Introduction aux Systèmes Industriels et Critiques
-Temps Réel-
(SIC)

Maria Zrikem

GI3 - GS3 ENSA de Marrakech

2024 - 2025

Overview

Ce module vise à apporter une compréhension des systèmes temps réel et critiques en donnant les clés de la modélisation et de la programmation d'applications multi-tâches sur des systèmes d'exploitation temps réel couramment utilisés et des outils permettant de garantir la sûreté de fonctionnement.

Partie 1 : Introduction aux systèmes temps réel

Partie 2 : Initiation sur des Plateformes industrielles temps réel embarquées

Overview

Partie 1 :

- Généralités sur les systèmes temps réel et critiques
- Les modèles d'ordonnancement
- Protocols de communication et de partages de ressources
- Spécification, Conception et Développement de Systèmes Temps Réel
- Un exemple d'exécutif temps réel : TORNADO de VxWorks

Q’est- ce que le temps réel?

Q’est- ce que le temps réel?

Exemple 1 : Système d’édition automatique des relevés de notes d’un service de scolarité.

Relevés produits pour les étudiants concernés
chaque relevé reprend l’ensemble des notes obtenues par l’étudiant
Moyennes calculées sont exactes

Exemple 2 : Système de navigation maritime pour calculer la position d’un bateau en mouvement.

Un résultat même exacte, obtenu après un intervalle de temps trop important, fournit une position considérée non exacte

L’intervalle de temps toléré par le système fait partie de ces spécifications

Le temps réel : historique

Fin des années 50 : premières apparitions de machines numériques dans des applications temps-réel (système de commande d'une unité de polymérisation d'une raffinerie Texaco).

Années 60 : les systèmes temps-réel s'étendent à des secteurs plus critiques tel que l'aérospatiale (systèmes de contrôle de vol NASA dans le projet Appolo).

Années 70 : l'arrivée des mini-calculateurs accélère l'insertion d'outils informatisés dans les environnements temps-réel.

Début des années 80 : accès de tous les secteurs industriels à l'informatique temps-réel grâce à la micro-informatique (modularisation de l'architecture du système informatique ⇒ intégration dans des applications géographiquement décentralisées + extensibilité & fiabilité)

Des définitions du temps réel?

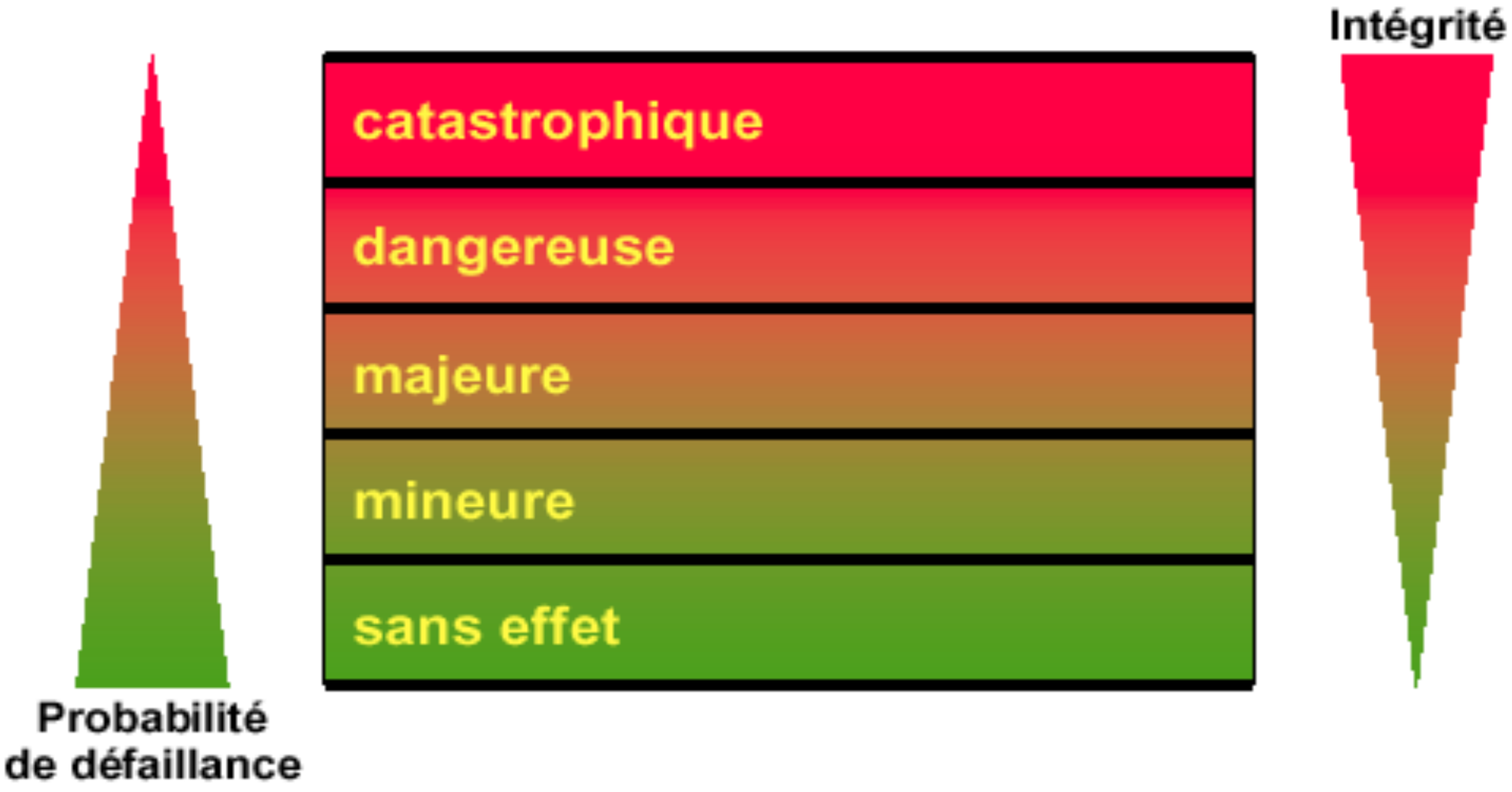
Définition [Martin 65] : Un système temps réel peut être défini comme un système qui contrôle un environnement en recevant des données , en les traitant, et en produisant une action ou des résultats de façon suffisamment rapide pour intervenir dans le fonctionnement du système à ce moment là.

Définition [Gillies 95] : Un système temps réel est un système dans lequel l'exactitude des applications ne dépend pas seulement de l'exactitude des résultats mais aussi du temps auquel ce résultat est produit. Si les contraintes temporelles de l'application ne sont pas respectées, on parle de défaillance du système. Il est donc essentiel de pouvoir garantir le respect des contraintes temporelles du système. Ceci nécessite que le système permette un taux d'utilisation élevé, tout en respectant les contraintes temporelle identifiées.

Notion de criticité en temps réel

Notion de criticité

Criticité = gravité de la situation qui résulterait d'un manquement au respect des spécifications temporelles



Temps réel dur et temps réel lâche

Temps réel dur ou critique (Hard Real Time)

• On parle de Temps Réel dur (Hard Real Time) quand les événements traités trop tardivement ou perdus provoquent des conséquences catastrophiques pour la bonne marche du système (vies humaines, faillites économiques, perte d'informations cruciales, plantage...).

• Les systèmes à contraintes temporelles dures ne tolèrent qu'une gestion stricte du temps afin de conserver l'intégrité du service rendu.

• Nécessitent des Logiciels et matériels spécifiques.

Exemple : Système de contrôle de centrale nucléaire, les systèmes embarqués utilisés dans l'aéronautique

Temps réel dur et temps réel lâche

Temps réel mou ou lâche (Soft Real Time)

• On parle de Temps Réel mou (Soft Real Time) quand les événements traités trop tardivement ou perdus sont sans conséquence catastrophique pour la bonne marche du système. On ne garantit qu'un pourcentage moyen d'utilisation du temps CPU.

• Les systèmes à contraintes temporelles souples ou molles (soft real time) acceptent des variations dans le traitement des données. On parle alors de Qualité de Services.

• Se satisfont de logiciel et matériel généralistes

Exemple : Système de visioconférence

Ces systèmes se rapprochent fortement des systèmes d'exploitation classiques à temps partagé

Temps réel dur et temps réel lâche

Temps réel mou :

La fonction est plus complexe et dépend de l'application : ordonnancer les activités et définir un test d'acceptabilité des résultats en fonction du retard

Activités du système :

• Élire une application ⇒ accorder du temps de calcul et faire progresser la qualité du résultat qu'il produit avant son échéance.

• Politique d'ordonnancement en fonction des échéances et de la qualité des résultats déjà obtenus.

Temps réel dur :

Le critère de respect des contraintes temporelles est simple : fonction booléenne

Activités du système :

Connaître le temps d'exécution des primitives fournies par le système (les instructions élémentaires)

Temps d'exécution des primitives

Élément clefs pour déterminer le temps d'exécution d'un programme.

Difficultés :

Primitives en temps d'exécution incertain
E.g. gestionnaire de mémoire virtuelle, allocateur de mémoire, système de cache, ….

⇒ Calcul du temps d'exécution des primitives « au pire des cas »

⇒ Résultats très pessimistes

⇒ Proposer des formes simplifiées des primitives du système avec de meilleurs temps de réponse au pire cas.

Conclusion

Le système temps réel ≠ Système hautes performances

Notion de tâche critique

- Un système temps réel se compose de plusieurs sous-systèmes.
  Tous ne relèvent pas du temps réel !!
  Exemple : altimètre, jauge et gestion du bar dans un avion

- Notion de tâche critique : chaque tâche se voit attribuer une priorité en fonction de son caractère critique.

- Classification des temps de réponse :
  - *Critique : moins de quelques micro-secondes de temps de réponse*
  - *Moyen : quelques centaines de ms*
  - *Non critique : à partir de quelques dizaines de secondes*

Tâche périodique / apériodique

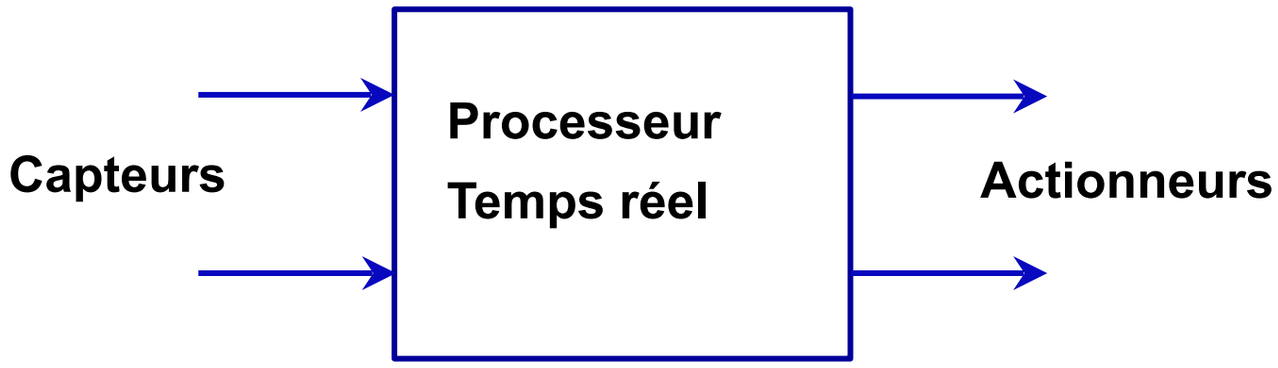
- Un système temps réel comporte souvent des tâches périodiques, mais il peut aussi comporter des tâches *apériodiques*, par exemple la gestion d'un événement imprévu tel que la chute d'un objet devant un robot de stockage dans un entrepôt.

- Typiquement, les tâche *périodiques* sont celles de bas-niveau dans l'application (celles de lecture des capteurs notamment) qui s'effectuent en fonction des caractéristiques physiques de l'environnement qui impose leur rythme.

Un système temps réel et son environnement

Architecture générale d'un STR

Système : ensemble d'activités correspondant à un ou plusieurs traitements effectués en séquence ou en concurrence et qui communiquent éventuellement entre eux.



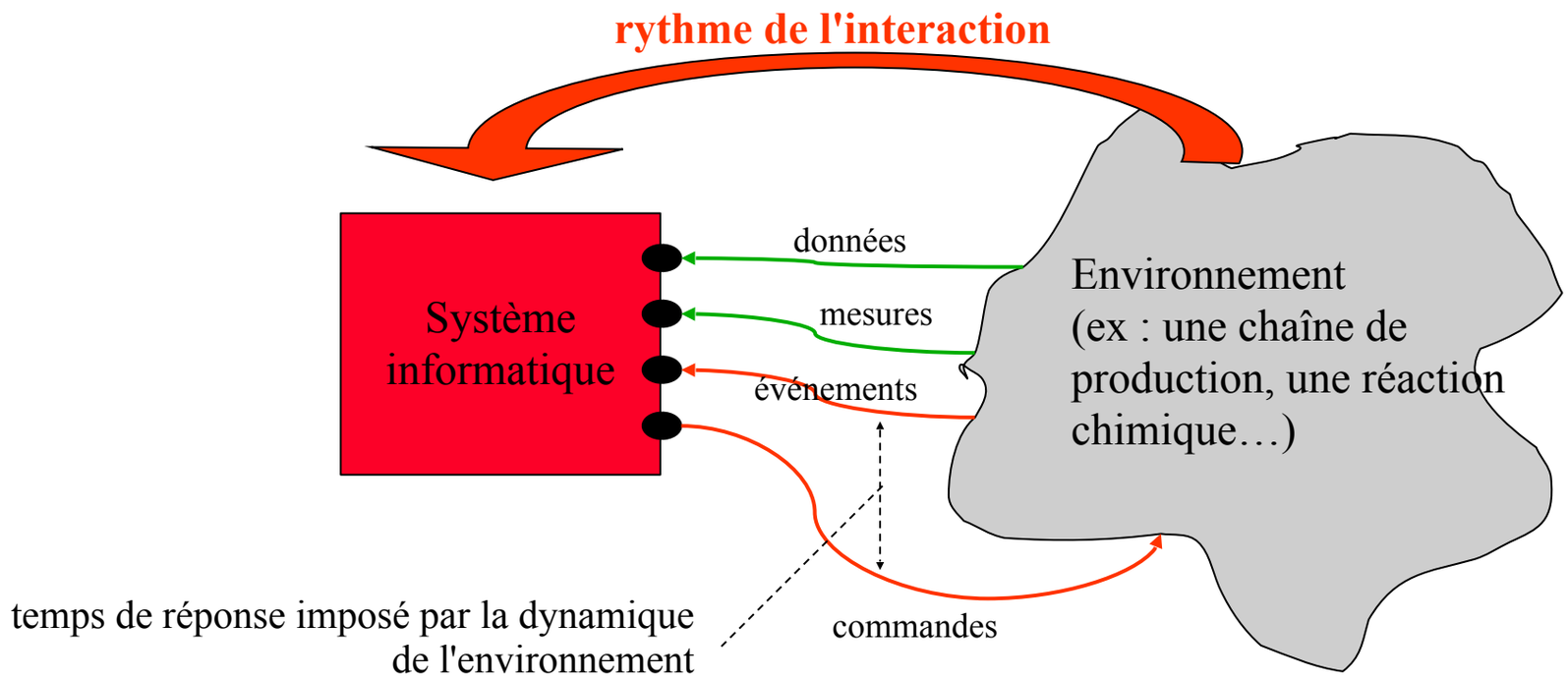
Capteurs : collecter les signaux émis par l'environnement (événements) ou prélever l'état de l'environnement (mesures).

Actionneurs : reçoivent la réaction du système à travers des commandes

Réaction entre système et environnement s'effectuent à des moments déterminés par :

- le système ⇒ système piloté par le temps (time-driven system)
- l'environnement ⇒ système piloté par l'événement (event-driven system)

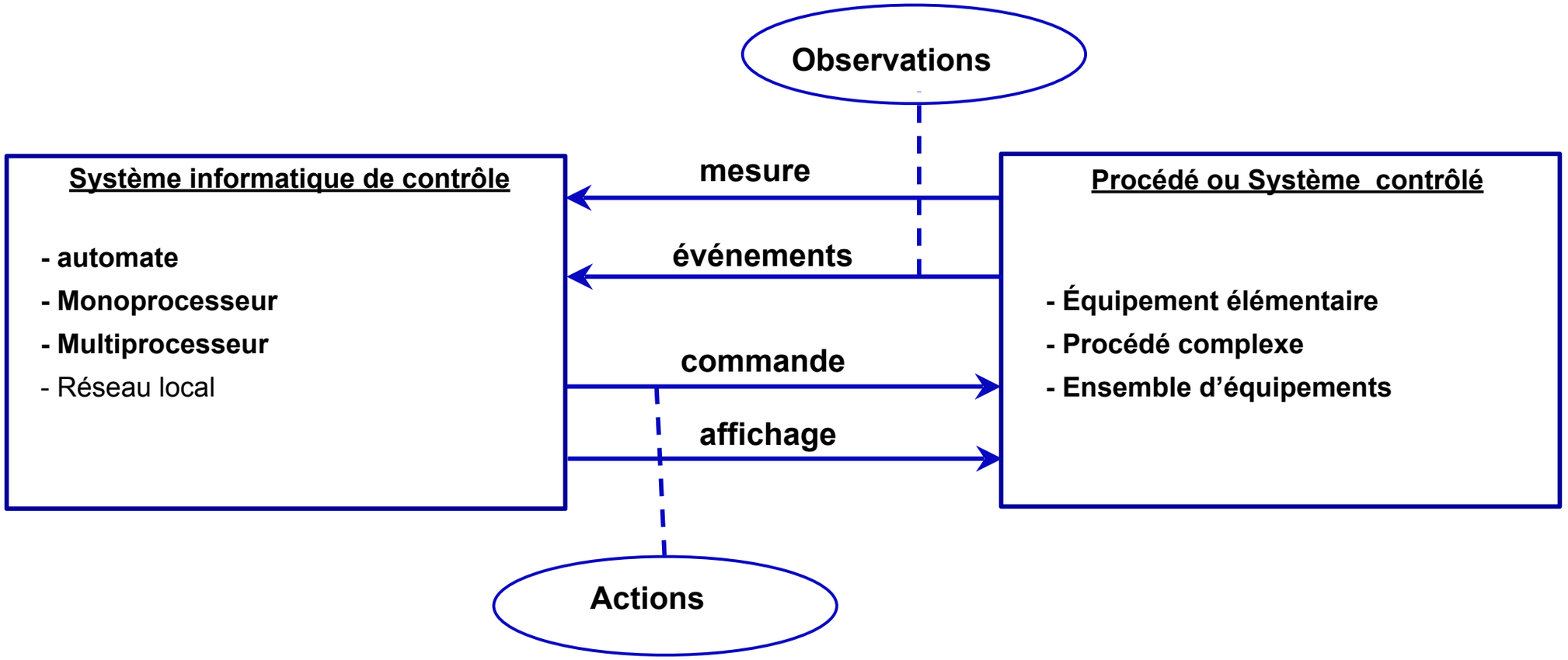
Architecture générale d'un STR



• Acquisition des mesures périodique à une cadence compatible avec les dynamiques du procédé

• Réaction du système de contrôle (émission des commandes) dans un laps de temps contraint

Le système temps réel et son environnement



La relation entre les deux sous-systèmes est décrite par trois opérations : échantillonnage, calcul et réponse.

Ces opérations doivent se réaliser à l'intérieur d'intervalles de temps : ce sont les contraintes de temps.

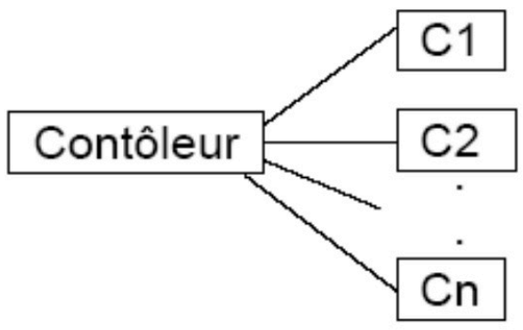
L'échelle du temps varie selon les applications : 1 μs (radar), 1 s (interface homme), 1 mn (chaîne de production) et 1h (réaction chimique)

Qualité requises des structures d'accueil temps réel

Un système temps réel doit être capable d'interrompre l'activité en cours d'exécution pour répondre à un événement

Prévisibilité (predictability) : c'est ce qui permet de déterminer à l'avance si un système va respecter ses contraintes temporelles

Déterminisme : Enlever toute incertitude sur le comportement des activités individuelles et sur leurs comportements lorsque elles sont mises ensemble. Un systèmes déterministe si à partir de son état actuel, sa réaction à un stimulé est « prédictible »



Que fait-on si deux signaux arrivent « simultanément » ?

Pb : communication synchrone/asynchrone

En aéronautique :
non déterminisme = méfiance
problème : preuve de déterminisme sur bus AFDX de l'A380 (thèse de doctorat)

⇒ robustesse des protocoles et des allocations des ressources, en présence de variation de trafic, de pannes matérielles et de surcharge
⇒ déterminisme absolu difficile à atteindre

Qualité requises des structures d'accueil

Exactitude :

⇒ Exactitude logique (Logical correctness) :

- Sorties adéquates en fonction des entrées
- Le se comporte comme prévu en réaction des entrées

⇒ Exactitude temporelles (Timeliness)

- Satisfaction des contraintes temporelles
- Sorties présentées « au bon moment »

Fiabilité (reliability) : Difficulté de prédire le comportement d'un système dont les composantes matérielles et logicielles ne sont pas fiables

⇒ Tolérance aux fautes : la tolérance aux pannes doit être intégrée dès la conception dans le matériel comme dans le logiciel tout en prenant en compte les contraintes temporelles

⇒ Implantation de points de reprises, de duplication matérielle et logicielle et de techniques de recouvrement après fautes.

Précision suffisante de la mesure du temps et des datation des événements.

Les systèmes embarquées

• Dispositif informatique intégré au dispositif physique dont il assure le contrôle et (ou) la commande et dont il partage les contraintes d'environnement (spatial, automobile...). Caractérisé par :

- ➢ l'environnement dans lequel il fonctionne (température, humidité, radiations, vibrations, chocs…, taille, poids…)
- ➢ la performance qu'on attend de lui
- ➢ ses interfaces avec l'environnement

• Systèmes logiciel-matériel servant à résoudre des fonctions spécifiques, allant du simple contrôleur de lave-vaisselle au complexe système de guidage de missiles.

• Embarqué dans le sens où il fait partie d'un système complet qui n'est pas nécessairement un ordinateur.

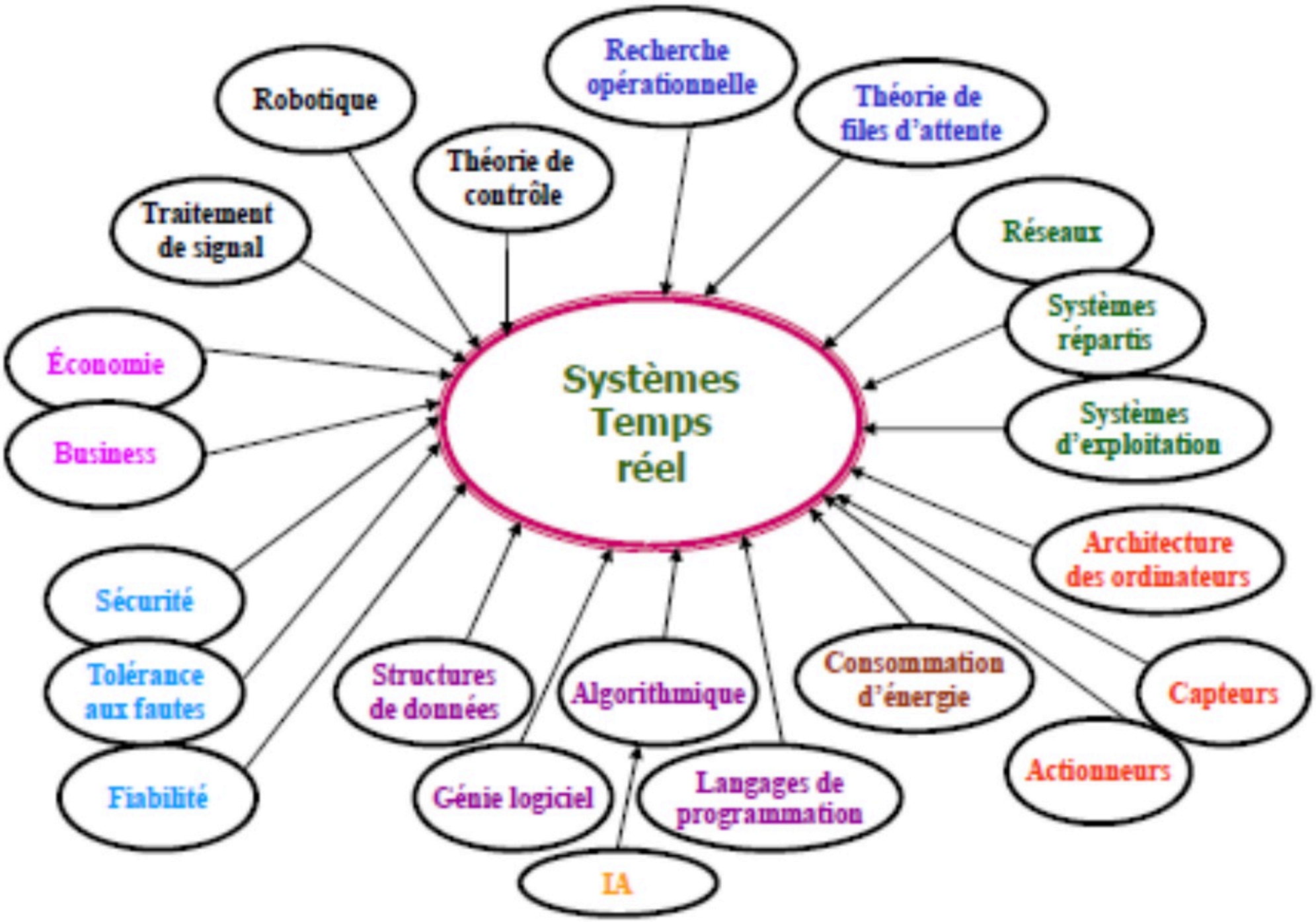
Les systèmes embarquées (temps réel)

Caractéristiques des systèmes embarqués:

- Utilisé dans un environnement réactif et contraint en temps
- Réalise un nombre fixe et limité de tâches par opposition à un ordinateur général (*general purpose*)
- Très souvent des systèmes haute performance soumis à des contraintes temps réel
- La puissance dissipée, le coût, le temps de développement et la fiabilité sont souvent des métriques qui influencent la conception

Exemples d’applications temps réel

Disciplines impliquées dans les STR



Domaines d'applications Temps Réel

- Transports : métro, aéronautique (avions, satellites, spacial), trains, automobile, …
- Multimédias : décodeurs numériques openTV, décodeurs TNT, MPEG, jeux vidéo. films d'animation.
- Services téléphoniques : téléphone mobile, auto-commutateur.
- Supervision médicale, écologique.
- Système de production industriel : centrale nucléaire, chaîne de montage, usine chimique.
- Robotique.
- etc...

Exemples d'applications temps réel

Exemple de robot : Un robot doit prendre des objets qui défilent sur un tapis roulant. L'objet se déplace à une certaine vitesse et possède une certaine taille qui laisse au robot « une fenêtre temporelle » pour le prendre. Si le robot agit trop tard, il manquera l'objet. S'il agit trop tôt, il risque de bloquer l'objet et par la suite tout le système de défilement

Système de vidéoconférence sur un réseau local : Le système commence par numériser le signal vidéo pour le transférer en séquence d'images. Pour avoir une image acceptable, on traite 30 images / seconde. Vu le débit du réseau, le système doit procéder à la compression des images avant de les transmettre.

$\Rightarrow$ Temps de latence = temps(numérisation)+temps(compression)+ temps(transmission)

Assurer une vision de bonne qualité $\Leftarrow$ Minimiser le temps de latence

Exemples d'applications temps réel

Système de control de centrale nucléaire qui vérifié le niveau de radioactivité d'une zone contrôlée. Si le système ne produit pas l'alarme attendue lors du franchissement du seuil toléré, les conséquences peuvent être catastrophiques

⇒ La tâche de traitement de l'alarme est critique

Système de contrôle arien ce système rassemble l'information sur l'état de chaque avion via un ou plusieurs radars actifs. Ce radar interroge chaque avion périodiquement. Le système traite les messages reçus de l'avion et stocke les informations dans une base de données. Ces informations sont prises en compte et traité par des processeurs graphiques. Au même temps, un système de surveillance analyse de façon continue le scenario et alerte les opérateurs toutefois qu'il détecte une collision possible.

Exemples d'applications temps réel

Logiciel de contrôle d'un Combiné GSM

– Embarqué

– Gestion des opérations de la couche physique

Emission, réception, mesure des niveaux de réception, etc...

– Gestion des procédures logiques

Localisation, mesures de qualité du lien radio,

– Gestion de la conversation

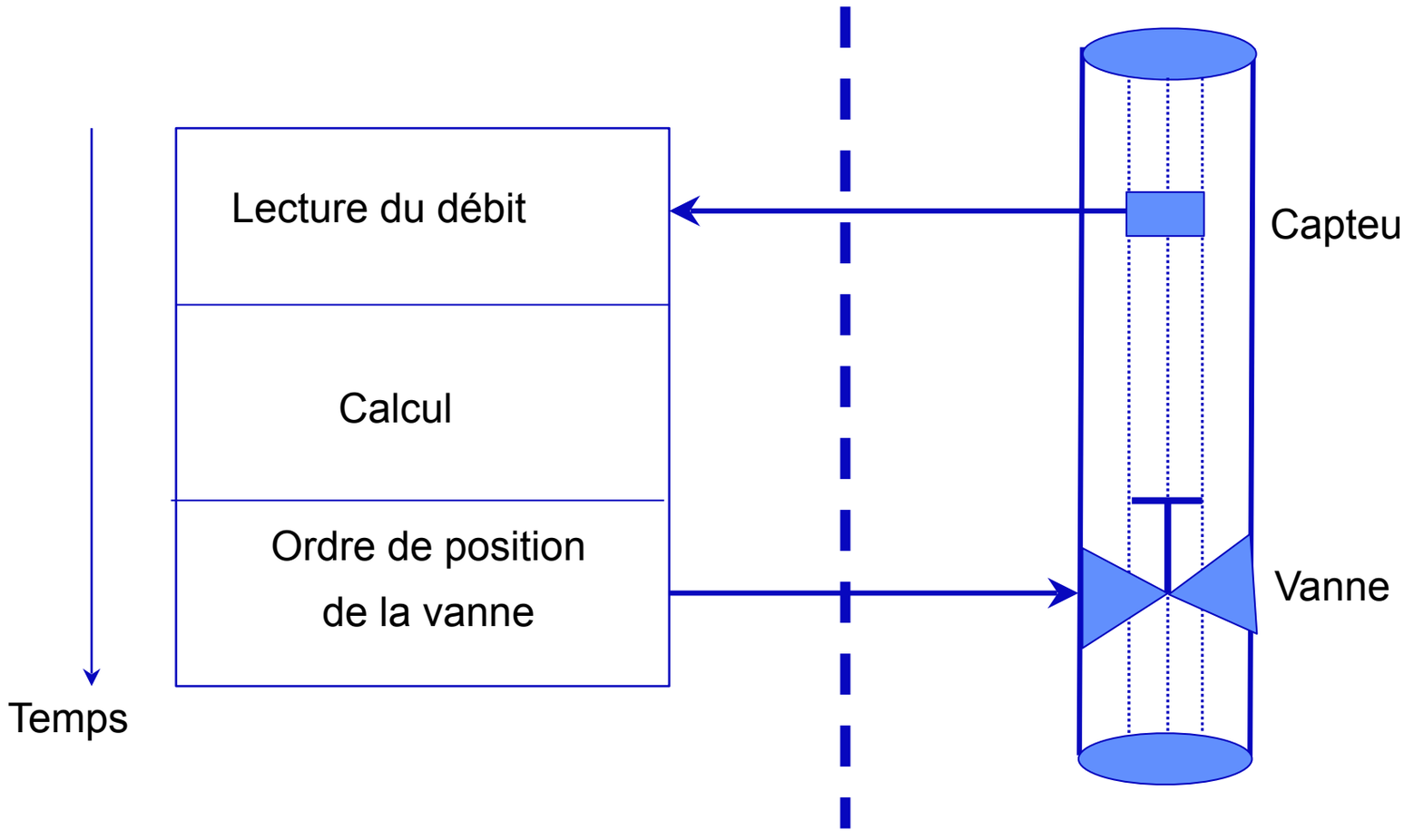
Relais entre deux réseaux

– IHM

Rédaction, navigation, etc…

Exemples d'applications temps réel

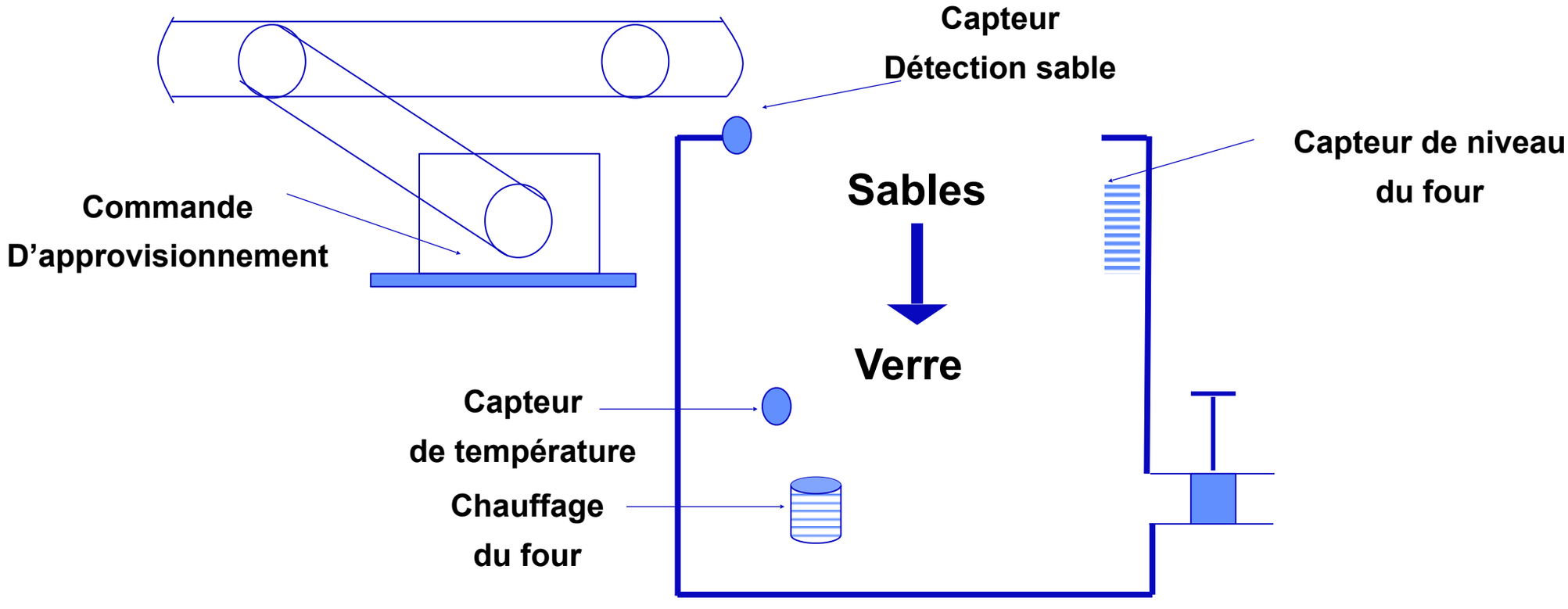
Système de contrôle de débit



Exemples d'applications temps réel

Gestion d'un four à verre

Un four pour la fabrication de verre fonctionne en quasi-continu pour l'approvisionnement en matière première (sable) et aussi pour l'utilisation du produit (verre). En effet, le four doit rester en fonctionnement permanent avec un niveau toujours de matières fondues à température constante, une évacuation du trop plein étant prévue en cas d'attente prolongée d'utilisation du verre.



Exemples d'applications temps réel

Gestion d'un four à verre

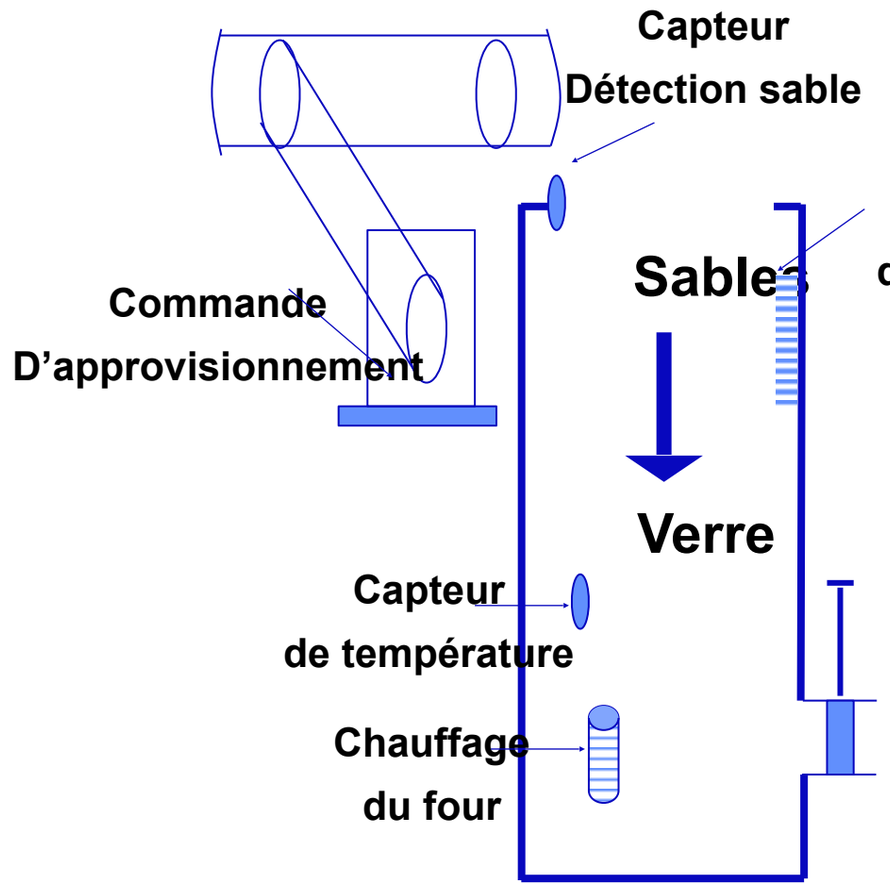
Le contrôle de cette application est fait par l'intermédiaire de 3 capteurs (température, niveau du four, détection de l'arrivée de matière première) et 2 actionneurs (commande d'approvisionnement en sable, chauffage du four)

⇒ L'acquisition de la température doit se faire à des moments réguliers (utilisation de l'horloge temps réel interne du système).

⇒ l'acquisition du niveau de matière est liée à l'interruption générée de façon apériodique par les tombées successives détectées non régulières de la matière première.

⇒ Le traitement du "signal température" permet de faire un calcul précis de la température et lance la tâche de commande de chauffage si la température du four est inférieure à une température de consigne.

⇒ Le paramètre "niveau de matière" va impliquer l'approvisionnement ou non en matière première en commandant la vitesse d'approvisionnement en fonction du paramètre niveau du four, mais aussi de la valeur du capteur de température afin d'obtenir une régulation plus fine.

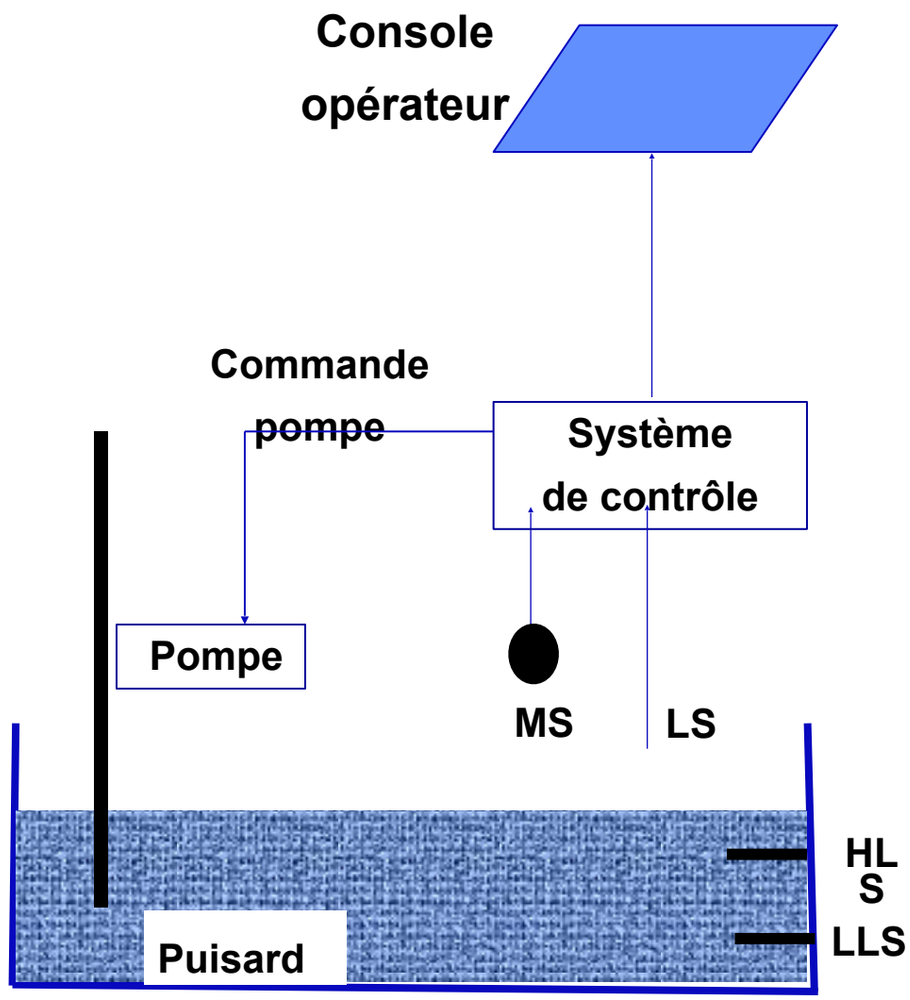


Exemples d'applications temps réel

Gestion de la sécurité d'une mine

La mine doit fonctionner tant que la sécurité max est maintenue. Les indications générales de fonctionnement sont :

- **une alarme doit être lancée vers la console de l'opérateur dés que le niveau limite du capteur MS est franchi afin de pouvoir évacuer la mine.**
- **la pompe doit être mise en route si le niveau d'eau dépasse le niveau max (LS >HLS). La pompe s'arrête dés que le niveau descend en dessous de la valeur inf (LS < LLS).**
- **la pompe ne doit pas fonctionner quand le niveau du capteur de méthane dépasse un certain seuil afin d'éviter le risque d'explosion.**



Exemples d'applications temps réel

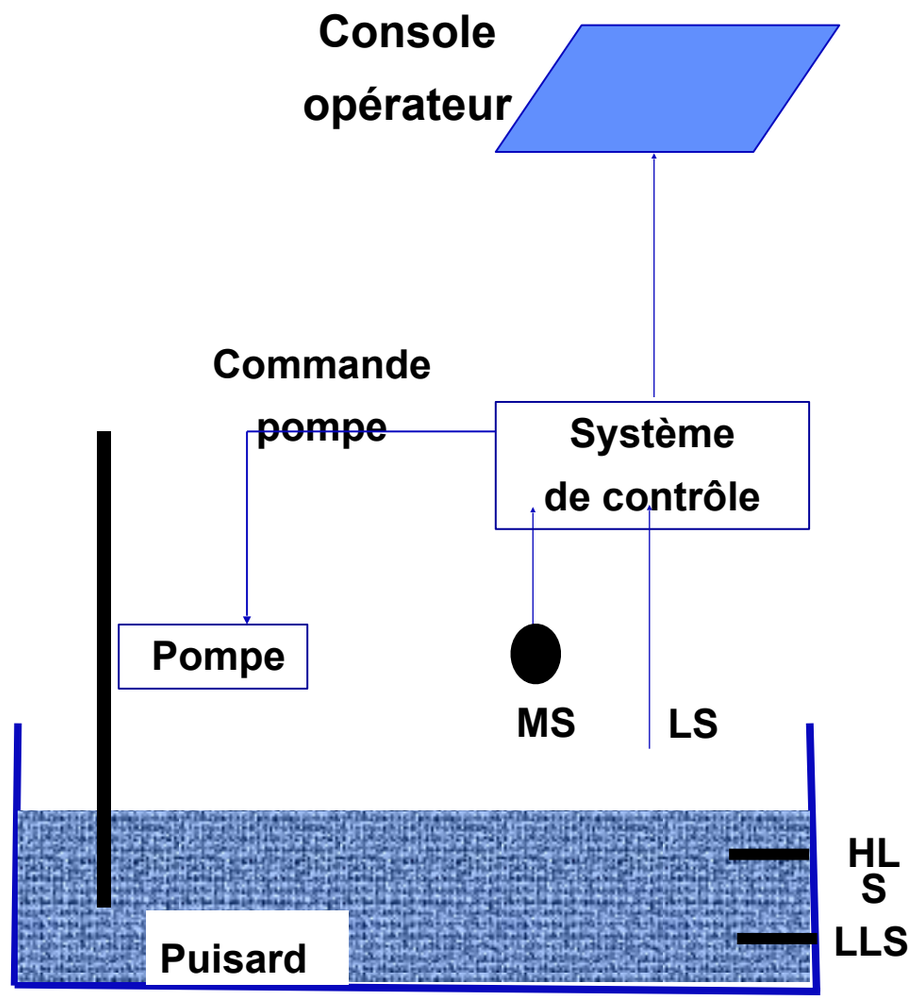
Gestion de la sécurité d'une mine

La mine doit fonctionner tant que la sécurité max est maintenue. Les indications générales de fonctionnement sont :

• une alarme doit être lancée vers la console de l'opérateur dés que le niveau limite du capteur MS est franchi afin de pouvoir évacuer la mine.

• la pompe doit être mise en route si le niveau d'eau dépasse le niveau max (LS >HLS). La pompe s'arrête dés que le niveau descend en dessous de la valeur inf (LS < LLS).

• la pompe ne doit pas fonctionner quand le niveau du capteur de méthane dépasse un certain seuil afin d'éviter le risque d'explosion.



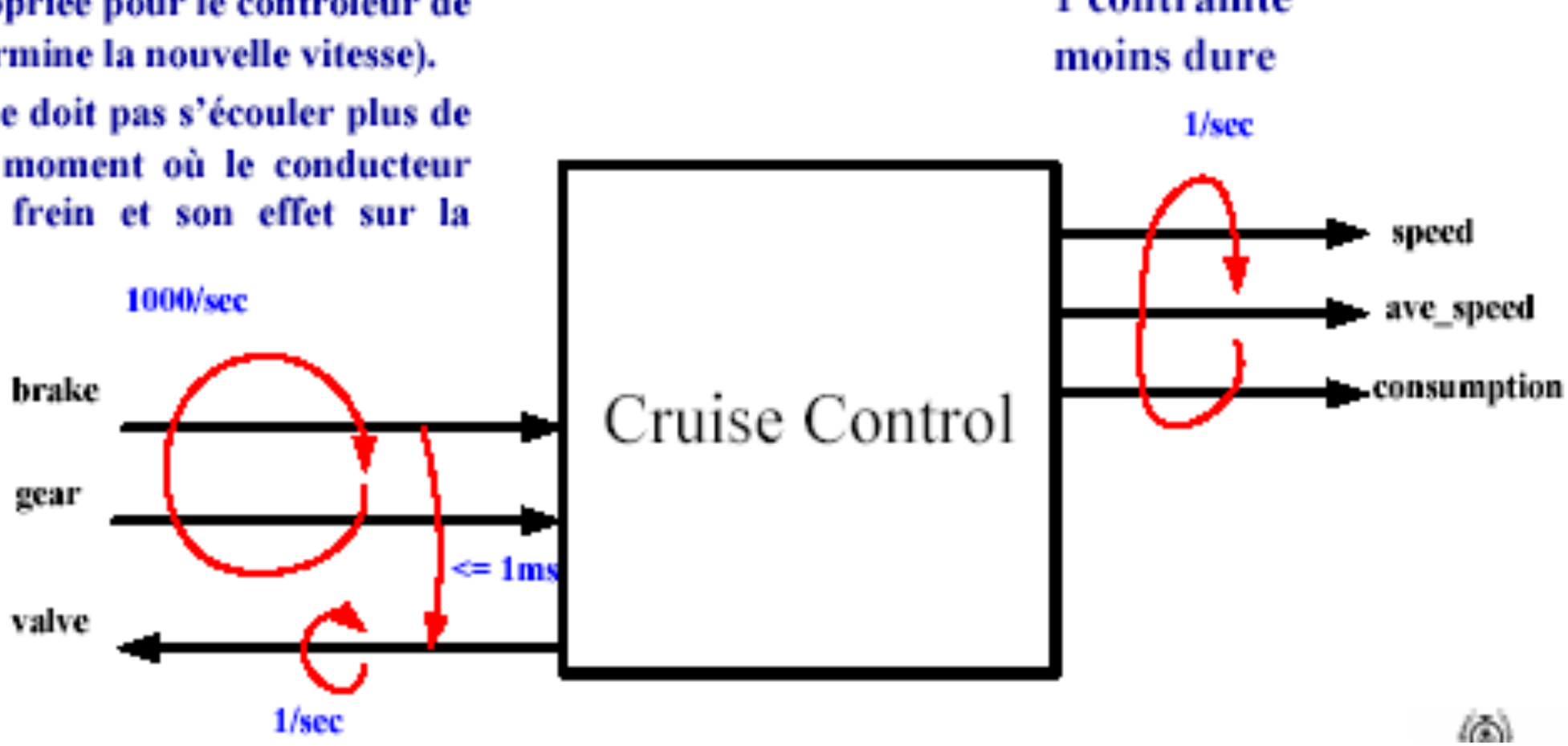
Exemples d'applications temps réel

Conduite d'automobile

- **Spécifications temporelles d'un sous-ensemble de signaux:**
  - **Via des sensors, échantillonnage du signal embrayage (*gear*) et frein (*brake*) au moins à toutes les *ms*.**
  - **Suite aux valeurs obtenus précédemment, on doit calculer au moins à toutes les *sec* la valeur appropriée pour le contrôleur de valve (qui détermine la nouvelle vitesse).**
  - **Également, il ne doit pas s'écouler plus de 1 ms entre le moment où le conducteur appuie sur le frein et son effet sur la valve.**
  - **Finalement, le tableau de bord doit être mis à jour, au moins à toutes les secondes.**

3 contraintes dures

1 contrainte moins dure



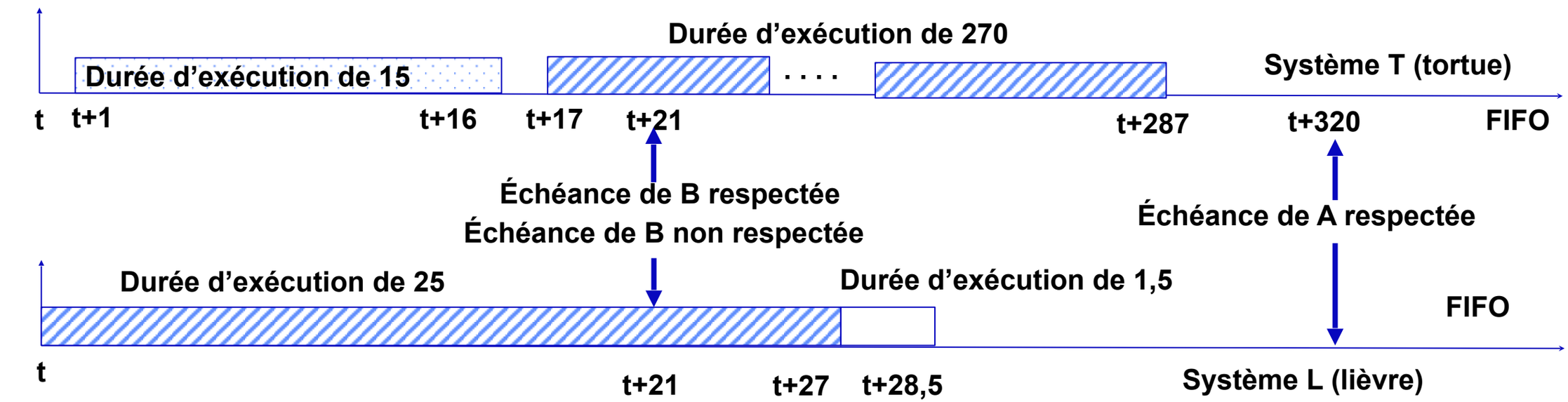
Temps Réel ≠ aller vite !

Un exemple simple et éclairant

• Système T, avec processeur à vitesse 1, surcoût 1 et ordonnancement EDF sur l'échéance
• système L, avec processeur à vitesse 10, surcoût 0 et ordonnancement à l'ancienneté (FIFO)
Surcoût système = ord. + changement de contexte; les tâches sont non préemptives

L est 10 fois plus rapide et infiniment efficace

Une application de conduite avec 2 tâches :
• tâche A envoie des commandes périodiques pour entretenir la commande moteur du véhicule
sa durée = 270 sur T et 27 sur L ; son délai critique de 320
• tâche B qui réagit à une commande du volant
sa durée = 15 sur T et 1,5 sur L ; son délai critique de 21
Au temps t, la tâche A est déclanchée pour son exécution périodique, au même instant B est demandée en réaction à un événement imprévu. A et B ont la même importance.



Un exemple simple et éclairant

Le système temps réel, ce n'est pas

La vitesse de traitement (l'unité centrale la plus rapide)

La meilleure efficacité de la plate-forme (le surcoût système le plus faible).

Le système temps réel, c'est

Le respect des contraintes temporelles

Les contraintes temporelles proviennent

des spécifications de l'application et de la dynamique du procédé

Des spécifications dérivées pour la structure d'accueil informatique

Temps Réel ≠ aller vite

» besoin d'un temps de réaction court (1 ms) pour le contrôle d'un avion de combat
» besoin d'un temps de réaction moins court (10 ms) pour le contrôle d'un avion de transport civil
» besoin d'un temps de réaction moins court (1s) pour une IHM
» besoin d'un temps de réaction moins court (1mn) pour le contrôle d'une chaîne de production (lente)
» besoin d'un temps de réaction moins court (1h) pour le contrôle d'une réaction chimique (lente)
»…
» besoin d'un temps de réaction de quelques heures pour l'établissement d'une prévision météorologique
» besoin d'un temps de réaction de quelques jours pour le calcul de la paie…

Résumé; ce qu'il faut retenir ..

La demande des applications en temps réel, c’est

Optimiser la ponctualité

Assurer le respect des contraintes temporelles

Elle diffère de la demande des applications non temps réel ou on veut :

Optimiser le partage des ressources : efficacité et performance

Réduire le temps de réponse : vitesse et rapidité