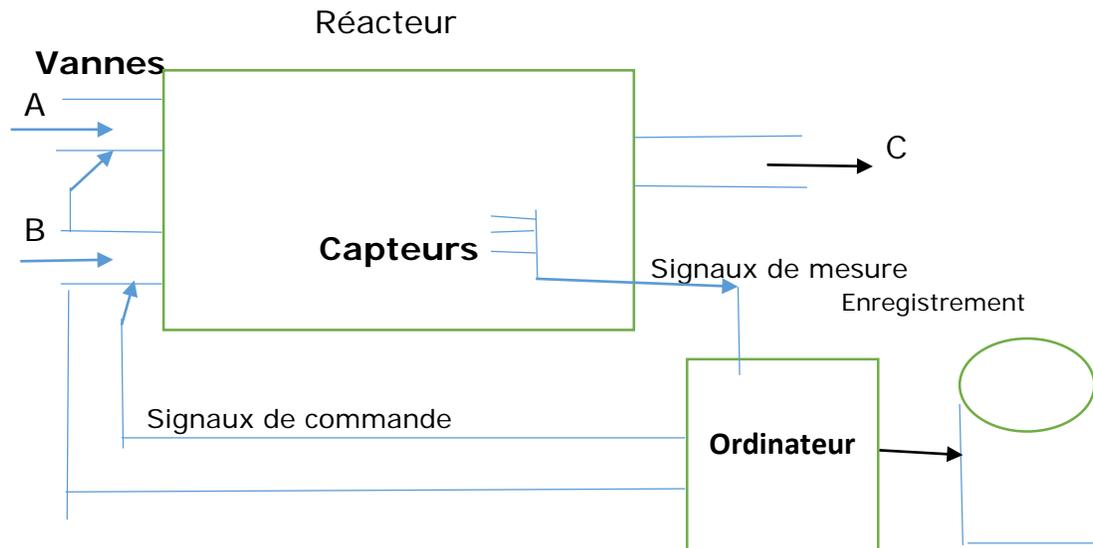


Chapitre 1 : Les Systèmes Temps Réel (STR)

1. Exemple introductif : Conduite d'un réacteur chimique.

- Dans une usine de produits chimiques, la synthèse d'un produit C à partir de 2 produits A et B s'effectue dans un réacteur.



- Cette fabrication est conduite par un ordinateur qui réalise 3 fonctions :

a) Régulation :

- La bonne marche de la fabrication nécessite que les paramètres de fonctionnement (température, pression, concentration, ...) soient maintenus dans des limites fixées → Pour cela, on agit sur les débits d'entrée des produits.
- Les paramètres de fonctionnement sont mesurés au moyen de capteurs. L'ordinateur prélève ces mesures et agit en conséquence sur les vannes d'admission, selon un programme de régulation.

b) **Enregistrement** :

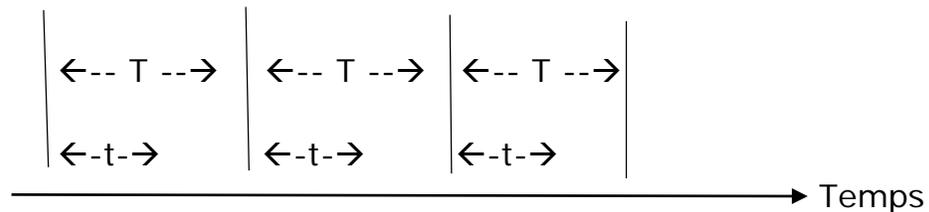
Les différents résultats de mesure sont périodiquement enregistrés. Leur valeur est affichée sur un tableau de bord, et recopiée dans un fichier (journal) en vue d'un traitement ultérieur (statistiques d'exploitation).

c) **Sécurité** :

Si certains paramètres mesurés dépassent une valeur critique prédéfinie (cas d'incident) → Le réacteur doit être arrêté d'urgence.

• **Contraintes** :

- a) Les mesures sont prélevées périodiquement. Soit T : la période de prélèvement et soit t le temps total nécessaire à l'ordinateur pour traiter un ensemble de mesures (prélèvement, enregistrement, détermination et exécution de la commande des vannes).



Le système ne peut fonctionner que si :

$$t \leq T$$

- b) La fonction de sécurité doit être prioritaire sur toutes les autres → le dépassement d'une valeur critique doit pouvoir être détecté à tout moment et le traitement de cet incident doit interrompre les opérations en cours.

- **Fonctions principales du Système d'Exploitation** :
 - a. Action sur les organes externes : Lecture des capteurs et commande des vannes ;
 - b. Prise en compte du temps physique : déclenchement périodique du cycle de traitement ;
 - c. Réaction aux événements extérieurs : arrêt d'urgence ;
 - d. Gestion d'information : conservation et entretien du fichier journal.

- **Caractéristiques des applications informatiques en temps-réel** :
 - Fixation d'une limite physique à la durée d'un traitement informatique ;
 - Existence d'échéances ;
 - Notion de traitement prioritaire ;
 - Connexion aux organes de commande et de mesure d'un dispositif extérieur.

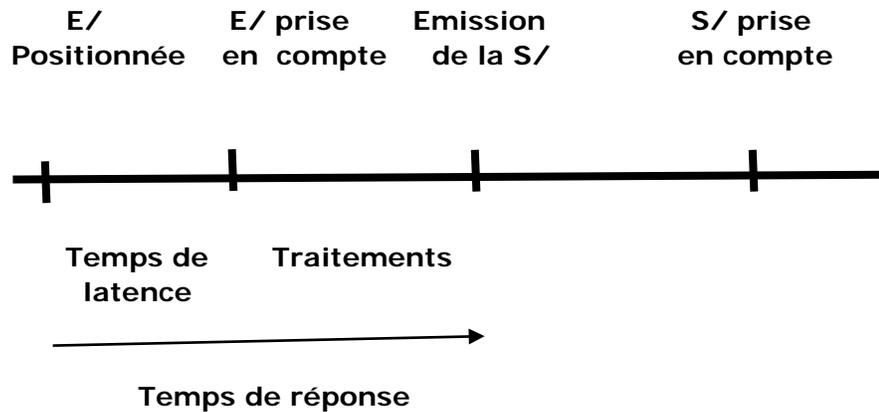
2. Définitions de base

2.1 Définitions :

- **Systeme** : Il est modélisé sous forme de boîte noire.
Les E/S peuvent être :
 - Des signaux, ITs, ... Si le système dialogue avec des périphériques matériels, ou
 - Des variables, sémaphores, événements, ... si le système dialogue avec d'autres composants logiciels.



- **Temps de réponse** : C'est le temps qui sépare la présentation des Entrées à un système de l'apparition des Sorties suite aux traitements effectués sur ces Entrées.
- **Temps de latence** : C'est le temps qui sépare la présentation des Entrées de leur prise en compte par le système.



2.2 Systeme Temps Réel :

- **Définition d'un STR** :
 - Un STR est un système qui interagit avec un environnement, qui a des propriétés temporelles variables, qui affiche un comportement dépendant du temps et prévisible et ce, par l'exploitation de ressources limitées.
 - La validité d'un STR dépend du résultat logique du traitement et du temps auquel le résultat est produit → Un STR doit satisfaire 2 contraintes importantes :
 - **Exactitude logique** : Sorties adéquates en fonction des entrées, assurant le comportement désiré pour le système suite à des événements et aux données communiquées.
 - **Exactitude temporelle** : Rencontre des contraintes temporelles.

➤ **Autres définitions :**

- Un STR est une association logiciel/matériel où le logiciel permet entre autre, une gestion adéquate des ressources matérielles en vue de remplir certaines tâches ou fonctions dans des limites temporelles bien précises.
- Les STR sont des systèmes qui possèdent des contraintes temporelles fortes. Lors de l'arrivée d'un stimulus en provenance de l'environnement externe, le système doit réagir dans des délais précis.

3. Classification des STR

En général, on distingue 2 catégories de STR :

- **STR non critique** = STR permissif, STR à contraintes relatives, Soft Real Time, STR à contraintes souples (ou molles).
- **STR critique** = STR à contraintes strictes, Hard Real Time, STR à contraintes dures.

3.1 **STR non critique** :

- ✓ C'est 1 système dont la performance est dégradée mais sans engendrer des conséquences graves si les contraintes temporelles ne sont pas rencontrées (résultat avec un certain retard ou parfois traitement effectué mais avec une limite maximale).
- ✓ Une certaine probabilité de dépassement de la limite supérieure de temps de réponse est tolérée.
- ✓ Le système n'est pas assujetti au respect des dates critiques.
- ✓ → On s'intéresse donc au temps de réponse moyen → Ces systèmes se rapprochent fortement des SE classiques à Temps Partagé (Time Sharing) (i.e répartition équitable du temps CPU entre processus).

✓ **Eg. :**

- Multimédia : Si quelques images ne sont pas affichées, cela ne met pas en péril le fonctionnement correct de l'ensemble du système.
- Jeux vidéo
- Système de réservation de billet
- ...

3.2 **STR critique** :

- ✓ C'est un système où tout dépassement de la limite supérieure de temps de réponse est susceptible d'avoir des conséquences catastrophiques.
- ✓ Chaque action doit se terminer avant une date prédéfinie appelée **ECHEANCE** → La violation d'une seule de ces échéances conduit le système dans un état de défaillance.
- ✓ **Eg. :**
 - Système de contrôle d'un avion (systèmes embarqués de l'aéronautique) ;
 - Système de surveillance d'un processus industriel (Eg. La régulation dans 1 centrale nucléaire).

4. Domaines d'application

Cf. Chapitre 2.

5. Aspects divers des STR

- **Parallélisme implicite**
- **Répartition**
- **Tolérance aux fautes :**

Il est typiquement très difficile et même impossible de tester un système de façon exhaustive, dès que sa complexité ou son coût dépasse un certain niveau.

Par exemple, on ne peut tester un exemplaire réel trop dispendieux et une simulation n'est pas complètement représentative du vrai système.

Ainsi, il se produira nécessairement des fautes matérielles ou logicielles.

Lorsque la fréquence d'arrivée des fautes est trop élevée ou que la réparation de ces fautes est impossible (e.g. système aérospatial) ou trop longue, on doit s'en remettre à une technique de tolérance aux fautes.

« La tolérance aux fautes se définit comme la capacité d'un système de continuer à fonctionner en présence de problèmes de matériel ou de logiciel i.e. à ne pas se retrouver en situation de défaillance malgré les problèmes au niveau des éléments de base et des sous-systèmes. »