Projet Systèmes Hybrides

05/01/2023

Hypothèse : Il n'existe pas d'évènements qui changeraient de manière brutale la dynamique d'un syst. à temps continu

- Aucun évènement durant le temps de l'expérience
- Si un évènement survient, alors les dynamiques internes ont convergées.

Conséquences :
- Evènements modélisés comme des processus survenant à des échelles de temps différentes (réponse de système pas sur la même échelle)
- Plusieurs niveaux d'abstraction possible entre les SED et les STC

⚠ Il existe de nombreux phénomènes où les évènements (dépendants ou non d'autres variables) interagissent de manière significative sur la dynamique du système à temps continu => difficile de ne pas le prendre en compte (si le syst. est encore en régime transitoire)

Rappel SED : Trajectoire = séquence d'évènements non datés mais ordre important)

- état discret : abstractions d'états en classe discrète d'état
- dynamique évènementielle : l'occurence d'un évènement peut provoquer un changement d'état (asynchron/synchrone)
- Un évènement
  - ↳ instantané / sans durée
  - ↳ exogène ou endogène
  - ↳ contrôlable (action) ou spontané (perturbation / panne)

    $\mathcal{E}_c \subset \mathcal{E}$ | $\mathcal{E}_{nc} \subset \mathcal{E}$

  - ↳ observables ou non observable

    $\mathcal{E}_o \subset \mathcal{E}$ | $\mathcal{E}_{no}$

    informations issues des capteurs/actionneurs, poussoirs, alarme, intervention opérateur | évènements internes, évènements de faute

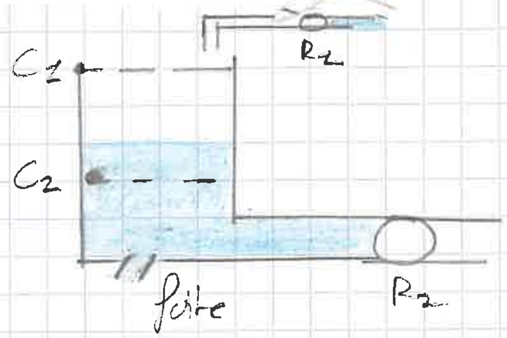
Exemple : Système discrétisable

Soit un réservoir d'eau pouvant être représenté par un système discret

- R1 : { ouvert ; fermé }
- R2 : { ouvert ; fermé }
- C1 : { niv_atteint ; niv_non_atteint }
- C2 : { niv_atteint ; niv_non_atteint }

=> espace d'état discret : $R_1 \times R_2 \times C_1 \times C_2$

évènements discret : ouverture_R1 ; Fermeture_R2 ; Niv_dessous C2



Détection d'un fuite grâce à :

$$\begin{cases} R_1 = \text{fermé} \; ; \; R_2 = \text{fermé} \\ C2 = \text{niv\_atteint} \\ \text{Evènement : niv\_dessous } C2 \end{cases}$$

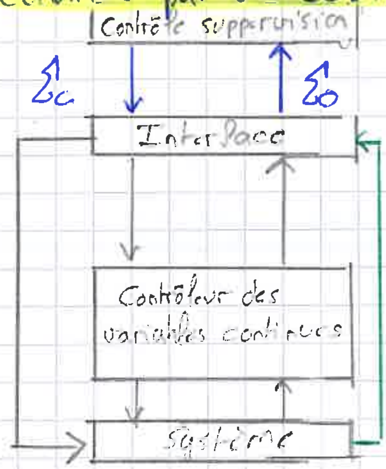
Autre exemple : [Sampath et al. 1995] Système d'air comprimé

. Modélisation formelle ↳ Langage L représentant l'ensemble de séquences d'évènements possibles sur Σ^* pour le système

↳ Automate G défini par $G = (X; \Sigma; \delta; x_0; X_f)$
avec X : ensemble fini d'états discrets
Σ : ensemble des évènements (alphabet)
δ : fonction de transition $\delta : X \times \Sigma \longrightarrow 2^X$
(représente à la fois les automates déterministes et non déterministes)
x_0 : état initial
X_f : états finals ou états marqués

Remarque : Machine de Mealy ou Machine de Moore

Modélisation par abstraction 1) Quand on modélise un SED, on abstrait toute dynamique continue par des évènements discret



2) Si plusieurs valeurs possibles pour une variable continue alors on crée plusieurs états discrets correspondant à ces valeurs

3) Quand un évènement faisant évoluer le système dans un autre état survient, le système subit une nouvelle expérience avec une nouvelle commande (consigne)

Exemple des problèmes posés par la modélisation Soit une balle de masse $m = 1$ kg qui rebondit sur le sol. On considère que la balle est soumise à la seule attraction terrestre. Quand la balle est en l'air, on écrit le Principe Fondamental de la Mécanique $\ddot{x}(t) = -g$ où x est la position de la balle par rapport au sol $\Rightarrow \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -g \end{cases}$

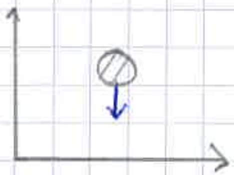
Quand la balle est au sol, on doit trouver une autre modélisation. Prenons

$\begin{cases} \dot{x}_1 = 0 \\ \dot{x}_2 = -\lambda x_2 \end{cases}$ ⚠ mais problème car on ne définit pas mathématiquement les évènements "en l'air" et "au sol" ⇒ Comment définir, on fonction des variables d'état ou autres ces notions ?

De plus, la définition des dérivées temporelles au sol est un peu étrange car la dérivée est mal définie dans ce cadre ⇒ on a besoin d'un nouveau formalisme pour mettre en évidence un changement instantanée de vitesse et d'accélération.

Bibliographie : Introduction to Discrete Event Systems, C.G Cassandras and S. Lafortune.
Hybrid dynamical systems : Modeling, Stability and Robustness, R. Goebel, R.G. Sanfelice and A.R. Teel.
The Theory of hybrid automata, T.A. Henzinger.
Hybrid Feedback Control R.G. Sanfelice.

Retour sur l'exemple de la balle : **En l'air** : mode 1



$m\ddot{x}(t) = -mg$ avec $m = 1\,kg$

$$\boxed{\ddot{x}(t) = -g}$$

Au sol : $\ddot{x}(t) = -g + \cancel{F_{Rsol}} + F_{rebond}$ mode 2

$$\ddot{x} = \underbrace{-\lambda \dot{x}(t)}_{F_{rebond} \text{ proportionnel à la vitesse}}$$

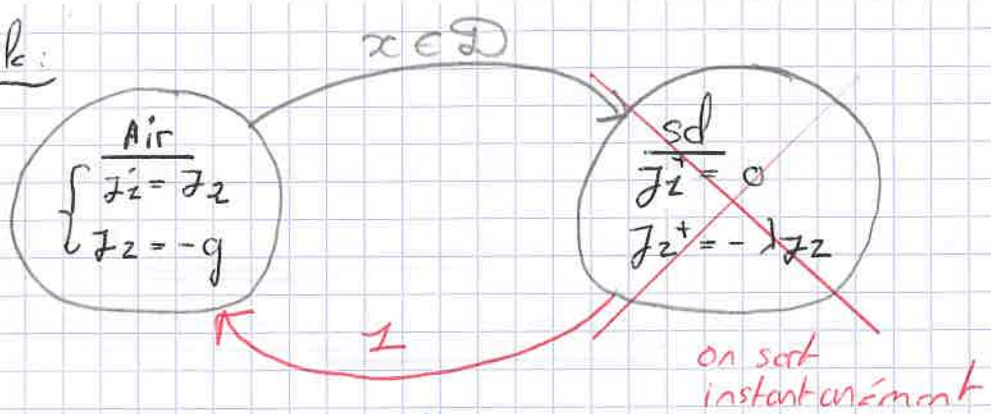
⇒ états en l'air sont $z(t) = \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix}$ donnant $\begin{cases} \dot{z}_1(t) = z_2(t) \\ \dot{z}_2(t) = -g \end{cases}$

⇒ états au sol sont $z(t) = \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix}$ donnant $\begin{cases} \dot{z}_1(t) = 0 \\ \dot{z}_2(t) = -\lambda z_2 \end{cases}$

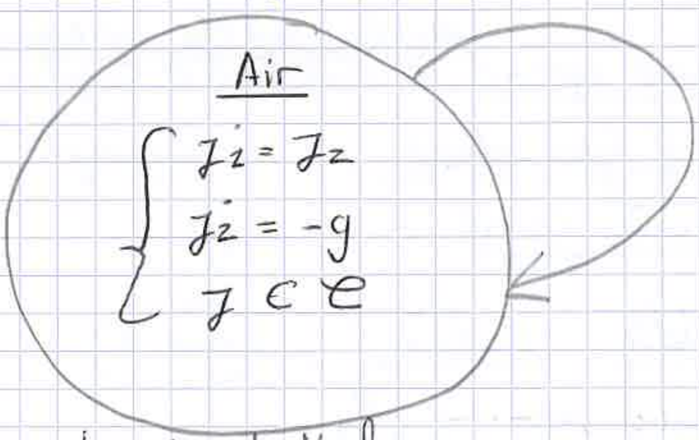
Ainsi

En l'air	Au sol
$C = \{z \in \mathbb{R}^2 \mid z_1 > 0 \text{ ou } z_1 = 0 \text{ et } z_2 > 0\}$	$D = \{z \in \mathbb{R}^2 / \text{et } z_1 = 0,\ z_2 \leq 0\}$ Phénomène de réinitialisation avec $z_1^+ = 0$ et $z_2^+ = -\lambda z_2$ L'évolution est instantané si D vérifié

1^er^ modèle :



$\Downarrow$



$x \in \mathcal{D} \Rightarrow \begin{cases} z^+ = 0 \\ \dot{z}^+ = -\lambda \dot{z} \end{cases}$

Formalisme avec les automates de Mealy

$b / x^+ = x^-$

capteur — action

Retour sur l'exemple des lucioles :

On a n lucioles $\Rightarrow$ n horloges : $z_i \; ; \; i \in \{1; \dots; n\}$

Hypothèse : $z_{i_{MAX}} = 1$

Cas nominal (pas de flash) : $\dot{z}_i = 1$

Cas avec flash : $\exists\, \ell$ tel que $z_\ell = 1$ avec $\begin{cases} z_\ell^+ = 0 \\ z_i^+ = z_i + \varepsilon \; ; \; i \neq \ell \end{cases}$

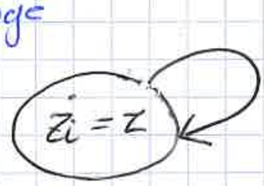
Ainsi

Sans flash	Avec flash
$\mathcal{C} = \{ z \,/\, z \in [0; 1[^n \}$	$\mathcal{D} = \{ z \,/\, z \in [0; 1]^n$ et $\max z_i = 1 \}$
$z = \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix}$ donnant $\dot{z}_i = 1$	donnant $z_i^+ = \begin{cases} z_i + \varepsilon & \text{si } z_i + \varepsilon < 1 \\ 0 & \text{sinon} \end{cases}$

$\Rightarrow$ nécessite l'utilisation d'horloge
$\Rightarrow$ fonction multivaluée

$z \in \mathcal{D} \Rightarrow z_i^+ = \begin{cases} z_i + \varepsilon & \text{si } z_i + \varepsilon < 1 \\ 0 & \text{sinon} \end{cases}$

Modélisation graphique



Retour sur l'exemple des lucioles fatigués

Modélisations possibles :

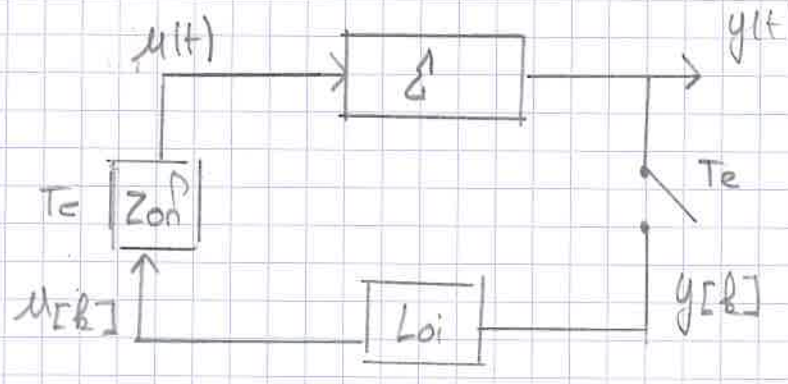
1) $z_i = e_i$ => e_i change à chaque réinitialisation -> c'est le modèle le plus simple

2) Un nouvel état à chaque réinitialisation => on garde trace de "l'historique"
=> infinité d'états discrets

Synthèse : Le choix du modèle dépend de ce que l'on veut faire

Retour sur l'exemple de l'échantillonneur d'ordre zéro :

Principe : Tant qu'il ne reçoit pas de nouvelle donnée, envoi de t_j (la même chose)



Ainsi

Entre deux instants d'échantillonnage	Aux instants d'échantillonnage
$z = \begin{bmatrix} n \longleftarrow T_c \\ u \longleftarrow \text{commande} \\ z \longleftarrow \text{horloge Zoh} \end{bmatrix}$	$z^+ = \begin{bmatrix} x \\ K(x(t)) \\ 0 \end{bmatrix}$
$\begin{cases} \dot{x}(t) = f(x(t); u(t)) \\ \dot{u}(t) = 0 \\ \dot{z}(t) = 1 \end{cases}$	
$\mathcal{C} = \begin{cases} z / x \in \mathbb{R}^n ; u(t) \in \mathbb{R}^m \\ z \in [0; T_c] \end{cases}$	$\mathcal{D} = \{ z / x \in \mathbb{R}^n ; u \in \mathbb{R}^m ; z = T_c \}$

Modélisation

$z \in \mathcal{D} \Rightarrow z^+ = \begin{bmatrix} x \\ K(x(t)) \\ 0 \end{bmatrix}$

$\dot{x} = f(x(t); u(t))$
$\dot{u} = 0$
$\dot{j} = 1$

=> permet de prouver la stabilité du système

=> dans ce système, la commande est continue et le système continu
=> modèle plus proche de la réalité

Mots clés pour la bibliographie : "Hybrid dynamical systems" ou "Hybrid Automation"

Bloc Simulink : Stateflow

Remarque :

↳ concernant les automates hybrides cf. Cassandras and Lafortune

↳ concernant les systèmes hybrides cf. Tael

Notation : pour la réinitialisation, x_i^+ avec + signifiant l'instant d'après

Projet Systèmes Hybrides
Modélisation

Modèle de la balle rebondissante
↳ modèle avec condition de réinitialisation :

Il existe pour ce modèle 1 mode ⇒ 1 état discret ! Et 2 états continus (position et vitesse) regroupés dans un vecteur x.

↳ un espace de flot $\mathcal{C} = \{x ; x_1 > 0 \text{ ou } (x_1 = 0 \text{ et } x_2 \geq 0)\}$ et l'équation dynamique $\begin{cases} \dot{x_1} = x_2 \\ \dot{x_2} = -g \end{cases}$

↳ un espace de saut $\mathcal{D} = \{x ; x_1 = 0 \text{ et } x_2 \leq 0\}$ et une équation de réinitialisation $\begin{cases} x_1^+ = x_2 \\ x_1^+ = -g \end{cases}$

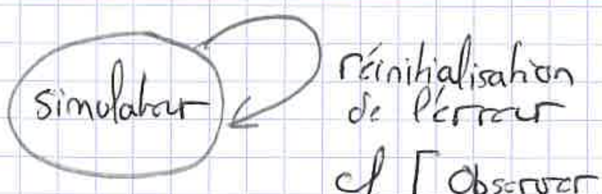
Observateur paresseux :

Problème : Synthétiser un observateur pour un système linéaire pour lequel les mesures sont obtenues de manière sporadique $\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = y(t_k) \end{cases}$

On sait que la première mesure arrive au pire à T_2 secondes et que deux mesures consécutives sont au moins espacées de T_1 secondes et au pire de T_2 secondes

Ainsi, $\begin{cases} 0 \leq t_1 \leq T_2 \\ T_1 \leq t_{k+1} \leq T_2 \end{cases}$ avec t_1 le temps de la première mesure

Comment construire un observateur ?



cf [Observer, F. Francesco]

Fonctionnement du thermostat
Deux modes de fonctionnement q_1 et q_2 correspondant au chauffage éteint et au chauffage allumé.

L'état continu x correspond à la température dans la salle.

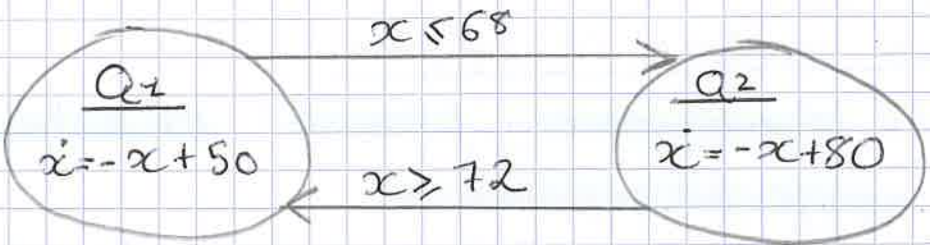
Le thermostat est programmé pour maintenir une température entre 68 et 72 degrés

Lorsque le chauffage est allumé (q_2), la condition pour l'éteindre est $x \geq 72$.

Si le chauffage est éteint (q_1), la condition pour l'allumer est $x \leq 68$.

En l'absence de chauffage, la température descend jusqu'à 50.

Ainsi,



On a l'automate hybride : $G = (Q, X, f, \phi, q_0, x_0)$

avec Q : ensemble des états discrets de CI. q_0

X : ensemble des états continus de CI. x_0

⚠ Il faut spécifier les conditions initiales

On définit $f : Q \times X \longrightarrow X$: la fonction dynamique temporelle

$$\begin{cases} f(q_1; x) = -x + 50 \\ f(q_2; x) = -x + 80 \end{cases}$$

$\phi : Q \times X \longrightarrow X$: la fonction dynamique discrète

$$\phi(q_1; x) = \begin{cases} q_2 & x \leq 68 \\ q_1 & x > 68 \end{cases} \quad \text{et} \quad \phi(q_2; x) = \begin{cases} q_1 & x \geq 72 \\ q_2 & < 72 \end{cases}$$

Fenêtre de Zénon : switch infini entre deux valeurs d'état discret (ex : clocher)

A éviter et plutôt jouer sur les états ou sur le temps => ajout d'un timer.

Souvent, il n'y a pas de réinitialisation des états continus

N.B :
- Dans cet exemple, il faut spécifier les conditions de gardes sur les variables d'état continu.
- En revanche, pas de spécification des invariants associés aux modes discrets (définition des domaines par défaut).
- Il n'y a pas d'occurrence d'événement discret.
- Pas de condition de réinitialisation des variables d'état continu

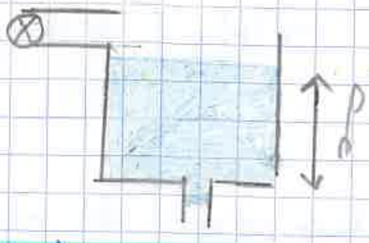
Fonctionnement du réservoir avec délais :

But de la commande : Eviter que le bac ne se vide ou se remplisse trop.

On a : Flux entrant $\lambda = 3$
Flux sortant $\mu = 1$
Délais entre l'envoi de la commande et son exécution $\delta = 0,5$

=> le temps est donc rajouté dans le vecteur d'état pour représenter ce délais => 4 états

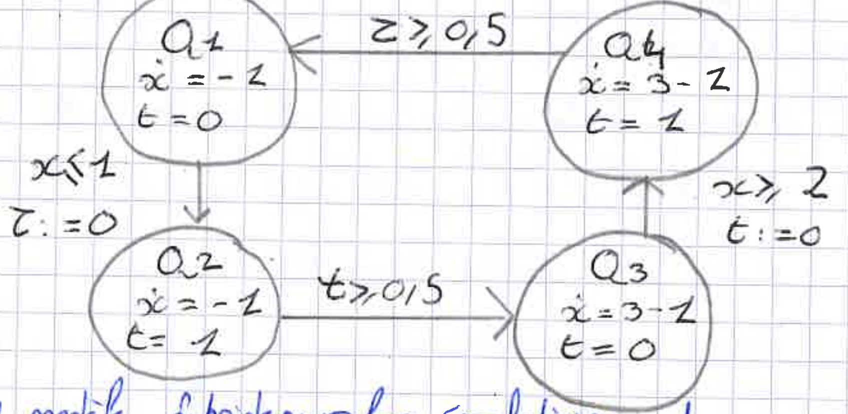
On distingue, le Timer virtuel : on peut le réinitialiser
le flot : évolution continue



La fonction multivariée F est $F \longrightarrow X$

$x \longmapsto X_x \subset X$ on associe à x un sous espace de X

Ainsi,



Intérêt du modèle hybride pour les évolution !
Dans cet exemple, : spécification de conditions de gards sur les variables d'état continu (x, τ)

. pas de spécification des invariants associés aux modes discrets (définition de domaines par défaut)

. définitions de condition de réinitialisation pour la variable temporelle τ.

Dans ce modèle, les états discrets sont modélisés comme étant états continus. Le changement d'états discret se fait pour $x^+ = a$, dans le flux et ces états n'évoluent pour $q = 0$.

Synthèse : Objectifs différents entre | l'automate hybride (insertion SED)
| système hybride (insertion temps continue)

. Pour les ~~automates~~ systèmes hybrides : prouver la convergence, la stabilité d'un exemple
. Pour les ~~systèmes~~ automates hybrides : contrôlabilité, détection de panne, commande si précisé

A faire : . modéliser avec les deux modèles le fonctionnement du dispositif
. utiliser state flow

et cela pour la halle

Exemple : Fonctionnement d'une machine avec réparation

On dispose d'une machine non fiable avec des délais :

↳ 3 modes de fonctionnement : idle (Q_1), busy (Q_2) et down (Q_3)

↳ l'état physique $x(t)$ est la température de la machine, $\tau(t)$ est une horloge qui permet d'obtenir un délais (timeout) => l'état continu est donc

$$x(t) = \begin{bmatrix} x(t) \\ \tau(t) \end{bmatrix}$$

↳ on rajoute des variables d'entrée de 2 types :

- des événements discrets $\Sigma' = \{\alpha, \beta, \gamma\}$ extension de la fonction ϕ aux conditions ne dépendant pas de l'état continu x.
  - -> α représente le démarrage de la machine quand elle est en veille
  - -> β représente l'arrêt de la machine lorsqu'elle est en cours d'utilisation
  - -> γ représente la réparation de la machine lorsqu'elle est en panne.
- des commandes continues : une entrée contrôlable $u \in U$ est inclus telle que $\dot{x} = f(q, x, u)$

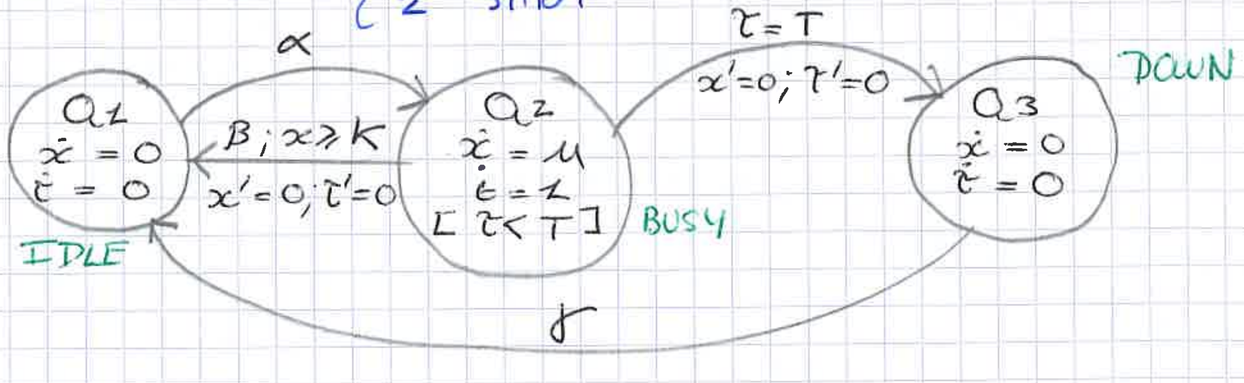
↳ si la machine reste dans Q_2 pendant une durée T ou plus, cela cause immédiatement une transition vers Q_3

↳ Dans Q_2, le chauffage de la machine est activé $\dot{x} = u$ où u est une entrée scalaire contrôlable. Une horloge est déclenchée : $\dot{\tau} = 1$.
Dans ce mode Q_2, une condition d'invariant est définie $\tau < T$.

↳ la condition de garde $\tau = T$ provoque le passage à Q_3. Pendant cette évolution, l'état continu x subit une condition de réinitialisation.

↳ Une autre transition est possible à partir de Q_2 lorsque la température est supérieure à K ou que l'événement β apparaît.

Ainsi, $\phi(Q_2, x, \tau, e) = \begin{cases} 3 & \tau = 3 \\ 1 & x \geq K \text{ et } e = \beta \\ 2 & \text{sinon} \end{cases}$



Remarques : -> conditions de garde sur X défini dans la fonction ϕ

-> condition d'invariant (domaine) : sous-ensemble de X associé au mode Q tel que x doit appartenir à cet ensemble pour rester dans ce mode

-> condition de réinitialisation sur x quand on change d'état discret

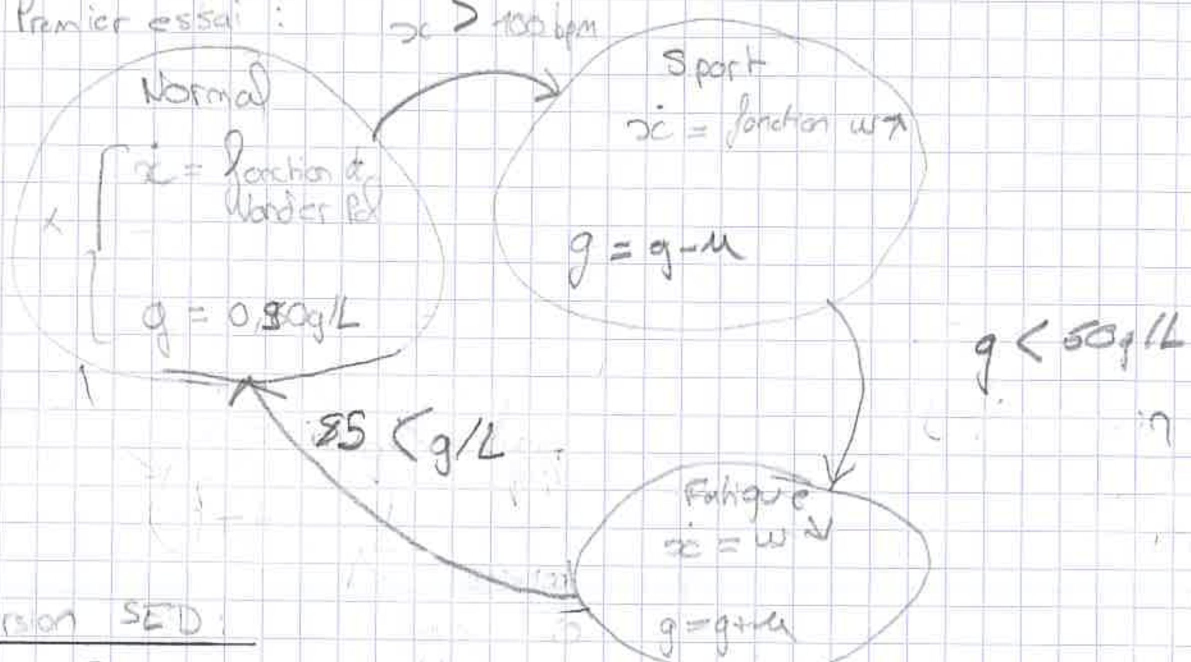
⚠ Distinction entre évènement exogène et endogène dans un ~~domaine~~ système hybride :

-> exogène : vient de l'extérieur pour forcer une transition discrète dans un automate classique

-> endogène : évènement apparaissant quand une variable continue dépendant du temps entre dans un ensemble particulier $x(t) \geq k$ ou $T \geq T$.

Modélisation : $G = (Q ; X ; E ; U ; f ; \phi ; Inv, Guard, \rho ; q_0 ; x_0)$

Premier essai :



Version SED :

$G = \{ \phi, Q, \delta, X ; x_0 ; E, U, f, Inv, Guard, \rho \}$

avec : $Q = \{ \text{Normal}, \text{Sport}, \text{Fatigue} \}$

$X = \begin{bmatrix} x \\ g \end{bmatrix}$

ϕ fonction de transition $\begin{cases} \phi(\text{Normal}) = \text{Sport si } x > 100 \text{ bpm} \\ \phi(\text{Sport}) = \text{Fatigue si } T > 60 \text{ min} \\ \phi(\text{Fatigue}) = \text{Normal si } x \leq 70 \text{ bpm} \end{cases}$

et $f = \{$ à définir !

ρ : fonction de réinitialisation $\rho = 0$

$q_0 =$ Normal : état discret initial

$x_0 = \begin{bmatrix} 70 \\ 0{,}90 \end{bmatrix}$ état continue initial

$E = \{ x > 100\text{bpm} ; \ g < 60\,g/L \ ; \ g > 85\,g/L \}$

$U = \{ u = 10\,g/L \}$

$\mathrm{Inv} = \emptyset$

Guard $= E$

<u>Version Temps Continu</u> : $\mathcal{H} = (\mathcal{C}, F, \mathcal{D}, G)$

Un espace d'état $X = \begin{bmatrix} x \\ g \end{bmatrix}$

Un espace de flot $\mathcal{C} = \{ X /$

un espace de saut $\mathcal{D} = \{ X /$

Equation de ré-initialisation $G = 0$

Trajectoires et Propriétés :

Guard = ensemble qui représente les conditions sur les variables continu

Trajectoire (au sens SED) : séquence d'états discrets

⚠ Phénomène de blocage (deadblock) apparait quand l'état ne peut plus évoluer. Par exemple lorsque la variable continue respecte la condition d'invariant dans un état Q_e mais s'oppose à la condition de Guard pour évoluer

=> par éviter ce problème : on s'assure que $Inv(q)$ est ouvert et quand $x \notin Inv(q)$, on a $x \in \cup\, Gard(q,q')$ (les ensembles $Inv(q)$ et Guard sont complémentaires)

Prévenir un comportement Zenon : l'évolution de l'état implique un nombre infini de transitions discrètes en un temps fini (on change d'état sans arrêt avant de voir les variables continues s'établir).

=> solution : introduire une borne inférieure sur le temps entre deux transitions discrètes successives (à définir selon l'application : attendre que les variables atteignent le régime permanent). Le franchissement de la Gard est conditionné par ce timer. Par exemple : il est possible de changer la fonction sign(.) par une hystéresis

Automates non déterministes : lorsqu'une transition discrète n'est pas unique. Il existe plusieurs façon de faire évoluer l'état. Trois différences dans la définition formelle donnée de l'automate hybride non déterministe :

$$\phi : Q \times X \times E \longrightarrow 2^{Q} \quad \text{(plusieurs états Q atteint)}$$

$$\rho : Q \times Q \times X \times E \longrightarrow 2^{X}$$

q_0 peut être un ensemble d'états $q_0 \subseteq Q$

cf. bibliographie [CLOG] Peut provenir de problème d'incertitude ou de modélisation. Ne peut pas être utilisé pour la commande mais plutôt pour des problèmes d'estimation, de reconstruction de trajectoire.

Trajectoire (des états TC)

Pour les syst. TC le temps ordinaire est $t \in \mathbb{R}_+$ } Pour les systèmes
Pour les syst. discret $j \in \mathbb{N}$ } hybrides : union t, j

=> défini le temps hybride par le domaine temporel $E = \bigcup_{j=0}^{j-1} ([t_j ; t_{j+1}], j)$
Pour j incrémenté à chaque franchissement de garde.

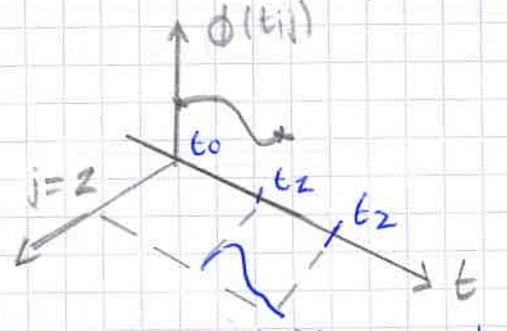
Remarques : • cette union est constitué de couple en nombre fini ou infini
• le dernier intervalle $[t_j ; t_{j+1}]$ peut être infini, signifiant que le système hybride finit dans le flot
• on peut définir un ordre $(t, j) \leq (t', j')$ si $(t \leq t')$ ou $((t = t')$ et $j \leq j')$
• on introduit deux bornes $\sup_t E = \sup \{ t \in \mathbb{R}_+ ; \exists j \in \mathbb{N}, (t, j) \in E \}$

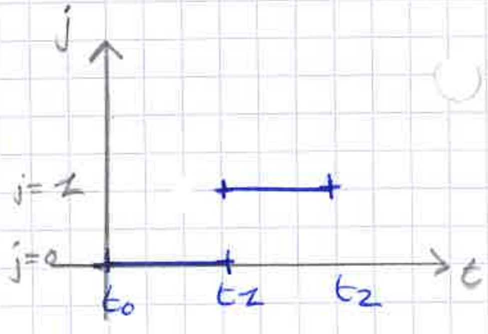
$\sup_j E = \sup \{ j \in \mathbb{N} ; \exists t \in \mathbb{R}_+ ; (t, j) \in E \}$

On appelle arc hybride une fonction $\phi : E \longrightarrow \mathbb{R}^N$ avec E un domaine hybride et $\phi(t, .)$ localement absolument continu sur t, $(t, j) \in E$.
Ainsi, sur des intervalles $[t_j ; t_{j+1}]$, la fonction ϕ est différentiable. On note le domaine de définition de ϕ $\mathrm{dom}(\phi) =$ cf diapo

Types d'arc hybride :

① Non trivial si $\mathrm{dom}(\phi)$ contient au moins 2 points





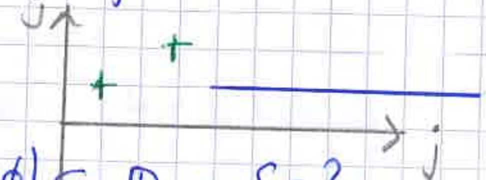
② Complet si $\mathrm{dom}(\phi)$ n'est pas borné (en temps ordinaire ou en j)

③ Zéno si l'arc est complet et que $\sup_t (\mathrm{dom}(\phi))$ n'est pas borné

④ Eventuellement discret si $T = \sup (\mathrm{dom}(\phi))$ est borné et $\mathrm{dom}(\phi) \cap (T) \times \mathbb{N}$ contient au moins deux points

⑤ Discret si non trivial et $\text{dom}(\phi) \subset \{0\} \times \mathbb{N}$

⑥ Éventuellement continue si $J = \sup_j (\text{dom}(\phi))$ est borné et $\text{dom}(\phi) \cap \mathbb{R}_+ \times J$ contient au moins deux points



⑦ Continu si non trivial et $\text{dom}(\phi) \subset \mathbb{R}_+ \times \{0\}$

<u>Solutions</u> Un arc hybride est une solution du système hybride $\mathcal{H} = (\mathcal{C}; F; \mathcal{D}; G)$ si la condition initiale $\phi(0;0) \in \overline{\mathcal{C}} \cup \mathcal{D}$

fermeture de $\mathcal{C}$ (domaine C et la frontière)

par tout les j tel que $\underbrace{\text{l'intérieur de } I}_{\text{domaine sans la frontière}}{}^{j} = \{t \,;\, (t;j) \in \text{dom}(\phi)\}$ est non vide.

?

$$\begin{cases} \phi(t;j) \in \mathcal{C} \,;\ \forall t \in \text{int}(I^j) \\ \dot{\phi}(t;j) \in F(\phi(t;j)) \,;\ \forall t \,(pp) \in I^j \end{cases}$$

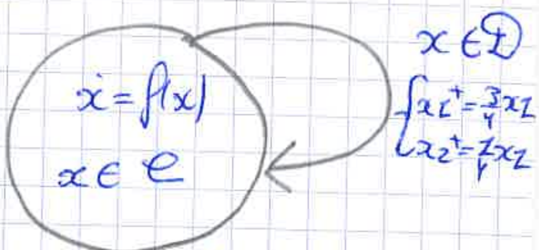
Pour tout les $(t;j) \in \text{dom}(\phi)$ tel que $(t;j+1)$ cf. diapo

<u>Exemple</u>: On considère le système hybride défini par $\mathcal{C} = \mathbb{R}^2 \setminus \mathcal{D}$; $f(x) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$$\mathcal{D} = \{x,\ 0 \leq x_2 \leq -\frac{1}{5} x_1 + \frac{2}{5}\} \quad \text{et} \quad g(x) = \begin{bmatrix} \frac{3}{4} x_1 \\ \frac{1}{4} x_2 \end{bmatrix}$$

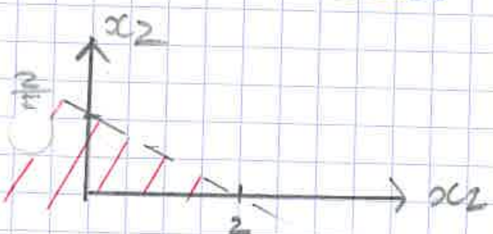
D'après les données $\begin{cases} \dot{x}_1 = 1 \\ \dot{x}_2 = 1 \end{cases}$; $x \in \mathcal{C}$

et par ré-initialisation $\begin{cases} x_1^+ = \frac{3}{4} x_1 \\ x_2^+ = \frac{1}{4} x_2 \end{cases}$ ainsi



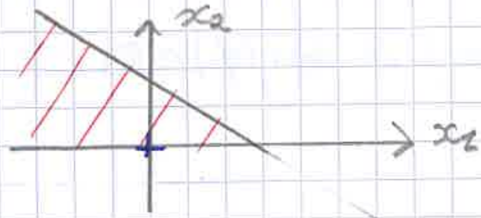
$\Rightarrow$ 1 mode discret

<u>Représentation du domaine $\mathcal{D}$</u>:



$$\begin{cases} \dot{x}_1 = 1 \\ \dot{x}_2 = 1 \end{cases} \Leftrightarrow \begin{cases} x_1(t) = t + x_{10} \\ x_2(t) = t + x_{20} \end{cases}$$

Arc hybride CI en (0,0)

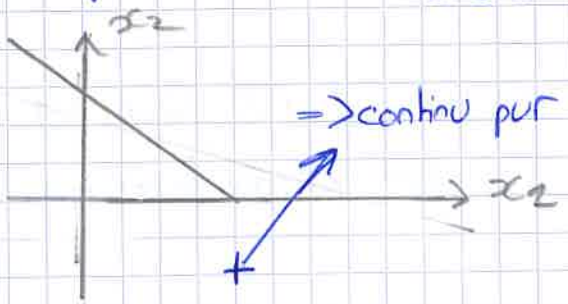


$\Rightarrow$

on reste discret pur

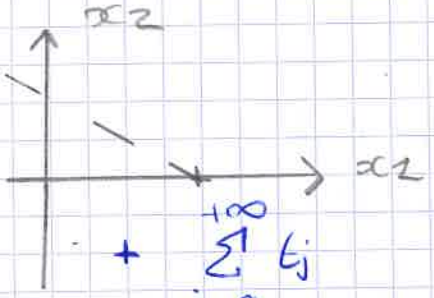
On vient de mettre en évidence un **point d'équilibre** sur la dynamique discrète

Arc hybride CI en (2;-1)



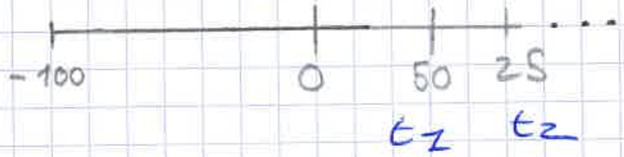
$\Rightarrow$ continu pur

Arc hybride CI (1;-1)



$$\sum_{j=0}^{+\infty} t_j$$

Paradoxe de Zénon :



$$\sum_{i=1}^{+\infty} \left(\frac{1}{2}\right)^i$$ suite géométrique de raison 1

$$= \frac{1}{1-q} = 2$$

Toolbox HyEQ : hybrid.soe.ucsc.edu/software