

1.1. Définitions :

Nous pouvons définir un système de contrôle-commande comme un système informatique en relation avec l'environnement physique réel externe par l'intermédiaire de capteurs et/ou d'actionneurs, **contrairement** aux systèmes d'informatiques scientifiques (gestion de base de données, CAO, bureautique...) qui ont des entrées constituées de données fournies par des fichiers ou éventuellement un opérateur. Les grandeurs physiques acquises permettent au système de contrôle-commande de piloter un procédé physique quelconque. Donnons ainsi une définition générale d'un système de contrôle-commande (**figure 1.1**) :

Un système de contrôle-commande reçoit des informations sur l'état du procédé externe, traite ces données et, en fonction du résultat, évalue une décision qui agit sur cet environnement extérieur afin d'assurer un état stable

Cette notion d'état stable peut être différente selon les applications ou procédés. Il dépend du cahier des charges de l'application (maintien d'une température de consigne, régime moteur, qualité de service...).

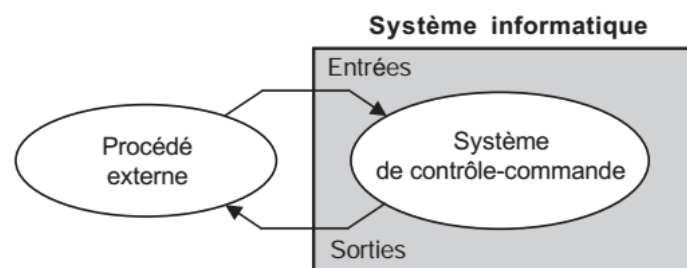


Figure.1.1. Représentation schématique d'un système de contrôle-commande.

L'interaction du système de contrôle-commande avec le procédé extérieur à piloter se décompose en deux parties (figure 1.2) :

- Observations par l'intermédiaire de capteurs (**sensors**)
- Actions réalisées par l'intermédiaire d'actionneurs (**actuators**)

Cette définition des systèmes de contrôle-commande ayant été faite, nous pouvons replacer ces systèmes par rapport aux autres systèmes informatiques en faisant trois catégories :

- **les systèmes transformationnels** qui utilisent des données fournies à l’initialisation par l’utilisateur. Ces données, leurs traitements et l’obtention du résultat n’ont aucune contrainte de temps ;
- **les systèmes interactifs** dans le sens où les données sont produites par interaction avec l’environnement sous différentes formes (clavier, fichier, réseaux, souris, etc.). Mais le temps n’intervient pas en tant que tel si ce n’est avec un aspect confort de travail ou qualité de service ;
- **les systèmes de contrôle-commande ou réactifs** qui sont aussi en relation avec l’environnement physique réel pour les données en entrée.

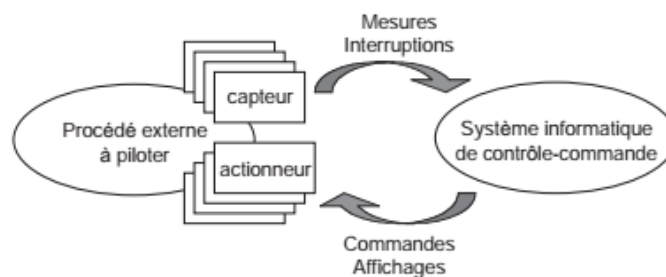


Figure 1.2. Représentation schématique de l’interaction du procédé physique piloté et du système de contrôle-commande.

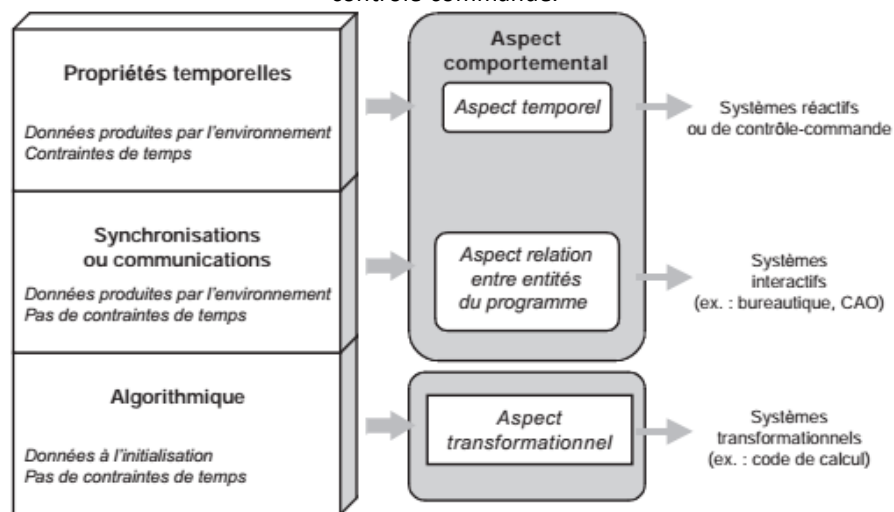
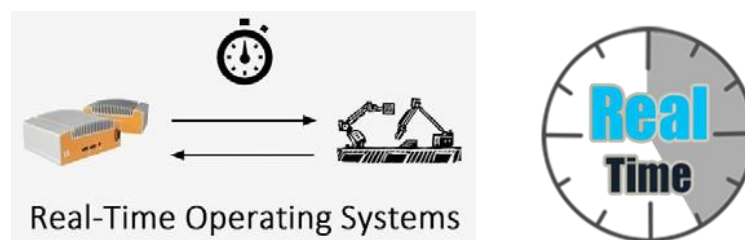


Figure 1.3 – Comparaison des systèmes de contrôle-commande par rapport aux autres applications Informatiques.

1.2. **Système Temps Réel (Autre définition) :**



- Un système capable de contrôler un procédé physique à une vitesse adaptée à l'évolution du procédé contrôlé.
- Il fournit un service dans un contexte où le temps intervient
- Il subit une contrainte : temps et l'exactitude des résultats (c à d le système doit délivrer des résultats exacts dans les délais imposés, compatible avec l'évolution du système.
- Un système TR n'est pas un système qui va vite mais un système qui satisfait à des contraintes temporelles.
- C'est un système qui interagit avec un environnement externe qui lui-même évolue avec le temps. Qui réalise certaines fonctionnalités en relation avec cet environnement.
- Dans un système temps réel, Un résultat de calcul mathématique exact, mais arrivant au-delà d'une échéance prédéfinie est un résultat faux.

1.3. Principales caractéristiques des systèmes de contrôle-commande

Considérons un exemple représentatif d'une application de contrôle-commande représenté sur la **figure 1.4**.

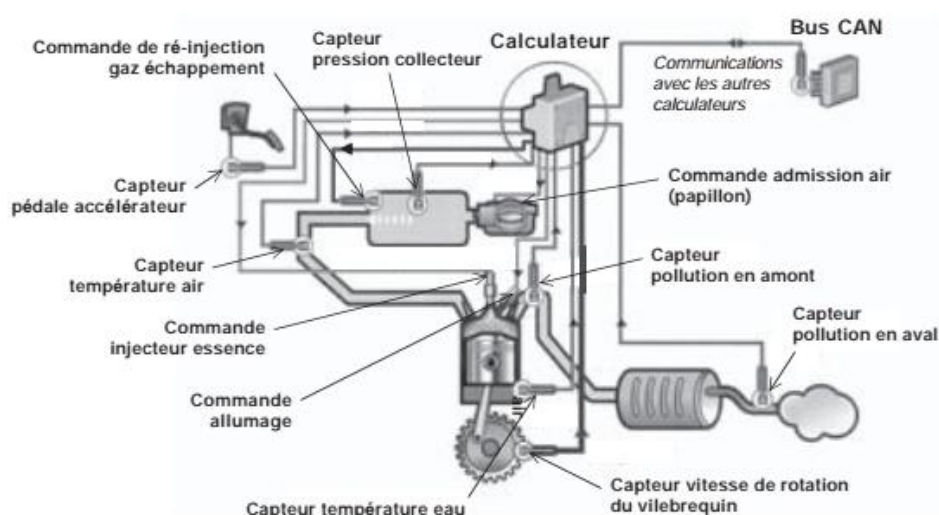


Figure 1.4 – Exemple d'une application de contrôle-commande d'un moteur à combustion

Cet exemple de contrôle-commande d'un moteur à combustion. Le contrôle-commande de cette application est fait par l'intermédiaire d'un ensemble de capteurs et d'actionneurs (pédale d'accélérateur, température air, pression air, température eau, rotation vilebrequin, capteurs de pollution, injection essence, allumage, admission air, etc.) et d'une connexion au réseau interne à l'automobile. L'analyse de cet exemple d'application permet de mettre en exergue **les principales caractéristiques** des systèmes de contrôle-commande.

- **Grande diversité des dispositifs d'entrées/sorties :**
- **Prise en compte des comportements concurrents :**
- **Respect des contraintes temporelles :**
- **Sûreté de fonctionnement :**

1.3.1. Caractéristique temporelle des systèmes de contrôle-commande

Le respect **des contraintes temporelles** d'une application de contrôle-commande dépend essentiellement de la dynamique du procédé. Cette caractéristique temporelle peut être très différente suivant l'application (**figure 1.5**) :

- **Milliseconde** : systèmes radar, systèmes vocaux, systèmes de mesures...
- **Seconde** : systèmes de visualisation, robotique...
- **Minute** : chaîne de fabrication...
- **Heure** : contrôle de réactions chimiques... Ce paramètre temporel correspond à l'ordre de grandeur de la capacité de réponse ou de traitement du système de contrôle-commande.

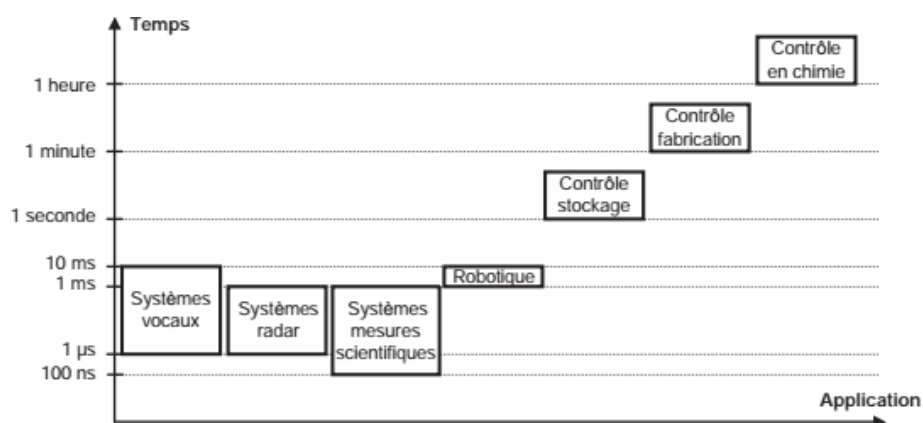


Figure 1.5 – Comparaison de la dynamique de différentes applications de contrôle-commande

1.3.2. Plusieurs contraintes à vérifier pour qu'un système temps réel soit fonctionnel

- **Exactitude logique** : (logical correctness) : sorties adéquates en fonction des entrées, assurant le comportement désiré pour le système suite à des événements et aux données communiquées.
- **Déterminisme logique** : les mêmes entrées appliquées au système produisent les mêmes résultats.
- **Exactitude temporelle** : (timeliness) respect des contraintes temporelles (ex : échéance). Les sorties sont présentées au bon moment.
- **Fiabilité** : le système répond à des contraintes de disponibilité (fiabilité du logiciel et du matériel).
- **Systèmes prédictibles** : on cherche à déterminer a priori si le système va répondre aux exigences temporelles.

Un système temps réel n'est pas un système "qui va vite" mais un système qui satisfait à des contraintes temporelles.

1.3.3. Echelle de temps et temps de réponse

Il est nécessaire de préciser et de formaliser **cette caractéristique temporelle** qui peut prendre de **nombreuses formulations**. Ainsi, nous pouvons définir de manière non exhaustive :

Durée d'exécution d'une activité : l'activité d'une application, qui peut être l'enchaînement de plusieurs activités élémentaires (acquisition, traitement, commande, affichage...), possède une durée d'exécution qui peut être mesurée de diverses manières.

Cadence de répétition ou **périodicité** d'une activité : l'acquisition d'une donnée ou la commande d'un actionneur peuvent nécessiter une régularité liée par exemple à la fréquence d'échantillonnage.

Date au plus tôt ou **date de réveil** : dans certains cas, un traitement doit être déclenché à une date précise relative par rapport au début de l'exécution de l'application ou absolue (plus rarement).

Date d'activation : cet instant correspond à l'exécution effective de l'activité.
Date au plus tard ou **échéance** : le traitement ou la commande d'un actionneur doivent être terminés à un instant fixé par rapport au début de l'exécution de l'application.

Temps de réponse : cette caractéristique peut s'appliquer à une activité de régulation ou à un ensemble d'activités de régulation ; elle est directement liée à la dynamique du système. Ce paramètre correspond à la différence entre la date de réveil et la date de fin de l'activité.

Gigue temporelle : ce paramètre caractérise la répétabilité d'une activité au fur et mesure de ses occurrences.

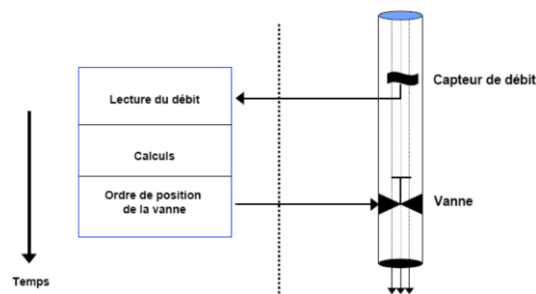
1.4. Applications de systèmes temps réel et embarqués

1.4.1. Domaines d'applications

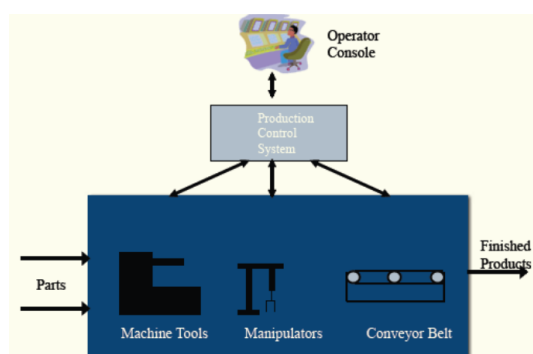
Jusqu'à une date récente, les systèmes temps réel et embarqués étaient destinés quasi-exclusivement aux applications de contrôle/commande de procédés (ou de phénomènes) physiques (tels que des laminoirs ou des usines de fabrication de voitures) et applications militaires. La notion de temps réel se confondait alors avec ce type d'applications. Le développement de l'informatique aidant, les applications temps réel et embarqués sont présentes aujourd'hui dans des domaines très variés comme le montre la liste suivante, même si elle n'est pas exhaustive :

- Télécommunications (transport de la parole, systèmes de commutation)
- Domaine médical (assistance et supervision de malades, ...)
- Contrôle de différents équipements dans les voitures, bus, trains, avions,
- Contrôle et régulation de trafic en milieu urbain,
- Guidage de véhicules en milieu urbain,
- Industrie (contrôle/commande de procédés manufacturiers ou continus, domaine militaire (suivi de trajectoires de missiles, ...))
- Aérospatial (suivi de satellites, simulation de vols, pilotage automatique,
- Multimédia (transport d'images et de voie, téléconférences, ...)
- Finance (suivi du cours des devises et actions, ...)
- Loisirs (consoles de jeu, ...), domestique (sécurité d'habitations, ...)
- Contrôle/commande d'appareils électroménagers

a. Exemple typique 1 : Système de contrôle de débit



b. Exemple typique 2 : Système de contrôle de production



1.5. Catégories de systèmes en temps réel :

Les systèmes temps réel peuvent se classer, selon le respect des contraintes temporelles, en **trois catégories** :

1.5.1. Temps réel dur ou critique (hard real time) : lorsqu'un résultat arrivant en **retard** provoque **un accident fatal** au système (génération d'une exception).

C'est aussi un système dans lequel certaines échéances ne doivent en aucun cas être dépassées.

Exemples : système de contrôle d'avion, système de conduite de missile,

1.5.2. Temps réel ferme (firm real time) : lorsqu'un résultat arrivant **en retard n'a plus d'importance** pour le système.

C'est aussi un système dans lequel le dépassement occasionnel des échéances ne met pas le système en difficulté.

Exemple : projection vidéo

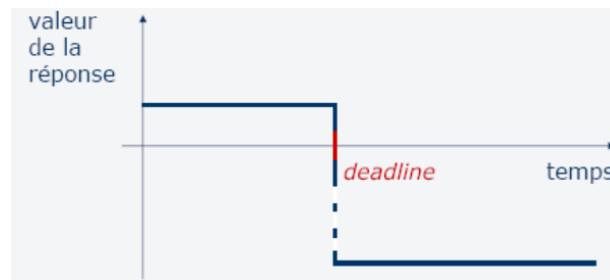
1.5.3. Temps réel mou ou souple (soft real time) : réduction progressive de l'intérêt d'un résultat arrivant en retard pour le système. L'arrivée exceptionnelle après échéance d'un événement attendu ne mettra pas le système en danger (erreur).

Exemple : système de distribution de son (streaming), système d'acquisition de données pour affichage.

- **Temps réel dur (*Hard real time*) :**

La réponse du système dans les délais est vitale.

L'absence de réponse est catastrophique (plus qu'une réponse incorrecte).



Exemples : Contrôle aérien, contrôle d'une centrale nucléaire...

- **Temps réel ferme (*Firm real time*):**

La réponse du système dans les délais est essentielle.

Le résultat ne sert plus à rien une fois le deadline passe.



Exemples : transactions en bourse... (Définition alternative : la pénalité de non-réponse est dans le même ordre de magnitude que la valeur de la réponse.

(**C'est subjectif** : le temps réel ferme de l'un peut être le temps réel dur de l'autre).

- **Temps réel mou (*Soft real time*) :**

La réponse du système après les délais réduit progressivement son intérêt.

Les pénalités ne sont pas catastrophiques.



Exemples : logiciel embarqué du téléphone portable, iPod, ...etc.

Exemples pratique.

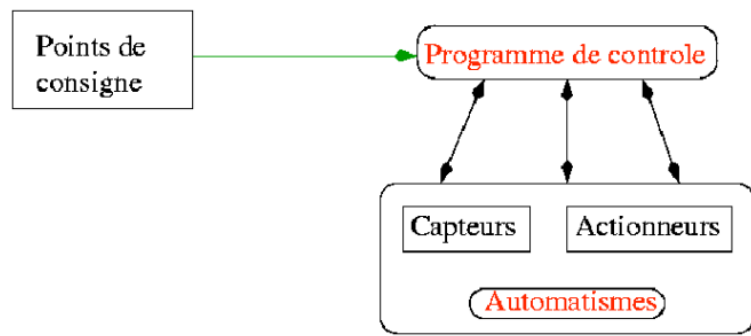
- a. Les systèmes d'interception de missiles ou de contrôle d'atterrissage d'avions sont considérés comme étant des systèmes à contraintes temporelles critiques. En effet, dans un système d'interception de missiles, si un missile n'est pas intercepté à temps, cela peut conduire à des conséquences graves. Si une opération d'atterrissage d'un avion commencée n'est pas achevée au bout d'un délai borné, un accident grave peut survenir.
- b. Les systèmes de téléconférences et les systèmes de réservation des compagnies aériennes sont considérés comme étant des systèmes à contraintes temporelles souples. Dans un système de téléconférence, son et image doivent être synchronisés, mais des dérives de synchronisation (dus à des messages tardifs) sont souvent tolérées. Pour les systèmes de réservation des compagnies aériennes, si une requête de réservation dure un peu plus longtemps que prévu, la seule conséquence regrettable peut être la perte d'un client qui quitte l'agence, car il ne souhaite plus attendre davantage.
- c. Certains systèmes de supervision d'installations industrielles sont considérés comme des systèmes à contraintes strictes. Par exemple, des pièces fabriquées sur une chaîne de production sont déposées sur un convoyeur et passent devant une caméra qui doit détecter certaines anomalies de fabrication. En fonction des anomalies détectées par le système de reconnaissance, les pièces sont dirigées vers la cellule de fabrication appropriée. La détection des anomalies doit se faire pendant que la pièce est en mouvement. Si la détection d'anomalie ne s'est pas faite dans les temps, la pièce est recyclée sur le convoyeur pour une deuxième analyse. On accepte que le système recycle une pièce sur dix car s'il recycle trop de pièces la chaîne de production s'en trouve ralentie.

1.6. Structures générales d'un STR

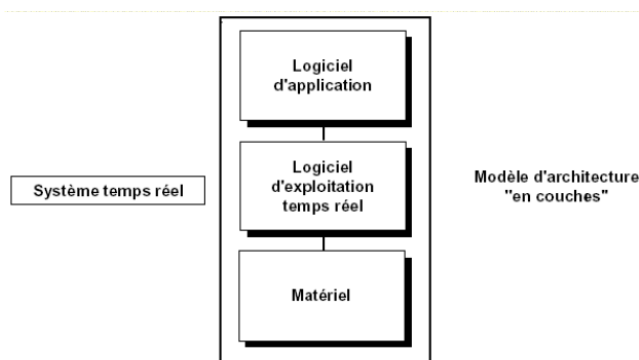
1.6.1. Boucle ouverte :



1.6.2. Boucle fermée :



- Un système temps réel est une association logicielle matérielle 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.
- La partie du logiciel qui réalise cette gestion est le système d'exploitation ou noyau temps réel.
- Ce noyau temps réel va offrir des services au(x) logiciel(s) d'application ; ces services seront basés sur les ressources disponibles au niveau du matériel.
- L'architecture d'un système informatique temps réel est la suivante :



1.7. Objets de base des applications temps réel et embarquées

Toute application temps réel est composée d'un ensemble d'actions qui opèrent sur des données en tenant compte de l'écoulement du temps et des événements qui apparaissent au

niveau du système informatique ou de son environnement. Les trois concepts (actions, données et événements) sont les composants de base de toutes les applications temps réel.

- **Une action** peut correspondre à tout traitement ayant une fonctionnalité bien précise au niveau de l'application, cela peut être une tâche dans les systèmes temps réel, une transaction dans les bases de données temps réel, une opération de transmission sur un réseau, etc. Au cours de la phase de conception, des actions sont introduites, par raffinement d'autres actions. On peut alors distinguer des actions simples et des actions composées.
- **Une donnée** représente toute information simple ou composée utilisée par une ou plusieurs actions pour s'exécuter. Une mesure de température, un fichier contenant une image ou un paquet véhiculé par un réseau sont des exemples de données.
- **Un événement** désigne un changement d'état dans le système informatique ou dans son environnement. "température > 100° C", " Fin de la tâche T1 ", " réception d'une image vidéo ", " Fin de transaction " sont des exemples d'événements.

Résumé, ce qu'il faut retenir

1. Un système de contrôle-commande

Nous pouvons définir un système de contrôle-commande comme un système informatique en relation avec l'environnement physique réel externe par l'intermédiaire de capteurs et/ou d'actionneurs, contrairement aux systèmes d'informatiques scientifiques (gestion de base de données, CAO, bureautique...) qui ont des entrées constituées de données fournies par des fichiers ou éventuellement un opérateur.

Un système de contrôle-commande reçoit des informations sur l'état du procédé externe, traite ces données et, en fonction du résultat, évalue une décision qui agit sur cet environnement extérieur afin d'assurer un état stable.

2. Systèmes embarqués

Voici quelques définitions pour cerner le concept de système embarqué :

Un système embarqué (SE) est un système informatisé spécialisé qui constitue une partie intégrante d'un système plus large ou une machine. Typiquement, c'est un système sur un seul processeur et dont les programmes sont stockés en ROM (Read-Only Memory). A priori,

tous les systèmes qui ont des interfaces digitales (i.e. montre, caméra, voiture...) peuvent être considérés comme des SE. Certains SE ont un système d'exploitation et d'autres non.

Un système embarqué est une combinaison de logiciel et matériel, avec des capacités fixes ou programmables, qui est spécialement conçu pour un type d'application particulier. Les distributeurs automatiques de boissons, les automobiles, les équipements médicaux, les caméras, les avions, les jouets et les téléphones portables sont des exemples de systèmes qui abritent (refuge) des SE. Les SE programmables sont dotés d'interfaces de programmation et leur programmation est une activité spécialisée.

Un système embarqué est une composante primordiale d'un système (i.e. un avion, une voiture...) dont l'objectif est de commander, contrôler et superviser ce système.

Un système embarqué est un système enfoui (enterré) (embedded system).

3. Systèmes temps réel

Selon l'aspect abordé dans les systèmes temps réel, on choisit une définition qui s'approche le plus de la problématique traitée. Ainsi, on assimile, selon le cas, un système temps réel à un système rapide, à un système en interaction directe avec un procédé physique, à un système réactif, à un système qui ne fournit pas de réponse en différé, à un système avec un comportement prévisible (attendu), à un système qui travaille sur des données fraîches, à un système robuste, etc

Cependant, toutes les applications temps réel (dites aussi applications contraintes par le temps ou encore applications temps contraint) ont en commun la prépondérance (supériorité) du facteur temps.

Un système en temps réel est défini comme un système dont l'exactitude du système dépend non seulement des résultats logiques des calculs, mais aussi de temps à laquelle les résultats sont produits.

Cette définition a conduit, en quelque sorte, à l'adage suivant que toute personne intervenant dans le cycle de vie d'une application temps réel connaît ou devrait connaître : Un résultat juste mais hors délai est un résultat faux.

Un système temps réel n'est pas un système "qui va vite" mais un système qui satisfait à des contraintes temporelles.



**L'information c'est bien,
mais l'information en
temps réel, c'est mieux !**