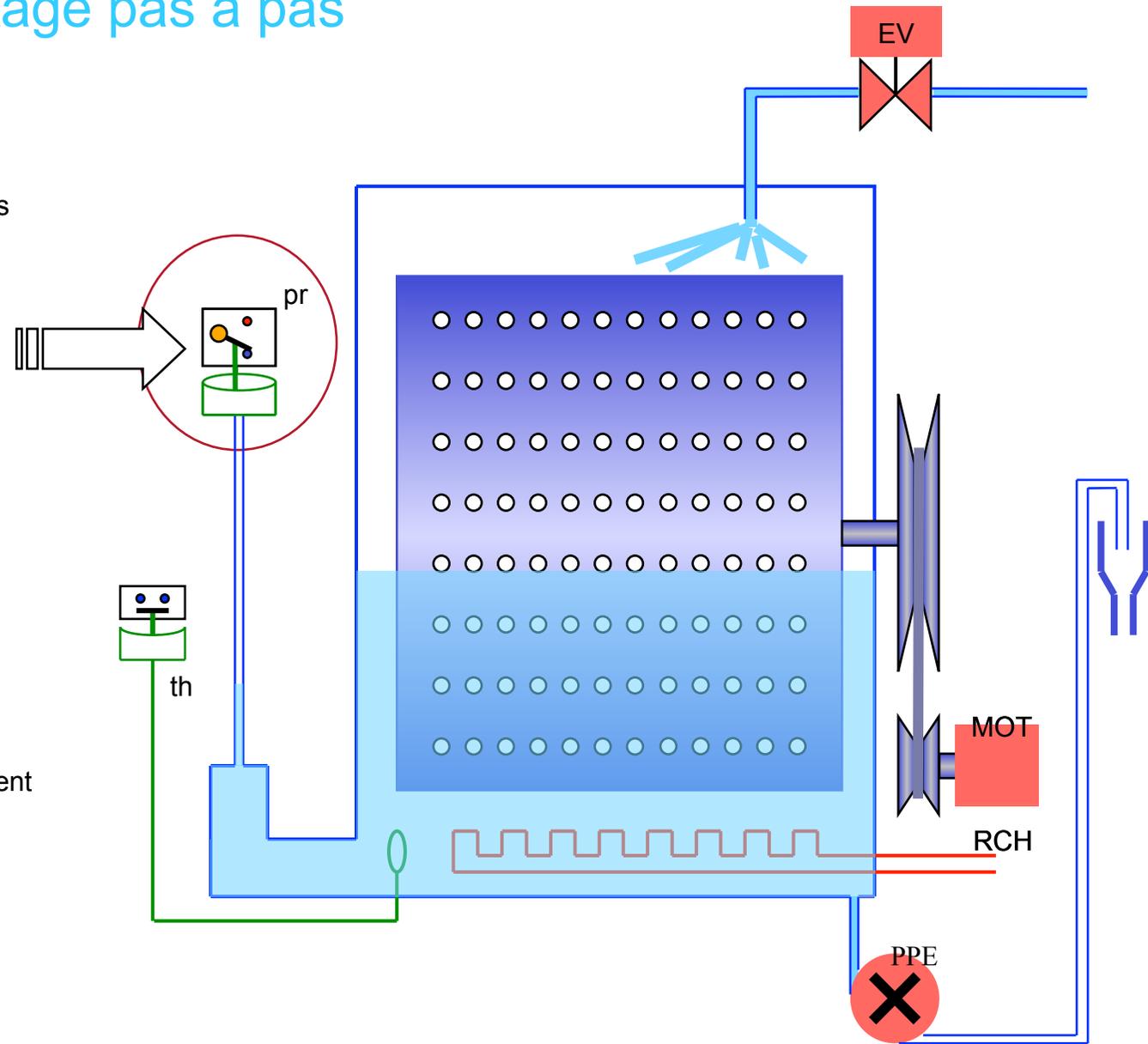
## Remplissage : pilotage pas à pas

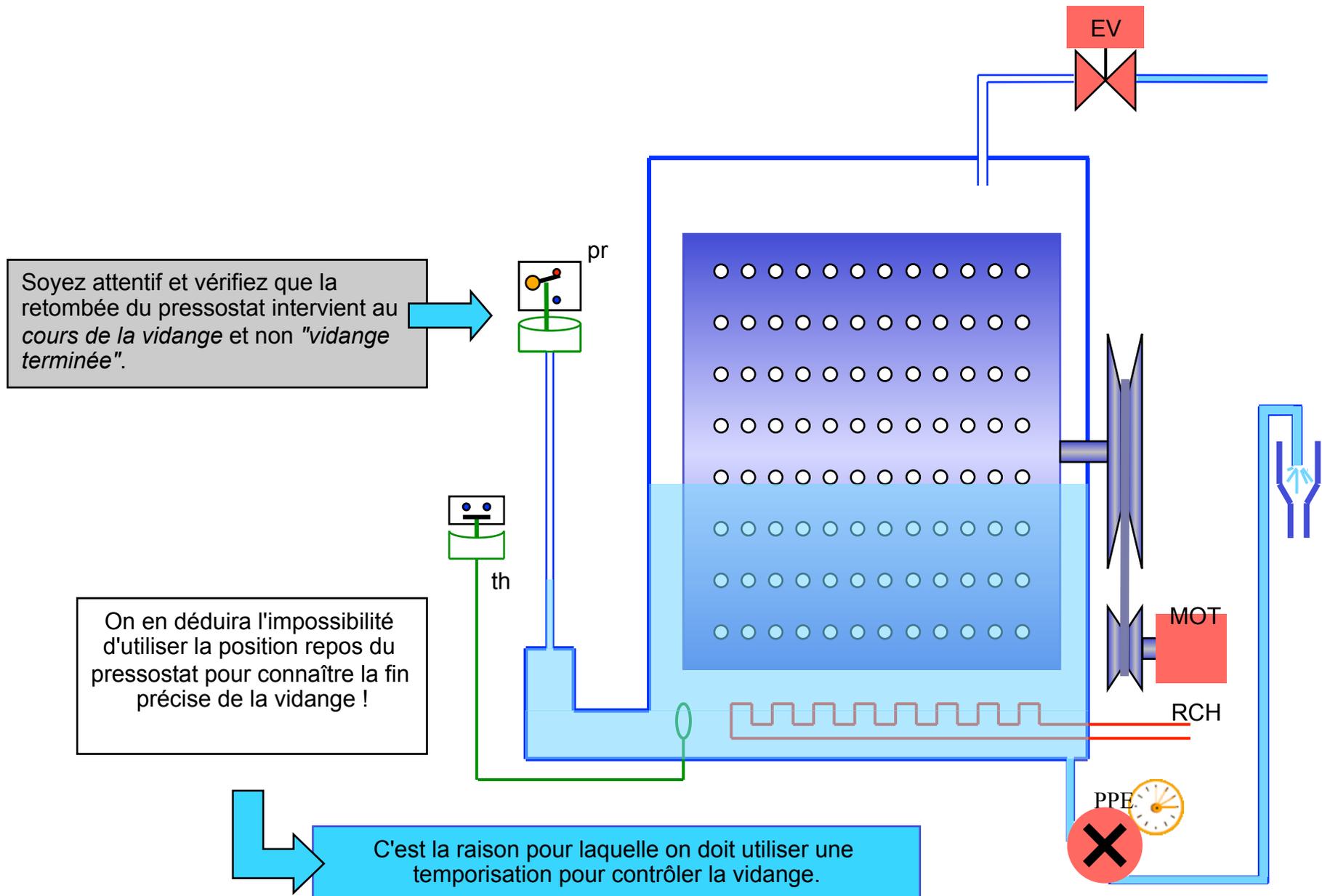
Par un clic, ouvrez l'électrovanne

En montant dans le fond de la cuve, l'eau vient d'isoler le puits d'accès au pressostat : un certaine quantité d'air y est emprisonnée.

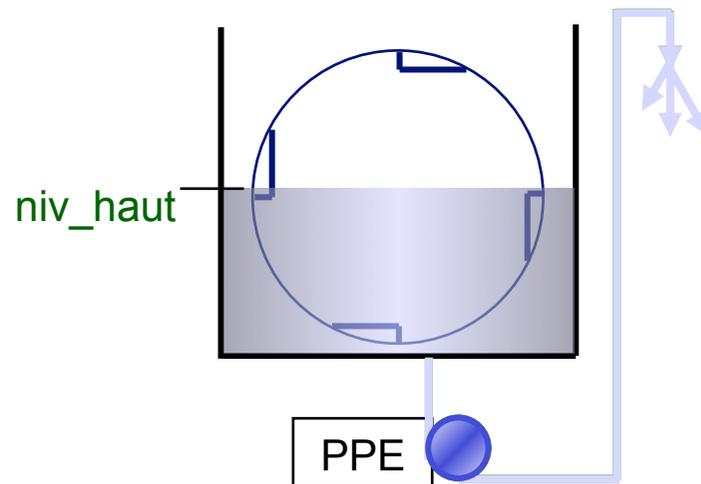
Observez bien la fin du remplissage de cette cuve. Dans le puits du pressostat l'eau ne peut monter aussi vite que dans la cuve : au dessus d'elle la pression de l'air emprisonné augmente !

La pression sous la membrane du pressostat vient de faire basculer le contacteur / inverseur





# Machine à laver : le fonctionnement simplifié



Remplissage (jusqu'à niv\_haut)

Brassage dans un sens =  $t_1$  secondes

Brassage dans l'autre sens =  $t_2$  secondes

Vidange =  $t_3$  secondes

Il faut découper le problème en sous-problèmes, certains sont du domaine de l'automatisme, d'autres de celui de la régulation.

On cherchera à décomposer en petites parties. Chaque partie (ou sous-système) est représenté par une boîte noire à laquelle on associe des signaux d'entrées et de sorties qui affectent cette boîte.

Le but est d'optimiser le comportement de chaque sous-système vis à vis de ces signaux.

1) Généralités

**2) Les automatismes**

3) L'automatique

Ils concernent les systèmes à événements discrets :

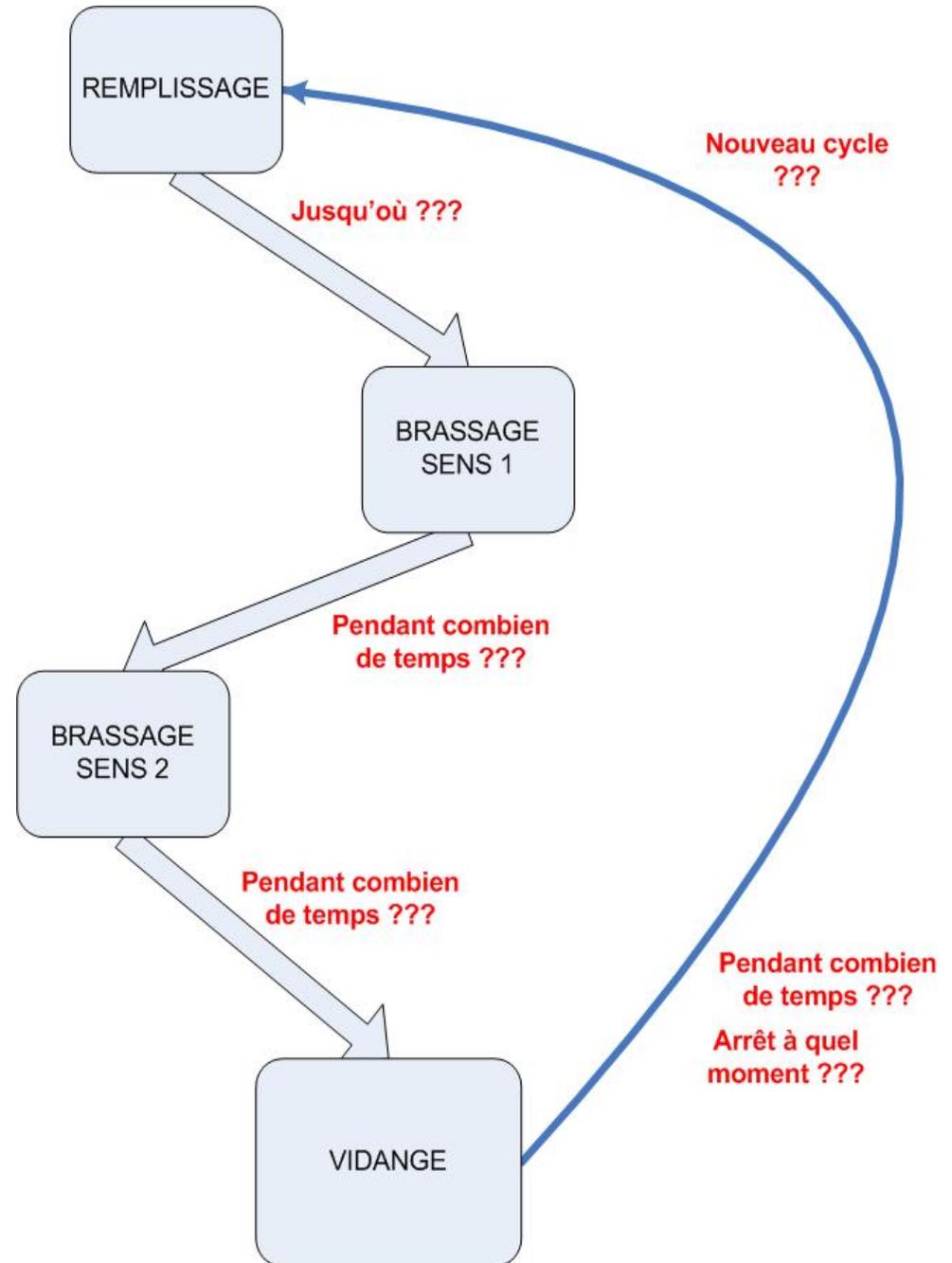
- ✓ Ils réagissent sur un événement qui peut intervenir n'importe quand (exemple une alarme).
- ✓ L'évènement est de type binaire et sa présence déclenche une action.
- ✓ Les signaux entre partie opérative et partie commande sont de type binaire.
- ✓ Ces systèmes sont en majorité gérés par des automates programmables ou un ordinateur

plusieurs questions se posent :

comment décider le passage  
d'une action à une  
autre ????

Comment modéliser ce  
problème ?

Réponse : Réseau de Pétri



Le fonctionnement d'un système à événements discrets peut se décrire par un graphique composé de :

- **Places** qui symbolisent l'état dans lequel se trouve le système
- **Transitions** qui symbolisent les événements discrets susceptibles d'intervenir
- **Arcs** : lignes orientées qui relient les places et les transitions



PLACE



TRANSITION



ARC

Interrupteur marche/arrêt

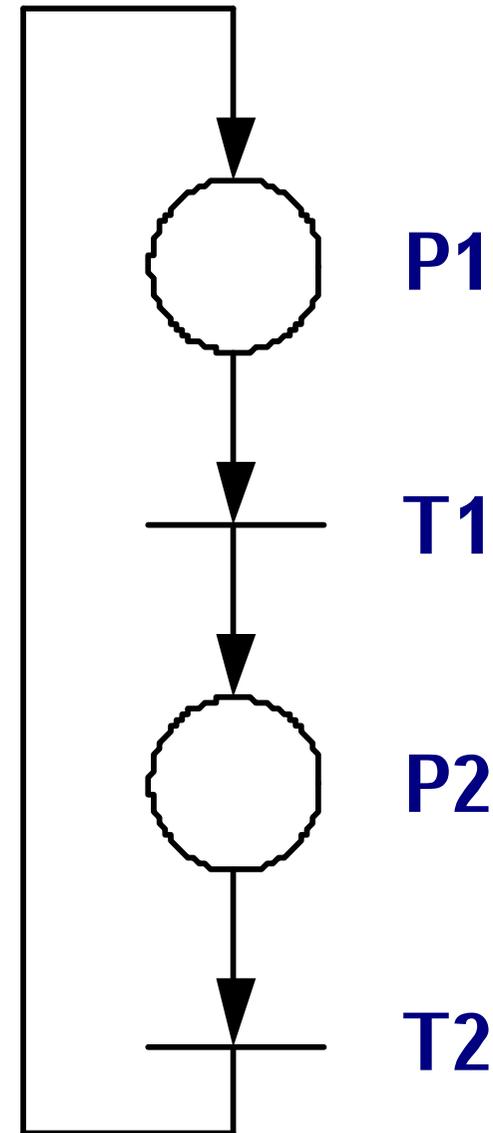
Places (2 états) :

- P1 : le système est à l'arrêt
- P2 : le système est en fonctionnement

Transitions (2 évènements) :

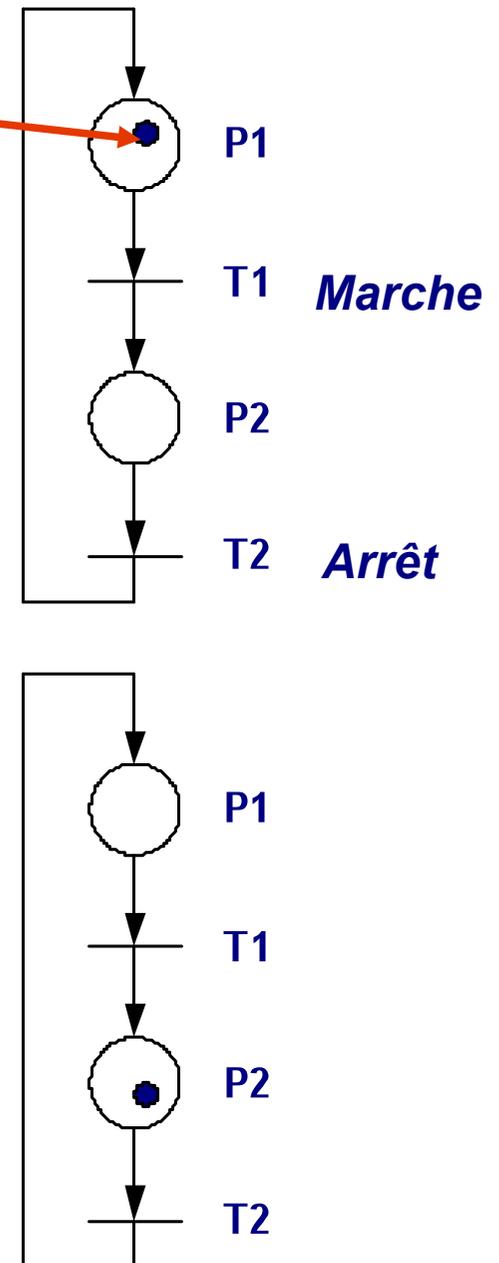
- T1 : le bouton marche est enclenché
- T2 : le bouton arrêt est enclenché

**Exercice** : donner le RDP de la machine à laver en précisant la signification des places et transitions

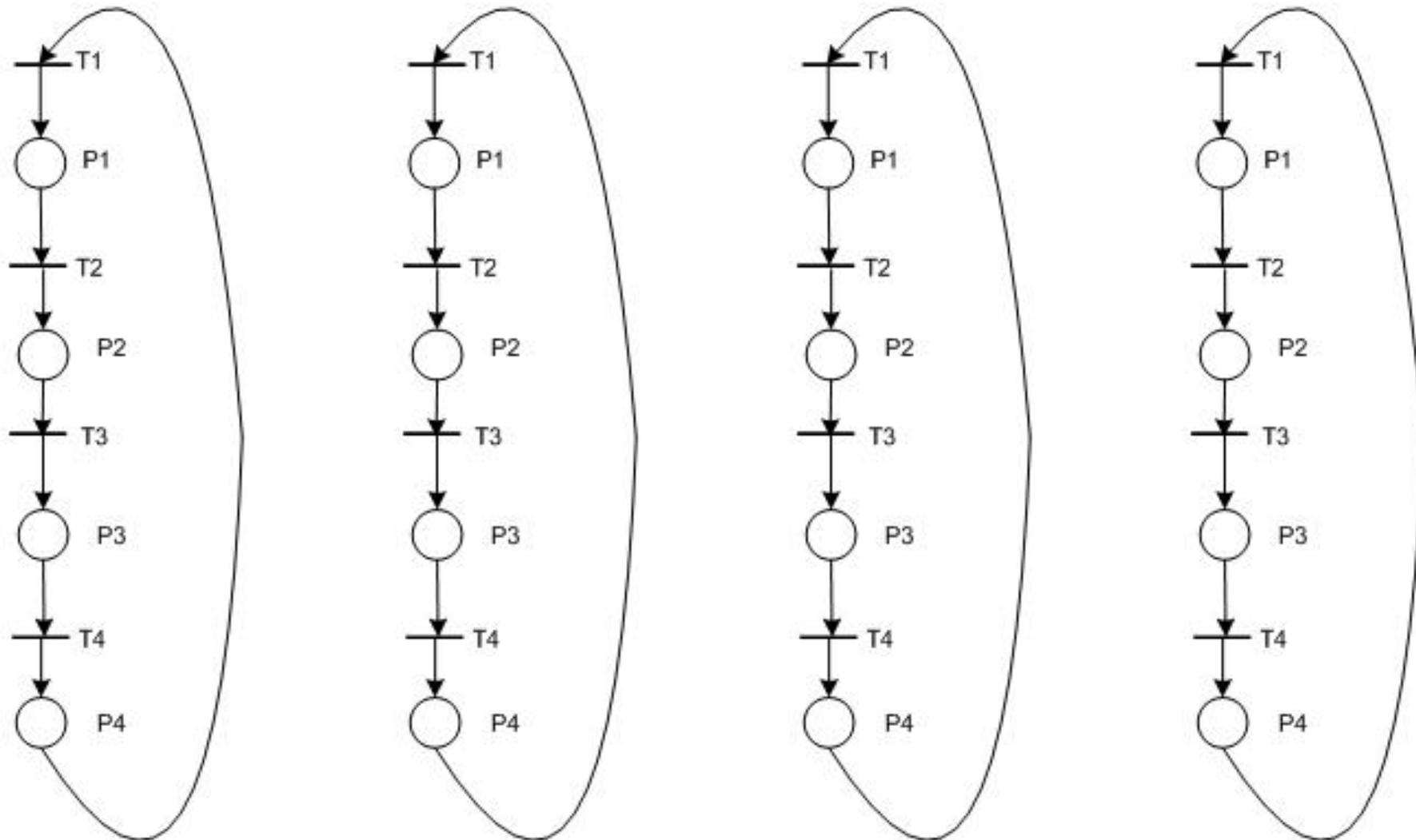


La présence d'un jeton signifie que cet état est actif

- Pour passer d'un état à un autre (d'une place à une autre) il faut franchir une transition,
- **Conditions** de franchissement : pour qu'une transition soit franchie il faut que toutes ses places amont contiennent au moins 1 jeton et que l'évènement associé à la transition se produise.
- **Effet** du franchissement : lorsqu'une transition est franchie, chaque place amont perd un jeton et chaque place aval en récupère un.



# Un exercice : compléter l'évolution du RDP Machine à laver



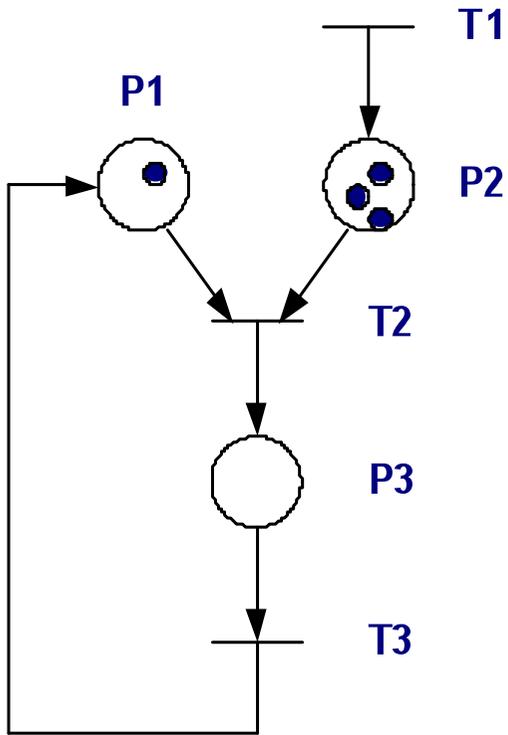
Un atelier est constitué d'une machine de coupe et d'un stockage. Quand une commande arrive, la pièce est stockée et la coupe est effectuée lorsque la machine est disponible.

## Les places :

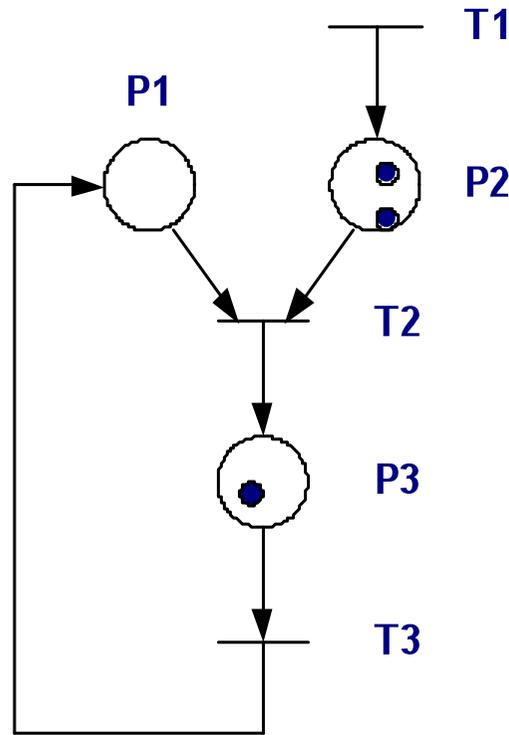
- P1 : la machine est en attente.
- P2 : des commandes sont en attente.
- P3 : la machine traite une commande.

## Les transitions :

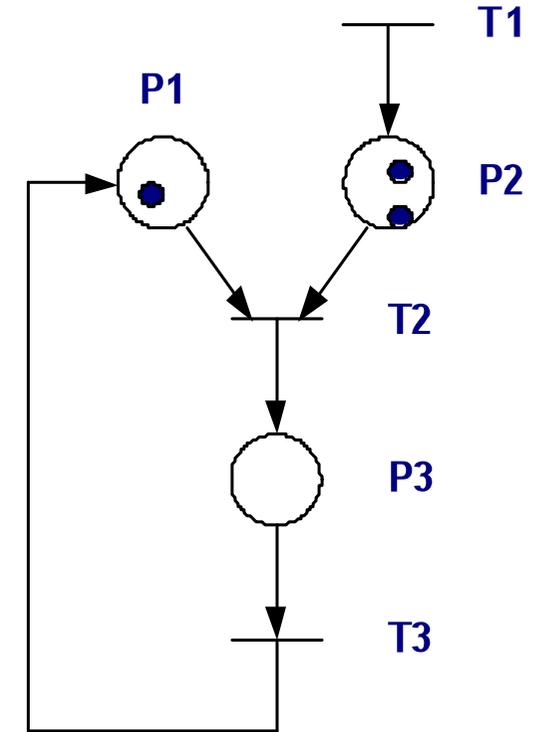
- T1 : arrivée d'une commande.
- T2 : prise d'une commande et début de la découpe.
- T3 : fin de la découpe.



3 commandes en attente,  
machine disponible



Traitement d'une  
commande



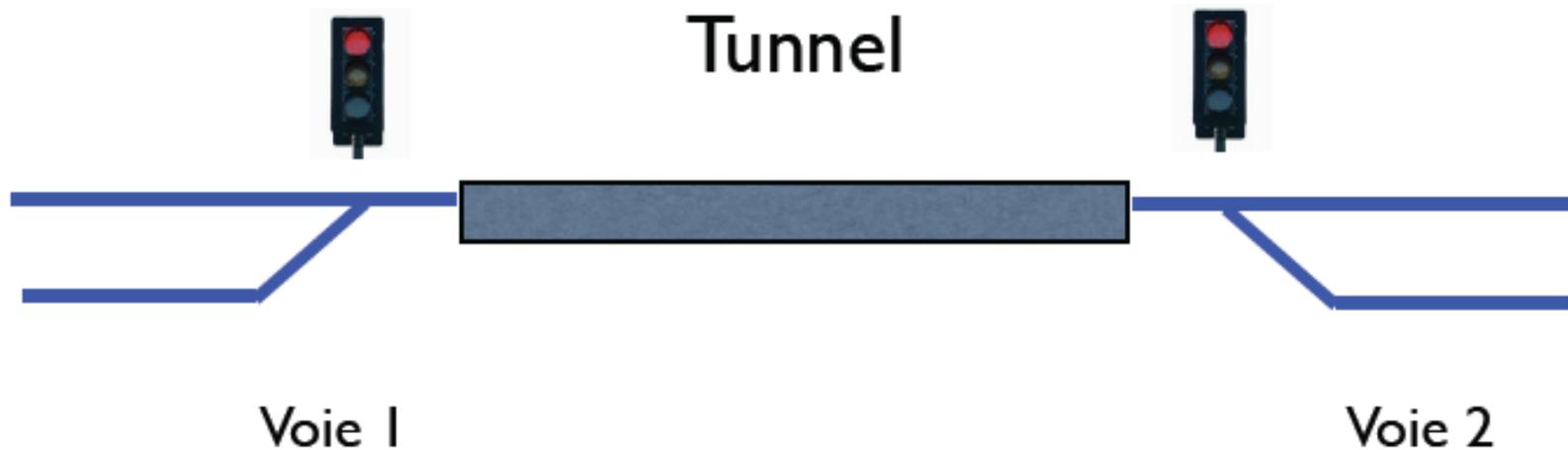
1 Commande traitée

La suite : voir exercice 1

*Deux lignes de chemin de fer utilisent un même tunnel qui ne comporte qu'une seule voie. Aux extrémités des voies sont situés des détecteurs de présence de trains et des aiguillages permettent de sélectionner la voie utilisatrice.*

## *Fonctionnement :*

- ✓ *La circulation des trains peut avoir lieu dans les deux sens*
- ✓ *En amont les lignes sont régulées pour qu'il n'y ait sur chacune d'elles qu'un seul train*
- ✓ *Quand un train est détecté on positionne correctement les aiguillages et les éventuels trains de l'autre ligne sont mis en attente (feux de signalisation)*
- ✓ *En cas de conflit, la voie 1 est prioritaire sur la voie 2*



T1 : arrivée d'un train

P2 : train en attente

T2 : voie utilisable

P3 : position aiguillage

T3 : Aiguillage prêt

P4 : passage du train

T4 : le train est passé



T5 : arrivée d'un train

P6 : train en attente

T7 : voie utilisable

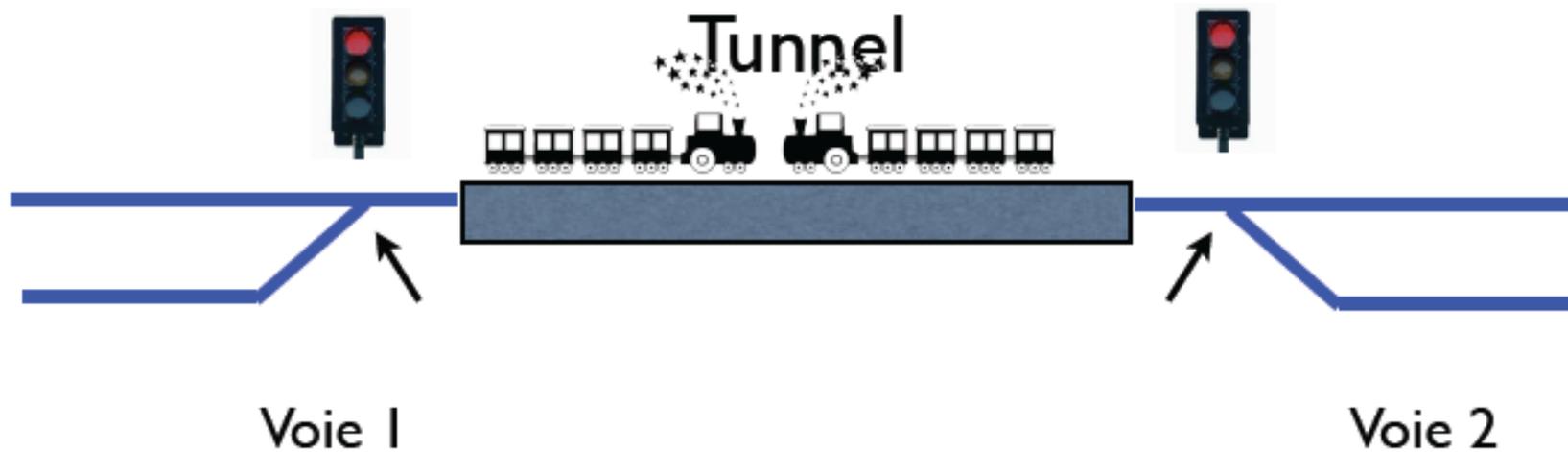
P7 : position aiguillage

T8 : Aiguillage prêt

P8 : passage du train

T9 : le train est passé





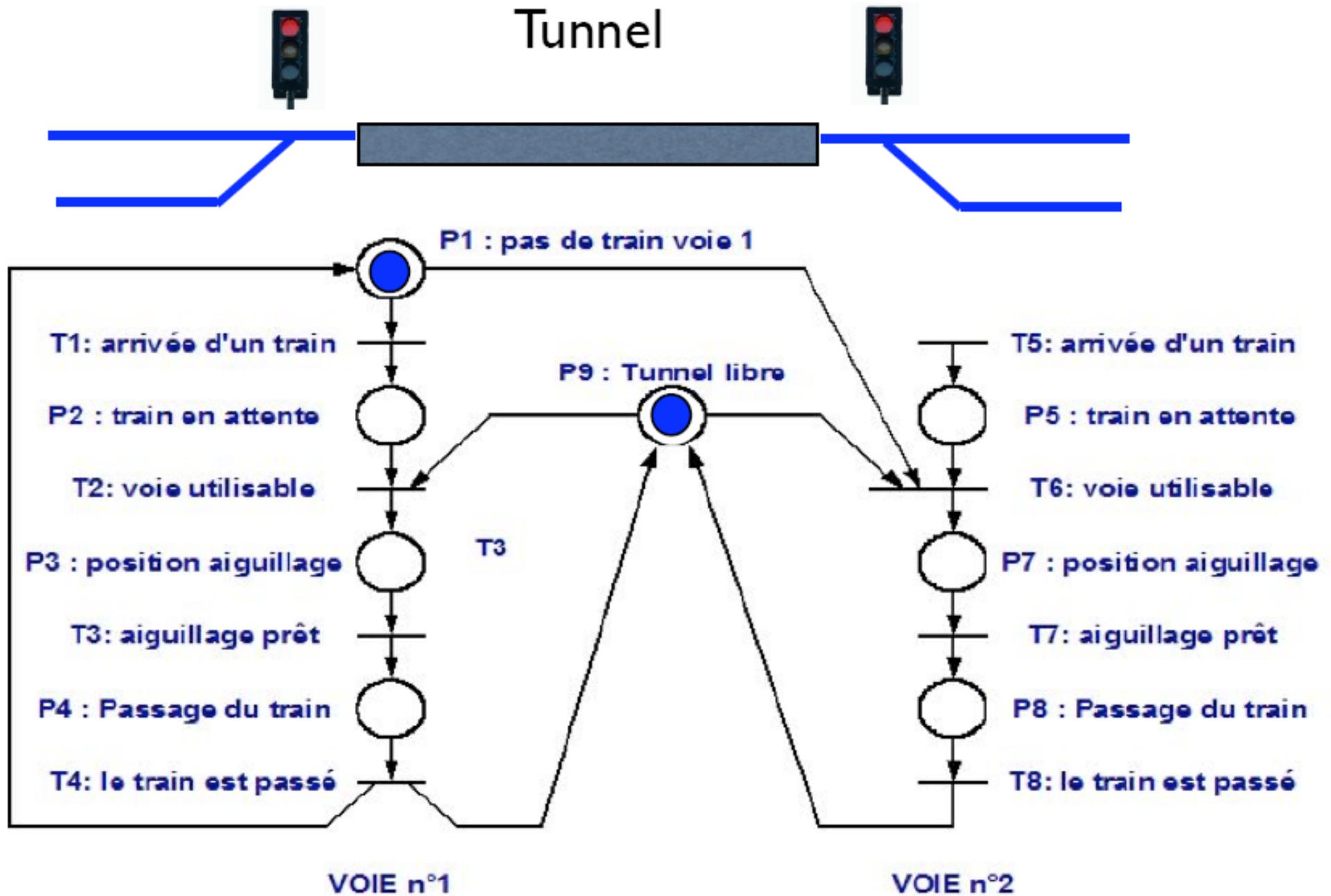
T1 : arrivée d'un train  
 P2 : train en attente  
 T2 : voie utilisable  
 P3 : position aiguillage  
 T3 : Aiguillage prêt  
 P4 : passage du train  
 T4 : le train est passé



T5 : arrivée d'un train  
 P6 : train en attente  
 T7 : voie utilisable  
 P7 : position aiguillage  
 T8 : Aiguillage prêt  
 P8 : passage du train  
 T9 : le train est passé



**collision !!!**



- RDP : outil qui permet une modélisation (plus ou moins fine selon la complexité voulue)
- Cet outil est une aide à la conception et à la réalisation
- Cet outil permet aussi de garantir une sûreté de fonctionnement (défauts)
- D'autres outils plus ou moins semblables existent et sont parfois plus adaptés au caractère spécifique de certains automatismes

- 1) Généralités
- 2) Les automatismes
- 3) L'automatique**

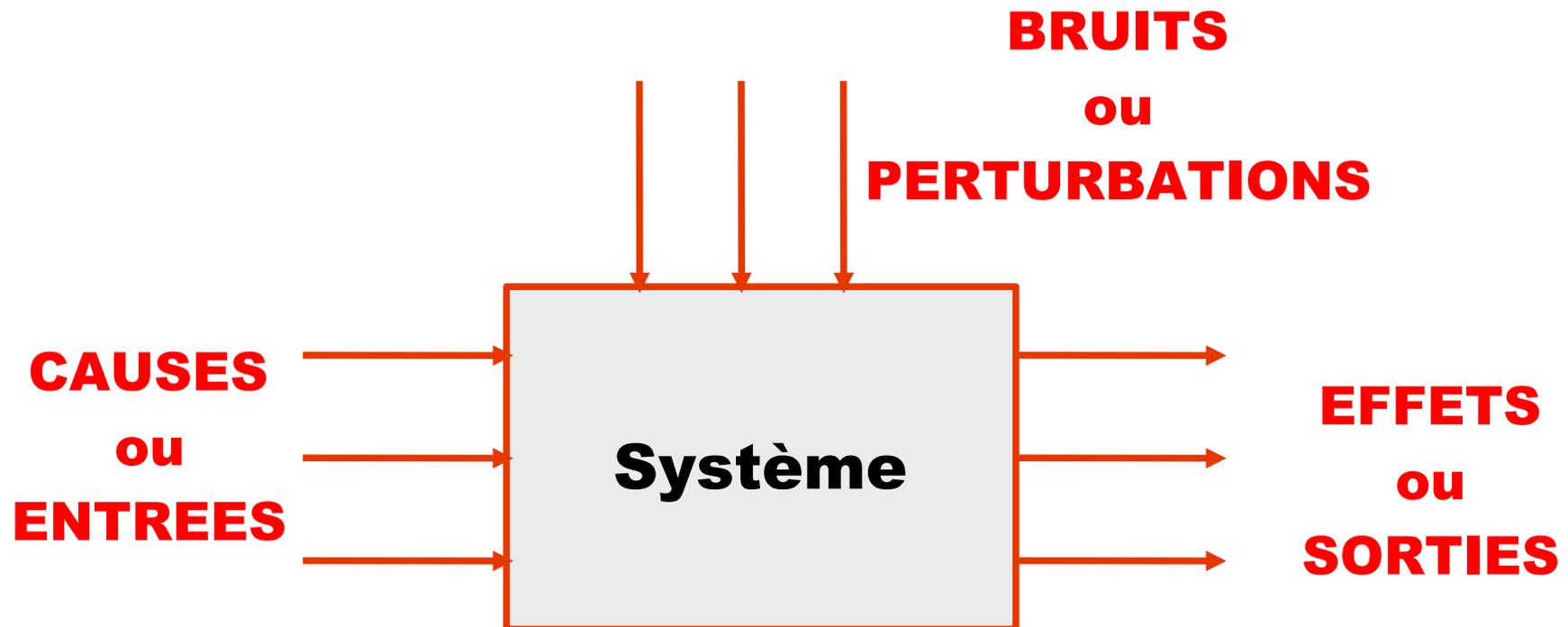
Concerne les systèmes continus :

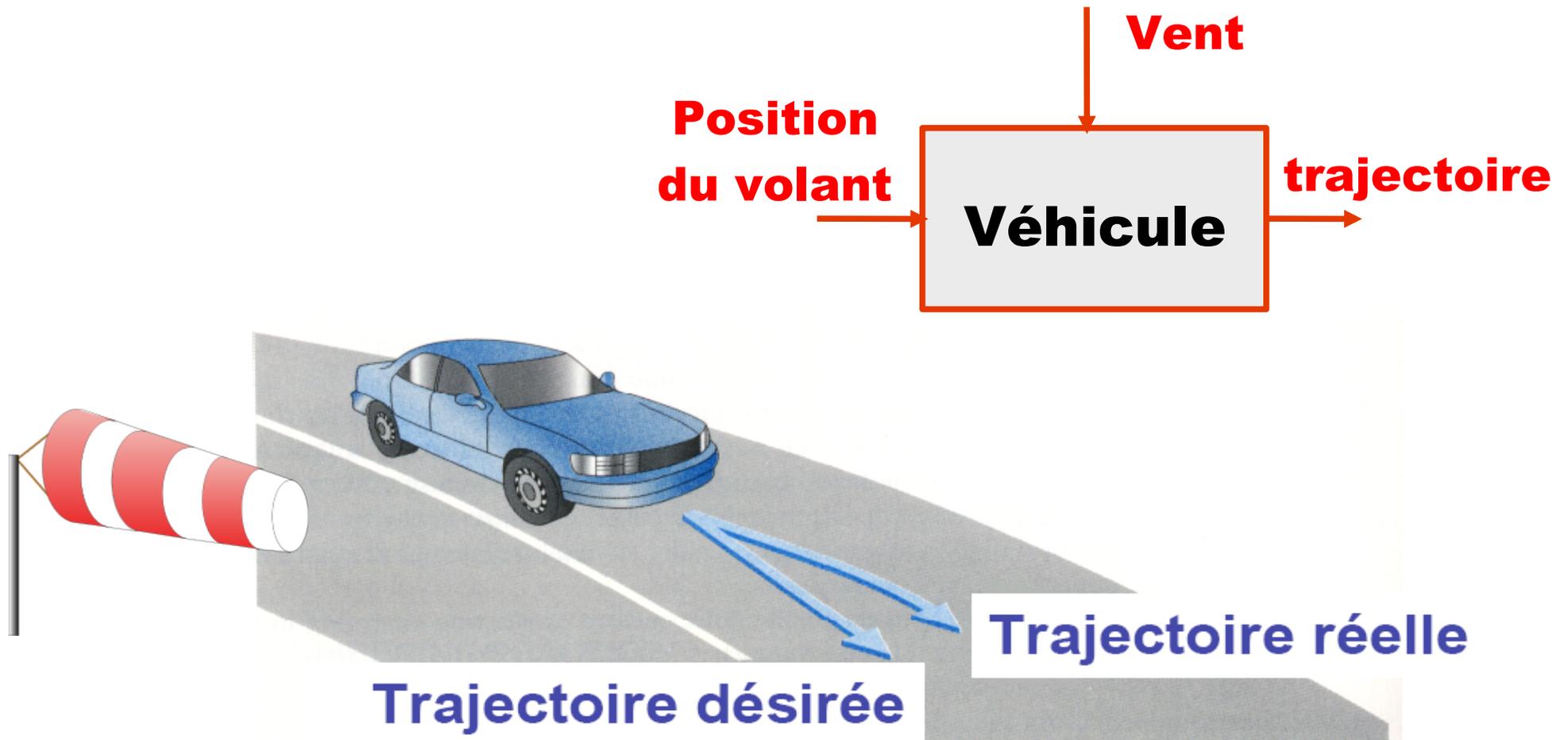
- ✓ Des grandeurs sont mesurées en permanence (température,...).
- ✓ Pour traiter les signaux, on utilise souvent l'échantillonnage et le numérique.
- ✓ Les signaux entre partie opérative et partie commande sont en majorité de type continu échantillonné.
- ✓ Ces systèmes sont généralement gérés par des ordinateurs, des cartes spécifiques à base de microprocesseur ou de DSP (processeur de traitement du signal). Certains Automates Programmables ont souvent une partie qui permet d'effectuer de petites actions de cette catégorie.



Un système est un dispositif physique mettant en jeu un ensemble de phénomènes.

Du point de vue extérieur, un système est soumis à des signaux d'entrées : les **causes**. Il y aura des effets : les **sorties**. Son fonctionnement peut être perturbé par des bruits ou perturbations.





C'est ce qui se produit si le volant est maintenu en position fixe : pas d'intervention du conducteur, le système est dit en **Boucle Ouverte (BO)**.

L'exemple précédent illustre la boucle ouverte, la commande (volant) est fixée et la trajectoire espérée peut être perturbée par le bruit (vent).

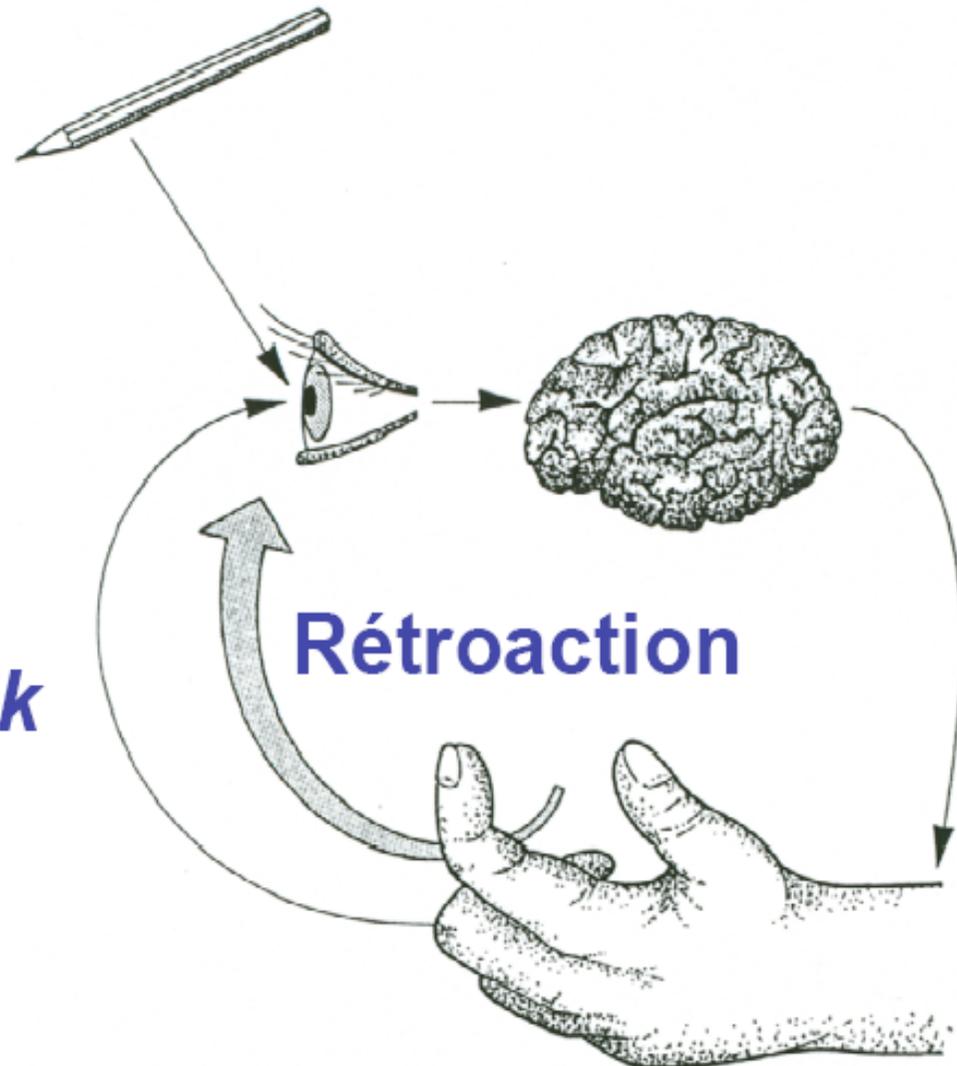
Si le conducteur intervient il se produit plusieurs choses :

- Il se rend compte que la trajectoire n'est pas correcte : mesure de la trajectoire par un **capteur**, ici l'oeil du conducteur.
- Il corrige la position du volant en fonction de ses observations : il joue un rôle de **correcteur**. Il y a commande de la trajectoire par un fonctionnement en boucle fermée.

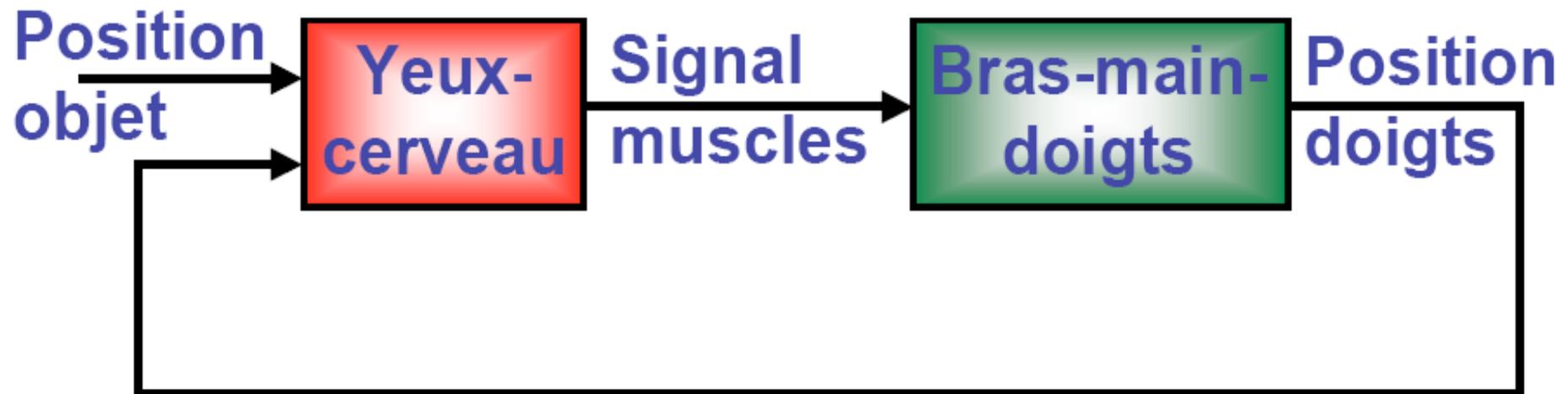
Nous avons un exemple élémentaire d'un **asservissement**, d'une **Boucle Fermée (BF)**.

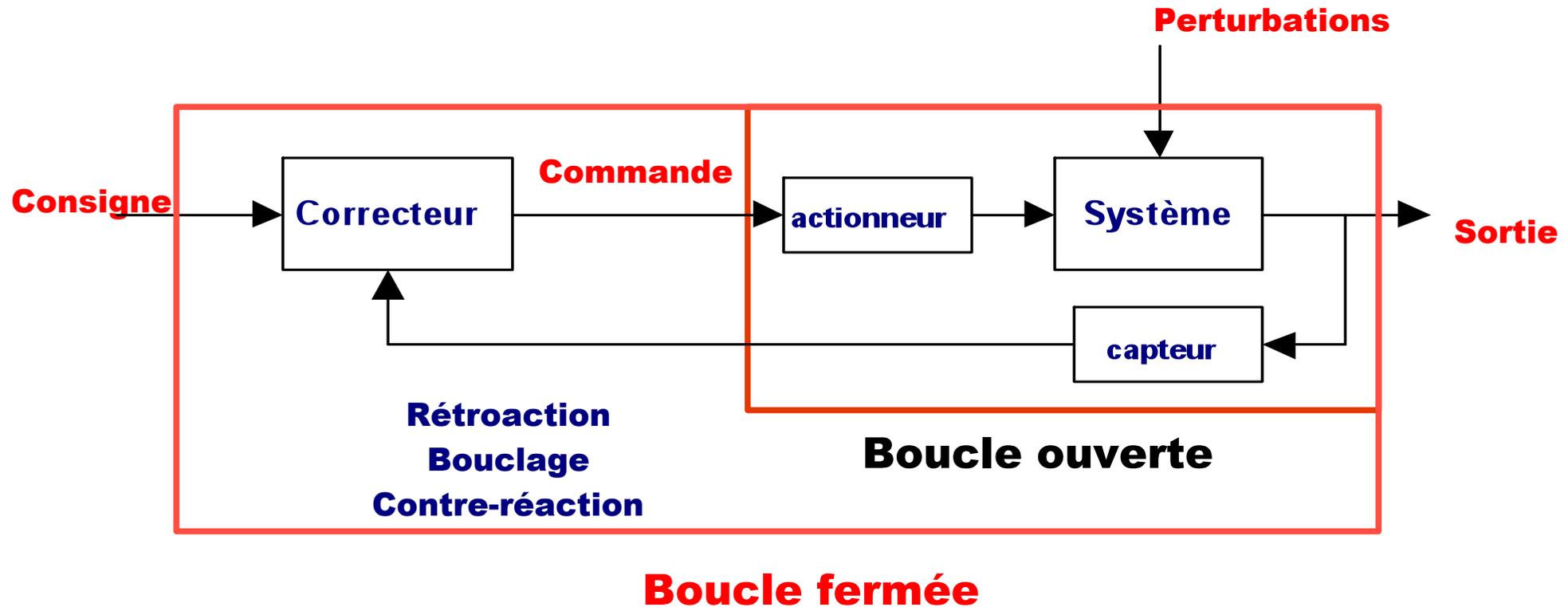
## ILLUSTRATION DU PRINCIPE

Principe de base:  
rétroaction, *feedback*



## ILLUSTRATION DU PRINCIPE





Remarque : Les actionneurs et capteurs sont considérés dans la partie opérative c'est à dire le système en boucle ouverte. Leur choix est important, ils doivent être tels que leur action puisse être considérée comme idéale et de ce fait n'apparaissent pas dans les schémas de principe.

# Les systèmes en présence et le langage associé



SYSTEMES	ENTREES	SORTIE
<i>Boucle Ouverte</i>	<ul style="list-style-type: none"> <li>➤ <i>Commande</i></li> <li>➤ <i>Perturbations (non désirées)</i></li> </ul>	<i>Grandeur à commander</i>
<i>Correcteur</i>	<ul style="list-style-type: none"> <li>➤ <i>Consigne</i></li> <li>➤ <i>Grandeur à commander</i></li> </ul>	<i>Commande</i>
<i>Boucle fermée</i>	<ul style="list-style-type: none"> <li>➤ <i>Consigne</i></li> <li>➤ <i>Perturbations (non désirées)</i></li> </ul>	<i>Grandeur à commander</i>

Un asservissement a deux objectifs de base :

- La grandeur à commander ou sortie doit suivre le signal de référence ou consigne : c'est l'objectif de **poursuite**.
- La sortie doit être le moins possible influencée par les perturbations : c'est l'objectif de **régulation**.

Pour cela on doit concevoir le correcteur qui est le véritable cerveau du système.

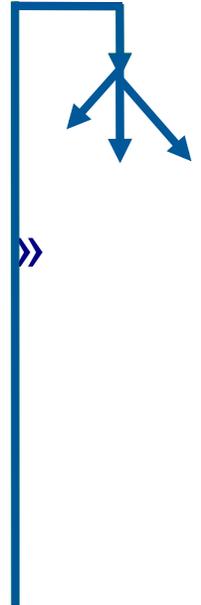
Cela nécessite une formulation correcte du problème qui oblige une modélisation mathématique permettant de rendre compte des comportements.

Des méthodes de synthèse du correcteur doivent être élaborées.



## Température de la douche :

- On règle la température voulue
- Fonctionnement classique : on doit attendre un « certain temps » que l'eau atteigne cette température
- Problèmes possibles pour un système mal régulé :
  1. le temps est « trop long » : problème de **rapidité** de réponse.
  2. la température n'est pas « exactement » celle souhaitée : problème de **précision** statique.
  3. On doit passer par des phases trop chaudes puis trop froides avant d'obtenir la bonne température (oscillations transitoires) : problème de **précision** dynamique.
  4. Une prise d'eau ailleurs sur le réseau modifie la température de la douche : problème de rejet rapide et précis de **perturbation**.



**Problème 1 à 3 : poursuite**

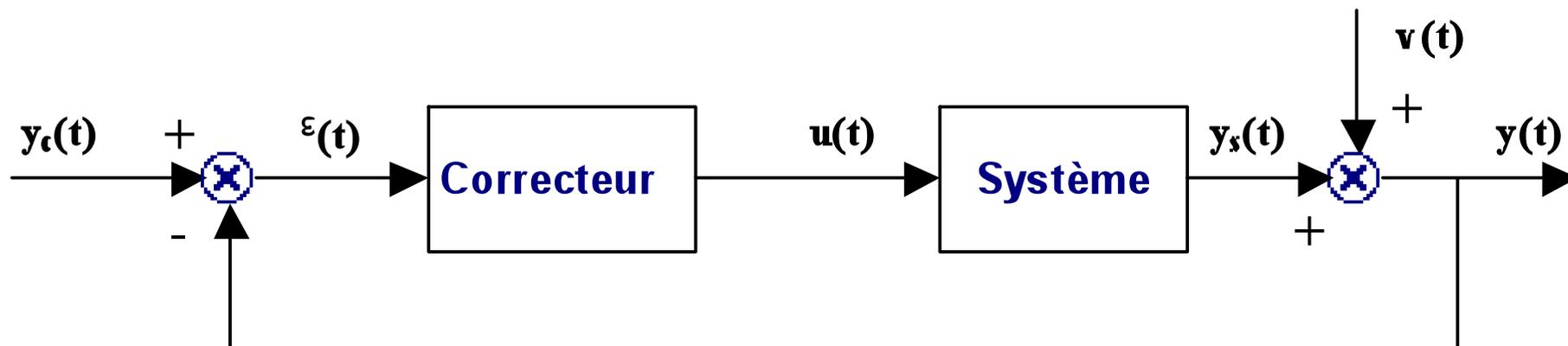
**Problème 4 : régulation**

La structure la plus simple consiste en un correcteur qui prend en compte l'erreur = écart entre la consigne et la sortie pour élaborer sa commande.

Consigne  $\rightarrow y_c(t)$

Commande  $\rightarrow u(t)$

Perturbation de sortie  $\rightarrow v(t)$



Un nouveau symbole : le sommateur dont les signes indiquent l'action :

Sortie  $\rightarrow y(t) = y_s(t) + v(t)$

Erreur  $\rightarrow \varepsilon(t) = y_c(t) - y(t)$

Une seule perturbation est ici prise en compte, la perturbation de sortie.

# Un premier modèle mathématique : le système linéaire



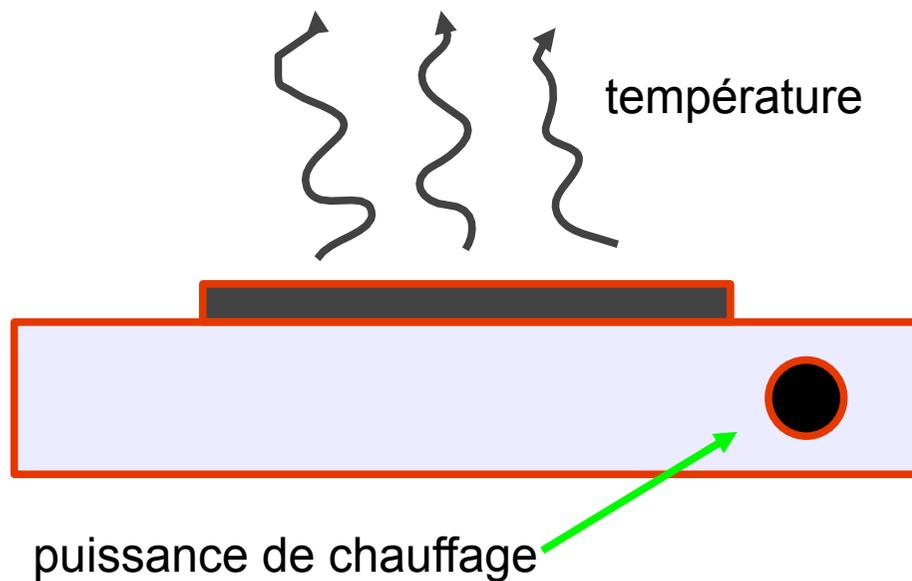
Tous les signaux pris en compte sont les plus simples qui soient : ce sont des signaux constants.

Le système en boucle ouverte est supposé linéaire c'est à dire que la sortie est proportionnelle à l'entrée (la commande) ce qui se traduit mathématiquement par :

$$y_s = G_0 u$$

$G_0$  est le gain statique du système.

## Description du système



Une température stable s'établit peu de temps après le changement de consigne

On peut identifier l'évolution de la température en fonction de la consigne.



$$T(C_{statique}) = G_0 C_{statique}$$

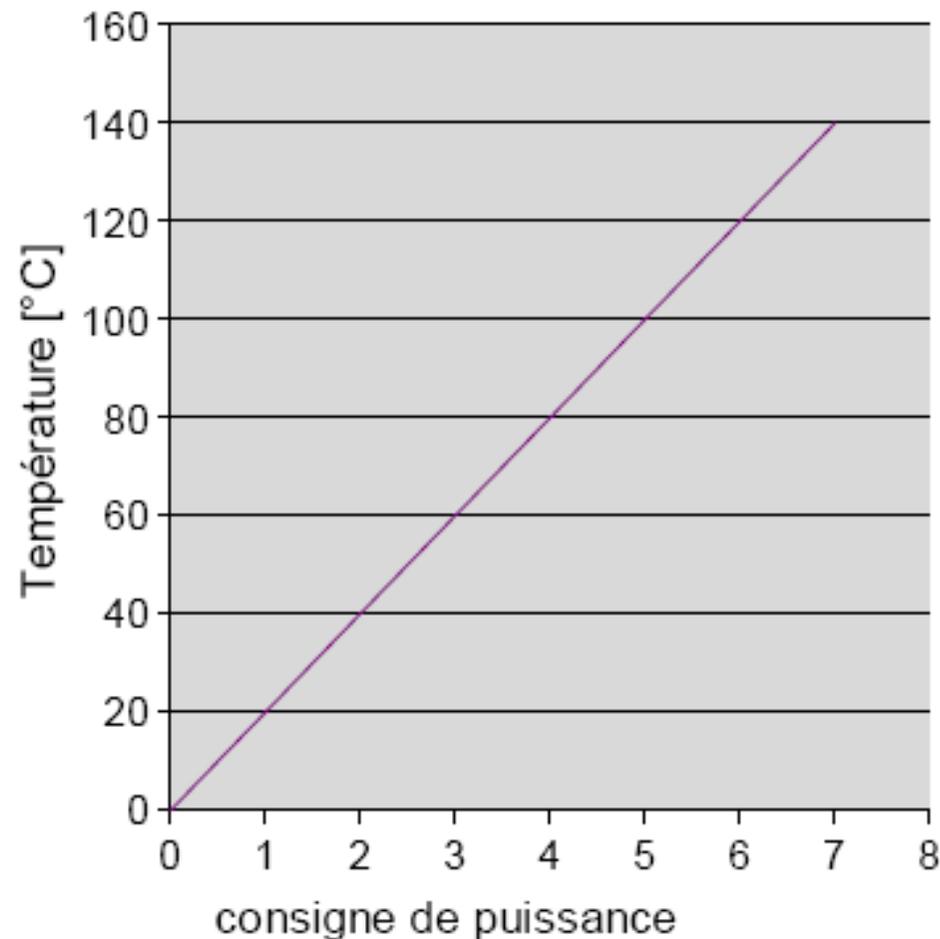
## Commande en Boucle Ouverte (sans charge)

Réponse linéaire

On identifie :

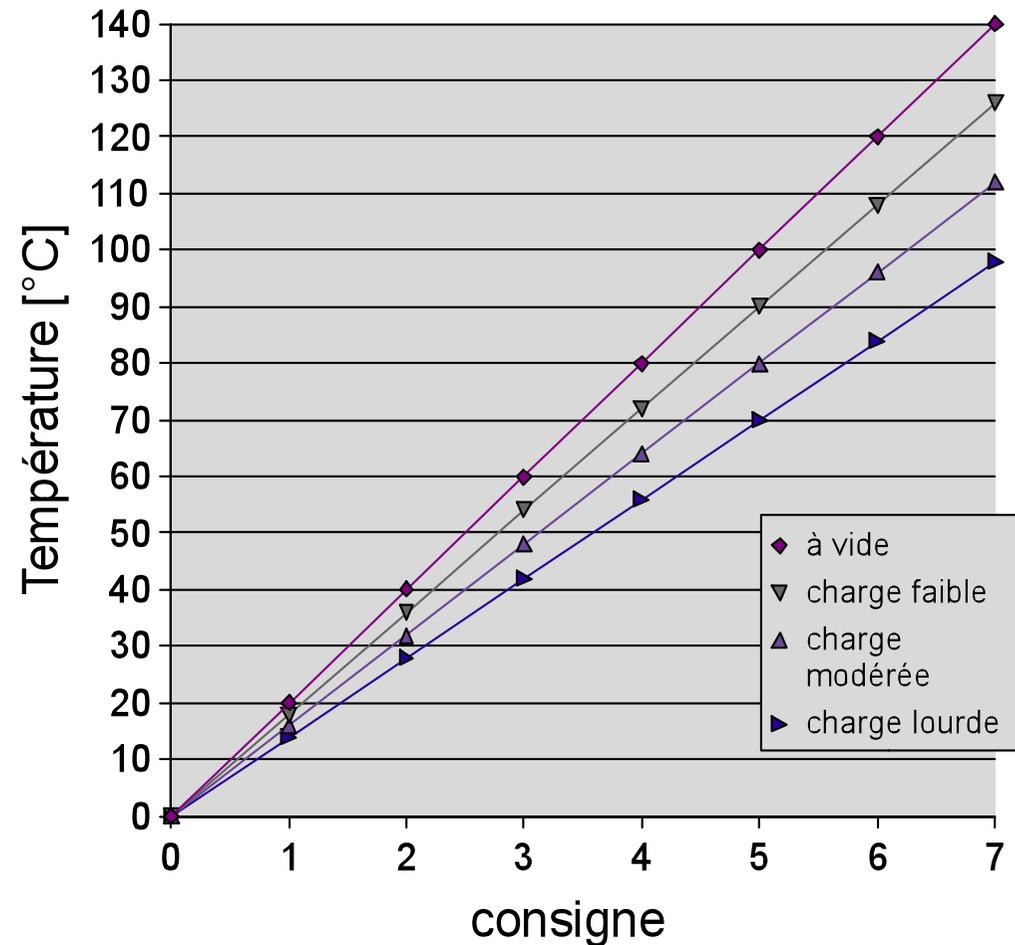
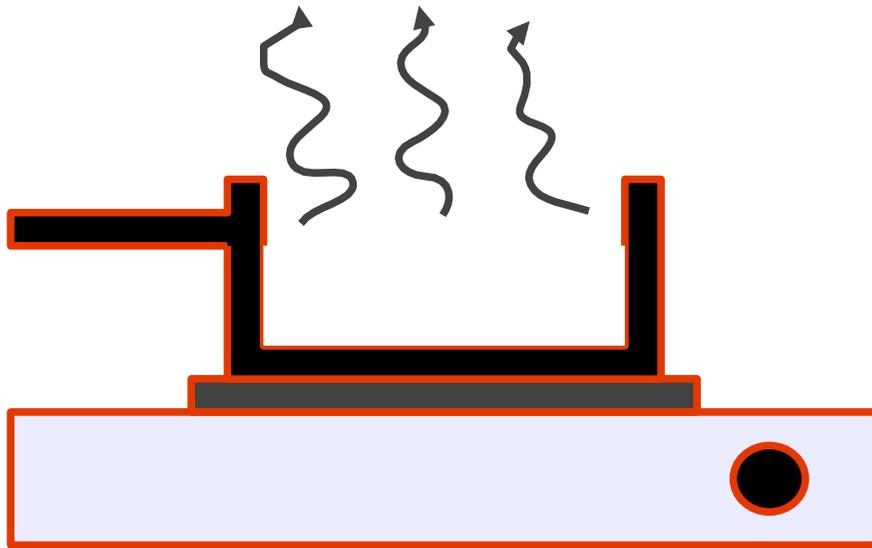
$$G_0 = \frac{T(C_{statique})}{C_{statique}} = 20 \text{ } ^\circ\text{C}$$

Identification Statique

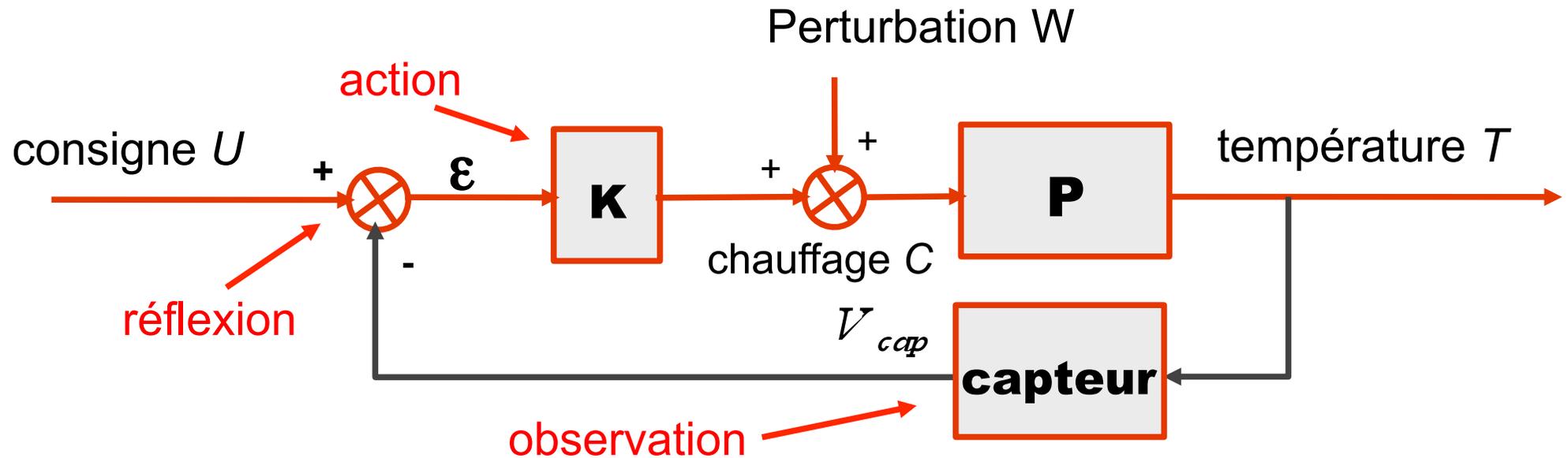


## Commande en Boucle Ouverte (en charge)

### Identification Statique



## Commande en Boucle Fermée



$$\epsilon = U - V_{cap} \Rightarrow \epsilon = U - K G_0 \epsilon \Rightarrow \epsilon = \frac{U}{1 + K G_0} \Rightarrow T = \frac{K G_0}{1 + K G_0} U$$

Pour un gain  $K$  infini, il n'y a plus d'erreur quelque soit le gain statique du système

Le plus simple correcteur qui puisse être envisagé est un correcteur proportionnel c'est à dire :

$$u(t) = K ( y_c - y ) = K \varepsilon \text{ (avec } K=\text{cste)}$$

La sortie du système bouclé est *(la calculer à titre d'exercice)*:

$$y = \left( \frac{G_0 K}{1 + G_0 K} \right) y_c + \left( \frac{1}{1 + G_0 K} \right) v$$

Une analyse simple montre que si  $K \rightarrow +\infty$  alors  $y \rightarrow y_c$  avec :

$$\left( \frac{1}{1 + G_0 K} \right) \rightarrow 0$$

L'objectif de régulation est atteint

$$\left( \frac{G_0 K}{1 + G_0 K} \right) \rightarrow 1$$

L'objectif de poursuite est atteint

Notions et exemples volontairement « simplistes » : étude en statique.

Les performances (ici la précision) sont « bonnes » si  $K$  infini (pas réaliste !)

Les systèmes réels sont plutôt dynamiques, prise en compte de l'inertie des systèmes, description par des équations différentielles (ou systèmes d'équations différentielles)

L'automatique dispose d'outils adaptés (transformation de Laplace, étude fréquentielles, représentation d'état, etc...)

Un exemple de système dynamique : le pendule inversé ... →



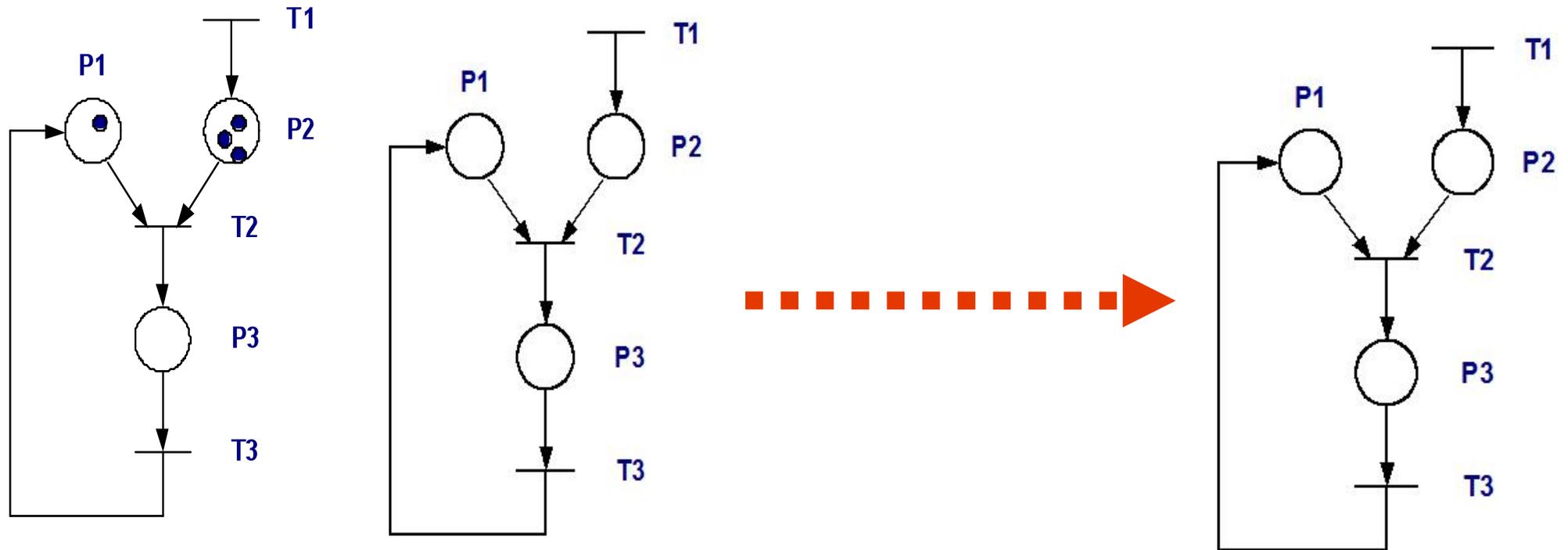


- Outils de modélisation à disposition aussi bien pour les automatismes que pour l'automatique et ses asservissements
- Exemples et systèmes considérés volontairement simpliste mais ont permis présentation des principaux concepts
- Pour décrire plus « finement », outils plus approfondis, maths appliquées (transformations, etc...)
- Utilisation de logiciels de simulations et d'analyse
- Prise en compte de la dynamique des systèmes (L2, L3, etc...)

# Culture Scientifique de Base en Sciences pour l'ingénieur

***Automatique***

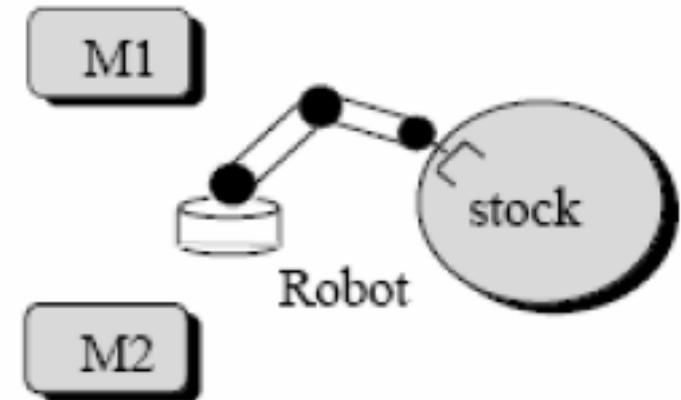
***Les exercices  
à préparer  
pour la semaine prochaine***



3 commandes en  
attente, machine  
disponible

**Tracer l'ensemble de l'évolution du RDP  
Combien de commandes ont été traitées ?**

- **Usinage** : de 2 pièces par 2 machines M1 & M2
- **Transport** : des pièces par un seul robot
- **Entreposage** : 1 stock de pièces
- **Fonctionnement** : dès qu'elles sont usinées, les pièces sont évacuées manuellement de façon immédiate (pas de gestion)



- **Capteurs disponibles** :

Piece\_M1 (M2) : fin du transport d'une pièce sur M1 (M2)

Fin\_US1 (US2) : fin de l'action usinage de la machine M1 (M2)

- **Actionneurs disponibles** :

TR1 (TR2) : transport d'une pièce sur M1(M2)

US1(US2) : usinage de la machine M1 (M2)

- **Attention** : pas de capteurs "Robot libre" ou "M1 libre" (gestion par partage de ressource).

Pas de précision sur la quantité du stock : on le suppose infini

**Proposer un RDP permettant de modéliser le fonctionnement**

Deux programmes informatiques (programme 1 et programme 2) partagent un espace mémoire unique. Quand un des deux programmes l'utilise, l'autre ne peut pas y avoir accès. Le programme 2 fonctionne de la même manière que le programme 1.

L'état du système est caractérisé par :

- le programme 1 n'a pas besoin de la mémoire : P1
- le programme 1 est en attente pour utiliser la mémoire : P2
- le programme 1 utilise la mémoire : P3
- le programme 2 n'a pas besoin de la mémoire : P4
- le programme 2 est en attente pour utiliser la mémoire : P5
- le programme 2 utilise la mémoire : P6
- la mémoire est disponible (non utilisée par les programmes) : P7

L'état du système va évoluer quand :

- le programme 1 accède à la mémoire : T1
- le programme 1 libère la mémoire : T2
- le programme 1 demande l'accès à la mémoire : T3
- le programme 2 accède à la mémoire : T4
- le programme 2 libère la mémoire : T5
- le programme 2 demande l'accès à la mémoire : T6

A l'instant initial :

- les deux programmes n'ont pas besoin de la mémoire qui est donc disponible
- les deux programmes sont en attente d'utilisation de la mémoire.

**A.1-** Modéliser par le RDP n°1 le comportement du programme n°1 avec la mémoire.

**A.2-** Modéliser par le RDP n°2 le comportement du programme n°2 avec la mémoire.

**A.3-** Fusionner les RDP 1 et 2 pour modéliser l'utilisation de la mémoire commune. Positionner des jetons dans les places adéquates (à l'instant initial) pour respecter le comportement du système.