

ne résident pas trop longtemps dans le four. Notons que le GET de la figure 1 n'intègre pas de dispositif permettant de satisfaire une telle contrainte, il modélise simplement le fonctionnement au plus tôt du système².

Le travail présenté dans ce papier vise à synthétiser une loi de commande, sous la forme d'un précompensateur filtrant la dynamique des entrées u_1 et u_2 ,

- telle que le comportement du système corrigé soit aussi proche que possible de celui d'un modèle de référence donné (problème classique de commande par modèle de référence),
- et qui, de plus, garantisse un temps de séjour des jetons borné dans certaines places (problème de temps critique).

Nous verrons que cette dernière contrainte se formalise grâce à une équation implicite satisfaite par le vecteur d'état. La synthèse de la correction repose en conséquence sur des outils de recherche de points fixes d'applications définies sur des dioïdes.

Notons que les *systèmes à temps critique* ont déjà fait l'objet d'études, notamment *via* le formalisme des réseaux de Petri temporels [Merlin, 1974], [Berthomieu and Diaz, 1991]. En revanche, à notre connaissance, aucune étude n'a encore été menée au travers de la théorie des systèmes linéaires sur les dioïdes.

II. OUTILS ALGÈBRIQUES

A. Algèbre des dioïdes

Un dioïde \mathcal{D} est un semi-anneau pour lequel la loi additive, notée \oplus est idempotente ($\forall a \in \mathcal{D}, a \oplus a = a$) [Baccelli et al., 1992]. Les éléments neutres des lois additive et multiplicative (notée \otimes) sont notés respectivement e et ε . En raison de l'idempotence de \oplus , une relation d'ordre naturelle est définie par $a \preceq b \Leftrightarrow b = a \oplus b$. L'élément $a \oplus b$ correspond à la borne supérieure de la paire (a, b) , et un dioïde admet une structure de demi-treillis [Davey and Priestley, 1990]. Si un dioïde est complet, c'est-à-dire, s'il est fermé pour les sommes infinies et si le produit distribue sur les sommes infinies, alors celui-ci admet une structure de treillis complet. Cela signifie que toute partie finie ou infinie de l'ensemble admet à la fois une borne supérieure (correspondant à la somme \oplus des éléments de la partie) et une borne inférieure. Pour une paire (a, b) la borne inférieure, notée \wedge , est définie par : $a \wedge b = \bigoplus_{\{x|x \preceq a, x \preceq b\}} x$.

Soit f une application définie d'un dioïde complet \mathcal{D} dans lui-même, elle est dite *isotone* si $a, b \in \mathcal{D}, a \preceq b \Rightarrow f(a) \preceq f(b)$ et *semi-continue supérieurement* (resp. *inférieurement*) si $f(a \wedge b) = f(a) \wedge f(b)$ (resp. $f(a \oplus b) = f(a) \oplus f(b)$). La *théorie de la résiduation* permet de définir une *pseudo-inverse* pour certaines applications isotones définies sur des ensembles ordonnés [Blyth and Janowitz, 1972] et, par spécialisation, sur des dioïdes [Baccelli et al., 1992]. Plus précisément, si le plus grand élément de l'ensemble $\{x|f(x) \preceq b\}$ existe pour tout $b \in \mathcal{D}$, alors il est noté $f^\sharp(b)$ et f^\sharp est appelée la *résiduée* de f . L'application $L_a : x \mapsto a \otimes x$ définie sur un dioïde complet est résiduable, autrement dit, pour tout $b \in \mathcal{D}$, l'inéquation

$L_a(x) = ax \preceq b$ admet une plus grande solution notée

$$L_a^\sharp(b) = a \backslash b. \quad (1)$$

B. Plus grands points fixes d'applications définies sur des dioïdes complets

Dans cette section, on étudie les bornes supérieures³ des ensembles définis ci-dessous.

Notation 1 Soit f une application isotone d'un dioïde complet \mathcal{D} dans lui-même. On note \mathcal{F}_f et \mathcal{P}_f les ensembles des *points fixes* et des *points post-fixes* de f :

$$\mathcal{F}_f = \{x|f(x) = x\}, \quad \mathcal{P}_f = \{x|f(x) \succeq x\}. \quad (2)$$

On rappelle ci-dessous un premier théorème qui établit que l'ensemble \mathcal{P}_f admet une structure de treillis complet. Le deuxième théorème montre qu'il est possible de substituer la recherche de la borne supérieure de \mathcal{F}_f à celle de \mathcal{P}_f . Exploitant cela, on peut proposer ensuite un algorithme qui, s'il converge, aboutit à cette borne supérieure. Cet algorithme sera utilisé par la suite pour la synthèse de correcteurs garantissant une contrainte de temps critique.

Théorème 1: [Baccelli et al., 1992, thm 4.72]

L'ensemble \mathcal{P}_f a une structure de treillis complet.

Proof: Puisque f est isotone, si $x, y \in \mathcal{P}_f$, alors $x \oplus y \preceq f(x) \oplus f(y) \preceq f(x \oplus y)$, et donc $x \oplus y \in \mathcal{P}_f$. Ce résultat s'étend naturellement à tout sous-ensemble fini ou infini de \mathcal{P}_f , ce qui montre la structure de sup-demi-treillis complet de ce dernier. De plus, $\varepsilon \preceq f(\varepsilon)$, et donc $\varepsilon \in \mathcal{P}_f$, ce qui suffit à affirmer que \mathcal{P}_f a une structure de treillis complet. ■

Un théorème énoncé dans [Tarski, 1955] montre qu'une application f isotone définie sur un treillis complet admet (au moins) un point fixe. De plus, on peut montrer que le plus grand point post-fixe (*i.e.* la plus grande solution de l'inéquation $f(x) \succeq x$) est aussi le plus grand point fixe (voir, par exemple, [Davey and Priestley, 1990, théorème 8.20]). Sa spécification sur un dioïde complet donne le théorème suivant.

Théorème 2:

Soit f une application définie sur un dioïde complet \mathcal{D} . Si f est isotone, alors on a

$$\text{Sup } \mathcal{P}_f = \text{Sup } \mathcal{F}_f, \quad (3)$$

$$\text{et } \text{Sup } \mathcal{F}_f \in \mathcal{F}_f. \quad (4)$$

Proof:

(3) Soit $x' = \text{Sup } \mathcal{P}_f$. D'après le théorème 1, x' appartient à \mathcal{P}_f , et il s'agit donc du plus grand x tel que $f(x) \succeq x$. Par isotonie de f , on a $f \circ f(x') \succeq f(x')$. Autrement dit, $f(x')$ est solution de $f(x) \succeq x$, et comme x' en est la plus grande solution, on en déduit que $x' \succeq f(x')$. De $f(x') \succeq x'$ et $x' \succeq f(x')$, il découle que x' est un point fixe de f , *i.e.* $x' \in \mathcal{F}_f$. Puisque tous les points fixes de f sont aussi solutions de $f(x) \succeq x$ (*i.e.* $\mathcal{F}_f \subseteq \mathcal{P}_f$), il vient que $x' = \text{Sup } \mathcal{P}_f$ est aussi le plus grand point fixe de f .

(4) Provient directement de $\mathcal{F}_f \subseteq \mathcal{P}_f$, $\text{Sup } \mathcal{F}_f = \text{Sup } \mathcal{P}_f$ et du fait que \mathcal{P}_f a une structure de treillis complet. ■

²Pour s'en convaincre, supposons qu'à l'instant initial l'entrée u_1 soit tirée cinq fois. La cinquième pièce ne quittera alors le four que lorsqu'une pièce sera extraite du buffer aval. Une pièce peut donc rester un temps infini dans le four si u_2 n'est jamais tirée.

³On aurait pu de façon symétrique étudier les bornes inférieures. Les résultats présentés peuvent d'ailleurs être transposés pour étudier les plus petits points fixes d'applications définies sur des dioïdes.

Proposition 1: Soit h une application isotone d'un dioïde complet \mathcal{D} dans lui-même, et $w \in \mathcal{D}$; on définit l'application f par $\forall x \in \mathcal{D}, f(x) = h(x) \wedge w$. Si l'algorithme itératif

$$\begin{aligned} y_0 &= w \\ y_{k+1} &= h(y_k) \wedge y_k \end{aligned} \quad (5)$$

converge en un nombre fini k_f d'itérations, alors y_{k_f} est le plus grand point fixe de f .

Proof: On montre que s'il existe k_f tel que $y_{k_f} = f(y_{k_f}) \wedge y_{k_f}$, alors y_{k_f} est le plus grand point post-fixe de f , ce qui, d'après (3), permet d'affirmer qu'il s'agit aussi du plus grand point fixe de f .

• Vérifions tout d'abord que y_{k_f} est point post-fixe de f ($y_{k_f} \in \mathcal{P}_f$). Notons que $y_{k+1} = f(y_k) \wedge y_k \preceq y_k$, et par conséquent l'itération définit une chaîne $y_0 = w \succeq y_1 \dots \succeq y_{k_f}$. D'où

$$y_{k_f} = h(y_{k_f}) \wedge y_{k_f} \preceq h(y_{k_f}) \wedge w = f(y_{k_f}).$$

• On montre maintenant que tout point post-fixe de f est inférieur à y_{k_f} , ce qui, puisque \mathcal{P}_f a une structure de treillis (cf. théorème 1), finit de montrer que $y_{k_f} = \text{Sup } \mathcal{P}_f$.

On montre par induction que si $x \in \mathcal{P}_f$ alors $\forall k \geq 0, y_k \succeq x$ (et donc *a fortiori* que $y_{k_f} \succeq x$).

1. L'inégalité est vérifiée au rang 0 ($x \preceq y_0$). En effet, si $x \in \mathcal{P}_f$, alors $x \preceq f(x)$, c.-à-d. $x \preceq h(x) \wedge w = y_0$.

2. Supposons que $y_k \succeq x$, on montre l'inégalité au rang $k+1$:

$$y_{k+1} = h(y_k) \wedge y_k \succeq h(x) \wedge x = x,$$

la dernière égalité découlant de $x \preceq f(x) = h(x) \wedge w \preceq h(x)$. ■

Remarque 1 On a noté dans la preuve de la proposition que l'itération définit une chaîne $y_0 \succeq y_1 \succeq \dots \succeq y_k \succeq \dots$. Pour l'algorithme, il est requis que cette chaîne soit de longueur finie.

Une condition suffisante à garantir cette propriété est que le dioïde \mathcal{D} satisfasse la *condition de chaîne descendante*⁴ [Davey and Priestley, 1990, def. 2.37].

Notons que si cette condition suffisante n'est pas vérifiée (ce qui est le cas lors des mises en œuvre pour la synthèse de correcteurs ci-après), constater la convergence (que ce soit par un calcul formel ou numérique) suffit à expliciter le plus grand point fixe. □

Remarque 2 Si h est semi-continue supérieurement, on remarque que

$$\begin{aligned} y_{k+1} &= h(y_k) \wedge y_k \\ &= h(h(y_{k-1}) \wedge y_{k-1}) \wedge h(y_{k-1}) \wedge y_{k-1} \\ &= h^2(y_{k-1}) \wedge h(y_{k-1}) \wedge y_{k-1} \\ &\vdots \\ &= \bigwedge_{i=0}^{j+1} h^i(y_{k-j}), \quad j \geq 0, \end{aligned}$$

ou encore, pour $j = k$,

$$y_{k+1} = \bigwedge_{i=0}^{k+1} h^i(y_0) = \bigwedge_{i=0}^{k+1} h^i(w).$$

⁴Où, au plus juste, un sous-ensemble contenant la chaîne.

Il apparaît alors que l'itération tend vers $h_*(w)$ ⁵, qui est le plus grand point fixe de f . Ce résultat est bien connu ; sa spécification aux dioïdes est donnée notamment dans [Baccelli et al., 1992, point 5 du théorème 4.72 p.187]. □

III. MODÉLISATION DES GET DANS $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$

Nous expliquons ici brièvement comment le comportement dynamique des GET peut être représenté par des éléments du dioïde $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ (se reporter par ex. à [Cohen et al., 1989], [Baccelli et al., 1992] pour des exposés complets sur ce sujet).

Les éléments de $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ sont des séries formelles en deux variables commutatives γ et δ . Pour la modélisation de GET, γ et δ servent à modéliser les décalages événementiels/temporels induits par les temporisations et les marquages initiaux des places. L'opérateur γ correspond à un décalage dans le comptage d'événements et l'opérateur δ à un décalage temporel.

A titre d'exemple, pour le GET décrit figure 1, les éléments de $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ associés aux trajectoires de tir des transitions t_1, t_2, u_1 et x vérifient les relations suivantes:

$$\begin{aligned} t_1 &= \gamma^4 x \oplus u_1 \\ x &= \delta^3 t_1 \oplus \gamma^3 t_2. \end{aligned}$$

Plus formellement, le dioïde $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ correspond au dioïde des séries formelles en γ et δ (à coefficients booléens et exposants dans \mathbb{Z}) quotienté par la relation d'équivalence suivante : $x \equiv y \iff (\gamma \oplus \delta^{-1})^* x = (\gamma \oplus \delta^{-1})^* y$. Ce quotientage permet de prendre en compte le caractère monotone des trajectoires de tir d'un GET⁶.

Pour la figure 1, en notant $U = (u_1 \ u_2)^t$ le vecteur des transitions d'entrée, $X = (t_1 \ x \ t_2)^t$ le vecteur d'état (transitions internes) et y la trajectoire de sortie, le comportement dynamique du GET est décrit par la représentation d'état suivante

$$\begin{cases} X &= \begin{pmatrix} \varepsilon & \gamma^4 & \varepsilon \\ \delta^3 & \varepsilon & \gamma^3 \\ \varepsilon & \delta^4 & \varepsilon \end{pmatrix} X \oplus \begin{pmatrix} e & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & e \end{pmatrix} U \\ y &= (\varepsilon \ \varepsilon \ e) X \end{cases}.$$

Plus généralement, le comportement dynamique des GET peut s'exprimer dans $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ par une représentation d'état de la forme

$$\begin{cases} X &= AX \oplus BU \\ Y &= CX \end{cases}.$$

La plus petite solution de ce modèle d'état, qui traduit la dynamique de fonctionnement au plus tôt du GET, fournit la relation entrée-sortie du système :

$$Y = CA^*BU,$$

où $H = CA^*B$ est appelée matrice de transfert du GET⁷.

Remarque 3 On dispose désormais d'outils informatiques (bibliothèque sous Scilab) permettant le calcul dans le dioïde $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ [L. Hardouin and M. Lhommeau, 2001]. □

⁵Avec $h_* = \bigwedge_{i \in \mathbb{N}} h^i$ et $h^0 = id$.

⁶Le $k+1$ ^{ème} tir d'une transition ne peut pas avoir lieu avant le k ^{ème} tir.

⁷Avec $A^* = \bigoplus_{i \in \mathbb{N}} A^i$ et $A^0 = id$.

IV. COMMANDE SOUS LA CONTRAINTE D'UNE ÉGALITÉ IMPLICITE

A. Motivations et modélisation du problème

Nous considérons par la suite les GET ayant la structure⁸ décrite par le schéma-bloc donné sur la figure 2. Le système est constitué de 4 sous-systèmes notés a, b, c et d dont on peut établir une relation de transfert entrée-sortie. Le symbole \oplus désigne différents points de synchronisation.

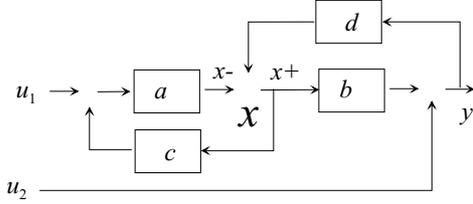


Fig. 2. Structure du système (hors contrôle)

Le problème de commande traité ici porte sur le comportement interne du système au point x . En pratique, x correspondra à une transition interne du GET traduisant la synchronisation de deux trajectoires, l'une provenant de la sortie du système a , l'autre de la sortie du système d . Du fait de la synchronisation, les jetons disponibles en sortie du système a ne franchissent pas nécessairement immédiatement la transition x . On notera donc x^- la trajectoire de disponibilité des jetons ayant quitté le système a mais en attente du franchissement de la transition x , et x^+ la trajectoire de franchissement effectif de la transition⁹.

L'objectif de commande est ici de garantir que tous les jetons ayant quitté le système a franchissent *immédiatement* la transition x . En d'autres termes, on souhaite éviter les blocages de jetons en amont de la transition x , ceci de façon à ce que le temps de séjour entre u_1 et l'entrée du système b (c.-à-d. x^+) n'excède pas le délai imposé par le système a . Ceci revient donc à imposer une contrainte temporelle sur le chemin allant de u_1 à x^+ .

D'un point de vue mathématique, garantir cette contrainte temporelle revient simplement à satisfaire l'égalité

$$x^- = x^+. \quad (6)$$

Notre démarche prend pour point de départ la détermination des relations de transfert entre le vecteur d'entrée $U = (u_1 \ u_2)^t$ et les trajectoires x^- et x^+ . Pour le système décrit figure 2, les relations de transfert (en fonction de a, b, c et d) sont les suivantes :

$$x^- = (a \oplus ac(ac \oplus db)^* a \quad ac(ac \oplus db)^* d) \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = H^- U \quad (7)$$

$$x^+ = ((ac \oplus db)^* a \quad (ac \oplus db)^* d) \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = H^+ U \quad (8)$$

En absence de loi de commande particulière, puisque $H^- \preceq H^+$, la contrainte (6) ne peut pas être respectée pour toute

⁸Comme nous le mentionnons en perspectives dans la conclusion, les techniques proposées peuvent s'étendre directement à d'autres structures.

⁹Il est à noter que les jetons ne peuvent pas franchir la transition x avant d'avoir quitté le système a , aussi l'inégalité $x^- \preceq x^+$ sera vérifiée quelle que soit l'entrée appliquée.

entrée U . On va donc chercher à satisfaire la contrainte (6) au moyen d'un correcteur P , c'est-à-dire un système dynamique linéaire, placé ici en amont du système ; la structure du système contrôlé est donnée figure 3. Ce correcteur aura pour effet d'éliminer les entrées susceptibles de transgresser la contrainte (6).

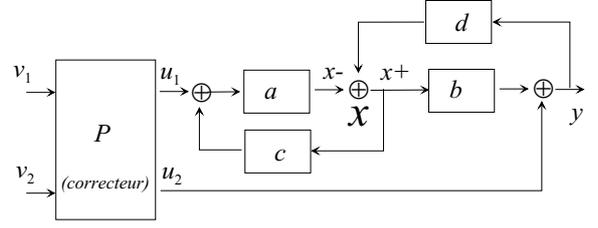


Fig. 3. Système sous-contrôle

En présence du précompensateur P , les relations de transfert caractérisant le comportement de x^- et x^+ deviennent

$$x^- = H^- PV \quad (9)$$

$$x^+ = H^+ PV, \quad (10)$$

où $V = (v_1 \ v_2)^t$ représente la consigne d'entrée.

Le problème de commande, avec contrainte temporelle, revient donc à calculer un correcteur P vérifiant la propriété

$$H^- P = H^+ P \quad (11)$$

de sorte que pour toute consigne V l'égalité (6) soit respectée.

B. Synthèse du correcteur

Pour la synthèse du correcteur, on considère plusieurs propriétés ou contraintes s'exprimant sous la forme d'égalités ou d'inégalités. Notons déjà que seule la plus grande solution à ces relations nous intéresse car celle-ci génère le correcteur permettant de déclencher les entrées le plus tardivement possible¹⁰.

Tout d'abord, la contrainte temporelle décrite précédemment nous amène à rechercher un correcteur, noté ici X en tant qu'indéterminée du problème, qui satisfasse, d'après la relation (11), l'équation implicite

$$H^- X = H^+ X. \quad (12)$$

En outre, on va traiter le problème dans l'objectif d'atteinte de modèle de référence G_{ref} (voir [Cottenceau et al., 2001]), ce qui conduit à l'ajout des contraintes suivantes:

$$H^- X \preceq G_{ref} \quad (13)$$

$$H^+ X \preceq G_{ref} \quad (14)$$

Notons que, puisque $H^- \preceq H^+$, si (14) est vérifié alors (13) l'est également.

Enfin, on souhaite obtenir un correcteur causal afin que l'élaboration de la trajectoire de commande U ne nécessite

¹⁰Dans le cas de système de production, cette gestion permet de limiter au maximum les en-cours et satisfait donc à l'objectif de gestion en juste-à-temps.

semble *a priori* envisageable. Nous projetons également d'utiliser une correction par feedback (plutôt que par précompensation) afin de bénéficier des "apports" d'une boucle fermée. Lors de ces prospections, il serait également intéressant de tenter de mettre à jour des conditions suffisantes à garantir la convergence de l'algorithme proposé (voir remarque 1). Enfin, au-delà des contraintes de temps critique, les techniques présentées (de commande sous une contrainte d'égalité implicite) devraient permettre d'appréhender d'autres problèmes inhérents aux SDED et non encore traités *via* l'algèbre des diïdes.

RÉFÉRENCES

- [Baccelli et al., 1992] Baccelli, F., Cohen, G., Olsder, G., and Quadrat, J. (1992). *Synchronization and Linearity : An Algebra for Discrete Event Systems*. Wiley and Sons.
- [Berthomieu and Diaz, 1991] Berthomieu, B. and Diaz, M. (1991). Modeling and verification of time dependent systems using time petri nets. *IEEE transactions on software engineering*, 17(3):259–273.
- [Blyth and Janowitz, 1972] Blyth, T. and Janowitz, M. (1972). *Residuation Theory*. Pergamon press.
- [Bonhomme et al., 2001] Bonhomme, P., Aygalinc, P., and Calvez, S. (2001). Robust control for time critical systems. In *Proceedings of the IEEE ETFA'01*, Antibes, France.
- [Cohen et al., 1989] Cohen, G., Moller, P., Quadrat, J., and Viot, M. (1989). Algebraic Tools for the Performance Evaluation of Discrete Event Systems. *IEEE Proceedings: Special issue on Discrete Event Systems*, 77(1):39–58.
- [Cottenceau et al., 1997] Cottenceau, B., Hardouin, L., and Boimond, J.-L. (1997). Dynamic Control of a Kanban System in Dioid Algebra. *Proc. of 5th IEEE Medit. Conf. on Cont. and Syst., Paphos, Cyprus*.
- [Cottenceau et al., 2001] Cottenceau, B., Hardouin, L., Boimond, J. L., and Ferrier, J. L. (2001). Model reference control for timed event graphs in Dioids. *Automatica*, 37:1451–1458.
- [Davey and Priestley, 1990] Davey, B. A. and Priestley, H. A. (1990). *Introduction to Lattices and Order*. Cambridge University Press, Cambridge.
- [L. Hardouin and M. Lhommeau, 2001] L. Hardouin and M. Lhommeau (2001). Bibliothèque de fonctions de manipulation de séries périodiques utilisable sous scilab. Disponible sur <http://www.istia.univ-angers.fr/~hardouin>.
- [Maia et al., 2004] Maia, C., Hardouin, L., Santos-Mendes, R., and Cottenceau, B. (to appear in 2004). Optimal closed-loop control of timed event graphs in dioid. *IEEE TAC*.
- [Menguy et al., 2000] Menguy, E., Boimond, J. L., Hardouin, L., and Ferrier, J. L. (2000). Just in time control of linear systems in dioid: cases of an update of reference input and uncontrollable input. *IEEE TAC*, 45(11):2155–2159.
- [Merlin, 1974] Merlin, P. M. (1974). *A study of the recoverability of computing systems*. PhD thesis, Department of Information and Computer Science, University of California, Irvine, CA.
- [Tarski, 1955] Tarski, A. (1955). A lattice theoretical fixed point theorem and its applications. *Pacific Journal of Maths.*, (5):285–309.

VII. ANNEXE

Nous rappelons ici que le calcul de l'exemple peut se faire au moyen de la boîte à outils MinMaxGD sous Scilab (disponible sur [L. Hardouin and M. Lhommeau, 2001]). Nous donnons ici le script Scilab conduisant au résultat.

```

a=series(eps,[0 3],e);
b=series(eps,[0 4],e);
c=series(eps,[4 0],e);
d=series(eps,[3 0],e);

s=stargd((a*c)+(d*b)); //s=(ac+db)*

Hpre(1,1)=a+(a*c*s*a); // H-
Hpre(1,2)=(a*c*s*d);

Hpost(1,1)=s*a; // H+
Hpost(1,2)=s*d;

```

```

Gref=Hpost // Gref=H+

// initialisation de l'algorithme (prop. 2)
yk=(Hpost\Gref)
continue=%T
while(continue) // itérations
    ykp=(Hpost\((Hpre*yk))^prcaus(yk)^yk
    if(ykp==yk) // teste la convergence
        continue=%F;
    end
    yk=ykp;
end

P=yk // P=ykf

(Hpre*P)==(Hpost*P) // vérification de (5): H+P==H-P

```