stability verification, scheduling, and synthesis of cyber-physical systems

Mohammad Al Khatib

Communauté UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Mathématiques appliquées**

Arrêté ministérial : 7 aout 2006

Présentée par

Mohammad Al Khatib

Thèse dirigée par **Antoine Girard**

et par **Thao Dang**

préparée au sein du **Laboratoire des signaux et systèmes**

et **Laboratoire Jean Kuntzmann**

dans l'école doctorale **Mathématiques, Sciences et technologies de l'information, Informatique**

Analyse de stabilité, ordonnancement, et synthèse des systèmes cyber-physiques

Thèse soutenue publiquement le **29 septembre 2017**,

devant le jury composé de :

Mme. Sophie Tarbouriech

Directeur de Recherche CNRS, Université de Toulouse, Président

M. Maurice Heemels

Professeur, Eindhoven University of Technology, Rapporteur

M. Laurentiu Hetel

Chargé de Recherche CNRS, HDR, Ecole Centrale de Lille , Rapporteur

M. Manuel Mazo Jr.

Professeur Assistant, Delft University of Technology, Examinateur

M. Antoine Girard

Directeur de Recherche CNRS, Université Paris-Saclay, Directeur de thèse

Mme. Thao Dang

Directeur de Recherche CNRS, Université Grenoble Alpes, Directeur de thèse

Abstract

This is a study conducted on cyber-physical systems on three main aspects: stability verification, scheduling, and parameter synthesis. Embedded control systems (ECS) acting under timing contracts are the considered class of cyber-physical systems in the thesis. ECS refers to integrations of a computing device with the physical system. As for timing contracts they are time constraints on the instants where some events happen such as sampling, actuation, and computation. These contracts are used to model issues that arise in modern embedded control systems: uncertain sampling to actuation delays, uncertain sampling periods, and interaction of several physical systems with shared computational resources (CPUs). Now given an ECS and a timing contract we reformulate the system into an impulsive one and verifies stability of the system, under all possible bounded uncertainties given by the contract, using safe convex approximation techniques and new generalized results for the problem on a class of systems modeled in the framework of difference inclusions. Second given a set of controllers implemented on a common computational platform (CPUs), each of which is subject to a timing contract, and best and worst case execution times on each CPU, we synthesize a dynamic scheduling policy, which guarantees that each timing contract is satisfied and that each of the shared CPUs are allocated to at most one embedded controller at any time. The approach is based on a timed game formulation that allows us to write the scheduling problem as a timed safety game. Then using the tool UPPAAL-TIGA, a solution to the safety game provides a suitable scheduling policy. In addition, we provide a novel necessary and sufficient condition for schedulability of the control tasks based on a simplified timed game automaton. Last, we solve a parameter synthesis problem which consists of synthesizing an under-approximation of the set of timing contracts that guarantee at the same time the schedulability and stability of the embedded controllers. The synthesis is based on a re-parameterization of the timing contract to make it monotonic, and then on a repeatedly sampling of the parameter space until reaching a predefined precision of approximation.

In the name of god. May peace be upon the soul of whom has left us too soon. In memory of my aunt Sikneh.

Acknowledgments

Thanks to Maurice Heemels and Laurentiu Hetel for accepting to review my thesis report in details. I also thank Sophie Tarbouriech and Manuel Mazo Jr. for accepting to be part of my thesis committee.

I would like to express my sincere gratitude to my advisors: Antoine and Thao. I appreciate your guidance, availability, and support. I am forever grateful for the precious knowledge and skills you've imparted to me!

I extend my gratitude to the members of the COMPACS project for the valuable discussions, presentations, and ideas we shared during the various meetings we had.

In addition, it was my pleasure to conduct my research in "Laboratoire Jean Kuntzmann" and "Laboratoire des signaux et système" and to visit the laboratory "VERIMAG". I am grateful towards the colleagues therein: permanents, non-permanents, post-docs, Ph.D. students, and graduate students for their great company.

And finally, thanks to my parents and numerous friends who endured this long process with me, always offering support and love.

Table of Contents

List of Tables

List of Figures

List of Symbols

$\mathbb{R}$	Set of real numbers.
$\mathbb{R}_0^+$	Set of non-negative reals.
$\mathbb{R}^+$	Set of positive reals.
$\mathbb{R}_0^-$	Set of non-positive reals.
$\mathbb{R}^-$	Set of negative reals.
$\mathbb{N}$	Set of non-negative integers.
$\mathbb{N}^+$	Set of positive integers.
$\mathbb{N}_I$	Set of integers $\mathbb{N} \cap I$, where $I \subseteq \mathbb{R}_0^+$.
$dom(f)$	Domain of a function f.
$ch(\mathcal{S})$	Convex hull of a set $\mathcal{S}$.
$\text{int}(\mathcal{S})$	The interior of a set $\mathcal{S}$.
H_i	The i-th row vector of a matrix H.
p_i	The i-th element of a vector p.
$2^{\mathbb{R}^n}$	Set of all subsets of $\mathbb{R}^n$.
$\mathcal{K}(\mathcal{S})$	Set of bounded subsets of a set $\mathcal{S}$.
$\mathcal{K}_0(\mathcal{S})$	Set of bounded subsets of a set $\mathcal{S}$ containing 0 in their interior.
$\mathcal{B}(\mathcal{S})$	Set of compact subsets of a set $\mathcal{S}$.
$\mathcal{B}_0(\mathcal{S})$	Set of compact subsets of a set $\mathcal{S}$ containing 0 in their interior.
$L_2^n[0,\infty)$	Set of the square integrable functions that map from $[0,\infty)$ to $\mathbb{R}^n$.
$\mathcal{P}$	A polytope or the intersection of a finite number of closed half-spaces, that is $\mathcal{P} = \{x \in \mathbb{R}^n : Hx \leq b\}$ where $H \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ and the vector of inequalities is interpreted component-wise.
$\lceil x \rceil$	Smallest integer not less that x.
$c \leq c'$	This is true for $c, c' \in \mathbb{R}^n$, if and only if $c_i \leq c'_i$, $i = 1, \ldots, n$.
$\|x\|$	Norm of x.
$\emptyset$	Empty set.
$\forall$	For all.
$\exists$	There exists.
$\in$	Set membership.

$\subset$ Subset.

$\cup$ Set-theoretic union.

$\cap$ Set-theoretic intersection.

$\vee$ Logical or.

$\wedge$ Logical and.

Résumé de la thèse

Chapitre 1. Motivation

Système cyber-physique: La théorie du contrôle repose principalement sur les boucles de retour d'état pour diriger un système réel vers un objectif souhaité. Les contrôleurs modernes sont aujourd'hui implémentés sur des plates-formes numériques. Cependant, la théorie repose généralement sur des modèles qui ne tiennent pas compte de l'interaction des logiciels avec le système physique. En d'autres termes, toute exécution d'un programme prend du temps et doit être prise en compte pour améliorer la fidélité du modèle, au lieu de considérer simplement que ce délai est nul. Ce temps de calcul est réellement affecté par la vitesse de la CPU, le partage des CPU entre les tâches de contrôle, les retards du réseau, les échantillonneurs, etc. Par conséquent, des outils pour concevoir, analyser et contrôler les systèmes cyber-physiques (CPS) où l'interaction entre les processus cybernétiques et physiques est étroite, sont nécessaires de toute urgence car les applications CPS deviennent omniprésentes dans les sociétés modernes (véhicules autonomes, bâtiments intelligents, robots, etc.) et auront une incidence sur la vie des citoyens sous tous leurs aspects (logement, transport, santé, industrie, assistance aux personnes âgées, etc.). Dans cette thèse, nous examinons le cas du contrôle embarqué pour étudier l'un des aspects de contrôle dans CPS.

Contrôle embarqué: un système de contrôle embarqué consiste en des intégrations d'un dispositif informatique avec le système physique. Classiquement, on suppose que le système informatique implémentant le contrôleur réalise l'abstraction d'une équation aux différences en temps discret d'une manière idéale. Pendant ce temps, le fait que les calculs prennent du temps et que les calculs parallèles soient limités par le nombre de processeurs disponibles est souvent ignoré. En outre, lorsque la boucle de contrôle est fermée sur un réseau, ce qui est le cas dans les systèmes de commande embarqués en réseau (NECS), les retards, qui ne sont pas idéalement égaux à zéro comme supposé traditionnellement, existent et sont loin d'être constants et en particulier dans les réseaux sans fil où la perte de paquets est pertinente. Le résultat de ces différences est le non-déterminisme temporel qui est une situation courante dans les CPS. Dans les chapitres suivants, le sujet sera d'étudier ce non-déterminisme qui affecte l'analyse, la conception et la mise en œuvre d'un système de contrôle intégré.

Chapitre 2. Formulation du problème

Nous considérons les systèmes échantillonnés sous la forme suivante:

$$\dot{z}(t) = Az(t) + Bu(t), \quad \forall t \in \mathbb{R}_0^+ \tag{1a}$$

$$u(t) = Kz(t_k^s), \qquad t_k^a < t \leq t_{k+1}^a \tag{1b}$$

où $z(t) \in \mathbb{R}^p$ est l'état du système et $u(t) \in \mathbb{R}^m$ est le variable de contrôle. De plus, on suppose que K est défini de telle sorte que la matrice $A + BK$ est Hurwitz.

Nous supposons aussi que les séquences d'échantillonnage et les instants d'actionnement $(t_k^s)_{k\in\mathbb{N}}$ et $(t_k^a)_{k\in\mathbb{N}}$ satisfont un *contrat temporel* $\theta(\underline{\tau}, \overline{\tau}, \underline{h}, \overline{h})$ donné par

$$\begin{aligned} 0 \ &\leq t_0^s, \\ t_k^s \leq t_k^a &\leq t_{k+1}^s, && \forall k \in \mathbb{N} \\ \tau_k = t_k^a - t_k^s &\in [\underline{\tau}, \overline{\tau}], && \forall k \in \mathbb{N} \\ h_k = t_{k+1}^s - t_k^s &\in [\underline{h}, \overline{h}], && \forall k \in \mathbb{N} \end{aligned} \tag{2}$$

avec $\mathcal{C}$ donné par:

$$\mathcal{C} = \left\{(\underline{\tau}, \overline{\tau}, \underline{h}, \overline{h}) \in \mathbb{R}_0^+ \times \mathbb{R}_0^+ \times \mathbb{R}^+ \times \mathbb{R}^+ : \underline{\tau} \leq \overline{\tau} \leq \overline{h}, \underline{h} \leq \overline{h}\right\}.$$

Dans cette étude, nous considérons la notion de stabilité suivante pour le système (1)-(2), qui garantit la convergence exponentielle de l'état à l'origine, i.e. $z = 0$, avec un taux prédéfini $\beta \in \mathbb{R}^+$:

Definition 1 (β'-stability)**.** *Etant donné $\beta \in \mathbb{R}^+$, le système (1)-(2) est β'-stable s'il exist $C \in \mathbb{R}^+$ et $\varepsilon' \in \mathbb{R}^+$ tel que:*

$$|z(t)| \leq Ce^{-(\beta+\varepsilon')(t-t_0^s)}|z_0|, \ \forall t \in \mathbb{R}^+. \tag{3}$$

Par conséquent, dans ce travail, nous considérons le problème suivant:

Problème 1 (La vérification de stabilité)**.** *Etant donné $\beta \in \mathbb{R}^+$, $A \in \mathbb{R}^{p\times p}$, $B \in \mathbb{R}^{p\times m}$, $K \in \mathbb{R}^{m\times p}$, $(\underline{\tau}, \overline{\tau}, \underline{h}, \overline{h}) \in \mathcal{C}$, vérifier que (1)-(2) est β'-stable.*

Maintenant, considérons une collection de N systèmes $\{\mathcal{S}_1, \ldots, \mathcal{S}_N\}$ de la forme (1) où chaque système $\mathcal{S}_i = (A_i, B_i, K_i)$ est soumise à un contrat temporel $\theta(\underline{\tau}^i, \overline{\tau}^i, \underline{h}^i, \overline{h}^i)$ de la forme (2), avec des paramètres $(\underline{\tau}^i, \overline{\tau}^i, \underline{h}^i, \overline{h}^i) \in \mathcal{C}$, $i \in \mathbb{N}_{[1,N]}$.

De plus, nous supposons que ces systèmes partagent J CPUs, comme le montre Figure 1, pour calculer la valeur de leurs entrées de contrôle fournies par (1b). En outre, le temps requis par CPU j pour calculer

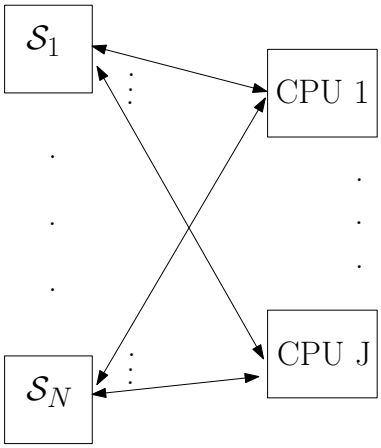


Figure 1: Schéma de N systèmes partageant J CPUs.

les entrées du systéme S_i est supposé appartenir à un intervalle connu $[\underline{c}_j^i, \overline{c}_j^i]$ avec $0 \leq \underline{c}_j^i \leq \overline{c}_j^i$, $i \in \mathbb{N}_{[1,N]}$, et $j \in \mathbb{N}_{[1,J]}$, où $\underline{c}_j^i$ and $\overline{c}_j^i$ désignent le meilleur et le pire cas d'exécution respectivement.

On indique par $\mathbb{N}(i,j)$ l'ensemble collectant les indices du cycle de contrôle où le système $\mathcal{S}_i$ accède au CPU j, avec $\bigcup_{j\in\mathbb{N}_{[1,J]}} \mathbb{N}(i,j) = \mathbb{N}$ pour tout $i \in \mathbb{N}_{[1,N]}$. En notant les instants où le calcul de l'entrée pour un système i commence et se termine par $t_k^{b_i}$ et $t_k^{e_i}$ respectivement, les instants des événements pour le système (1)-(2) satisfont:

$$
\begin{aligned}
& 0 \leq t_0^{s_i} \\
& t_k^{s_i} \leq t_k^{b_i} \leq t_k^{e_i} \leq t_k^{a_i} \leq t_{k+1}^{s_i}, && \forall k \in \mathbb{N} \\
& c_k^i = t_k^{e_i} - t_k^{b_i} \in [\underline{c}_j^i, \overline{c}_j^i], && \forall k \in \mathbb{N}(i,j), \forall j \in \mathbb{N} \\
& \tau_k^i = t_k^{a_i} - t_k^{s_i} \in [\underline{\tau}^i, \overline{\tau}^i], && \forall k \in \mathbb{N} \\
& h_k^i = t_{k+1}^{s_i} - t_k^{s_i} \in [\underline{h}^i, \overline{h}^i], && \forall k \in \mathbb{N}.
\end{aligned}
\tag{4}
$$

L'ordonnanceur contrôle les instants d'échantillonnage et d'actionnement $(t_k^{s_i})_{k\in\mathbb{N}}$, $(t_k^{a_i})_{k\in\mathbb{N}}$ et les instants $(t_k^{b_i})_{k\in\mathbb{N}}$ lorsque le calcul commence. De plus, l'ordonnanceur attribue un CPU pour calculer l'entrée de contrôle pour chaque système $\mathcal{S}_i$ à chaque cycle de contrôle $k \in \mathbb{N}$. Cependant, le temps d'exécution $(c_k^i)_{k\in\mathbb{N}}$, et donc les instants lorsque le calcul finit $(t_k^{e_i})_{k\in\mathbb{N}}$, est déterminé par l'environnement et est donc incontrôlable du point de vue de l'ordonnanceur. Étant donné qu'un ensemble de tâches $\mathcal{T}$, une tâche T_i, et un ensemble de contrats temporels Θ sont caractérisés par

$$
T_i = \left((\underline{c}_1^i, \overline{c}_1^i), \ldots, (\underline{c}_J^i, \overline{c}_J^i)\right), i \in \mathbb{N}_{[1,N]} \tag{5a}
$$

$$
\mathcal{T} = \{T_1, \ldots, T_N\}, \tag{5b}
$$

et

$$\Theta = \{\theta(\underline{\tau}^1, \overline{\tau}^1, \underline{h}^1, \overline{h}^1), \ldots, \theta(\underline{\tau}^N, \overline{\tau}^N, \underline{h}^N, \overline{h}^N)\}, \tag{6}$$

nous définissons le problème de l'ordonnancement pour le moment de maniére informelle, comme:

Problème 2 (L'ordonnancement)**.** *Etant donné d'un ensemble de tâches de contrôle $\mathcal{T}$ comme dans (5), et des contrats temporels $\Theta = \{\theta(\underline{\tau}^1, \overline{\tau}^1, \underline{h}^1, \overline{h}^1), \ldots, \theta(\underline{\tau}^N, \overline{\tau}^N, \underline{h}^N, \overline{h}^N)\}$, vérifier s'il existe ou non une politique d'ordonnancement avec des séquences d'événements temporels satisfaisant (4) et garantissant qu'au plus un contrôleur accède à chacun des CPU à chaque instant.*

Le troisième problème que nous traitons dans la thèse est le suivant:

Problème 3 (La synthèse des contrats temporels)**.** *Etant donné une collection de systèmes $\{\mathcal{S}_1, \ldots, \mathcal{S}_N\}$, où $\mathcal{S}_i = (A_i, B_i, K_i)$ avec $A_i \in \mathbb{R}^{n_i \times n_i}$, $B_i \in \mathbb{R}^{n_i \times m_i}$, et $K_i \in \mathbb{R}^{m_i \times n_i}$, $i \in \mathbb{N}_{[1,N]}$, J CPUs, un ensemble de tâches de contrôle $\mathcal{T}$ par (5b), un ensemble $\{\beta_i\}_{i \in \mathbb{N}_{[1,N]}} \subset \mathbb{R}^+$, et des ensembles $\mathcal{D}_i$, $i \in \mathbb{N}_{[1,N]}$, synthétiser un ensemble $\mathcal{P}^* \subseteq (\mathcal{C}^N) \cap (\mathcal{D}_1 \times \cdots \times \mathcal{D}_N)$ de sorte que pour tous $(\underline{\tau}^1, \overline{\tau}^1, \underline{h}^1, \overline{h}^1, \ldots, \underline{\tau}^N, \overline{\tau}^N, \underline{h}^N, \overline{h}^N) \in \mathcal{P}^*$:*

1. *Le système $\mathcal{S}_i = (A_i, B_i, K_i)$ est β_i'-stable sous le contrat temporel $\theta(\underline{\tau}_i, \overline{\tau}_i, \underline{h}_i, \overline{h}_i)$, pour tout $i \in \mathbb{N}_{[1,N]}$.*
2. *L'ensemble des tâches de contrôle $\mathcal{T}$ est ordonnancable sous les contrats temporels Θ donnés par (6).*

Chapitre 3. Vérification de la stabilité: une approche basée sur les inclusions aux différences et l'analyse d'atteignabilité

Nous présentons une classe d'inclusions aux différences par:

$$\xi_{k+1} \in \Phi(\{\xi_k\}),\ k \in \mathbb{N}. \tag{7}$$

Le système (1) sous le contrat temporel (2) est reformulé comme l'inclusion aux différences (7) avec Φ défini comme dans (8).

$$\Phi(\mathcal{S}) = \bigcup_{\tau \in [\underline{\tau}, \overline{\tau}]} \bigcup_{w \in [\max(0, \underline{h} - \tau), \overline{h} - \tau]} e^{(w+\tau)\beta} e^{wA_c} A_a e^{\tau A_c} A_s \mathcal{S}. \tag{8}$$

La stabilité (7) est donnée dans le sens suivant:

Definition 2 (GES). *Le système (7) est globalement stable de façon exponentielle (GES) s'il existe* $(C, \varepsilon) \in \mathbb{R}^+ \times (0,1)$ *de sorte que pour toutes les trajectoires* $(\xi_k)_{k\in\mathbb{N}}$ *de (7), on a*

$$|\xi_k| \leq C\varepsilon^k |\xi_0|, \ \forall k \in \mathbb{N}. \tag{9}$$

L'équivalence entre la stabilité de (1)-(2) et celui de (7) est ensuite établie en utilisant le résultat suivant:

Proposition 1. *Etant donné que* $\beta \in \mathbb{R}^+$*, le système (1)-(2) est* β'*-stable si et seulement si le système (7) est GES avec* Φ *donné par (8).*

Maintenant, nous devons seulement dériver des conditions de stabilité sur Φ comme fait dans le résultat suivant:

Theorem 1. *Supposons que* $\mathcal{S} \in \mathcal{B}_0(\mathbb{R}^n)$*, avec* $\mathcal{B}_0(\mathbb{R}^n)$ *l'ensemble de sous-ensembles de* $\mathbb{R}^n$ *compacts contenant l'origine, et* Φ *donné par (8), les suivants sont equivalents:*

(a) Le système (7) est GES;

(b) Il existe $(k, j, \rho) \in \mathbb{N}^+ \times \mathbb{N}_{[0,k-1]} \times (0,1)$ *tel que* $\Phi^k(\mathcal{S}) \subseteq \rho\Phi^j(\mathcal{S})$*;*

(c) Il existe $(k, \rho) \in \mathbb{N}^+ \times (0,1)$ *tel que* $\Phi^k(\mathcal{S}) \subseteq \rho \bigcup_{j=0}^{k-1} \Phi^j(\mathcal{S})$.

Le Théorème 1 montre l'existence d'un ensemble contractant généralement non convexe $\mathcal{S}'$ pour le système (7) chaque fois que ce dernier est GES.

Maintenant, nous établissons des conditions de stabilité basées sur une fonction dont les images sont des ensembles convexes. Considérons la fonction $\hat{\Phi} : 2^{\mathbb{R}^n} \to 2^{\mathbb{R}^n}$ donné par

$$\forall \mathcal{S} \subseteq \mathbb{R}^n, \ \hat{\Phi}(\mathcal{S}) = \mathrm{ch}(\Phi(\mathcal{S})).$$

Nous définissons également le système dynamique associé à $\hat{\Phi}$:

$$\xi_{k+1} \in \hat{\Phi}(\{\xi_k\}), \ k \in \mathbb{N}. \tag{10}$$

Nous pouvons maintenant prouver le résultat suivant, qui montre l'équivalence entre la stabilité des systèmes (7) et (10) et donne une caractérisation en termes de $\hat{\Phi}$.

Theorem 2. *Considérer que* $\mathcal{S} \in \mathcal{B}_0(\mathbb{R}^n)$*, et* Φ *est donné par (8), les suivantes sont équivalentes:*

(a) Le système (7) est GES;

(b) Il existe $(k, j, \rho) \in \mathbb{N}^+ \times \mathbb{N}_{[0,k-1]} \times (0,1)$ *tel que* $\hat{\Phi}^k(\mathcal{S}) \subseteq \rho\hat{\Phi}^j(\mathcal{S})$*;*

(c) Il existe $(k, \rho) \in \mathbb{N}^+ \times (0,1)$ *such that* $\hat{\Phi}^k(\mathcal{S}) \subseteq \rho ch\big(\bigcup_{j=0}^{k-1} \hat{\Phi}^j(\mathcal{S})\big)$*;*

(d) Le système (10) est GES.

Les fonctions Φ et $\hat{\Phi}$ impliquées dans les Théorèmes 1 et 2 sont en pratique difficilement calculable. Dans ce cas, nous pouvons utiliser une sur-approximation $\overline{\Phi} : 2^{\mathbb{R}^n} \to 2^{\mathbb{R}^n}$, qui est plus facile à calculer et satisfait l'hypothèse suivante:

Assumption 1. *Pour tous* $\mathcal{S} \subseteq \mathbb{R}^n$*, les assertions suivantes sont vraies:*

(i) $\Phi(\mathcal{S}) \subseteq \overline{\Phi}(\mathcal{S})$*;*

(ii) si $\mathcal{S}$ *est borné alors* $\overline{\Phi}(\mathcal{S})$ *est borné.*

Corollary 1. *Supposons que* Φ *est donné par (8), sous l'Assumption 1, s'il existe un ensemble* $\mathcal{S} \in \mathcal{B}_0(\mathbb{R}^n)$ *et* $(k, i, \rho) \in \mathbb{N}^+ \times \mathbb{N}_{[0,k-1]} \times (0,1)$ *tel que* $\overline{\Phi}^k(\mathcal{S}) \subseteq \rho\overline{\Phi}^i(\mathcal{S})$*, alors le système (7) est GES.*

Basé sur le corollaire 1, la méthode globale est ensuite résumée par un algorithme de vérification de stabilité qui, pour le cas de (1)-(2), calcule un ensemble initial $\mathcal{S}_0$ invariant à plusieurs sous-systèmes de (1)-(2) et ensuite dans la boucle principale propage $\mathcal{S}_0$ en utilisant $\overline{\Phi}$. Ensuite, si la condition de stabilité suffisante est vérifiée, la stabilité est garantie, sinon si un nombre maximum d'itérations est atteint, l'algorithme ne parvient pas à prouver la stabilité. À la fin du chapitre, des comparaisons de notre approche de vérification de stabilité avec celles de la littérature sont menées où notre algorithme produit des résultats prometteurs et competitifs. Deux sections, qui ne sont pas directement liées à la portée principale de la thèse, sont ajoutées puis étendent le travail présenté dans ce chapitre pour traiter deux autres problèmes, pour le cas spécial de (1)-(2) avec $\overline{\tau} = \underline{\tau} = 0$, qui sont le problème de contrôle "self-triggered" et le problème de vérification de la stabilité dans des contrats temporels stochastiques.

Chapitre 4. Ordonnancement des contrôleurs embarqué sous des contrats temporels

Pour résoudre le Problème 2, nous associons à chaque tâche de contrôle et contrat temporel un automate de jeu temporisé TGA_i, $i = 1, \ldots, N$, et nous définissons un réseau des automates de jeu temporisé TGA décrivant l'évolution parallèle de $\mathsf{TGA}_1, \ldots, \mathsf{TGA}_N$. Chaque automate temporisé est donné comme dans la Figure 2. Maintenant, notons les séquences $(t_k^{s_i})$, $(t_k^{a_i})$, $(t_k^{b_i})$, et $(t_k^{e_i})$ données par les instants des transitions discrètes marquées par les actions $sample^i$, $actuate^i$, $begin^i$ et end^i, respectivement. Il est facile de voir

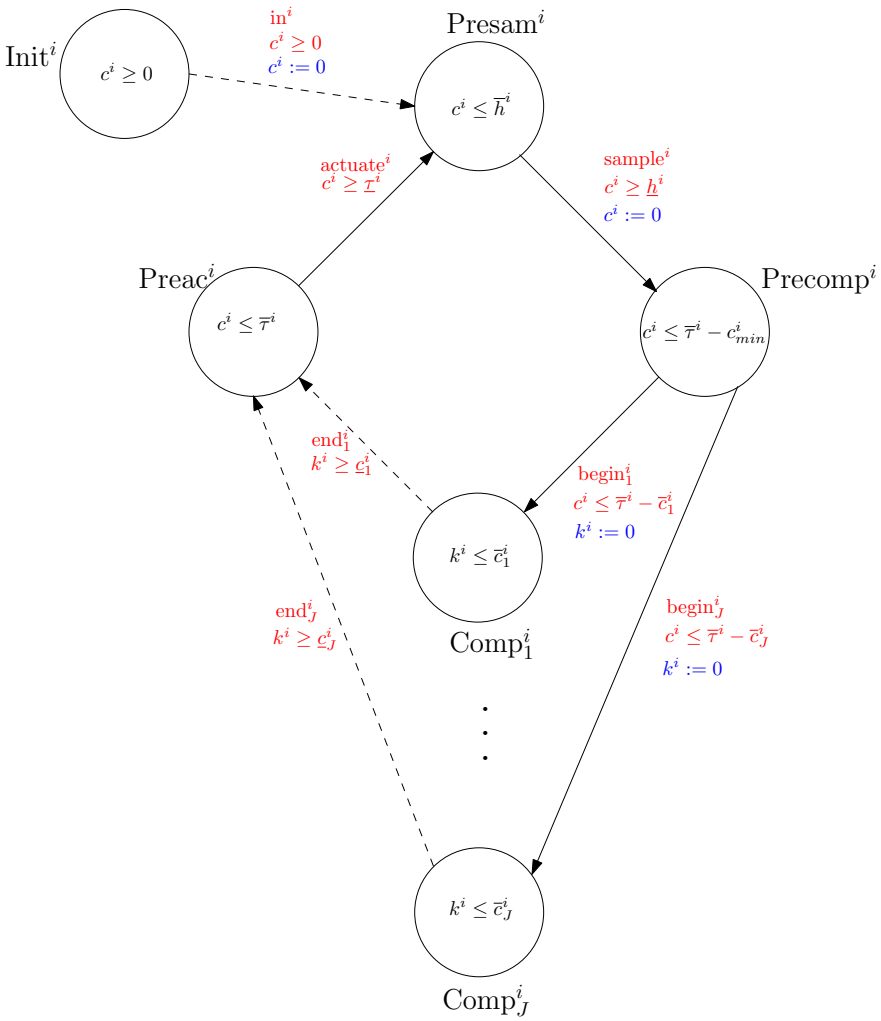


Figure 2: TGA_i, où les flèches plaines et pointillées correspondent respectivement à des actions contrôlables et incontrôlables.

que ces séquences satisfont les contraintes imposées par (4). Inversement, on peut vérifier que toutes les séquences satisfaisant (4) peuvent être générées par des exécutions de TGA_i.

De plus, rappelons que les actions contrôlables sont $sample^i$, $actuate^i$, $begin^i$, ce qui signifie que l'ordonnanceur détermine les instants lorsque l'échantillonnage et l'actionnement se produisent et lorsque le calcul commence. Cependant, end^i est incontrôlable, ce qui signifie que le temps d'exécution, et donc l'instant auquel le calcul finit est déterminé par l'environnement. En utilisant TGA pour reformuler le problème d'ordonnancement dans un jeu avec des spécifications de sûreté (TGA,$\overline{L}_u$), nous utilisons l'outil UPPAAL Tiga pour synthétiser une stratégie pour déclencher des actions contrôlables (échantillonnage, actionnement et début des calculs dans la CPU) afin que, pour toutes les actions possibles incontrôlables (fin de calcul dans la CPU), un

ensemble d'emplacements indésirables $\overline{L}_u$ soit évité, avec

$$\begin{aligned}\overline{L}_u = \{l \in \overline{L} : \exists (m,n,j) \in \mathbb{N}^2_{[1,N]} \times \mathbb{N}_{[1,J]}, m \neq n, \\ (l^m = Comp^m_j) \wedge (l^n = Comp^n_j)\}.\end{aligned} \tag{11}$$

À la fin du chapitre, des applications d'ordonnancement sur un CPU ou deux CPU sont utilisées pour évaluer l'approche.

Chapitre 5. Synthèse des paramètres

Dans ce chapitre, nous synthétisons un ensemble de contrats temporels qui garantissent en même temps l'ordonnancement et la stabilité des contrôleurs embarqués. Le flux de travail de l'approche est donné par la Figure 3.

Nous découpons le problème en deux parties. Tout d'abord, nous synthétisons des ensembles des contrats temporels, définis par $\mathcal{P}^i_{st}$, garantissant la stabilité de chaque système indexé par $i = 1, \ldots, N$. Dans la deuxième étape, nous synthétisons un ensemble $\mathcal{P}_{sched}$ donnant une garantie sur l'ordonnancement. Et, par conséquent, en prenant l'intersection de ce dernier avec l'ensemble augmenté $\mathcal{P}_{st} = \mathcal{P}^1_{st} \times \cdots \times \mathcal{P}^N_{st}$, nous résolvons le problème. Les approches suivies pour la première et la deuxième étape ont le même concept qui est d'abord de re-paramétrer les contrats temporels. Par exemple, le re-paramétrisation du contrat temporel $\theta(\underline{\tau}, \overline{\tau}, \underline{h}, \overline{h})$ est donné par η avec:

- $\alpha = (\underline{\tau}, \overline{\tau}, \underline{h}, \overline{h}) \in \mathcal{D} = [\tau_{min}, \tau_{max}] \times [\tau_{min}, \tau_{max}] \times [h_{min}, h_{max}] \times [h_{min}, h_{max}]$.
- $\eta = (\eta_1, \eta_2, \eta_3, \eta_4) \in \mathcal{D}'$ où $\mathcal{D}' = [\tau_{min}, \tau_{max}] \times [-\tau_{max}, -\tau_{min}] \times [h_{min}, h_{max}] \times [-h_{max}, -h_{min}]$.
- $f : \mathcal{D}' \to \mathcal{D}$ tel que $f(\eta) = \alpha = (\underline{\tau}, \overline{\tau}, \underline{h}, \overline{h})$ où

$$\underline{\tau} = \eta_1,\ \overline{\tau} = \min(-\eta_2, -\eta_4),\ \underline{h} = \eta_3,\ \overline{h} = -\eta_4.$$

Pour $\alpha \in \mathcal{C} \cap \mathcal{D}$ nous désignons la propriété:

$$\mathsf{Stab}(\alpha) \equiv \text{(1-2) est } \beta'\text{-stable avec le paramètre } \alpha.$$

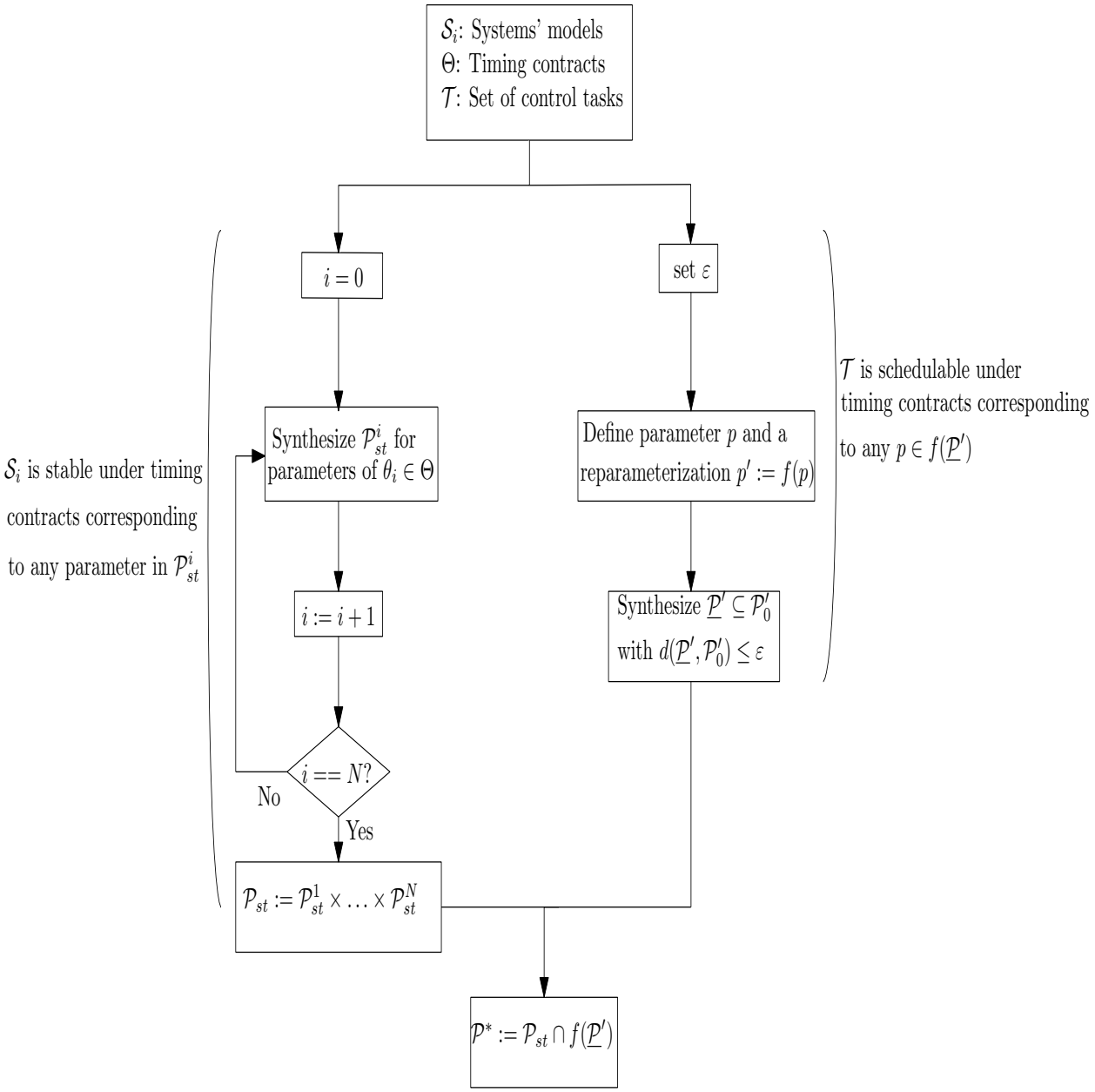


Figure 3: Flux de travail de l'approche proposée .

La synthèse d'un ensemble de contrats temporels garantissant la stabilité équivaut au calcul (d'un sous-ensemble) de $\mathcal{C}_o$ donné par

$$\mathcal{C}_o = \{\alpha \in \mathcal{C} \cap \mathcal{D} : \mathsf{Stab}(\alpha)\}.$$

Nous définissons un ensemble de contrainte pour le paramètre η:

$$\mathcal{C}' = \left\{ \eta \in \mathbb{R}_0^+ \times \mathbb{R}_0^- \times \mathbb{R}^+ \times \mathbb{R}^- : \begin{array}{l} \eta_1 \leq \min(-\eta_2, -\eta_4) \\ \eta_3 \leq -\eta_4 \end{array} \right\}. \tag{12}$$

Le résultat suivant tient:

Lemma 1. *Nous définissons $\mathcal{C}'_o$ par*

$$\mathcal{C}'_o = \{\eta \in \mathcal{C}' \cap \mathcal{D}' : \mathit{Stab}(f(\eta))\}.$$

Alors, $f(\mathcal{C}' \cap \mathcal{D}') = \mathcal{C} \cap \mathcal{D}$ et $f(\mathcal{C}'_o) = \mathcal{C}_o$.

Nous définissons encore l'ensemble suivant

$$\mathcal{E}'_o = \{\eta \in \mathcal{D}' : (\eta \notin \mathcal{C}') \vee ((\eta \in \mathcal{C}') \wedge \mathsf{Stab}(f(\eta)))\}.$$

On peut facilement vérifier la relation suivante:

$$\mathcal{C}'_o = \mathcal{C}' \cap \mathcal{E}'_o. \tag{13}$$

Par conséquent, par l'égalité précédente et le Lemme 1, on peut synthétiser les ensemble $\mathcal{P}^i_{st}$, où $i \in \mathbb{N}_{[1,N]}$, en calculant (un sous-ensemble de) l'ensemble $\mathcal{E}'_o$. De plus, $\mathcal{E}'_o$ satisfait la propriété de monotonie suivante:

Proposition 2. *Pour tout $\eta, \eta' \in \mathcal{D}'$, les implications suivantes tiennent:*

$$((\eta \leq \eta') \wedge (\eta \in \mathcal{E}'_o)) \implies \eta' \in \mathcal{E}'_o.$$

$$((\eta \leq \eta') \wedge (\eta' \notin \mathcal{E}'_o)) \implies \eta \notin \mathcal{E}'_o.$$

La propriété précédente est essentielle pour calculer un sous-ensemble de $\mathcal{E}'_o$ car il nous permet d'énoncer le théorème suivant:

Theorem 3. *On suppose que $\underline{\eta}^1, \ldots, \underline{\eta}^{M_1} \in \mathcal{E}'_o$, et $\overline{\eta}^1, \ldots, \overline{\eta}^{M_2} \in \mathcal{D}' \setminus \mathcal{E}'_o$. Nous définissons également*

$$\underline{\mathcal{E}}' = \bigcup_{j=1}^{M_1} \{\eta \in \mathcal{D}' : \underline{\eta}^j \leq \eta\},\ \overline{\mathcal{E}}' = D' \setminus \bigcup_{j=1}^{M_2} \{\eta \in \mathcal{D}' : \eta \leq \overline{\eta}^j\}.$$

Alors, $\underline{\mathcal{E}}' \subseteq \mathcal{E}'_o \subseteq \overline{\mathcal{E}}'$.

De plus, $\mathcal{P}^i_{st} = f\left(\mathcal{C}' \cap \underline{\mathcal{E}}'\right)$.

Comme indiqué, la re-paramétrisation permet la stabilité de (1) - (2) de devenir monotone par rapport aux nouveaux paramètres. Ensuite, en utilisant une recherche guidée pour échantillonner l'espace des paramètres,

le Théoreme 3 nous permet de synthétiser les contrats temporels en surveillant une sous-approximation de l'espace des paramètres et en échantillonnant à plusieurs reprises l'ensemble de paramètres inexploré jusqu'à ce que la distance entre les sur-et les sous-approximations est inférieure à un seuil donné. Des résultats similaires sur la re-paramétrage, la monotonie, et l'échantillonnage guidé sont établis pour synthétiser l'ensemble $\mathcal{P}_{sched}$. L'approche globale est évaluée finalement pour deux systèmes bidimensionnels programmés sur un uniprocesseur.

Conclusion

Dans cette thèse, nous traitons trois problèmes différents qui se posent dans les CPS et plus particulièrement pour les systèmes de contrôle embarqué. Pour le problème de la vérification de la stabilité, nous avons proposé dans le cadre de modélisation bien connu des inclusions aux differences une nouvelle approche basée sur l'analyse de l'atteignabilité. Les résultats de cette approche nous ont permis d'étendre notre travail à d'autres problèmes qui sont le problème de contrôle "self-triggered" et la vérification de la stabilité dans des contrats temporels stochastiques. Pour le problème d'ordonnancement, une nouvelle approche basée sur sur une reformulation qui nous permet d'écrire le problème d'ordonnancement comme un jeu temporele avec spécification de sureté, ce qui nous permet de synthétiser une stratégie d'ordonnancement garantissant que chaque processeur soit au maximum utilisé par une boucle de contrôle à la fois et que tous les contrats temporels soient satisfaits. En outre, nous fournissons une nouvelle condition nécessaire et suffisante pour l'ordonnancement des tâches de contrôle en fonction d'un jeu temporisé simplifié. Pour le dernier problème, qui est le problème de synthèse des paramètres, nous avons suggéré un nouveau paramétrage des contrats temporels qui nous permet d'explorer l'espace des paramètres à l'aide des ensembles monotones. Les contrats temporels synthétisés garantissent alors l'ordonnancement des tâches et la stabilité de chaque boucle de contrôle. Tout au long de la thèse, les résultats sont illustrés par des simulations numériques et des comparaisons avec les résultats existants dans la littérature, où notre algorithme de vérification de stabilité et notre stratégie de contrôle "self-triggered" ont des résultats prometteurs et compétitifs.

Nous décrivons ci-dessous des directions prometteuses pour développer les résultats présentés pour les trois problèmes principaux discutés dans la thèse.

- *Vérification de la stabilité:* L'approche de vérification de la stabilité présentée au Chapitre 3 ne fournit que des conditions suffisantes, pour verifier pratiquement la stabilité d'un système. Néanmoins, si la sur-approximation $\overline{\Phi}$ satisfait des hypothèses supplémentaires, la nécessité pourrait également être établie. Ces hypothèses sont liées aux schémas de sur-approximation utilisés pour calculer $\overline{\Phi}$ si ces

schemas ne souffrent pas de l'effet d'emballage ou l'effet d'une approximation approfondie de l'erreur, la première condition nécessaire et suffisante pratique pourrait être établie dans la littérature. Une autre direction pour améliorer les résultats est d'améliorer la stratégie "self-triggered". La stratégie à suivre n'est pas optimale dans le sens où il peut exister d'autres stratégies qui conduisent à moins d'échantillonnage dans une fenêtre de temps donnée. Un autre problème sur lequel souffrent les algorithmes existants dans la littérature est le passage à l'échelle computationnel, la plupart des résultats étant évalués sur des systèmes de dimension modeste. Bien que nous ne disposions pas d'idées précise pour gérer cela, nous pouvons proposer à long terme de développer des calculs d'atteignabilité qui reposent sur une représentation d'ensemble efficace pour calculer l'ensemble Φ donné par (8).

- *Ordonnancement:* Au Chapitre 4, nous synthétisons les stratégies d'ordonnancement à l'aide de l'outil UPPAAL Tiga. Cependant, la stratégie synthétisée est donnée simplement par le texte et ne peut donc pas être importée facilement à Matlab afin de simuler les systèmes supervisés. En conséquence, certains efforts de programmation devraient être faits pour surmonter cet obstacle, comme la traduction des modèles UPPAAL utilisant stateflow en Matlab comme fait dans [PLMS11]. Une autre direction consiste à prendre des actions contrôlables optimales dans le sens où la boucle de contrôle doit être fermée le plus tôt possible pour chaque tâche. Les outils qui utilisent des automates "priced" comme UPPAAL Cora [BLR05] pourraient être utilisés. À long terme, les algorithmes pour l'ordonnancement préemptive dans le cadre des contrats temporels doivent être gérés contrairement aux politiques synthétisées au Chapitre 4. Dans ce travail, les automates ne sont pas censés être capables de résoudre ce problème car une fois qu'une tâche est préemptée, le temps de la tâche devrait arrêtez ce qui ne pouvait être évidemment traduit avec des automates temporisés.

- *Synthèses des paramètres:* Le choix de l'échantillon suivant dans l'algorithme de synthèse du contrat temporels n'est pas tout à fait évident et nécessite l'utilisation de méthodes existantes comme celle dans [LLGCM10]. La complexité de ce dernier augmente de façon exponentielle par rapport au nombre de systèmes et, par conséquent, à la dimension de l'espace des paramètres, qui nécessite des travaux futurs avec d'autres méthodes pour suivre la distance entre la sous-et la sur-approximation dans l'espace des paramètres, et choisir à plusieurs reprises l'échantillon de l'ensemble inexploré.

Chapter 1

Introduction

1.1 Cyber-physical systems

Control theory relies mainly on feedback control loops to drive a physical dynamical system toward a desired goal. Modern controllers are nowadays implemented on digital platforms. However, the theory generally relies on models that do not take the interaction of software with the physical system into consideration. Indeed, any execution of a program takes time which needs to be taken into consideration to ensure correctness of the model, instead of simply considering such time to be null. This computational time is possibly affected by the CPU's speed, sharing of CPUs between control tasks, network delays, samplers, etc. Consequently, tools to design, analyze, and control *cyber-physical systems (CPSs)*, or systems where the interaction between the cyber and physical processes is tight, are urgently needed as CPS applications have become ubiquitous in modern societies (autonomous vehicles, smart buildings, robots, etc.) and will practically impact the life of citizens in all their aspects (housing, transportation, health, industry, assistance to the elderly, etc.). In this thesis we examine the case of embedded control to study one of the control aspects in CPSs.

Embedded control: An embedded control system consists of integrations of a computing device with the physical system. Classically, the computing system implementing the controller is assumed to be the abstraction of a discrete-time difference equation in an ideal way. Meanwhile, the fact that computations take time and parallel computations are limited by number of available processors is often disregarded. Also, when the control loop is closed over a network, which is the case in networked embedded control systems (NECS), delays, which are not ideally equal to zero as assumed traditionally, exist and are far from being constant, in particular in wireless networks where packet loss is relevant. The consequence of these differences is temporal nondeterminism which is a common situation in CPSs. In the next chapters, the topic will be to study this non-determinism which affects the analysis, design, and implementation of an embedded control system.

1.2 Contributions

The main contributions of this thesis are presented in three different chapters. Chapter 3 studies the stability of an embedded control system under timing contracts. An approach based on difference inclusions and reachability analysis is proposed where a novel stability verification algorithm is written at the end. The notion of a reachable set used throughout the chapter is seen to provide an intuitive tool that leads to conditions on the stability of a system, where in comparison to approaches that pre-define a Lyapunov function for each particular system, the reachability based approach is seen to shape this function by constructing the latters level sets using over-approximations of the reachable set around the origin. In Chapter 4, we address the scheduling problem that arises when multiple control loops need to be run on several single processors, assuming that for each control task, its best and the worst case execution times, on each processor, are given and that the sensing and actuation operations also happen in a known time window. A solution is achieved by mapping the problem to timed game automata in which the execution times of control tasks are modeled using uncontrollable actions (within bounds). Given the controllable actions (corresponding to sampling time, actuation time, and execution start time) a feasible dynamic schedule is synthesized (if it exists) for all possible outcomes of the uncontrollable actions. The contribution in Chapter 5 is a synthesis strategy to generate timing contracts for the control loops in such a way that the control computations are schedulable on a given number of processors and the control loops achieve stability. Also, two additional contributions are added to Chapter 3 as extensions to the work therein. The first extend the stability verification approach to the case of systems with uniformly distributed inter-sampling time and the other proposes a self-triggered implementation of the embedded controller based on the main results of the same chapter. We summarize below the results illustrated in each chapter.

1.2.1 Chapter 3. Stability verification: an approach based on difference inclusions and reachability analysis

We introduce a class of difference inclusions (3.15) by:

$$\xi_{k+1} \in \Phi(\{\xi_k\}),\ k \in \mathbb{N} \tag{3.15}$$

In parallel, we consider a sampled-data system given by

$$\dot{z}(t) = Az(t) + Bu(t), \quad \forall t \in \mathbb{R}_0^+ \tag{2.1a}$$

$$u(t) = Kz(t_k^s), \qquad t_k^a < t \leq t_{k+1}^a \tag{2.1b}$$

where $z(t) \in \mathbb{R}^p$ is the state of the system and $u(t) \in \mathbb{R}^m$ is the control input. In addition, it is assumed that K is designed such that the matrix $A + BK$ is Hurwitz.

We also assume that the sequences of sampling and actuation instants $(t_k^s)_{k\in\mathbb{N}}$ and $(t_k^a)_{k\in\mathbb{N}}$ satisfy a *timing contract* $\theta(\underline{\tau}, \overline{\tau}, \underline{h}, \overline{h})$ given by

$$\begin{aligned} 0 \ &\leq t_0^s, \\ t_k^s \leq t_k^a &\leq t_{k+1}^s, && \forall k \in \mathbb{N} \\ \tau_k = t_k^a - t_k^s &\in [\underline{\tau}, \overline{\tau}], && \forall k \in \mathbb{N} \\ h_k = t_{k+1}^s - t_k^s &\in [\underline{h}, \overline{h}], && \forall k \in \mathbb{N} \end{aligned} \tag{2.2}$$

System (2.1) under timing contract (2.2) is reformulated into an impulsive system and then embedded in the framework of the difference inclusion (3.15) with Φ defined as in (3.17).

$$\Phi(\mathcal{S}) = \bigcup_{\tau\in[\underline{\tau},\overline{\tau}]} \bigcup_{w\in[\max(0,\underline{h}-\tau),\overline{h}-\tau]} e^{(w+\tau)\beta} e^{wA_c} A_a e^{\tau A_c} A_s \mathcal{S}, \tag{3.17}$$

where $\mathcal{S} \subseteq \mathbb{R}^n$ and matrices A_c, A_a, and A_s are given in terms of the matrices A, B, and K. We note at this point that the constant β is added to the expression of Φ so that later on stability could be checked for (2.1)-(2.2) with a given decay rate β. Equivalence between the stability of (2.1)-(2.2) and that of (3.15) is then established. After that, under certain assumptions, necessary and sufficient conditions for the stability of (3.15), and consequently for (2.1)-(2.2), are established in Theorem 17. Furthermore another necessary and sufficient stability conditions are established in Theorem 18 for the case of Φ as given in (3.17). The latter conditions are stronger than those in Theorem 17 in the sense that they are given in terms of a map $\hat{\Phi}$ whose images are convex sets, i.e. $\hat{\Phi}(\mathcal{S}) = ch(\Phi(\mathcal{S})) \supseteq \Phi(\mathcal{S})$. Practical sufficient stability results are then given based on an over-approximation $\overline{\Phi}$ of Φ, and which is computed using new reachability analysis approximations and the support function representation from [LGG10]. At last, the overall method is summarized by a stability verification algorithm which, for the case of impulsive systems, computes an initial contracting set $\mathcal{S}_0$ to several subsystems of (2.1)-(2.2) and then in the main loop propagates $\mathcal{S}_0$ using the map $\overline{\Phi}$. Then if the sufficient stability condition is verified stability is guaranteed, otherwise if a

maximum number of iterations is reached the algorithm fails to prove stability. At the end of the chapter comparisons of our stability verification approach with those in literature are conducted where our algorithm yields promising and competitive results. Two sections, that are not directly related to the main scope of the thesis, are added then to extend the work presented in this chapter to handle two other problems, for a certain subclass of (2.1)-(2.2), which are the self-triggered control problem and the stability verification problem under stochastic timing contracts.

1.2.2 Chapter 4. Scheduling of embedded controllers under timing contracts

Now given N systems, each of the form (2.1)-(2.2), sharing J CPUs to compute the control input for each control loop, a scheduling strategy needs to be synthesized if possible so that for all possible execution times, which are bounded in given intervals, at most one control task accesses a CPU at a time and all timing contracts are guaranteed. After associating to each control task and timing contract a timed game automaton TGA_i, $i = 1, \ldots, N$, we define a network of timed game automata TGA describing the parallel evolution of $\mathsf{TGA}_1, \ldots, \mathsf{TGA}_N$. Using TGA to reformulate the scheduling problem into a safety game (TGA,$\overline{L}_u$), we use a the tool UPPAAL Tiga to synthesize a strategy to trigger controllable actions (sampling, actuating, and start of computations in the CPU) so that for all possible uncontrollable actions (end of computation in the CPU) a defined set of undesired locations $\overline{L}_u$, given by (4.1), is avoided. After that we exploit the specific structure of TGA to give a simplified scheduling condition based on TGA', a particular case of TGA. At the end applications of scheduling on a single CPU and two CPUs are given to evaluate the approach.

1.2.3 Chapter 5. Parameter synthesis

In this chapter, we synthesize a set of timing contracts that guarantee at the same time the schedulability and the stability of the embedded controllers. The work-flow of the approach is given by Figure 5.1. We decouple the problem into two parts. First, we synthesize a set of timing contracts, defined by $\mathcal{P}_{st}^i$ giving a guarantee on stability for every system indexed by $i = 1, \ldots, N$. In the second step, we synthesize a set $\mathcal{P}_{sched}$ giving guarantee on schedulability. And as a consequence taking the intersection of the latter set with the augmented set $\mathcal{P}_{st} = \mathcal{P}_{st}^1 \times \cdots \times \mathcal{P}_{st}^N$, we solve the problem. The approaches followed in the first and second steps have the same concept which is first to re-parameterize the timing contracts so that stability (respectively schedulability) of (2.1)-(2.2) (respectively the control task-set) becomes monotone with respect to the new parameters. Then by using a guided search to sample the parameter space, Theorem 20 (respectively Theorem 21) allows us to synthesize timing contracts by keeping track of an under- and over-approximation of the parameter space and repeatedly sampling from the unexplored parameter set

until the distance between the over- and under-approximations is smaller than a given threshold. The overall approach is implemented by the timing contract synthesis algorithm and evaluated finally for two 2-dimensional systems scheduled on a uniprocessor.

1.3 Publications

As of today, this thesis led to one journal and four conference publications:

Journal paper:

- Mohammad Al Khatib, Antoine Girard, and Thao Dang. Stability verification and timing contract synthesis for linear impulsive systems using reachability analysis. *Nonlinear Analysis: Hybrid Systems, 2016.*

International conference paper:

- Mohammad Al Khatib, Antoine Girard, and Thao Dang. Stability verification of nearly periodic impulsive linear systems using reachability analysis. In *IFAC Conference on Analysis and Design of Hybrid Systems,* pages 358-363,2015.
- Mohammad Al Khatib, Antoine Girard, and Thao Dang. Verification and synthesis of timing contracts for embedded controllers. In *Proceedings of the 19th International Conference of Hybrid Systems: Computation and Control,* pages 115-124. ACM, 2016.
- Mohammad Al Khatib, Antoine Girard, and Thao Dang. Scheduling of embedded controllers under timing contracts. In *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control,* pages 131-140. ACM, 2017.
- Mohammad Al Khatib, Antoine Girard, and Thao Dang. Self-triggered control for sampled-data systems using reachability analysis. In *IFAC World Congress,* 2017.

1.4 Outline and note from the author

The thesis discusses interesting topics to researchers having skills from both the control engineering community and computer science community. Chapters 1, 2, 3, and 6 involve knowledge and problems more specific to the former field whereas Chapters 1, 2, 4, and 6 could be followed by readers more interested to the latter field. Chapter 5 concerns researchers having both disciplines. Normally, any serious reader should/can go through all the chapters.

After introducing the main topics in Chapter 1, the thesis is organized as follows. The problem setting is given in Chapter 2, where we introduce the considered classes of systems and timing contracts and where we formulate the stability verification, schedulability verification and timing contract synthesis problems. Also, the chapter illustrates the related work in literature to each of the scheduling and stability verification problems. Chapter 3 provides a solution to the stability verification problem based on reachability analysis and difference inclusions. Chapter 4 proposes a methodology to solve the scheduling problem based on timed games. Then, the timing contract synthesis problem is addressed in Chapter 5 before concluding our work in Chapter 6. Note that at the end of Chapters 3, 4, and 5 we evaluate our results through examples which are realized on a desktop with i7 4790 processor of frequency 3.6 GHz and a 8 GB RAM.

Chapter 2

Problem formulation and related work

Abstract

Timing contracts for embedded controller implementation specify the constraints on the time instants at which certain operations are performed such as sampling, actuation, and computation. In the first part of the chapter, we introduce the model of the sampled-data system under timing contract as well as formulate the three major problems we tackle in the thesis: stability verification problem, scheduling problem, and parameter synthesis problem. The first problem asks to prove exponential stability of a linear sampled-data system under a given timing contract with a predefined rate of convergence. In the second problem, we are required to prove the existence of a dynamic scheduling strategy to run some given control loops on a multi-processor system in such a way that only one control loop runs on any of the processors at any time and the timing contracts of all the control loops are always satisfied. As for the last problem, we require to generate a set of timing contracts that guarantee at the same time the schedulability and the stability of the embedded controllers. In the second part, we discuss related works in literature that solve instances of the scheduling and stability verification problems.

2.1 Problem formulation

2.1.1 Modeling with timing contracts

The model considered in the thesis is represented by the block diagram given by Figure 2.1. Typically, the plant's state z flows under continuous dynamics. Then, at each sampling instant t_k^s, $k \in \mathbb{N}$, the plant's state is sampled by the sampler and is passed through a network to the controller. The latter computes the control input u based on $z(t_k^s)$ and update the plant's input at instant t_k^a, $k \in \mathbb{N}$. The plant's input is then held constant by a zero order hold until the next update arrives via the network.

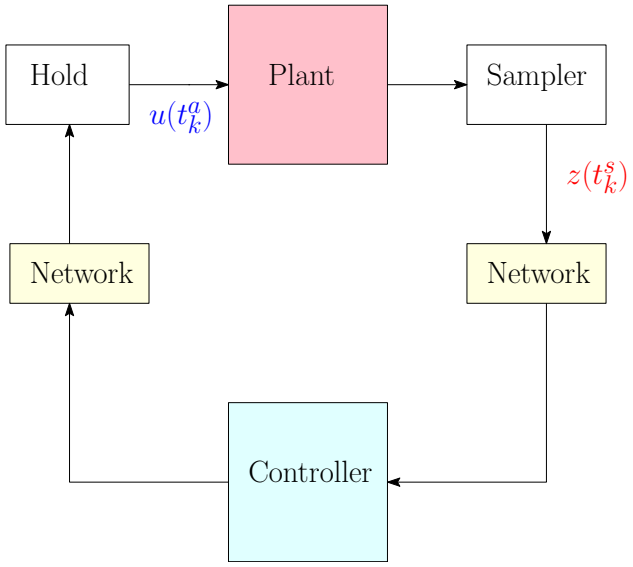


Figure 2.1: Block diagram of a sampled-data system.

Traditionally, controllers assumes that sampling is performed periodically and that actuation is performed with as little latency as possible. This scenario is shown in Figure 2.2 where the sampling instants are given by $t_k^s = kh$ for all $k \in \mathbb{N}$ and h being as the sampling period. However, when we want to formally capture the continuous dynamics of the plant and controller as well as the discrete dynamics, introduced by the sampler and hold for instance, in one model we use of *timing contracts*.

In other words, we use timing contracts to formulate an equivalent mathematical model to linear sampled-data systems that take into account the temporal nondeterministic of the sequences of sampling and actuation instants $(t_k^s)_{k\in\mathbb{N}}$ and $(t_k^a)_{k\in\mathbb{N}}$:

$$\dot{z}(t) = Az(t) + Bu(t), \quad \forall t \in \mathbb{R}_0^+ \tag{2.1a}$$

$$u(t) = Kz(t_k^s), \qquad t_k^a < t \leq t_{k+1}^a \tag{2.1b}$$

where $z(t) \in \mathbb{R}^p$ is the state of the system, $u(t) \in \mathbb{R}^m$ is the control input, the matrices $A \in \mathbb{R}^{p\times p}$, $B \in \mathbb{R}^{p\times m}$, $K \in \mathbb{R}^{m\times p}$ and $k \in \mathbb{N}$. In addition, it is assumed that K is designed such that the matrix $A+BK$ is Hurwitz and that for all $t \in [0, t_0^a]$, $u(t) = 0$.

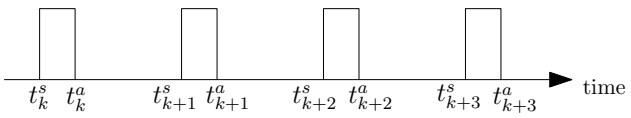


Figure 2.2: Periodic sampled-data systems.

We assume that the sequences of sampling and actuation instants $(t_k^s)_{k\in\mathbb{N}}$ and $(t_k^a)_{k\in\mathbb{N}}$ satisfy a *timing contract* $\theta(\underline{\tau},\overline{\tau},\underline{h},\overline{h})$ given by

$$\begin{aligned} 0 \ &\leq t_0^s, \\ t_k^s \leq t_k^a &\leq t_{k+1}^s, && \forall k \in \mathbb{N} \\ \tau_k = t_k^a - t_k^s &\in [\underline{\tau},\overline{\tau}], && \forall k \in \mathbb{N} \\ h_k = t_{k+1}^s - t_k^s &\in [\underline{h},\overline{h}], && \forall k \in \mathbb{N} \end{aligned} \tag{2.2}$$

where $\underline{\tau} \in \mathbb{R}_0^+$, $\overline{\tau} \in \mathbb{R}_0^+$, $\underline{h} \in \mathbb{R}^+$ and $\overline{h} \in \mathbb{R}^+$ provide bounds on the sampling-to-actuation delays (which includes time for computation of the control law) and sampling periods. Note that we impose $\underline{h} \neq 0$ to prevent Zeno behavior. Moreover, these parameters must belong to the following set $\mathcal{C}$ so that the time intervals given in (2.2) are always non-empty and it is always possible to choose $t_{k+1}^s \geq t_k^a$:

$$\mathcal{C} = \left\{(\underline{\tau},\overline{\tau},\underline{h},\overline{h}) \in \mathbb{R}_0^+ \times \mathbb{R}_0^+ \times \mathbb{R}^+ \times \mathbb{R}^+ : \underline{\tau} \leq \overline{\tau} \leq \overline{h}, \underline{h} \leq \overline{h}\right\}.$$

Contract (2.2) is a general timing contract which includes or over-approximates the different contracts introduced in [DLTT13]. Their relation to the timing contract (2.2) is described as follows:

1. **ZET Contract**: The Zero Execution Time contract is given by (2.2) with $\underline{\tau} = \overline{\tau} = 0$ and $\underline{h} = \overline{h} = h \in \mathbb{R}^+$. In other words, the contract states that the sampling and actuation instants are periodic and simultaneous such that $t_k^s = t_k^a = kh$ for $k \in \mathbb{N}$. As mentioned in [DLTT13], this contract is hardly achievable in practice since computation always takes time in between the sampling and actuation instants.

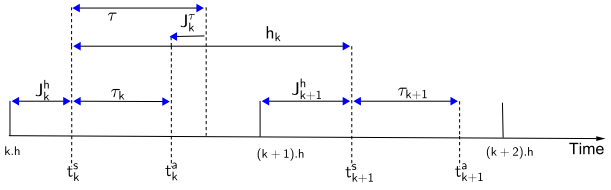


Figure 2.3: Time variables included in a *TOL* contract. $J_k^h \in [0, J^h]$ and $J_k^\tau \in [-J^\tau, J^\tau]$.

2. **LET Contract**: The Logical Execution Time contract is given by (2.2) with $\underline{\tau} = \overline{\tau} = \underline{h} = \overline{h} = h \in \mathbb{R}^+$. The contract states that the sampling and actuation instants are periodic such that $t_0^s = 0$ and $t_k^s = t_{k-1}^a = kh$ for $k \in \mathbb{N}^+$.
3. **DET Contract**: The Deadline Execution Time contract is given by (2.2) with $\underline{\tau} = 0$ and $\underline{h} = \overline{h} = h \in \mathbb{R}^+$. The contract states that the sampling instants are periodic, or $t_k^s = kh$ for $k \in \mathbb{N}$, and actuation instants are at some point t_k^a in the interval $[t_k^s, t_k^s + \overline{\tau}]$, with $\overline{\tau} \leq h$.
4. **TOL Contract**: The Timing Tolerance contract is defined by a nominal sampling period $h \in \mathbb{R}^+$, nominal sampling to actuation delay $\tau \in \mathbb{R}_0^+$, and two jitters $J^h, J^\delta \in \mathbb{R}_0^+$ with $J^\tau \leq \tau$ and $J^h + J^\tau + \tau \leq h$, such that $t_k^s \in [kh, kh + J^h]$ and $t_k^a \in [t_k^s + \tau - J^\tau, t_k^s + \tau + J^\tau]$, for $k \in \mathbb{N}$ (refer to Figure 2.3). We cannot exactly model this contract using (2.2). However we can over-approximate it using (2.2) with $\underline{\tau} = \tau - J^\tau$, $\overline{\tau} = \tau + J^\tau$, $\underline{h} = h - J^h$, and $\overline{h} = h + J^h$.

In the following, we formulate the three problems discussed in the thesis.

2.1.2 Stability verification problem

In our problem formulation, we consider the following notion of stability for system (2.1)-(2.2), that guarantees the exponential convergence of the state to the origin, i.e. $z = 0$, with a predefined rate $\beta \in \mathbb{R}^+$:

Definition 3 (β'-stability)**.** *Let $\beta \in \mathbb{R}^+$, system (2.1)-(2.2) is β'-stable if there exist $C \in \mathbb{R}^+$ and $\varepsilon' \in \mathbb{R}^+$ such that:*

$$|z(t)| \leq Ce^{-(\beta+\varepsilon')(t-t_0^s)}|z_0|, \ \forall t \in \mathbb{R}^+. \tag{2.3}$$

Consequently, in this work, we consider the following problem:

Problem 1 (Stability verification)**.** *Given $\beta \in \mathbb{R}^+$, $A \in \mathbb{R}^{p\times p}$, $B \in \mathbb{R}^{p\times m}$, $K \in \mathbb{R}^{m\times p}$, $(\underline{\tau}, \overline{\tau}, \underline{h}, \overline{h}) \in \mathcal{C}$, verify that (2.1)-(2.2) is β'-stable.*

The reader is referred to Chapter 3 where we provide an approach based on difference inclusions and reachability analysis to solve Problem 1 and to Section 2.2 for the state of the art concerning the stability verification problem.

2.1.3 Scheduling problem on multiple CPUs

Consider a collection of $N \in \mathbb{N}^+$ sampled-data systems $\{\mathcal{S}_1, \ldots, \mathcal{S}_N\}$ of the form (2.1) where each system $\mathcal{S}_i = (A_i, B_i, K_i)$ is subject to a timing contract $\theta(\underline{\tau}^i, \overline{\tau}^i, \underline{h}^i, \overline{h}^i)$ of the form (2.2), with parameters $(\underline{\tau}^i, \overline{\tau}^i, \underline{h}^i, \overline{h}^i) \in \mathcal{C}$, $i \in \mathbb{N}_{[1,N]}$.

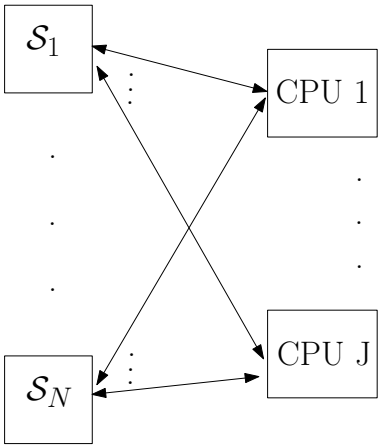


Figure 2.4: Block diagram of N sampled-data systems sharing J CPUs.

In addition, we assume that these systems share J CPUs, as shown in Figure 2.4, to compute the value of their control inputs given by (2.1b). Note that no communication exists in between the CPUs or the systems, but there exists communication only between the systems and all J CPUs. Furthermore, the time required by CPU j to compute inputs of system S_i is assumed to belong to some known interval $[\underline{c}_j^i, \overline{c}_j^i]$ with $0 \leq \underline{c}_j^i \leq \overline{c}_j^i$, $i \in \mathbb{N}_{[1,N]}$, and $j \in \mathbb{N}_{[1,J]}$, where $\underline{c}_j^i$ and $\overline{c}_j^i$ denote the best and worst case execution time respectively.

The timing of events in the k-th control cycle of system $\mathcal{S}_i$ starts at instant $t_k^{s_i}$ when sampling occurs. Then, system $\mathcal{S}_i$ gains access to the CPU j at instant $t_k^{b_i}$, at which computation of the control input value begins. The CPU is released at instant $t_k^{e_i}$, at which computation of the control input value ends. After that, actuation occurs at instant $t_k^{a_i}$. We denote by $\mathbb{N}(i,j)$ the set gathering indexes of the control cycles, system $\mathcal{S}_i$ accesses the CPU j, where $\bigcup_{j\in\mathbb{N}_{[1,J]}} \mathbb{N}(i,j) = \mathbb{N}$ for all $i \in \mathbb{N}_{[1,N]}$. Then, the sequences $(t_k^{s_i})_{k\in\mathbb{N}}$, $(t_k^{b_i})_{k\in\mathbb{N}}$, $(t_k^{e_i})_{k\in\mathbb{N}}$, and $(t_k^{a_i})_{k\in\mathbb{N}}$ satisfy the following constraints for all $i \in \mathbb{N}_{[1,N]}$:

$$
\begin{aligned}
0 \ &\leq t_0^{s_i} \\
t_k^{s_i} \leq t_k^{b_i} &\leq t_k^{e_i} \leq t_k^{a_i} \leq t_{k+1}^{s_i}, \quad \forall k \in \mathbb{N} \\
c_k^i = t_k^{e_i} - t_k^{b_i} &\in [\underline{c}_j^i, \overline{c}_j^i], \qquad \forall k \in \mathbb{N}(i,j), \forall j \in \mathbb{N} \\
\tau_k^i = t_k^{a_i} - t_k^{s_i} &\in [\underline{\tau}^i, \overline{\tau}^i], \qquad \forall k \in \mathbb{N} \\
h_k^i = t_{k+1}^{s_i} - t_k^{s_i} &\in [\underline{h}^i, \overline{h}^i], \qquad \forall k \in \mathbb{N}.
\end{aligned}
\tag{2.4}
$$

In addition, a conflict arises if several systems request access to one of the J CPUs at the same time. Let us define the following time sets, for $i \in \mathbb{N}_{[1,N]}$ and $j \in \mathbb{N}_{[1,J]}$:

$$\mathsf{Com}(\mathcal{S}_i, j) = \bigcup_{k \in \mathbb{N}(i,j)} [t_k^{b_i}, t_k^{e_i}).$$

$\mathsf{Com}(\mathcal{S}_i, j)$ is the union of time intervals when CPU j is used by system $\mathcal{S}_i$. Then, in order to prevent conflicting accesses to the CPU the following property must hold:

$$\begin{aligned} &\forall (m, n, j) \in \mathbb{N}^2_{[1,N]} \times \mathbb{N}_{[1,J]} \text{ with } m \neq n, \\ &\mathsf{Com}(\mathcal{S}_m, j) \cap \mathsf{Com}(\mathcal{S}_n, j) = \emptyset. \end{aligned} \tag{2.5}$$

Remark 1. *It is straightforward to verify that for any sequences $(t_k^{s_i})_{k\in\mathbb{N}}$, $(t_k^{b_i})_{k\in\mathbb{N}}$, $(t_k^{e_i})_{k\in\mathbb{N}}$, and $(t_k^{a_i})_{k\in\mathbb{N}}$ satisfying (2.4-2.5), the sequences $(t_k^{s_i})_{k\in\mathbb{N}}$ and $(t_k^{a_i})_{k\in\mathbb{N}}$ satisfy the timing contract $\theta(\underline{\tau}^i, \overline{\tau}^i, \underline{h}^i, \overline{h}^i)$.*

We aim at synthesizing a dynamic scheduling policy, generating sequences of timing events satisfying (2.4-2.5). The scheduler has control over the sampling and actuation instants $(t_k^{s_i})_{k\in\mathbb{N}}$, $(t_k^{a_i})_{k\in\mathbb{N}}$ and over the instants $(t_k^{b_i})_{k\in\mathbb{N}}$ when computation begins. Also, the scheduler assigns a CPU to compute the control input for each system $\mathcal{S}_i$ at each control cycle $k \in \mathbb{N}$. However, the execution time $(c_k^i)_{k\in\mathbb{N}}$, and thus the instants when computation ends $(t_k^{e_i})_{k\in\mathbb{N}}$, is determined by the environment and is therefore uncontrollable from the point of view of the scheduler. Next, given that a task-set $\mathcal{T}$, a task T_i, and timing contract Θ are characterized as

$$T_i = \left((\underline{c}_1^i, \overline{c}_1^i), \ldots, (\underline{c}_J^i, \overline{c}_J^i)\right), i \in \mathbb{N}_{[1,N]} \tag{2.6a}$$

$$\mathcal{T} = \{T_1, \ldots, T_N\}, \tag{2.6b}$$

and

$$\Theta = \{\theta(\underline{\tau}^1, \overline{\tau}^1, \underline{h}^1, \overline{h}^1), \ldots, \theta(\underline{\tau}^N, \overline{\tau}^N, \underline{h}^N, \overline{h}^N)\}, \tag{2.7}$$

we define the scheduling problem informally, at this point of the manuscript, as:

Problem 2 (Schedulability verification)**.** *Given a set of control tasks $\mathcal{T}$ as in (2.6), and timing contracts $\Theta = \{\theta(\underline{\tau}^1, \overline{\tau}^1, \underline{h}^1, \overline{h}^1), \ldots, \theta(\underline{\tau}^N, \overline{\tau}^N, \underline{h}^N, \overline{h}^N)\}$, verify whether or not there exists a scheduling policy with sequences of timing events satisfying (2.4-2.5).*

A precise formulation of the schedulability of the task-set $\mathcal{T}$ is provided in Chapter 4 along with a solution to the schedulability verification problem based on safety games over timed game automata.

2.1.4 Parameter synthesis problem

In this section we define the problem of synthesizing a set of timing contracts that guarantee at the same time the stability of the systems and the schedulability of control tasks.

Given the bounds on the parameters $0 \leq \underline{\tau}^i_{min} \leq \underline{\tau}^i_{max} \leq \overline{\tau}^i_{max}$, $\underline{\tau}^i_{min} \leq \overline{\tau}^i_{min} \leq \overline{\tau}^i_{max}$, $0 < \underline{h}^i_{min} \leq \underline{h}^i_{max} \leq \overline{h}^i_{max}$, $\underline{h}^i_{min} < \overline{h}^i_{min} \leq \overline{h}^i_{max}$, with $\underline{\tau}^i_{min} \leq \underline{h}^i_{min}$, $\overline{\tau}^i_{min} \leq \overline{h}^i_{min}$, $\underline{\tau}^i_{max} \leq \underline{h}^i_{max}$, $\overline{\tau}^i_{max} \leq \overline{h}^i_{max}$, let

$$\mathcal{D}_i = [\underline{\tau}^i_{min}, \underline{\tau}^i_{max}] \times [\overline{\tau}^i_{min}, \overline{\tau}^i_{max}] \times [\underline{h}^i_{min}, \underline{h}^i_{max}] \times [\overline{h}^i_{min}, \overline{h}^i_{max}], \;\; i \in \mathbb{N}_{[1,N]}, \tag{2.8}$$

with $N \in \mathbb{N}^+$, the timing contract synthesis problem is formalized as follows:

Problem 3 (Timing contract synthesis)**.** *Given a collection of systems $\{\mathcal{S}_1, \ldots, \mathcal{S}_N\}$, where $\mathcal{S}_i = (A_i, B_i, K_i)$ with $A_i \in \mathbb{R}^{n_i \times n_i}$, $B_i \in \mathbb{R}^{n_i \times m_i}$, and $K_i \in \mathbb{R}^{m_i \times n_i}$, $i \in \mathbb{N}_{[1,N]}$, J CPUs, a set of control tasks $\mathcal{T}$ by (2.6b), a set $\{\beta_i\}_{i \in \mathbb{N}_{[1,N]}} \subset \mathbb{R}^+$, and parameter sets $\mathcal{D}_i$, $i \in \mathbb{N}_{[1,N]}$, synthesize a set $\mathcal{P}^* \subseteq (\mathcal{C}^N) \cap (\mathcal{D}_1 \times \cdots \times \mathcal{D}_N)$ such that for all $(\underline{\tau}^1, \overline{\tau}^1, \underline{h}^1, \overline{h}^1, \ldots, \underline{\tau}^N, \overline{\tau}^N, \underline{h}^N, \overline{h}^N) \in \mathcal{P}^*$:*

1. *System $\mathcal{S}_i = (A_i, B_i, K_i)$ is β'_i-stable under timing contract $\theta(\underline{\tau}_i, \overline{\tau}_i, \underline{h}_i, \overline{h}_i)$, for all $i \in \mathbb{N}_{[1,N]}$.*
2. *The set of control tasks $\mathcal{T}$ is schedulable under timing contracts Θ given by (2.7).*

After defining the three main problems we are addressing in this thesis we discuss and highlight on existing work in literature that solves instances of the first two problems in the next section. After that we illustrate in three separate chapters our approaches that solve Problems 1, 2, and 3. It is noteworthy that as far as we know, there is no available approach for addressing Problem 3 besides our preliminary work [AKGD15] where we impose $\underline{\tau} = \overline{\tau} = 0$ in (2.2).

2.2 Related work to the stability verification problem

This section is dedicated, for simplicity, to present an overview of the different approaches that solve an instance of the stability verification problem, Problem 1. More precisely, this is the case of sampled-data systems with aperiodic sampling period given by (2.1-2.2) with $\underline{\tau} = \overline{\tau} = 0$. In other words, (2.1) are simplified to

$$\dot{z}(t) = Az(t) + Bu(t), \tag{2.9a}$$

$$u(t) = Kz(t^s_k), \quad t^s_k < t \leq t^s_{k+1}, \tag{2.9b}$$

where the initial state is given as $z(0) = z_0$.

Systems with aperiodic sampling can be seen as time-delay systems, hybrid systems, input/output interconnections, or discrete-time systems with time-varying parameters. For each of these point of views we present the basic concepts and fundamental results of the respective stability verification approaches. Moreoer, at the end of this section, we summarize some of the methods that extend the presented approaches to solve instances of Problem 1. The reader is referred to an extended overview of the work, presented in this section, in [HFO+17].

2.2.1 Time-delay approach

In this section we consider the following timing contract given by

$$\begin{aligned} 0 \ &\leq t_0^s, \\ h_k = t_{k+1}^s - t_k^s &\in (0, \overline{h}], \quad \forall k \in \mathbb{N}. \end{aligned} \tag{2.10}$$

2.2.1.1 Reformulation

We first remark that the control input in (2.9b) could be rewritten as

$$\begin{aligned} u(t) \quad &= Kz(t_k^s) = Kz(t - \tau(t)), \\ \tau(t) \quad &= t - t_k^s, \quad \forall t \in [t_k^s, t_{k+1}^s), \end{aligned} \tag{2.11}$$

with τ a piecewise-linear function satisfying $\dot{\tau}(t) = 1$ for $t \neq t_k^s$ and $\tau(t_k^s) = 0$. Then, (2.9) is reformulated as a linear time invariant system with a time-varying delay

$$\dot{z}(t) = Az(t) + BKz(t - \tau(t)), \quad \forall t \geq 0. \tag{2.12}$$

Let the functional $z_t(\theta) = z(t+\theta) \in C^0([-\underline{h}, 0], \mathbb{R}^p)$ for all $\theta \in [-\overline{h}, 0)$, where $C^0([-\underline{h}, 0], \mathbb{R}^p)$ is the set of continuous functions mapping $[-\underline{h}, 0]$ to $\mathbb{R}^p$. Also denote by $W([-\overline{h}, 0], \mathbb{R}^p)$ the Banach space of continuous functions $\phi : [-\overline{h}, 0] \to \mathbb{R}^p$ with $\dot{\phi} \in \mathcal{L}_2^p(-\overline{h}, 0)$ (set of the square integrable functions that map from $(-\overline{h}, 0)$ to $\mathbb{R}^p$) with the norm

$$|\phi|_W = \max_{s \in [-\overline{h}, 0]} |\phi(s)| + \left(\int_{-\overline{h}}^{0} |\dot{\phi}(s)|^2 ds \right)^{\frac{1}{2}}. \tag{2.13}$$

2.2.1.2 Theoretical foundation

The theoretical foundation of the presented time-delay approach is based on a popular generalization of the direct Lyapunov method for time-delay systems proposed by Krasovskii [Kra63].

Theorem 4. *(Lyapunov-Krasovskii Theorem[Kra63]) Consider $f : \mathbb{R}^+ \times C^0([-\underline{h}, 0], \mathbb{R}^p) \to \mathbb{R}^p$ continuous in both arguments and locally Lipschitz in the second argument. Assume that $f(t, 0) = 0$ for all $t \in \mathbb{R}^+$ and that f maps $\mathbb{R} \times \mathcal{K}(C^0([-\underline{h}, 0], \mathbb{R}^p))$ into $\mathcal{K}(\mathbb{R}^p)$. Suppose that $v, w, q : \mathbb{R}^+ \to \mathbb{R}^+$ are continuous nondecreasing functions, $v(s)$, $w(s)$, and $q(s)$ are positive for $s > 0$, $\lim_{s\to\infty} v(s) = \lim_{s\to\infty} w(s) = \lim_{s\to\infty} q(s) = \infty$ and $v(0) = w(0) = q(0) = 0$. The trivial solution of*

$$\dot{z}(t) = f(t, z_t),$$

is β^*-stable, with $\beta = 0$, if there exists a continuous functional $V : \mathbb{R} \times W([-\overline{h}, 0], \mathbb{R}^p) \times \mathcal{L}_2^p(-\overline{h}, 0) \to \mathbb{R}^+$, which is positive-definite, i.e.

$$v(|\phi(0)|_W) \leq V(t, \phi, \dot{\phi}) \leq w(|\phi|_W), \tag{2.14}$$

for all $\phi \in W([-\overline{h}, 0])$, $t \in \mathbb{R}^+$, and such that its derivative along the system's solutions is non-positive

$$\dot{V}(t, z_t, \dot{z}_t) \leq -q(|z_t(0)|_W). \tag{2.15}$$

A functional V satisfying the conditions of Theorem 4 is called a Lyapunov-Krasovskii Functional (LKF).

2.2.1.3 Practical conditions

Now let us summarize the time-delay LKF based approach, to verify stability of sampled data systems with aperiodic sampling, in four main steps. Note that sufficient practical conditions are derived at the end to guarantee stability of the system at hand.

1. *Propose a LKF:* A candidate LKF that satisfies (2.14) is given by

$$V(z_t, \dot{z}_t) = z^\top(t) P z^\top(t) + \overline{h} \int_{-\overline{h}}^{0} \int_{t+\theta}^{t} \dot{z}^\top(s) R \dot{z}(s) ds d\theta, \tag{2.16}$$

   with $P, R \succ 0$.

2. *Compute the derivative of V:*

$$\dot{V}(z_t, \dot{z}_t) = 2\dot{z}^\top(t) P z^\top(t) + \overline{h}^2 \dot{z}^\top(t) R \dot{z}(t) - \overline{h} \int_{t-\overline{h}}^{t} \dot{z}^\top(s) R \dot{z}^\top(s) ds. \tag{2.17}$$

3. *Over-approximate the integral terms:* The integral term in (2.17), or

$$J(\dot{z}_t, \overline{h}) = -\int_{t-\overline{h}}^{t} \dot{z}^\top(s) R \dot{x}(s) ds = -\int_{t-\overline{h}}^{t-\tau(t)} \dot{z}^\top(s) R \dot{z}(s) ds - \int_{t-\tau(t)}^{t} \dot{z}^\top(s) R \dot{x}(s) ds \tag{2.18}$$

needs to be replaced by simple expressions for subsequent manipulations. For this reason we recall the following relevant tool.

Lemma 2. *(Jensen's inequality [GCK03]) Given $R \succ 0$, $\theta \geq 0$, and a differentiable function $z : [t-\theta, t] \to \mathbb{R}^p$, the following inequality holds:*

$$J(\dot{z}_t, \theta) = -\int_{t-\theta}^{t} \dot{z}(s)^\top R \dot{z}(s) ds \leq -\frac{1}{\theta}(z(t) - z(t-\theta))^\top R (z(t) - z(t-\theta)). \tag{2.19}$$

After applying Jensen's inequality in (2.18) and substituting the result in (2.17) we get:

$$\dot{V}(z_t, \dot{z}_t) \leq 2\dot{z}^\top(t) P z^\top(t) + \overline{h}^2 \dot{z}^\top(t) R \dot{z}(t) - \xi(t)^\top \overline{R}(\tau(t)) \xi(t), \tag{2.20}$$

where $\xi(t) = \begin{pmatrix} z(t) - z(t-\tau(t)) \\ z(t-\tau(t)) - z(t-\overline{h}) \end{pmatrix}$, and $\overline{R}(\tau(t)) = \begin{pmatrix} \frac{\overline{h}}{\tau(t)} R & 0 \\ 0 & \frac{\overline{h}}{\overline{h}-\tau(t)} R \end{pmatrix}$.

4. *Over-approximate the delay dependent terms:* Any stability condition that involves $\overline{R}(\cdot)$ needs to be checked for all $\tau(t) \in [-\overline{h}, 0]$ due to its dependence on $\tau(t)$, which is not practical. Then by noting that $\frac{\overline{h}}{\tau(t)} \geq 1$ and $\frac{\overline{h}}{\overline{h}-\tau(t)} \geq 0$ we have:

$$\overline{R}(\tau(t)) \succeq \begin{pmatrix} R & 0 \\ 0 & 0 \end{pmatrix},$$

which permits us to over-approximate the terms in (2.20) leading to

$$\dot{V}(z_t, \dot{z}_t) \leq \begin{pmatrix} z(t) \\ z(t-\tau(t)) \end{pmatrix}^\top \Psi(P, R) \begin{pmatrix} z(t) \\ z(t-\tau(t)) \end{pmatrix}, \tag{2.21}$$

with

$$\Psi(P, R) = \begin{pmatrix} PA + A^\top P & P(BK) \\ (BK)^\top P & 0) \end{pmatrix} + \overline{h}^2 \begin{pmatrix} A^\top \\ (BK)^\top \end{pmatrix} R \begin{pmatrix} A & BK \end{pmatrix} - \begin{pmatrix} I \\ -I \end{pmatrix} R \begin{pmatrix} I & -I \end{pmatrix}. \tag{2.22}$$

Consequently, sufficient conditions on the stability of the system are given by the following theorem:

Theorem 5. *[HFO+17] Assume that there exist $P, R \succ 0$, such that the linear matrix inequality $\Psi(P, R) \prec 0$ holds with Ψ given by (2.22). Then, the sampled-data system (2.9-2.10) is β'-stable, with $\beta = 0$.*

2.2.1.4 Improvements and further reading

Following the presented approach above, we discuss the following three points.

1. *Conservativeness:* Theorem 5 presents just a sufficient condition for the stability of system (2.9-2.10). The main sources of conservatism are the choice of the LKF (step 1) and the over-approximation of its derivative (steps 3 and 4). Improvements for this approach are suggested by choosing other LKF [SLCR10], by using alternatives to Jensen's inequality such as Wirtinger's inequality, Bessel's inequality, or Legendre polynomials [LPJ+14, SG13, SG14], and by providing more accurate over-approximation of the delay dependent terms in step 4 [PKJ11].
2. *Extensions:* An advantage of this methodology is the possibility of extending the results in the case of linear systems to control design [LF12], scheduling [LFH15], stability verification for the case of systems with parameter uncertainties [PHYT11], and to stability verification for classes of nonlinear systems [MMD13].
3. *Other methods:* In the time-delay community, methods other than those based on LKFs could be cited like the one proposed by Razumikhin [Raz56].

2.2.2 Hybrid system approach

System (2.9) integrates both discrete and continuous dynamics and thus could be effectively modeled in the hybrid system formulation, $\mathcal{H} = (F, C, J, D)$, proposed by [GST09]:

$$\dot{\bar{z}}(t,k) \in F(\bar{z}(t,k)), \qquad \bar{z}(t,k) \in \mathcal{C} \tag{2.23a}$$

$$\bar{z}(t,k+1) \in J(\bar{z}(t,k)). \quad \bar{z}(t,k) \in \mathcal{D} \tag{2.23b}$$

where $\bar{z}(t,k) \in \mathbb{R}^{\bar{p}}$ represents the state of the hybrid system after t time units and k jumps. Solutions of (2.23) are parametrized by both the continuous time t and discrete time k and are thus defined on a hybrid time domain *dom* $\bar{z}$.

Definition 4. *(hybrid time domain) A compact hybrid time domain is a set* $\mathcal{E} = \cup_{k=0}^{\bar{k}-1}([t_k, t_{k+1}], k) \subset \mathbb{R}_0^+ \times \mathbb{N}$ *with* $\bar{k} \in \mathbb{N}^+$ *and* $0 = t_0 \leq \cdots \leq t_{\bar{k}}$. *A hybrid time domain is a set dom* $\bar{z} \subset \mathbb{R}_0^+ \times \mathbb{N}$ *such that dom* $\bar{z} \cap ([0,T] \times \{0,\ldots,\bar{k}\})$ *is a compact hybrid time domain for each* $(T,\bar{k}) \in$ *dom* $\bar{z}$.

The system's state evolves according to an ordinary differential inclusion (2.23a) when the state is in $\mathcal{C} \subseteq \mathbb{R}^{\bar{p}}$ and according to (2.23b) when the state is in $\mathcal{D} \subseteq \mathbb{R}^{\bar{p}}$. Note that $\bar{z}(t,k+1)$ denotes the value of the

state after the reset. For the hybrid system's approach we consider the following timing contract given by

$$\begin{aligned} 0 \ &\leq t_0^s, \\ h_k = t_{k+1}^s - t_k^s &\in [\underline{h}, \overline{h}], \quad \forall k \in \mathbb{N}. \end{aligned} \tag{2.24}$$

2.2.2.1 Reformulation

We can reformulate the aperiodic sampled-data system (2.9)-(2.24) in the hybrid formulation (2.23) with $\bar{z}^\top(t,k) = [\ z^\top(t) \quad z^\top(\theta^s(t,k)) \quad \tau(t)\ ] = [\ \chi^\top(t) \quad \tau(t)\ ] \in \mathbb{R}^{\bar{p}}$, $\bar{p} = 2p+1$, $\theta^s(t,k) = t_k^s$, for all $t \in (t_k^s, t_{k+1}^s]$, and

$$\mathcal{C} = \{\bar{z} \in \mathbb{R}^{\bar{p}} : \tau \in [0, \overline{h}]\}, \tag{2.25a}$$

$$\mathcal{D} = \{\bar{z} \in \mathbb{R}^{\bar{p}} : \tau \in [\underline{h}, \overline{h}]\}, \tag{2.25b}$$

$$F(\bar{z}(t,k)) = \begin{pmatrix} Az + BKz(\theta^s(t,k)) \\ 0 \\ 1 \end{pmatrix}, \ J(\bar{z}(t,k)) = \begin{pmatrix} z(t) \\ z(t) \\ 0 \end{pmatrix}. \tag{2.25c}$$

We refer with $\mathcal{H} = (A, B, K, C, D)$, to the hybrid system (2.23) where matrices are given by (2.25). After defining the distance of a vector $\bar{z} \in \mathbb{R}^{\bar{p}}$ to a compact set $\mathcal{A} \in \mathcal{B}(\mathbb{R}^p)$ by

$$|\bar{z}|_{\mathcal{A}} = \min\{|\bar{z} - y| : y \in \mathcal{A}\}, \tag{2.26}$$

stability of (2.23) is given in the following sense:

Definition 5. *(pre-asymptotic stability) We say that a set $\mathcal{A} \in \mathcal{B}(\mathbb{R}^{\bar{p}})$ is pre-asymptotically stable if:*

1. *$\mathcal{A}$ is stable: $\forall \epsilon > 0, \exists \delta > 0 : |\bar{z}(0,0)|_{\mathcal{A}} \leq \delta \Rightarrow |\bar{z}(t,k)|_{\mathcal{A}} < \epsilon$ for all $(t,k) \in dom\ \bar{z}$ and all possible solutions $\bar{z}$ of (2.23),*
2. *$\mathcal{A}$ is pre-attractive: $|\bar{z}(t,k)|_{\mathcal{A}} \to 0$ as $t + k \to +\infty$ where $(t,k) \in dom\ \bar{z}$.*

Note that the pre-asymptotic stability of system $\mathcal{H} = (A, B, K, C, D)$ is directly related to the β'-stability, with $\beta = 0$.

2.2.2.2 Theoretical foundation

General sufficient theoretical conditions for the stability of hybrid systems, $\mathcal{H} = (F, C, J, D)$, are given using a common Lyapunov function:

Theorem 6. *(common Lyapunov function [GST09]) Consider the hybrid system (2.23) and the set $\mathcal{A} \in \mathcal{B}(\mathbb{R}^{\bar{p}})$ such that the reset $J(\mathcal{A} \cap D) \subset \mathcal{A}$. If there exists a candidate Lyapunov function[1] V such that*

$$\frac{\partial V}{\partial t} F(\bar{z}) < 0 \quad \textit{for all } \bar{z} \in C \setminus \mathcal{A}, \tag{2.27a}$$

$$V(J(\bar{z})) - V(\bar{z}) < 0 \quad \textit{for all } \bar{z} \in D \setminus \mathcal{A}, \tag{2.27b}$$

then the set $\mathcal{A}$ is pre-asymptotically stable.

We consider now the system $\mathcal{H} = (A, B, K, C, D)$ and establish asymptotic stability[2] of the compact set

$$\mathcal{A} = \{\bar{z} : \chi(t) = 0, \tau \in [0, \overline{h}]\}, \tag{2.28}$$

with the candidate Lyapunov function $V(\chi) = \chi^\top P(\tau)\chi$ where $P : [0, \overline{h}] \to \mathbb{P}_{\bar{p}}$ and $P_{\bar{p}}$: the set of symmetric positive definite matrices. Then sufficient conditions could be obtained from Theorem 6:

Theorem 7. *(adapted from [HFO+17]) If there exists a differentiable matrix function $P : [0, \overline{h}] \to \mathbb{P}_{\bar{p}}, c_1 \prec P(\tau) \prec c_2 I$, satisfying the parametric set of LMIs*

$$F^\top P(\theta_1) + P(\theta_1)F + c_3 P(\theta_1) + \frac{\partial P}{\partial \tau}(\theta_1) \prec 0 \quad \forall \theta_1 \in [0, \overline{h}], \tag{2.29a}$$

$$J^\top P(0) J - P(\theta_2) \prec 0, \quad \forall \theta_2 \in [\underline{h}, \overline{h}], \tag{2.29b}$$

with $c_1, c_2, c_3 \in \mathbb{R}^+$ and F, J given by (2.25), then set $\mathcal{A}$ given by (2.28) is pre-asymptotically stable.

2.2.2.3 Practical conditions

Stability criteria (2.29), given by Theorem 7, is parametrized in τ which means that these conditions are infinite number of LMIs that need to be checked for all values of $\tau \in [0, \overline{h}]$ and thus are intractable.

Alternatively, let $P(\tau)$ be linear with respect to τ :

$$P(\tau) = P_1 + (P_2 - P_1)\frac{\tau}{\overline{h}}, \tag{2.30}$$

with $P_1, P_2 \succ 0$. Thus, a tractable sufficient stability criteria, derived from (2.29)-(2.30), is given by the following theorem in terms of a finite number of LMI conditions:

[1] V is *i)* continuous and non-negative on $C \cup D \setminus \mathcal{A} \subset dom\ V$, *ii)* $\lim_{\bar{z} \to \mathcal{A}, \bar{z} \in dom\ V \cap (C \cup D)} V(\bar{z}) = 0$, and *iii)* V is continuously differentiable on an open set $\mathcal{O}$ satisfying $C \setminus \mathcal{A} \subset \mathcal{O} \subset dom\ V$.

[2] For sampled-data systems (2.9)-(2.24) asymptotic and pre-asymptotic stability are equivalent since solutions are complete. However, we only define pre-asysmptotic stability for systems where solutions are not defined as $t \to +\infty$.

Theorem 8. *(adapted from [HFO+17]) Let P_1, P_2 be given by (2.30) and F, J by (2.25). If there exists $c_3 \in \mathbb{R}^+$ such that*

$$F^\top P_1 + P_1 F + c_3 P_1 + \tfrac{P_2 - P_1}{\bar{h}} \prec 0, \tag{2.31a}$$

$$F^\top P_2 + P_2 F + c_3 P_2 + \tfrac{P_2 - P_1}{\bar{h}} \prec 0, \tag{2.31b}$$

$$J^\top P_1 J \prec P_2, \tag{2.31c}$$

$$J^\top P_1 J \prec P_1 + \tfrac{(P_2 - P_1)\underline{h}}{\bar{h}}, \tag{2.31d}$$

then set $\mathcal{A}$ given by (2.28) is pre-asymptotically stable.

2.2.2.4 Improvements and further reading

Improvements to the illustrated method for studying the stability of aperiodic sampled-data system could be explained based on the conservatism introduced by the two main Theorems of the approach, i.e. Theorems 7 and 8:

1. *Conservatism due to Theorem 7:* The results in this theorem are based on the existence of a common candidate Lyapunov function satisfying the conditions in Theorem 6 and which are just sufficient for stability. Similar functions in literature, that could reduce conservatism, are proposed in [HLCS03, NHT08, NTC09, GST12, FGNZ14]. Alternatively, one may rely on the existence of multiple Lyapunov functions as suggested by Theorem 32 in [GST09] and which is necessary and sufficient for the stability of general hybrid systems $\mathcal{H} = (F, C, J, D)$.
2. *Conservatism due to Theorem 8:* An alternative derivation of tractable conditions based on polynomial matrix functions $P(\tau)$, which could lead to less conservatism, is suggested in [Bri13] where the authors use Sum-of-Squares (SOS) programming [PPP02].

Remark 2. *Other work in literature could be classified within the hybrid systems approach for solving the stability verification problem for aperiodic sampled-data systems like the studies that are based on an impulsive system reformulation of (2.20) such as [BS12, HDTP13, YMH98, NT04, CTN07].*

2.2.3 Discrete-time approach

We present an approach to solve the stability verification problem for system (2.9) under timing contract (2.24). Here, the study is based on the convex embedding of the transition matrix between sampling times. Theoretical results are presented before explaining techniques to derive tractable conditions on stability.

2.2.3.1 Reformulation

It is direct to show that the state of the system at sampling times evolve according to a discrete-time linear parameter varying dynamics [KK84]:

$$z(t_{k+1}^s) = \Lambda(h_k) z(t_k^s), \tag{2.32a}$$

$$\Lambda(\theta) = e^{A\theta} + \int_0^\theta e^{As} ds B K, \quad \forall \theta \in \mathbb{R} \tag{2.32b}$$

with $h_k \in [\underline{h}, \overline{h}]$.

Note that within the same discrete-time approach the system could be viewed as a difference inclusion. We adopt this viewpoint in Chapter 3. However for this section, we adopt (2.32) to study the stability verification problem.

2.2.3.2 Theoretical foundation

The following theorem gives necessary and sufficient conditions on the stability of system (2.32)-(2.24):

Theorem 9. *(adapted from [HKPR11]) Consider System (2.9)-(2.24) as well as (2.32). The following statements are equivalent:*

1. *The equilibrium point $z = 0$ of (2.9)-(2.24) is β'-stable, with $\beta = 0$.*
2. *There exist $P \succ 0$ and $N > 0$ such that:*

$$\left(\prod_i^N \Lambda(\theta_i) \right)^\top P \left(\prod_i^N \Lambda(\theta_i) \right) - P \prec 0, \tag{2.33}$$

for any N-length sequence $(\theta_i)_{i \in \mathbb{N}_{[1,N]}}$ with values in $[\underline{h}, \overline{h}]$.

Theorem 9 guarantees the equivalence between the stability of (2.9)-(2.24) and the existence of a *non-monotonic Lyapunov function* $V(z) = z^\top P z$, which decreases every N samples. Furthermore, if we restrict ourselves to quadratic Lyapunov functions which decreases at every sampling, i.e. $N = 1$, then Condition 2 in Theorem 9 becomes only sufficient.

2.2.3.3 Practical conditions

Now based on Theorem 9 let us give an idea on how to obtain practical **sufficient** conditions for the stability verification problem using the Taylor approximation method [HDI06, HVDWG+10].

Note that the transition matrix Λ is parametrized in h_k, $h_k \in [\underline{h}, \overline{h}]$, and could be rewritten as

$$\Lambda(h_k) = \Lambda(\underline{h}) + \Delta(\rho_k)\Psi(\underline{h}), \quad \text{with} \tag{2.34a}$$

$$\Delta(\rho) = \textstyle\int_0^\rho e^{As} ds, \ \forall \rho \in \mathbb{R}; \quad \Psi(\underline{h}) = A\Lambda(\underline{h}) + BK, \tag{2.34b}$$

and $\rho_k = h_k - \underline{h} \in [0, \overline{h} - \underline{h}]$.

Using the definition of the matrix exponential,

$$e^{A\rho} = \sum_{i=0}^{\infty} \frac{A^i}{i!}\rho^i, \quad \forall \rho \in \mathbb{R}$$

$\Delta(\rho)$ in (2.34b) can be expressed as

$$\Delta(\rho) = \int_0^\rho e^{As} ds = \sum_{i=0}^{\infty} \frac{\rho A^i}{(i+1)!}\rho^i. \tag{2.35}$$

Consider now the M-order Taylor approximation of $\Delta(\rho)$ in (2.35) given by

$$\Delta^M(\rho) = \sum_{i=0}^{M-1} \frac{A^i \rho}{(i+1)!}\rho^i. \tag{2.36}$$

The approximation error, $\delta^M(\rho) = \Delta(\rho) - \Delta^M(\rho)$, is then given by

$$\delta(\rho)^M = \sum_{i=M}^{\infty} \frac{\rho A^i}{(i+1)!}\rho^i. \tag{2.37}$$

The idea of the approach at this level is to embed $\Delta^M(\rho)$ in a convex polytope and to bound the approximation error $\delta^M(\rho)$. In other words,

Lemma 3. *[HDI06] Consider a polynomial matrix*

$$\Delta^M(\rho) = \sum_{i=0}^{\bar{M}} \rho^i \Delta_i, \tag{2.38}$$

with $\rho \in \mathbb{R}$ and $L_i \in \mathbb{R}^{p\times p}$. For each upper bound $\overline{\rho} \geq 0$ on ρ there exist matrices $U_i \in \mathbb{R}^{p\times p}$, $i \in \mathbb{N}_{[1,\bar{M}+1]}$ such that for all $\rho \in [0, \overline{\rho}]$ there exist parameters $\mu_i(\rho)$, $i \in \mathbb{N}_{[1,\bar{M}+1]}$ with:

$$\Delta^M(\rho) = \sum_{i=1}^{\bar{M}+1} \mu_i(\rho) U_i, \tag{2.39}$$

where $\mu_i(\rho) \geq 0$, *for all* $i \in \mathbb{N}_{[1,M+1]}$, *and* $\sum_{i=1}^{\bar{M}+1} \mu_i(\rho) = 1$. *A choice for* U_i, $i \in \mathbb{N}_{[1,\bar{M}+1]}$, *is:*

$$\begin{aligned} U_1 &= \Delta_0, \\ U_2 &= \overline{\rho}\Delta_1 + \Delta_0, \\ &\vdots \\ U_{\bar{M}+1} &= \overline{\rho}^{\bar{M}}\Delta_{\bar{M}} + \overline{\rho}^{\bar{M}-1}\Delta_{\bar{M}-1} + \cdots + \overline{\rho}\Delta_1 + \Delta_0. \end{aligned} \tag{2.40}$$

Lemma 3 suggests that we could embed $\Delta^M(\rho)$, given by (2.36), in a polytope with $M+1$ vertices such that:

$$\Delta^M(\rho) \in ch\{U_i : i \in \mathbb{N}_{[1,M+1]}\}, \quad \forall \rho \in [0, \overline{h} - \underline{h}], \tag{2.41}$$

where U_i are given by (2.40), $\bar{M} = M - 1$, $\overline{\rho} = \overline{h} - \underline{h}$, and $\Delta_i = \frac{\rho A^i}{(i+1)!}$.

As for the error $\delta^M(\rho)$, given by (2.37), its Euclidean norm could be bounded effectively from above by an arbitrary $\varepsilon > 0$, or $\|\delta^M(\rho)\|_2 < \varepsilon$ for all $\rho \in [0, \overline{h} - \underline{h}]$, if the approximation order M is chosen such that

$$\frac{\|A\|_2(\overline{h}-\underline{h})}{M+2} < 1 \quad \text{and} \tag{2.42a}$$

$$\frac{\|A^M\|_2(\overline{h}-\underline{h})^{M+1}}{(M+1)!}\frac{M+2}{M+2-\|A\|_2(\overline{h}-\underline{h})} \leq \varepsilon. \tag{2.42b}$$

For details on bounding $\delta^M(\rho)$ the reader is referred to [Lio66]. Consequently, using the polytopic embedding (2.41) and the error bounding conditions in (2.42) the reader could follow the steps in [HDI06] to construct a polytopic embedding for $\Lambda(h_k)$, given in (2.34), of the form:

$$\Lambda(h_k) \in \mathcal{W} = ch\{W_1, \ldots, W_{M+1}\}. \tag{2.43}$$

After constructing the needed polytopic approximation $\mathcal{W}$ we can directly obtain sufficient numerical conditions on the stability of (2.9)-(2.24) from Theorem 9.

Theorem 10. *(Theorem 2 in [HKPR11]) Consider system (2.9)-(2.24), (2.32), the polytopic set* $\mathcal{W}$ *given in (2.43), and the set*

$$\mathcal{Y}(\mathcal{W}) = \{Y : Y = \prod_{i=0}^{N-1} W_{\mu_i}, \;\; W_{\mu_i} \in \mathcal{W}, \;\; \mu_i \in \mathbb{N}_{[1,M+1]}\}. \tag{2.44}$$

If there exist a matrix $P = P^\top \succ 0$ *and* $N \in \mathbb{N}^+$, *such that:*

$$P \succ Y^\top P Y, \;\; \forall Y \in \mathcal{Y}(\mathcal{W}), \quad \textit{then} \tag{2.45}$$

the equilibrium point $z = 0$ of (2.9) is β'-stable, with $\beta = 0$.

2.2.3.4 Improvements and further reading

The discrete-time method presented above is one of many methods that exist in literature within the same framework. Decreasing the conservativeness of this approach could be possible at two levels:

1. *Alternatives to Theorem 9*: Although Theorem 9 gives necessary and sufficient conditions for stability, however the trade-off between the numerical complexity and the tightness of the derived LMIs later on, which gives only sufficient conditions, may lead to conservative results more than if we considered from the beginning other conditions for stability. In this scope we cite work in literature that analyze *the joint spectral radius* [AJPR14], check the existence of certain *Lyapunov functions* [HB10, DB01, AP08], or checks *set invariance* [FM16].
2. *Alternatives to Theorem 10*: Theorem 10 presents sufficient conditions for stability based on a quite tight approximation of the transition matrix Λ. On the other hand such an approximation becomes complex as the dimension of the system increases and as the required error tolerance ϵ decreases. Improvements to this approach suggests to divide the interval $[\underline{h}, \overline{h}]$ into several subintervals and apply the embedding procedure locally [HDTP13]. Comparisons of this method with the Cayley Hamilton, Jordan normal form, and gridding and norm bounding approximation-methods are given in [HVDWG$^+$10]. On the other hand, one may use sum of squares to approximate the transition matrix [BMH12] or other norm bounded criteria as in [Fuj09, KW14].

2.2.4 Robust control stability approach

This approach views the sampling error as a perturbation with respect to a nominal continuous-time control loop. In the following we present the basic idea of the interconnected system reformulation, give small gain stability conditions, and present numerical solutions for the stability of an aperiodic sampled-data system (2.9) under timing contract (2.10).

$$\begin{aligned} 0 \ &\leq t_0^s, \\ h_k = t_{k+1}^s - t_k^s &\in (0, \overline{h}], \quad \forall k \in \mathbb{N}. \end{aligned} \tag{2.10 revisited}$$

2.2.4.1 Reformulation

Let us rearrange equation (2.9)

$$\dot{z}(t) = A_K z(t) + B_K \vartheta(t) \tag{2.46a}$$

$$\vartheta(t) = \Delta_r \dot{z}(t). \tag{2.46b}$$

The operator $\Delta_r : \mathcal{L}_2^p[0,\infty) \to \mathcal{L}_2^p[0,\infty)$ is defined as

$$\Delta_r \eta = -\int_{t_k}^{t} \eta(s) ds, \tag{2.47}$$

for all $t \in [t_k, t_{k+1})$, $k \in \mathbb{N}$, $A_K = A + BK$, and $B_K = BK$.

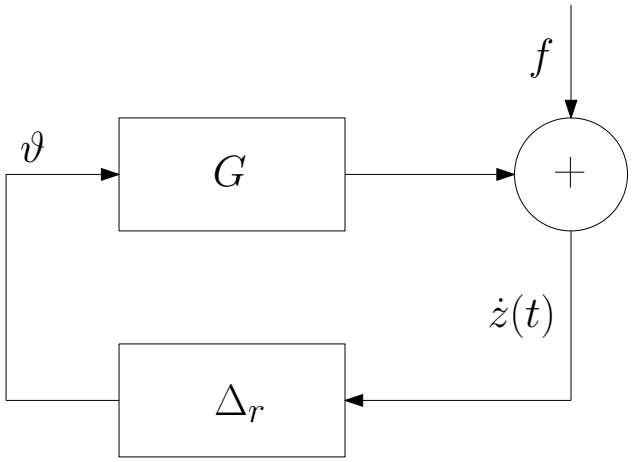


Figure 2.5: Representation of the interconnected system (2.48).

2.2.4.2 Theoretical foundation

The main theoretical tool used in this approach is the *small gain theorem* [ZDG+96]. But in order to apply the *small gain theorem*, we rewrite (2.46), after some derivations[3], in the form of the interconnected system expressed by Figure 2.5:

$$\dot{z}(t) = G\vartheta(t) + f(t) \tag{2.48a}$$

$$\vartheta(t) = \Delta_r \dot{z}(t). \tag{2.48b}$$

[3]The solution of (2.46) could be written as: $z(t) = e^{A_K t} z_0 + \int_0^t e^{A_K (t-s} B_K \vartheta(s) ds$. Then, $\dot{z}(t) = A_K e^{A_K t} z_0 + \frac{d}{dt} \int_0^t e^{A_K (t-s} B_K \vartheta(s) ds$.

where $f(t) = A_K e^{A_K t} z_0$ and the operator $G : \mathcal{L}_2^p[0,\infty) \to \mathcal{L}_2^p[0,\infty)$ is defined by

$$G\vartheta(t) = \frac{d}{dt}\int_0^t e^{A_K(t-\tau)} B_K \vartheta(\tau) d\tau. \tag{2.49}$$

Then we define the following notion of stability for interconnected systems:

Definition 6. *($\mathcal{L}_2$ stability) The interconnected system shown in Figure 2.5 is $\mathcal{L}_2$ stable if there exists a positive scalar C such that*

$$\int_0^t \left(|\dot{z}(\theta)|^2 + |\vartheta(\theta)|^2\right) d\theta \leq C \int_0^t |f(\theta)|^2 d\theta.$$

Finally, a consequence of the *small gain theorem* is the following:

Theorem 11. *[Fuj09] Consider the interconnected system (2.48) and the following:*

1. $\|G\|_2 \|\Delta_r\|_2 < 1$[4]*;*
2. *(2.48) is $\mathcal{L}_2$ stable;*
3. *(2.9)-(2.10) is β'-stable with, $\beta = 0$.*

Suppose that A_K is Hurwitz then 1) $\Rightarrow$ 2) $\Rightarrow$ 3).

2.2.4.3 Practical condition

Based on Theorem 11, Mirkin [Mir07] provided the following practical stability verification conditions:

Theorem 12. *(adapted from [Mir07]) Interconnected system (2.48) is $\mathcal{L}_2$-stable if*

1. *$\|G(s)\|_2 < \frac{\pi}{2\overline{h}}$, with $G(s) = s(sI - A_K)^{-1}B_K$ as the transfer function representing the operator G given by (2.49).*
2. *there exist $X, Y \succ 0$ such that:*

$$\begin{pmatrix} XA_K + A_K^\top X & \frac{2}{\pi}\overline{h}XB_K & A_K^\top Y \\ * & -Y & \frac{2}{\pi}\overline{h}B_K^\top Y \\ * & * & -Y \end{pmatrix} \prec 0; \tag{2.50}$$

where $*$ represents elements induced by symmetry.

Condition 1 could be verified by plotting the Bode diagram corresponding to $G(s)$ and verifying graphically that $\|G(s)\|_2 < \frac{\pi}{2\overline{h}}$. As for condition 2 it could be verified using existing LMI tools.

[4]Given an operator $G : \mathcal{L}_2^p[0,\infty) \to \mathcal{L}_2^p[0,\infty)$ its induced $\mathcal{L}_2$ norm is defined as $\|G\|_2 = sup_{u\neq 0}\frac{\|Gu\|_2}{\|u\|_2}$ where $u \in \mathcal{L}_2^p[0,\infty)$ and $\|u\|_2 = \left(\int_0^\infty |u(t)|^2 dt\right)^{\frac{1}{2}}$

Table 2.1: Methods that can solve instances of Problem 1 with description of the modeling and computational approaches, list of restrictions and possible extensions not included in the thesis.

	Models	Algorithm	Restrictions	Extensions
[CHVDW$^+$10]	difference inclusion	LMI	–	$\tau_k > h_k$; controller synthesis
[DHVDWH11]		LMI	–	scheduling
[HDTP13]		LMI	$\underline{\tau} = \overline{\tau} = 0$	controller synthesis
[HKPR11]		LMI	$\underline{\tau} = \overline{\tau} = 0$	–
[SP13]		SOS	$\underline{\tau} = \overline{\tau} = 0$	–
[FM14]		Invariance	$\underline{\tau} = \overline{\tau} = 0$	–
[LFH15]	time-delay systems	LMI	$\underline{h} = 0$	$\tau_k > h_k$; scheduling
[GMCL10]		LMI	$\underline{h} = \overline{h}$, $\underline{\tau} = 0$	controller synthesis; quantization
[LSF10]		LMI	$\underline{\tau} = \overline{\tau} = 0$	–
[Fuj09]	interconnected systems	LMI	$\underline{h} = \underline{\tau} = \overline{\tau} = 0$	–
[BMH12]	hybrid systems	SOS	–	nonlinear dynamics; scheduling
[HTVdWN10]		LMI	$\underline{\tau} = 0$, $\underline{h} = 0$	scheduling

Remark 3. *(Improvements) For the case of linear systems (2.9)-(2.10) the properties of Δ_r, given by (2.48), can be exploited in the framework of integral quadratic constraints (IQC) [MR97]. Some recent results are published in [KW14].*

After going through the four different frameworks that solve Problem 1 for the case of aperiodic sampled-data systems with no sampling to actuation delay, let us summarize in a table the literature that extends the discussed approaches to solve instances of Problem 1.

2.2.5 Extensions solving instances of Problem 1

Several approaches are developed within the modeling frameworks tackled in the previous sections (time-delay systems, hybrid systems, difference inclusions, or interconnected systems) to solve instances of Problem 1. A non-exhaustive list is given in Table 2.1. On the computational side, most of the approaches are based on semi-definite programming using either Linear Matrix Inequalities (LMI) or Sum Of Squares (SOS) formulations. This makes a clear distinction with our approach which relies on reachability analysis. Let us remark that only a few approaches [CHVDW$^+$10, DHVDWH11, BMH12] appear to be able to address all instances of Problem 1. It is noticeable that [CHVDW$^+$10, DHVDWH11] have been implemented in the Networked Control Systems (NCS) toolbox [BvLD$^+$12] whose results will be compared to those of our approach. We should also acknowledge that some of these approaches are able to handle problems that we do not consider in the present work (possibility of having $\tau_k > h_k$, controller synthesis, scheduling protocols, quantization, nonlinear dynamics).

2.3 Related work to the scheduling problem

In the first part of this section, we examine basic real-time scheduling techniques, using single processors, for tasks with fixed timing parameters (execution time, period, deadline,). Also, we extend the discussion to advanced scheduling techniques with multi-processors and for tasks with varying timing parameters. The reader is referred to an extended overview of this work in [AdNLR17].

In the second part, we summarize the existing work in literature that uses timed automata to solve scheduling problems.

2.3.1 Basic real-time scheduling

We consider in this section scheduling problems on a single processor of real-time control tasks T defined by a worst-case execution time (WCET) C, fixed period h, and deadline D (referring to the maximum time allowed for the task to finish its execution), i.e.

$$T = (C, h, D). \tag{2.51}$$

In the sequel, T executes periodically, where each of these periodic executions is known as a *job*, and therefore $D = h$. We denote by U the utilization, or the proportion of the processor that a task uses over time, and it is computed as:

$$U = \frac{C}{h}. \tag{2.52}$$

Also, we denote the set of N tasks, or task-set, and the total utilization of the task-set by $\mathcal{T} = \{T_1, \ldots, T_N\}$ and $\mathcal{U}$ respectively, where $T_i = (C_i, h_i, D_i)$, $i \in \mathbb{N}_{[1,N]}$, and

$$\mathcal{U} = \sum_{i \in \mathbb{N}_{[1,N]}} \frac{C_i}{h_i}. \tag{2.53}$$

Two types of priority assignments scheduling policies are studied next: fixed-priority assignments and dynamic-priority assignments.

2.3.1.1 Fixed-priority assignment

In this type of policy, we assign priorities to tasks at the design time and all their jobs inherit the same priority. In other words, a task's job delays or *preempt* all other tasks' jobs once it arrives knowing that the former task has a higher priority than the preempted ones. Then the task-set is said to be schedulable based on three different tests.

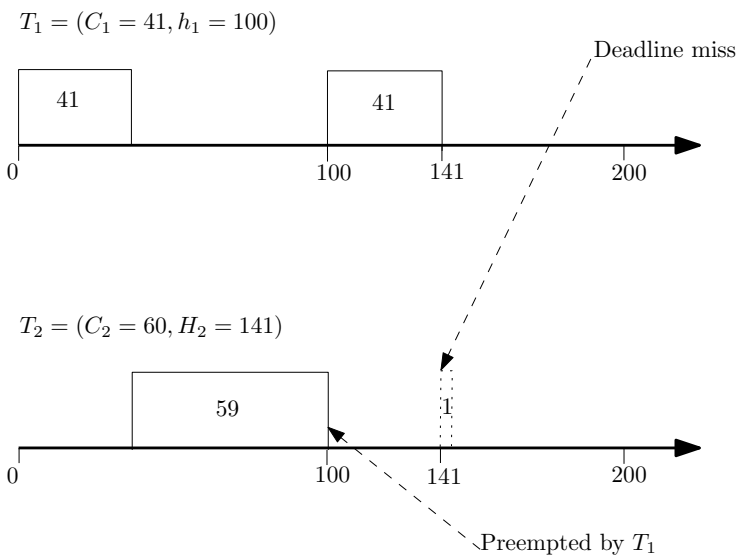


Figure 2.6: Suboptimal job priority assignment in RM scheduling.

Theorem 13. *(adapted from [LL73, JP86]) Consider a task-set* $\mathcal{T} = \{T_1, \ldots, T_N\}$ *where* T_i *is given by (2.51),* T_i *has a higher priority than* T_j *for all* $i < j$*, and* $i, j \in \mathbb{N}_{[1,N]}$*. Then* $\mathcal{T}$ *is schedulable if one of these conditions is true:*

1. $\mathcal{U} \leq \ln 2$.
2. $\mathcal{U} \leq N(2^{\frac{1}{N}} - 1)$.
3. *for all* $i \in \mathbb{N}_{[1,N]}$ *there exists* $k_i \in \mathbb{N}^* : (R_i^{k_i} = R_i^{[k_i-1]}) \wedge (R_i^{k_i} < h_i)$*, where*

$$R_i^0 = C_i;$$
$$R_i^{k_i} = C_i + \sum_{j<i} \left\lceil \frac{R_i^{k_i-1}}{h_j} \right\rceil .$$

Condition 1 is called an absolute bound on processor utilization and condition 2 it is a parametrized bound since it depends on N, the number of tasks. As for condition 3, it is a response time test that determines the worst-case finish time of a task. This test evaluates for every task the recurrence equation until convergence, i.e. $R_i^{k_i} = R_i^{[k_i-1]}$, or till $R_i^{k_i} > h_i$ which implies that T_i is not schedulable.

Last it is worth to note that the optimal fixed-priority assignment for tasks with implicit deadlines is the rate-monotonic (RM) priority assignment where tasks with higher rates are given higher priorities. Also, the scheduling policy discussed above could be generalized for tasks with $D < h$ [ABRW91].

2.3.1.2 Dynamic-priority assignment

In dynamic-priority assignment, we assign the priorities to the jobs when they arrive. Figure 2.6 shows that using a fixed priority assignment, task-set $\mathcal{T} = \{T_1, T_2\}$ is not schedulable where T_1 has a higher priority. However, with dynamic priority assignment, when $t = 100$ the first executing job in the second task could be assigned a higher priority than the arrived second job of the first task, therefore avoiding a deadline miss. This kind of policy is known as earliest deadline first (EDF) assignment and which is the optimal dynamic priority assignment policy. Thus a sufficient condition on schedulability is given next.

Theorem 14. *(Theorem 7 in [LL73]) In the case of EDF, a task-set is schedulable if and only if its utilization is less than or equal* 1*.*

2.3.2 Multiprocessor scheduling and advanced issues

In this section, we discuss global and partitioned multiprocessor scheduling as well as recent work on scheduling of tasks with varying timing parameters.

When a task-set executes on a platform with several CPUs it becomes more relevant to consider the multiprocessor scheduling problem. Two main forms of scheduling exists in this framework: global and partitioned scheduling. We focus here on scheduling with preemption of implicit-deadline tasks as the reader is referred to [DB11] for tasks with explicit or arbitrary deadlines.

2.3.2.1 Global multiprocessor scheduling

In global scheduling, the $J \in \mathbb{N}$ given CPUs share a single queue of jobs that are ready for execution. Then at each time, at maximum the highest J priority jobs are selected to execute on the processors. Such a priority could be assigned based on the task static priority scheduling or job static priority.

In the former each task is assigned a priority and each job inherit the priority of the task. An interesting aspect here is that RM scheduling is no more the optimal policy as noted for single processor scheduling where the best algorithm in this case is found in [And08]. In the latter, each job is assigned a fixed priority, where EDF is also no more the optimal priority assignment as in the case of single processor where a better assignment is cited from [SA02].

2.3.2.2 Partitioned multiprocessor scheduling

The task-set, in partitioned scheduling, is partitioned and each of these partitions is assigned to a fixed CPU based on task-static or job-static priority scheduling. In the former, each task is assigned to a processor and

cannot migrate during runtime. Then for each processor task-static priorities are used for scheduling the task-set. As for the assignments of tasks we can either handle tasks one by one and assign the currently considered task to the processor that has the lowest utilization or build assignment schemes based on the two ideas: Assign a task to a processor on which other tasks have been assigned and use a uniprocessor schedulability test to ensure that after a task has been assigned the task-set on each processor is schedulable. For job-static priority scheduling the same reasoning works except that when assigning tasks for processors we use schedulability tests for uniprocessor scheduling algorithms like EDF. Then in the latter case if the sum of utilization of all tasks on a processor does not exceed 1, the task-set on this processor is schedulable.

Other algorithms also exist in the literature for multiprocessor scheduling where jobs can migrate to any processor at any time [RLM+11] or task splitting is used [AT06].

2.3.2.3 Tasks with varying timing parameters

So far the scheduling problem was discussed for tasks with fixed timing parameters. Recently, theories were developed for tasks having timing requirements that vary as a function of the physical process; this is the case in modern control tasks. Given that this area is not yet well developed the reader is referred to some work where schedules are synthesized on a single processor for task-sets containing event-triggered control tasks [Tab07] or tasks whose timing parameters vary according to the state of the control system [KLR12]. In this context, we illustrate in Chapter 4 an approach to synthesize schedules; without preemption for control tasks that have varying timing requirements. These timing requirements are defined by a worst and best case execution time in addition to a timing contract (2.4) that defines implicit deadlines the control tasks must guarantee.

2.3.3 Scheduling with timed automata

Scheduling with timed automata is examined in literature [DLG10, DILS09, Feh99, MAT09, AAM+06] where in [Feh99] an application on a steel plant is studied and [AAM+06] shows how efficient shortest path algorithms for timed automata can find optimal schedules for the classical job-shop problem. In the latter tasks are characterized in a different manner to (2.6), i.e. each task is characterized by its duration, by the resources it needs in order to execute and by precedence relationships it has with other tasks. In that case, a scheduler has to resolve conflicts, arising between two or more tasks when they simultaneously demand access for a resource exceeding its availability, by deciding to which of the competing tasks to give the resource first and which tasks will have to wait until the resource is released.

Another study of a scheduling problem is depicted in [MAT09] where the problem is reformulated in terms of timed game automata [CDF$^+$05] and a safety game. Although, therein the approach has the advantage of employing event-triggered controllers and eventually improves the resource utilization over the network, however it can only solve instances of Problem 2 where $t_k^{s_i} = t_k^{b_i}$, $t_k^{e_i} = t_k^{a_i}$ and $\underline{\tau} = \overline{\tau} = \overline{c}^i = \underline{c}^i = c \in \mathbb{R}^+$ for all $k \in \mathbb{N}$ and $i \in \mathbb{N}_{[1,N]}$. In our approach, presented in Chapter 4, we use similar ideas to solve Problem 2. Precisely, we reformulate the latter as a safety game, given in terms of a network of timed game automaton and a set of undesired locations, which is then solved using the tool UPPAAL-TIGA [BCD$^+$07].

Chapter 3

Stability verification: an approach based on difference inclusions and reachability analysis

Abstract

We report a significant mathematical theory to analyze the stability of modern control systems operating under timing contracts. Reformulating the system into an impulsive one, the proposed work verifies stability of such systems with bounded timing uncertainties using safe convex approximation techniques and new generalized results for the problem on a class of systems modeled in the framework of difference inclusions. Extensions, published in [AKGD15, AKGD17b], into self-triggered control and stability verification under stochastic timing contracts are discussed at the end. In the first extension we design the sampling strategy in a sampled-data system where the state at the sampling instants is only required to be known which results in less intensive on-line computations. Sufficient stability conditions are presented in the second extension, for stochastic impulsive linear systems where the random durations between resets are independent and identically distributed. The main results of this chapter, which are published in [AKGD16a, AKGD16b], are evaluated on several systems found in literature where comparisons with existing results show that our stability verification algorithm is promising and competitive in providing tight bounds for uncertainties in the timing contract for stable systems and also in its execution time.

Solving the stability verification problem or Problem 1 is our main concern in this chapter. In the sequel, we solve the same problem for a more general class of dynamic systems, given by a difference inclusion, and conclude on the stability of system (2.1)-(2.2). Meanwhile, one essential contribution that is used in our study is the approximation schemes developed for over-approximating the reachable set of (2.1)-(2.2) from a given initial set. Then for the sake of clarity, we explain first these schemes in Section 3.1 and provide some examples so that the reader could follow with the approach introduced in Section 3.2. At the end of

the chapter, we extend the work to related topics that had to remain on the sidelines for the purposes of this thesis: stability verification under stochastic timing contracts and self-triggered control.

3.1 Reachability analysis

In the last decade, hybrid system reachability has had an important breakthrough in computing the reachable set corresponding to a linear continuous dynamics where the developed algorithms are based on representing the reachable sets by ellipsoids [KV00, BT00], zonotopes [Gir05, ASB10] or by support functions [LGG10, FLGD$^+$11]. In the following we take advantage of the reachable sets constructed for continuous linear time invariant (LTI) systems [LG09], and which we introduce in the next section, in order to provide an efficient and accurate algorithmic scheme to compute the reachable sets for system (2.1)-(2.2). In the following two sections we over-approximate first the reachable set corresponding to the case of Nearly Periodic Impulsive Linear Systems (NPILS) given by (2.1)-(2.2), where $\underline{\tau} = \overline{\tau} = 0$, then that which corresponds to the general timing contract (2.2).

3.1.1 Case of continuous LTI systems

In this section, we summarize the approach followed in [LG09] to construct reachable sets for autonomous continuous linear time invariant systems. But let us first define the following notations. Given a real matrix $A \in \mathbb{R}^{n\times n}$, $|A|$ is the matrix whose elements are the absolute values of the elements of A. Given $\mathcal{S} \subseteq \mathbb{R}^n$ and a real matrix $A \in \mathbb{R}^{n\times n}$, the set $A\mathcal{S} = \{x \in \mathbb{R}^n : (\exists y \in \mathcal{S} : x = Ay)\}$; for $a \in \mathbb{R}$, $a\mathcal{S} = (aI_n)\mathcal{S}$ where I_n is the $n \times n$ identity matrix. The interval hull of $\mathcal{S}$ is the smallest n-dimensional interval containing the set $\mathcal{S}$ and is denoted by $\square(\mathcal{S})$. The symmetric interval hull of $\mathcal{S}$ is the smallest symmetric (with respect to 0) n-dimensional interval containing $\mathcal{S}$ and is denoted by $\boxdot(\mathcal{S})$. Given $\mathcal{S}, \mathcal{S}' \subseteq \mathbb{R}^n$, the Minkowski sum of $\mathcal{S}$ and $\mathcal{S}'$ is $\mathcal{S} \oplus \mathcal{S}' = \{x + x' : x \in \mathcal{S}, x' \in \mathcal{S}'\}$.

We intend to over-approximate the reachable set defined as follows:

Definition 7. *Given a continuous-time dynamical system*

$$\dot{x}(t) = Ax(t),\ t \in \mathbb{R}_0^+,\ x(t) \in \mathbb{R}^n \tag{3.1}$$

the reachable set on $[t, t'] \subseteq \mathbb{R}_0^+$ *from the set* $\mathcal{S} \subseteq \mathbb{R}^n$ *is*

$$\mathcal{R}^A_{[t,t']}(\mathcal{S}) = \bigcup_{\tau\in[t,t']} e^{\tau A}\mathcal{S}. \tag{3.2}$$

After remarking that $\mathcal{R}^A_{[t,t']}(\mathcal{S}) = e^{tA} \bigcup_{\tau\in[0,t'-t]} e^{\tau A}\mathcal{S}$, we state the following result from [LG09] which gives an over-approximation scheme for the reachable set given by (3.2).

Theorem 15. *[LG09] Let $T \in \mathbb{R}^+$, $A \in \mathbb{R}^{n\times n}$, $\mathcal{S} \in \mathcal{K}(\mathbb{R}^n)$ and $N \in \mathbb{N}^+$, let*

$$\overline{\mathcal{R}}^A_{[0,T]}(\mathcal{S}) = \bigcup_{i=1}^{N} \overline{\mathcal{R}}^A_{[(i-1)\delta,i\delta]}(\mathcal{S})$$

where $\delta = T/N$ is the time step, and $\overline{\mathcal{R}}^A_{[(i-1)\delta,i\delta]}(\mathcal{S})$ is defined by the recurrence equation:

$$\begin{aligned} \overline{\mathcal{R}}^A_{[0,\delta]}(\mathcal{S}) &= ch(\mathcal{S}, e^{\delta A}\,\mathcal{S}) \,\oplus\, 1/4\;\epsilon_\delta(\mathcal{S}), \\ \overline{\mathcal{R}}^A_{[i\delta,(i+1)\delta]}(\mathcal{S}) &= e^{\delta A}\,\overline{\mathcal{R}}^A_{[(i-1)\delta,i\delta]}(\mathcal{S}),\; i \in \mathbb{N}_{[1,N-1]} \end{aligned} \tag{3.3}$$

with

$$\epsilon_\delta(\mathcal{S}) = \square(|A|^{-1}(e^{\delta|A|} - I) \boxdot (A(I - e^{\delta A})\mathcal{S})) \oplus \square(|A|^{-2}(e^{\delta|A|} - I - \delta|A|) \boxdot (A^2 e^{\delta A}\mathcal{S})).$$

Then, $\mathcal{R}^A_{[(i-1)\delta,i\delta]}(\mathcal{S}) \subseteq \overline{\mathcal{R}}^A_{[(i-1)\delta,i\delta]}(\mathcal{S})$, for all $i \in \mathbb{N}_{[1,N]}$ and $\mathcal{R}^A_{[0,T]}(\mathcal{S}) \subseteq \overline{\mathcal{R}}^A_{[0,T]}(\mathcal{S})$.

In conclusion, Theorem 15 suggests that an over-approximation of the reachable set, of the autonomous continuous linear time invariant system (3.1), from $\mathcal{S}$ in the time interval $[0, T]$ is given by $\overline{\mathcal{R}}^A_{[0,T]}(\mathcal{S})$. As for the practical computation of such an over-approximation many set representations could be used for the implementation such as zonotopes [Gir05], support functions [LGG10], etc.

3.1.2 Case of Nearly Periodic Impulsive Linear Systems (NPILS)

In the case of NPILS we are not considering sampling to actuation delays, then system (2.1a-2.1b) could be rewritten as

$$\dot{z}(t) = Az(t) + Bu(t) \tag{2.9a revisited}$$

$$z(t) = Kz(t^s_k) \quad t^s_k < t \le t^s_{k+1}, \tag{2.9b revisited}$$

where the initial state is given as $z(0) = z_0$.

Consequently, the impulsive system reformulation of (2.9) is given, after setting $e(t) = z(\alpha(t)) - z(t)$ with $\alpha(t) = t^s_k$ for $t^s_k < t \le t^s_{k+1}$, by:

$$\begin{aligned} \dot{x}(t) &= A'_c x(t) \quad t \neq t^s_k \\ x(t^{s+}_k) &= A'_s x(t^s_k), \end{aligned} \tag{3.4}$$

where

$$A'_c = \begin{pmatrix} A+BK & BK \\ -A-BK & -BK \end{pmatrix},\ A'_s = \begin{pmatrix} I_{n'} & 0 \\ 0 & 0 \end{pmatrix},\ x(t) = \begin{pmatrix} z(t) \\ e(t) \end{pmatrix}.$$

We then have $A'_c, A'_s \in \mathbb{R}^{n'\times n'}$ and the state of the impulsive system (3.4) $x \in \mathbb{R}^{n'}$, with $n' = 2p$ and $x(0) = x_0$. Moreover, the timing contract (2.2) is simplified to

$$\begin{aligned} 0 \ &\leq t_0^s, \\ h_k = t_{k+1}^s - t_k^s &\in [\underline{h}, \overline{h}], \quad \forall k \in \mathbb{N}. \end{aligned} \tag{2.24 revisited}$$

In this section we compute an over-approximation of the reachable set of the NPILS (3.4)-(2.24) starting from an initial set $\mathcal{S} \subseteq \mathbb{R}^{n'}$ that represents all the states of the system at sampling instant t_k^s. The exact reachable set at instant t_{k+1}^s is given by the following:

$$\Phi_{NPILS}(\mathcal{S}) = \bigcup_{h\in[\underline{h},\overline{h}]} e^{hA'_c}A'_s\mathcal{S} = \mathcal{R}^{A'_c}_{[\underline{h},\overline{h}]}(A'_s\mathcal{S}) = \mathcal{R}^{A'_c}_{[0,\overline{h}-\underline{h}]}(e^{\underline{h}A'_c}A'_s\mathcal{S}). \tag{3.5}$$

Using Theorem 15 in Section 3.1.1 it appears that an over-approximation of Φ_{NPILS} can be given by

$$\Phi_{NPILS}(\mathcal{S}) \subseteq \overline{\mathcal{R}}^{A'_c}_{[0,\overline{h}-\underline{h}]}(e^{\underline{h}A'_c}A'_s\mathcal{S}). \tag{3.6}$$

This over-approximation is given by the union of N sets which may be quite impractical for subsequent manipulations. For that reason, it will be over-approximated by a single polytope.

Given a matrix $H \in \mathbb{R}^{r\times n'}$, let H_i, $i \in \mathbb{N}_{[1,r]}$ denote the row vectors of H. For a set $\mathcal{S} \subseteq \mathbb{R}^{n'}$, let us define the polytope $\Gamma_H(\mathcal{S}) = \{x \in \mathbb{R}^{n'} : Hx \leq b\}$ where $b_i = \sup_{x\in\mathcal{S}} H_i x$, $i \in \mathbb{N}_{[1,r]}$. In other words, $\Gamma_H(\mathcal{S})$ is the smallest polytope whose facets directions are given by H and containing $\mathcal{S}$. Let us remark that if $\mathcal{S}$ is bounded and if 0 is in the interior of $\mathrm{ch}(\{H_1, \ldots, H_r\})$, then $\Gamma_H(\mathcal{S})$ is bounded. In addition, if $\mathcal{S}$ is convex, then it can be approximated arbitrarily close by $\Gamma_H(\mathcal{S})$ by taking a sufficient number of facets directions $H_1, \ldots, H_r$. The over-approximation of Φ_{NPILS} is then given as follows:

Corollary 2. *Let the matrix $H \in \mathbb{R}^{r\times n'}$ be such that 0 is in the interior of $ch(\{H_1, \ldots, H_r\})$ and $\mathcal{S} \subseteq \mathbb{R}^{n'}$. Let $\overline{\Phi}_{NPILS}$ be given by*

$$\overline{\Phi}_{NPILS}(\mathcal{S}) = \Gamma_H\left(ch(\overline{\mathcal{R}}^{A'_c}_{[0,\overline{h}-\underline{h}]}(e^{\underline{h}A'_c}A'_s\mathcal{S}))\right), \tag{3.7}$$

where $\overline{\mathcal{R}}^{A'_c}_{[0,\overline{h}-\underline{h}]}(e^{\underline{h}A'_c}A'_s\mathcal{S})$ is computed as in Theorem 15. Then, $\Phi_{NPILS}(\mathcal{S}) \subseteq \overline{\Phi}_{NPILS}(\mathcal{S})$ and $\overline{\Phi}_{NPILS}(\mathcal{S})$ is bounded if $\mathcal{S}$ is bounded.

Proof: By Theorem 15, (3.6), and (3.7), we have that for all $\mathcal{S} \subseteq \mathbb{R}^{n'}$, $\Phi_{NPILS}(\mathcal{S}) \subseteq \overline{\Phi}_{NPILS}(\mathcal{S})$. If $\mathcal{S}$ is bounded, then $\mathrm{ch}\big(\overline{\mathcal{R}}^{A'_c}_{[0,\overline{h}-\underline{h}]}(e^{\underline{h}A'_c}A'_s\mathcal{S})\big)$ is bounded. Furthermore, since 0 is in the interior of $\mathrm{ch}(\{H_1,\ldots,H_r\})$, $\overline{\Phi}_{NPILS}(\mathcal{S})$ is bounded. □

3.1.3 Systems under the general contract

3.1.3.1 Reformulation using impulsive systems

In our analysis it is more practical to transform (2.1) into an impulsive system with two types of resets each referring to a sampling or actuation instant. Such a reformulation is convenient to develop stability conditions based on reachability analysis. The system is thus given by:

$$\begin{aligned} \dot{x}(t) &= A_c x(t), \quad t \neq t_k^s, t \neq t_k^a \\ x(t_k^{s+}) &= A_s x(t_k^s) \\ x(t_k^{a+}) &= A_a x(t_k^a), \end{aligned} \tag{3.8}$$

where $x(t) \in \mathbb{R}^n$ is the state of the system with $n = p+2m$, (t_k^s) and (t_k^a) are given by (2.2), $x(t^+) = \lim_{\tau\to 0,\tau>0} x(t+\tau)$, and

$$\begin{aligned} A_c &= \begin{pmatrix} A & 0 & B \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \; A_s = \begin{pmatrix} I_p & 0 & 0 \\ K & 0 & 0 \\ 0 & 0 & I_m \end{pmatrix}, \\ A_a &= \begin{pmatrix} I_p & 0 & 0 \\ 0 & I_m & 0 \\ 0 & I_m & 0 \end{pmatrix}, \; x(t) = \begin{pmatrix} z(t) \\ Kz(\theta^s(t)) \\ u(t) \end{pmatrix}, \end{aligned} \tag{3.9}$$

with $\theta^s(t) = t_k^s$ for $t \in (t_k^s, t_{k+1}^s]$.

We consider in the following, system (3.8) under timing contract (2.2).

$$\begin{aligned} 0 \; &\leq t_0^s, \\ t_k^s \leq t_k^a &\leq t_{k+1}^s, && \forall k \in \mathbb{N} \\ \tau_k = t_k^a - t_k^s &\in [\underline{\tau}, \overline{\tau}], && \forall k \in \mathbb{N} \\ h_k = t_{k+1}^s - t_k^s &\in [\underline{h}, \overline{h}]. && \forall k \in \mathbb{N} \end{aligned} \tag{2.2 revisited}$$

This section tends to over-approximate the reachable set of system (2.2)-(3.8) at the next sampling instant, t^s_{k+1}, supposing that $\mathcal{S} \subseteq \mathbb{R}^n$ represents all the states of the system at sampling instant t^s_k. The exact reachable set at instant t^s_{k+1} is given by the following:

$$\Phi_{gen}(\mathcal{S}) = \bigcup_{\tau \in [\underline{\tau}, \overline{\tau}]} \bigcup_{w \in [\max(0, \underline{h}-\tau), \overline{h}-\tau]} e^{wA_c} A_a e^{\tau A_c} A_s \mathcal{S}. \tag{3.10}$$

From (3.10), one can easily check that

$$\begin{aligned} \Phi_{gen}(\mathcal{S}) &\subseteq \mathcal{R}^{A_c}_{[\max(0,\underline{h}-\overline{\tau}),\overline{h}-\underline{\tau}]} \Big(A_a \mathcal{R}^{A_c}_{[\underline{\tau},\overline{\tau}]}(A_s \mathcal{S}) \Big) \\ &\subseteq e^{\max(0,\underline{h}-\overline{\tau})A_c} \mathcal{R}^{A_c}_{[0,\min(\overline{h}-\underline{\tau},\overline{h}-\underline{h}+\overline{\tau}-\underline{\tau})]} \Big(A_a e^{\underline{\tau} A_c} \mathcal{R}^{A_c}_{[0,\overline{\tau}-\underline{\tau}]}(A_s \mathcal{S}) \Big), \end{aligned} \tag{3.11}$$

which in turn can easily be over-approximated using the result of Theorem 15. In the case of NPILS, the previous inclusion becomes an equality. This is the approach followed in Section 3.1.2 to find $\overline{\Phi}_{NPILS}$. However, for the general timing contract (2.2), the coupling in the timing uncertainties w and τ in (3.10) is totally disregarded in (3.11) and thus leads to conservatism. Therefore, in this section, to reduce conservatism, we present a specific approximation scheme for Φ_{gen}, that takes into consideration the coupling in the timing uncertainties. It is based on the following result:

Lemma 4. *Let $\mathcal{S} \in \mathcal{B}(\mathbb{R}^n)$, let $N_1, N_2 \in \mathbb{N}^+$, then*

$$\Phi_{gen}(\mathcal{S}) \subseteq \bigcup_{j_1=1}^{N_1} \bigcup_{j_2=1}^{n_2(j_1)} e^{(\theta(j_1)+(j_2-1)\delta_2)A_c} \mathcal{R}^{A_c}_{[0,\delta_2]} \Big(A_a e^{(\underline{\tau}+(j_1-1)\delta_1)A_c} \mathcal{R}^{A_c}_{[0,\delta_1]}(A_s \mathcal{S}) \Big) \tag{3.12}$$

where for $j_1 \in \mathbb{N}_{[1,N_1]}$

$$\begin{aligned} \delta_1 &= (\overline{\tau} - \underline{\tau})/N_1 \\ \delta_2 &= \min(\overline{h} - \underline{\tau}, \overline{h} - \underline{h} + \delta_1)/N_2 \\ \theta(j_1) &= \max(0, \underline{h} - \underline{\tau} - j_1\delta_1) \\ n_2(j_1) &= \lceil \min(\overline{h} - \underline{\tau} - (j_1 - 1)\delta_1, \overline{h} - \underline{h} + \delta_1)/\delta_2 \rceil. \end{aligned} \tag{3.13}$$

Proof. From (3.10), it follows that

$$\begin{aligned}\Phi_{gen}(\mathcal{S}) &= \bigcup_{j_1=1}^{N_1} \bigcup_{\tau\in[\underline{\tau}+(j_1-1)\delta_1,\underline{\tau}+j_1\delta_1]} \bigcup_{w\in[\max(0,\underline{h}-\tau),\overline{h}-\tau]} e^{wA_c}A_a e^{\tau A_c}A_s\mathcal{S}\\ &\subseteq \bigcup_{j_1=1}^{N_1} \mathcal{R}^{A_c}_{[\theta(j_1),\overline{h}-\underline{\tau}-(j_1-1)\delta_1]}\Big(A_a\mathcal{R}^{A_c}_{[\underline{\tau}+(j_1-1)\delta_1,\underline{\tau}+j_1\delta_1]}(A_s\mathcal{S})\Big)\\ &\subseteq \bigcup_{j_1=1}^{N_1} e^{\theta(j_1)A_c}\mathcal{R}^{A_c}_{[0,\overline{h}-\underline{\tau}-(j_1-1)\delta_1-\theta(j_1)]}\Big(A_a e^{(\underline{\tau}+(j_1-1)\delta_1)A_c}\mathcal{R}^{A_c}_{[0,\delta_1]}(A_s\mathcal{S})\Big).\end{aligned}$$

Remarking that

$$\overline{h}-\underline{\tau}-(j_1-1)\delta_1-\theta(j_1)=\min(\overline{h}-\underline{\tau}-(j_1-1)\delta_1,\overline{h}-\underline{h}+\delta_1)$$

one gets

$$\Phi_{gen}(\mathcal{S}) \subseteq \bigcup_{j_1=1}^{N_1}\bigcup_{j_2=1}^{n_2(j_1)} e^{\theta(j_1)A_c}\mathcal{R}^{A_c}_{[(j_2-1)\delta_2,j_2\delta_2]}\Big(A_a e^{(\underline{\tau}+(j_1-1)\delta_1)A_c}\mathcal{R}^{A_c}_{[0,\delta_1]}(A_s\mathcal{S})\Big)$$

which leads to (3.12). □

Remark 4. *N_1 and N_2 are parameters used to discretize time intervals. For $N_1 = N_2 = 1$, the over-approximation given by (3.12) is the same as the one in (3.11).*

We now present our over-approximation scheme for Φ_{gen}:

Theorem 16. *Let $\mathcal{S} \in \mathcal{B}(\mathbb{R}^n)$, $N_1, N_2 \in \mathbb{N}^+$, and $H \in \mathbb{R}^{r\times n}$, such that $0 \in int(ch(\{H_1,\ldots,H_r\}))$, let $\overline{\Phi}_{gen} : \mathcal{B}(\mathbb{R}^n) \to \mathcal{B}(\mathbb{R}^n)$ be given by*

$$\overline{\Phi}_{gen}(\mathcal{S}) = \Gamma_H\left(ch\left(\bigcup_{j_1=1}^{N_1}\bigcup_{j_2=1}^{n_2(j_1)} e^{(\theta(j_1)+(j_2-1)\delta_2)A_c}\overline{\Phi}_{j_1}(\mathcal{S})\right)\right)$$

where for $j_1 \in \mathbb{N}_{[1,N_1]}$,

$$\overline{\Phi}_{j_1}(\mathcal{S}) = \overline{\mathcal{R}}^{A_c}_{[0,\delta_2]}\Big(A_a e^{(\underline{\tau}+(j_1-1)\delta_1)A_c}\overline{\mathcal{R}}^{A_c}_{[0,\delta_1]}(A_s\mathcal{S})\Big)$$

with δ_1, δ_2, $\theta(j_1)$, $n_2(j_1)$ given by (3.13) , and $\overline{\mathcal{R}}^{A_c}_{[0,\delta_1]}$, $\overline{\mathcal{R}}^{A_c}_{[0,\delta_2]}$ computed as in (3.3). Then, $\Phi_{gen}(\mathcal{S}) \subseteq \overline{\Phi}_{gen}(\mathcal{S})$.

Proof. The proof is straightforward from Theorem 15 and Lemma 4. □

Remark 5. *In the previous results, the operation Γ_H is not necessary to guarantee over-approximation of Φ_{gen} and Φ_{NPILS}. On the other hand, without this operation, the over-approximation of Φ_{gen} and Φ_{NPILS}*

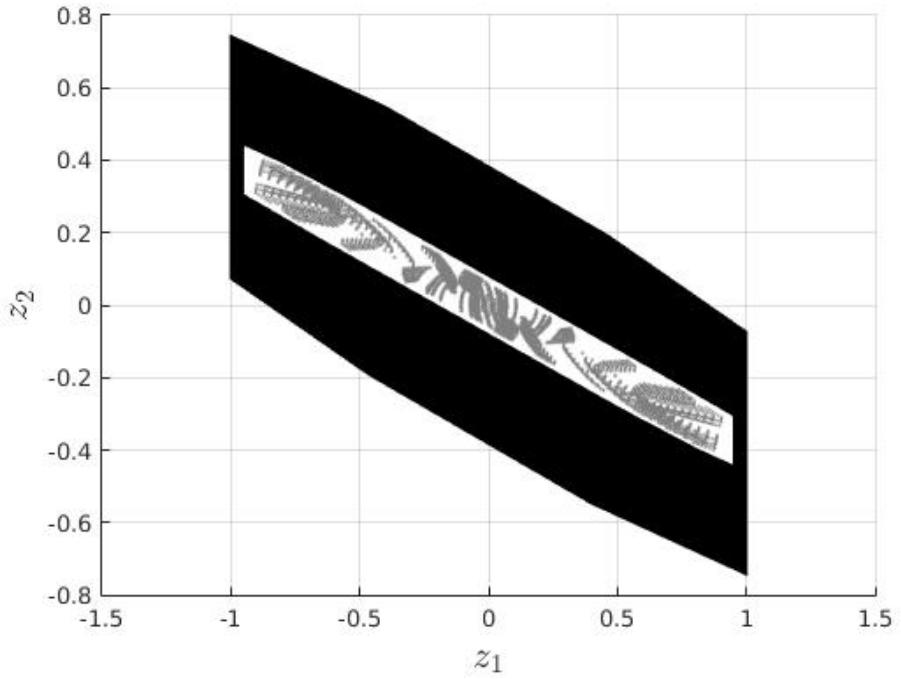


Figure 3.1: Sampled points of $\Phi_{gen}(\mathcal{S})$ (in grey), over-approximation $\overline{\Phi}_{gen}(\mathcal{S})$ given by Theorem 16 and over-approximation of (3.11) computed using Theorem 15 (in black).

would be impractical for subsequent manipulations. For that reason, this union is over-approximated by the smallest enclosing polytope whose facets direction are given by a matrix H. Moreover, if $\mathcal{S}$ is a polytope, then using the properties of support functions [LGG10], the computation of $\overline{\Phi}_{gen}(\mathcal{S})$ and $\overline{\Phi}_{NPILS}$ reduces to solving a set of linear programs.

We illustrate the tightness of our new approximation scheme using system (3.29) (see Section 6) with the timing contract given by $\underline{\tau} = 0$, $\overline{\tau} = 0.4$, $\underline{h} = 0.2$, $\overline{h} = 1.2$. We consider a polytope $\mathcal{S}$ defined by a matrix H with 44 rows. Figure 3.1 shows sampled points (in grey) from $\Phi_{gen}(\mathcal{S})$. The white polytope corresponds to the over-approximation $\overline{\Phi}_{gen}(\mathcal{S})$ given in Theorem 16 with $N_1 = 20$ and $N_2 = 50$. The black polytope is given by (3.11) over-approximated using Theorem 15 where $\overline{\mathcal{R}}^{A_c}_{[0,\overline{\tau}-\underline{\tau}]}$ and $\overline{\mathcal{R}}^{A_c}_{[0,\min(\overline{h}-\underline{\tau},\overline{h}-\underline{h}+\overline{\tau}-\underline{\tau})]}$ are computed with $N = 20$ and $N = 50$ respectively. One can check that the over-approximation given by Theorem 16 is quite tight and much less conservative than that given by (3.11). Keeping in mind that the reachability computations are used in the following section, we state now our main stability verification approach.

3.2 Main stability approach

Lets first rewrite Problem 1, in the form of impulsive systems, before stating the main results that solve it.

A notion for stability of the system that guarantees the exponential convergence of the state to the origin with a predefined rate $\beta \in \mathbb{R}^+$ is given by:

Definition 8 (β^*-stability)**.** *Let $\beta \in \mathbb{R}^+$, system (2.2)-(3.8) is β^*-stable if there exist $C \in \mathbb{R}^+$ and $\varepsilon^* \in \mathbb{R}^+$ such that:*

$$|x(t)| \leq Ce^{-(\beta+\varepsilon^*)(t-t_0^s)}|x_0|, \ \forall t \in \mathbb{R}^+. \tag{3.14}$$

Note that β^*-stability of system (2.2)-(3.8) is equivalent to the β'-stability of (2.1)-(2.2). We are now interested in verifying stability of embedded control systems in the form given by (3.8) under one of the general timing contracts defined previously in Section 2.1.1. Also note that we can easily show that system (3.8) under the ZET and LET contracts is stable if and only if the eigenvalues of the matrix $e^{hA_c}A_aA_s$ and $A_ae^{hA_c}A_s$ are inside the unit circle respectively. As for the DET or TOL contracts, we have that stability of system (2.1)-(2.2) is guaranteed by the stability of system (3.8)-(2.2) with an adequate choice of the timing contract parameters. It is noteworthy that in the case of the TOL contract stability of system (3.8-2.2) is only sufficient when the choice of the timing contract parameters are chosen as explained by the over-approximating contract in Section 2.1.1. Consequently, in this work, we consider an equivalent to Problem 1:

Problem 1*. *[Stability verification] Given $\beta \in \mathbb{R}^+$, A_c, A_s, $A_a \in \mathbb{R}^{n\times n}$, $(\underline{\tau}, \overline{\tau}, \underline{h}, \overline{h}) \in \mathcal{C}$, verify that (2.2)-(3.8) is β^*-stable.*

Our stability verification approach to solve Problem 1* is based on a reformulation of the linear impulsive systems (2.2)-(3.8) in the general framework of difference inclusions. Then, for a fairly large class of difference inclusions, we establish necessary and sufficient conditions for stability. These conditions are based on the successive images of a set under the dynamics of the difference inclusion, and generalize some previous conditions on the stability of discrete-time switched systems [LA09, AL14a]. For linear impulsive systems (2.2)-(3.8), these conditions allow us to design a stability verification algorithm using reachability analysis developed in Section 3.1.

Lets introduce first a general formulation based on difference inclusions and later show how linear impulsive systems in the form of (2.2)-(3.8) can be embedded in this framework.

3.2.1 Difference inclusions

We consider discrete-time dynamical systems modeled by the following difference inclusion:

$$\xi_{k+1} \in \Phi(\{\xi_k\}),\ k \in \mathbb{N} \tag{3.15}$$

where $\xi_k \in \mathbb{R}^n$ is the state of the system, and $\Phi : 2^{\mathbb{R}^n} \to 2^{\mathbb{R}^n}$ is a set-valued map. Stability for systems of the form (3.15) is considered in the following sense:

Definition 9 (GES)**.** *System (3.15) is globally exponentially stable (GES) if there exists $(C, \varepsilon) \in \mathbb{R}^+ \times (0, 1)$ such that for all trajectories $(\xi_k)_{k\in\mathbb{N}}$ of (3.15), we have*

$$|\xi_k| \leq C\varepsilon^k \, |\xi_0| \,,\ \forall k \in \mathbb{N}. \tag{3.16}$$

Next we verify the stability of a difference inclusion of the form (3.15). We make the following assumptions on the map Φ.

Assumption 2. *For all $\mathcal{S} \subseteq \mathbb{R}^n$, $\lambda \in \mathbb{R}_0^+$, the following assertions hold:*

(i) $\Phi(\mathcal{S}) = \bigcup_{z\in\mathcal{S}} \Phi(\{z\})$*;*

(ii) $\Phi(\lambda\mathcal{S}) \subseteq \lambda\Phi(\mathcal{S})$*;*

(iii) if $\mathcal{S}$ is bounded, then $\Phi(\mathcal{S})$ is bounded.

Under item (i) of Assumption 2, for all $\mathcal{S}$, $\mathcal{S}' \subseteq \mathbb{R}^n$, it follows that $\Phi(\mathcal{S} \cup \mathcal{S}') = \Phi(\mathcal{S}) \cup \Phi(\mathcal{S}')$. Also, if $\mathcal{S} \subseteq \mathcal{S}'$, then $\Phi(\mathcal{S}) \subseteq \Phi(\mathcal{S}')$. We define the iterates of Φ as $\Phi^0(\mathcal{S}) = \mathcal{S}$ for all $\mathcal{S} \subseteq \mathbb{R}^n$, and $\Phi^{k+1} = \Phi \circ \Phi^k$ for all $k \in \mathbb{N}$. Let $(\xi_k)_{k\in\mathbb{N}}$ be a trajectory of (3.15) such that $\xi_0 \in \mathcal{S}$, then under item (i) of Assumption 2, for all $k \in \mathbb{N}$, $\Phi^k(\mathcal{S})$ is the set of all possible values of ξ_k.

For some results of the paper, the following additional assumption related to the convexity of the map Φ is needed:

Assumption 3. *For all $\mathcal{S} \subseteq \mathbb{R}^n$, $\Phi(ch(\mathcal{S})) \subseteq ch(\Phi(\mathcal{S}))$.*

Then, the stability verification problem, for systems of the form (3.15), can be formulated as follows:

Problem 4 (Stability verification)**.** *Under Assumptions 2 and 3, verify that system (3.15) is GES.*

Let $\beta \in \mathbb{R}^+$, we define the map $\Phi : 2^{\mathbb{R}^n} \to 2^{\mathbb{R}^n}$, given for all $\mathcal{S} \subseteq \mathbb{R}^n$ by

$$\Phi(\mathcal{S}) = \bigcup_{\tau\in[\underline{\tau},\overline{\tau}]} \bigcup_{w\in[\max(0,\underline{h}-\tau),\overline{h}-\tau]} e^{(w+\tau)\beta} e^{wA_c} A_a e^{\tau A_c} A_s \mathcal{S}. \tag{3.17}$$

The following proposition establishes the equivalence between stability of systems (2.2)-(3.8) and (3.15).

Proposition 3. *Given $\beta \in \mathbb{R}^+$. System (2.2)-(3.8) is β^*-stable if and only if system (3.15) is GES with Φ given by (3.17).*

Proof: First we have that (2.2) is equivalent to $\tau \in [\underline{\tau}, \overline{\tau}]$ and $h \in [\max(\underline{h}, \underline{\tau}), \overline{h}]$. Then $w = h - \tau \in [\max(0, \underline{h} - \tau), \overline{h} - \tau]$. Consequently, for all t_k^s satisfying (2.2) there exists $(w_k)_{k\in\mathbb{N}}$, $(\tau_k)_{k\in\mathbb{N}}$ satisfying $\tau_k \in [\underline{\tau}, \overline{\tau}]$ and $w_k \in [max(0, \underline{h} - \tau_k), \overline{h} - \tau_k]$, $\forall k \in \mathbb{N}$, such that

$$x(t_{k+1}^s) = e^{w_k A_c} A_a e^{\tau_k A_c} A_s x(t_k^s) = \cdots = \left(\prod_{i=0}^{i=k} e^{w_i A_c} A_a e^{\tau_i A_c} A_s \right) x_0$$

Moreover, from (3.17) and (3.15) we always have that there exists a sequence $(\xi_k)_{k\in\mathbb{N}}$ such that $\xi_0 = x_0$ and

$$\xi_{k+1} = e^{\beta h_k} e^{w_k A_c} A_a e^{\tau_k A_c} A_s \xi_k = \cdots = \left(\prod_{i=0}^{i=k} e^{h_i \beta} e^{w_i A_c} A_a e^{\tau_i A_c} A_s \right) \xi_0 = e^{\sum_{i=0}^{k} h_i \beta} \left(\prod_{i=0}^{i=k} e^{w_i A_c} A_a e^{\tau_i A_c} A_s \right) \xi_0.$$

But $\sum_{i=0}^{k} h_i = t_{k+1}^s$ with $h_k = \tau_k + h_k$, for all $k \in \mathbb{N}$, then

$$\xi_{k+1} = e^{\beta t_{k+1}} x(t_{k+1}) \ \ \forall k \in \mathbb{N}. \tag{3.18}$$

To prove necessity, using (3.18), $k\underline{h} \leq t_k^s$, and the fact that (2.2)-(3.8) is β^*-stable, we have that there exists $C \in \mathbb{R}^+$ and $\varepsilon^* \in R^+$ such that

$$|\xi_k| = e^{\beta t_k^s} |x(t_k^s)| \leq e^{\beta t_k^s} C e^{-(\beta+\varepsilon^*) t_k^s} |x_0| \leq C (e^{-\varepsilon^* \underline{h}})^k |\xi_0|.$$

Since $\varepsilon \in \mathbb{R}^+$ then (3.15) is GES.

Next, to prove sufficiency, using (3.18) and the fact that (3.15) is GES, we have that there exist $C \in \mathbb{R}^+$ and $\varepsilon \in (0, 1)$ such that

$$|x(t_k^s)| = e^{-\beta t_k^s} |\xi_k| \leq e^{-\beta t_k^s} C \varepsilon^k |\xi_0| = C e^{(-\beta t_k^s + k ln \varepsilon)} |x_0|. \tag{3.19}$$

But $\frac{t_k^s}{\overline{h}} \leq k$ and $t \leq t_k^s + \overline{h}$, then, from (3.19), $\forall t \in [t_k^s, t_k^a], \tau_k \in [\underline{\tau}, \overline{\tau}]$, $k \in \mathbb{N}$,

$$|x(t)| \leq e^{|A_c| \tau_k} |A_s| |x(t_k^s)| \leq C e^{|A_c| \overline{\tau} + \beta \overline{h} - ln\varepsilon} |A_s| C e^{-(\beta - \frac{ln\varepsilon}{\overline{h}})t} |x_0|.$$

Then since $\varepsilon \in (0,1)$ we conclude that (2.2)-(3.8) is β^*-stable. □

The next proposition shows that the map Φ in (3.17) satisfies the previous assumptions.

Proposition 4. *Let Φ be given by (3.17), then Φ satisfies Assumptions 2 and 3.*

Proof: Let us prove that the different assumptions hold.

Assumption 2(i) : From the definition of reachable set, we have

$$\begin{aligned}\Phi(\mathcal{S}) &= \bigcup_{\tau\in[\underline{\tau},\overline{\tau}]}\ \bigcup_{w\in[\max(0,\underline{h}-\tau),\overline{h}-\tau]} e^{(w+\tau)\beta}e^{wA_c}A_a e^{\tau A_c}A_s\mathcal{S}.\\ &= \bigcup_{\tau\in[\underline{\tau},\overline{\tau}]}\ \bigcup_{w\in[\max(0,\underline{h}-\tau),\overline{h}-\tau]}\ \bigcup_{\xi\in\mathcal{S}} e^{(w+\tau)\beta}e^{wA_c}A_a e^{\tau A_c}A_s\{\xi\}\\ &= \bigcup_{\xi\in\mathcal{S}}\ \bigcup_{\tau\in[\underline{\tau},\overline{\tau}]}\ \bigcup_{w\in[\max(0,\underline{h}-\tau),\overline{h}-\tau]} e^{(w+\tau)\beta}e^{wA_c}A_a e^{\tau A_c}A_s\{\xi\} = \bigcup_{\xi\in\mathcal{S}}\Phi(\{\xi\}).\end{aligned}$$

Assumption 2(ii) : From the definition of reachable set, we have

$$\begin{aligned}\Phi(\lambda\mathcal{S}) &= \bigcup_{\tau\in[\underline{\tau},\overline{\tau}]}\ \bigcup_{w\in[\max(0,\underline{h}-\tau),\overline{h}-\tau]} e^{(w+\tau)\beta}e^{wA_c}A_a e^{\tau A_c}A_s\lambda\mathcal{S}\\ &= \lambda\bigcup_{\tau\in[\underline{\tau},\overline{\tau}]}\ \bigcup_{w\in[\max(0,\underline{h}-\tau),\overline{h}-\tau]} e^{(w+\tau)\beta}e^{wA_c}A_a e^{\tau A_c}A_s\mathcal{S} = \lambda\Phi(\mathcal{S}).\end{aligned}$$

Assumption 2(iii) : Let $\mathcal{S}\subseteq\mathbb{R}^n$, then let $\xi'\in\Phi(\mathcal{S})$, there exists $\xi\in\mathcal{S}$, $\tau\in[\underline{\tau},\overline{\tau}]$, and $w\in[\max(0,\underline{h}-\tau),\overline{h}-\tau]$ such that $\xi' = e^{(w+\tau)\beta}e^{wA_c}A_a e^{\tau A_c}A_s\xi$. Then,

$$|z'| \le e^{(w+\tau)\beta}e^{w|A_c|}|A_a|e^{\tau|A_c|}|A_s||z|.$$

Hence, if $\mathcal{S}$ is bounded, so is $\Phi(\mathcal{S})$.

Assumption 3 : Let $\xi'\in\Phi(\mathrm{ch}(\mathcal{S}))$, then there exist $\xi\in\mathrm{ch}(\mathcal{S})$, $\tau\in[\underline{\tau},\overline{\tau}]$, and $w\in[\max(0,\underline{h}-\tau),\overline{h}-\tau]$ such that $\xi' = e^{(w+\tau)\beta}e^{wA_c}A_a e^{\tau A_c}A_s\xi$. Since $\xi\in\mathrm{ch}(\mathcal{S})$, there exist $x,y\in\mathcal{S}$ and $\lambda\in[0,1]$ such that $\xi = \lambda x + (1-\lambda)y$. Then, by linearity

$$\xi' = \lambda e^{(w+\tau)\beta}e^{wA_c}A_a e^{\tau A_c}A_s x + (1-\lambda)e^{(w+\tau)\beta}e^{wA_c}A_a e^{\tau A_c}A_s y.$$

By remarking that $e^{(w+\tau)\beta}e^{wA_c}A_a e^{\tau A_c}A_s x\in\Phi(\mathcal{S})$ and $e^{(w+\tau)\beta}e^{wA_c}A_a e^{\tau A_c}A_s y\in\Phi(\mathcal{S})$, it follows that $\xi'\in\mathrm{ch}(\Phi(\mathcal{S}))$. Thus, $\Phi(\mathrm{ch}(\mathcal{S}))\subseteq\mathrm{ch}(\Phi(\mathcal{S}))$. □

It follows from Propositions 3 and 4 that Problem 1* can be reduced to 4. Therefore, in the next sections, we develop an algorithm to solve Problem 4.

3.2.2 Stability verification: theoretical results

This section proposes a framework to solve the stability verification problem. More precisely, it presents theoretical necessary and sufficient conditions for stability of system (3.15). Then, an algorithm is proposed to solve Problem 4.

3.2.2.1 Necessary and sufficient conditions for stability

The following result characterizes the stability of system (3.15) in terms of the map Φ given by (3.17).

Theorem 17. *Let $\mathcal{S} \in \mathcal{B}_0(\mathbb{R}^n)$, under Assumption 2, the following statements are equivalent:*

(a) System (3.15) is GES;

(b) There exists $(k, j, \rho) \in \mathbb{N}^+ \times \mathbb{N}_{[0,k-1]} \times (0,1)$ such that $\Phi^k(\mathcal{S}) \subseteq \rho\Phi^j(\mathcal{S})$;

(c) There exists $(k, \rho) \in \mathbb{N}^+ \times (0,1)$ such that $\Phi^k(\mathcal{S}) \subseteq \rho \bigcup_{j=0}^{k-1} \Phi^j(\mathcal{S})$.

Proof: It is obvious that $(b) \implies (c)$. Hence, it is sufficient to prove that $(a) \implies (b)$ and $(c) \implies (a)$.

$(a) \implies (b)$: We prove that there exists $(k, \rho) \in \mathbb{N}^+ \times [0,1)$ such that $\Phi^k(\mathcal{S}) \subseteq \rho\mathcal{S}$. This is a special case of (b) when $j = 0$. Since $\mathcal{S} \in \mathcal{B}_0(\mathbb{R}^n)$, then there exist $\underline{c}$, $\overline{c} \in \mathbb{R}^+$ such that $\underline{c}\mathcal{B} \subseteq \mathcal{S} \subseteq \overline{c}\mathcal{B}$. Let $(\xi_k)_{k\in\mathbb{N}}$ be a trajectory of (3.15) with $\xi_0 \in \mathcal{S}$, then $|\xi_0| \leq \overline{c}$. Under item (i) of Assumption 2, for all $k \in \mathbb{N}$, $\Phi^k(\mathcal{S})$ represents all the possible values of ξ_k. Since (3.15) is GES, then there exist $C \in \mathbb{R}^+$ and $\varepsilon \in (0,1)$ such that $|\xi_k| \leq C\varepsilon^k|\xi_0| \leq C\varepsilon^k\overline{c}$. This gives us for all $k \in \mathbb{N}$,

$$\Phi^k(\mathcal{S}) \subseteq C\overline{c}\varepsilon^k\mathcal{B} \subseteq C\frac{\overline{c}}{\underline{c}}\varepsilon^k\mathcal{S}.$$

For k sufficiently large, $C\frac{\overline{c}}{\underline{c}}\varepsilon^k < 1$ and therefore (b) holds.

$(c) \implies (a)$: Let $\varepsilon = \rho^{\frac{1}{k}}$; since $\rho \in (0,1)$ then for all $j \in \mathbb{N}_{[0,k-1]}$, $\rho \leq \varepsilon^{k-j}$ and

$$\Phi^k(\mathcal{S}) \subseteq \rho \bigcup_{j=0}^{k-1} \Phi^j(\mathcal{S}) \subseteq \bigcup_{j=0}^{k-1} \varepsilon^{k-j}\Phi^j(\mathcal{S}). \tag{3.20}$$

Let $\mathcal{S}' = \bigcup_{j=0}^{k-1} \varepsilon^{-j}\Phi^j(\mathcal{S})$, then using items (i) and (ii) of Assumption 2:

$$\Phi(\mathcal{S}') = \Phi\left(\bigcup_{j=0}^{k-1} \varepsilon^{-j}\Phi^j(\mathcal{S})\right) = \bigcup_{j=0}^{k-1} \Phi(\varepsilon^{-j}\Phi^j(\mathcal{S})) \subseteq \bigcup_{j=0}^{k-1} \varepsilon^{-j}\Phi^{j+1}(\mathcal{S}) = \left(\bigcup_{j=0}^{k-2} \varepsilon^{-j}\Phi^{j+1}(\mathcal{S})\right) \cup \varepsilon^{-k+1}\Phi^k(\mathcal{S}).$$

Making a change of index in the union and using (3.20) yield

$$\Phi(\mathcal{S}') \subseteq \left(\bigcup_{j=1}^{k-1} \varepsilon^{-j+1}\Phi^j(\mathcal{S})\right) \cup \varepsilon^{-k+1}\left(\bigcup_{j=0}^{k-1} \varepsilon^{k-j}\Phi^j(\mathcal{S})\right) \subseteq \varepsilon\left(\bigcup_{j=0}^{k-1} \varepsilon^{-j}\Phi^j(\mathcal{S})\right) = \varepsilon\mathcal{S}'. \tag{3.21}$$

Let us remark that $\mathcal{S} \subseteq \mathcal{S}'$, then $\underline{c}\mathcal{B} \subseteq \mathcal{S}'$. In addition, since $\mathcal{S}$ is bounded, from item (iii) of Assumption 2, $\mathcal{S}'$ is bounded and there exists $\overline{c}' \in \mathbb{R}^+$ such that $\mathcal{S}' \subseteq \overline{c}'\mathcal{B}$. Now consider a trajectory $(\xi_k)_{k\in\mathbb{N}}$ of (3.15), then $\xi_0 \in |\xi_0|\mathcal{B} \subseteq \frac{|\xi_0|}{\underline{c}}\mathcal{S}'$. Items (i) and (ii) of Assumption 2 and (3.21) give for all $k \in \mathbb{N}$

$$\xi_k \in \Phi^k\left(\frac{|\xi(0)|}{\underline{c}}\mathcal{S}'\right) \subseteq \frac{|\xi(0)|}{\underline{c}}\varepsilon^k\mathcal{S}' \subseteq \frac{|\xi(0)|}{\underline{c}}\varepsilon^k\overline{c}'\mathcal{B}.$$

In other words, it holds for all $k \in \mathbb{N}$,

$$|\xi_k| \leq \frac{\overline{c}'}{\underline{c}}\varepsilon^k|\xi_0|.$$

Since $\varepsilon \in (0,1)$, system (3.15) is GES.

□

Theorem 17 shows the existence of a generally non-convex contracting set $\mathcal{S}'$, with respect to the system (3.15) whenever the latter is GES. In addition, when Assumption 3 holds, it is possible to show the existence of a convex contracting set as well.

Corollary 3. *Let $\mathcal{S} \in \mathcal{B}_0(\mathbb{R}^n)$, under Assumptions 2 and 3, system (3.15) is GES if and only if there exists $(k,\varepsilon) \in \mathbb{N}^+ \times (0,1)$ such that $\Phi(\hat{\mathcal{S}}) \subseteq \varepsilon\hat{\mathcal{S}}$, where $\hat{\mathcal{S}} = ch\left(\bigcup_{j=0}^{k-1}\varepsilon^{-j}\Phi^j(\mathcal{S})\right)$.*

Proof: For sufficiency, we assume that there exists $(k,\varepsilon) \in \mathbb{N}^+ \times (0,1)$ such that $\Phi(\hat{\mathcal{S}}) \subseteq \varepsilon\hat{\mathcal{S}}$. Following the same steps after (3.21) in the proof of Theorem 17, we conclude that (3.15) is GES. For necessity, we assume that (3.15) is GES. Then, from the proof of Theorem 17, there exists $(k,\varepsilon) \in \mathbb{N}^+ \times (0,1)$ such that $\mathcal{S}' = \bigcup_{j=0}^{k-1}\varepsilon^{-j}\Phi^j(\mathcal{S})$ satisfies $\Phi(\mathcal{S}') \subseteq \varepsilon\mathcal{S}'$. Let $\hat{\mathcal{S}} = \mathrm{ch}(\mathcal{S}')$, then using Assumption 3 we have:

$$\Phi(\hat{\mathcal{S}}) = \Phi\left(\mathrm{ch}\left(\mathcal{S}'\right)\right) \subseteq \mathrm{ch}\left(\Phi\left(\mathcal{S}'\right)\right) \subseteq \mathrm{ch}\left(\varepsilon\mathcal{S}'\right) = \varepsilon\mathrm{ch}\left(\mathcal{S}'\right) = \varepsilon\hat{\mathcal{S}}.$$

□

Thus, with the addition of Assumption 3, the stable system (3.15) admits a convex contracting set $\hat{\mathcal{S}}$. In that case, we can further prove that a characterization of the stability of (3.15) can be given in terms of a convexified version of the set valued-map Φ. Let us consider the set-valued map $\hat{\Phi} : 2^{\mathbb{R}^n} \to 2^{\mathbb{R}^n}$ given by

$$\forall \mathcal{S} \subseteq \mathbb{R}^n,\ \hat{\Phi}(\mathcal{S}) = \mathrm{ch}(\Phi(\mathcal{S})).$$

The images of $\hat{\Phi}$ are convex sets and for all $\mathcal{S} \subseteq \mathbb{R}^n$; $\Phi(\mathcal{S}) \subseteq \hat{\Phi}(\mathcal{S})$. The iterates of $\hat{\Phi}$ are defined similarly to those of Φ. Let us also define the dynamical system associated to the set-valued map $\hat{\Phi}$:

$$\xi_{k+1} \in \hat{\Phi}(\{\xi_k\}),\ k \in \mathbb{N}. \tag{3.22}$$

Let us state some properties of the map $\hat{\Phi}$:

Lemma 5. *Let Assumptions 2 and 3 hold. For all $\mathcal{S}$, $\mathcal{S}' \subseteq \mathbb{R}^n$, $\lambda \in \mathbb{R}_0^+$, the following assertions hold:*

(i) if $\mathcal{S} \subseteq \mathcal{S}'$, then $\hat{\Phi}(\mathcal{S}) \subseteq \hat{\Phi}(\mathcal{S}')$;

(ii) $\hat{\Phi}(\lambda\mathcal{S}) \subseteq \lambda\hat{\Phi}(\mathcal{S})$;

(iii) if $\mathcal{S}$ is bounded, then $\hat{\Phi}(\mathcal{S})$ is a bounded;

(iv) $\hat{\Phi}(ch(\mathcal{S})) = \hat{\Phi}(\mathcal{S})$.

Proof: Let us prove the different assertions.

(i) : From item (i) of Assumption 2, we have that $\mathcal{S} \subseteq \mathcal{S}'$ implies $\Phi(\mathcal{S}) \subseteq \Phi(\mathcal{S}')$. Therefore, $\text{ch}(\Phi(\mathcal{S})) \subseteq \text{ch}(\Phi(\mathcal{S}'))$.

(ii) : From item (ii) of Assumption 2, $\text{ch}(\Phi(\lambda\mathcal{S})) \subseteq \text{ch}(\lambda\Phi(\mathcal{S})) = \lambda\text{ch}(\Phi(\mathcal{S}))$.

(iii) : From item (iii) of Assumption 2, if $\mathcal{S}$ is bounded then $\Phi(\mathcal{S})$ and thus $\text{ch}(\Phi(\mathcal{S}))$ are bounded.

(iv) : From the first item of the Lemma, $\mathcal{S} \subseteq \text{ch}(\mathcal{S})$ gives $\hat{\Phi}(\mathcal{S}) \subseteq \hat{\Phi}(\text{ch}(\mathcal{S}))$. Then, from Assumption 3,

$$\hat{\Phi}(\text{ch}(\mathcal{S})) = \text{ch}(\Phi(\text{ch}(\mathcal{S}))) \subseteq \text{ch}(\text{ch}(\Phi(\mathcal{S}))) = \text{ch}(\Phi(\mathcal{S})) = \hat{\Phi}(\mathcal{S}).$$

□

The previous result shows that items (ii) and (iii) of Assumption 2 are transferred from Φ to $\hat{\Phi}$. This is not the case of item (i) of Assumption 2, where only a weaker property can be stated for $\hat{\Phi}$ (item (i) in the lemma). In particular, for $k \in \mathbb{N}^+$, $\hat{\Phi}^k(\mathcal{S})$ generally contains values that are not reachable by any trajectory $(\xi_k)_{k\in\mathbb{N}}$ of (3.22) with $\xi_0 \in \mathcal{S}$. On the other hand, Assumption 3 gives a stronger property for $\hat{\Phi}$ than for the original map Φ (item (iv) in the lemma).

We can now prove the following result which shows equivalence between stability of systems (3.15) and (3.22) and gives a characterization in terms of the set-valued map $\hat{\Phi}$.

Theorem 18. *Let $\mathcal{S} \in \mathcal{B}_0(\mathbb{R}^n)$, under Assumptions 2 and 3, the following statements are equivalent:*

(a) System (3.15) is GES;

(b) There exists $(k, j, \rho) \in \mathbb{N}^+ \times \mathbb{N}_{[0,k-1]} \times (0, 1)$ such that $\hat{\Phi}^k(\mathcal{S}) \subseteq \rho\hat{\Phi}^j(\mathcal{S})$;

(c) There exists $(k,\rho)\in\mathbb{N}^+\times(0,1)$ *such that* $\hat{\Phi}^k(\mathcal{S})\subseteq\rho ch\big(\bigcup_{j=0}^{k-1}\hat{\Phi}^j(\mathcal{S})\big)$*;*

(d) System (3.22) is GES.

Proof: Obviously $(b)\implies(c)$. Moreover $(d)\implies(a)$, since all trajectories of (3.15) are also trajectories of (3.22). Hence, it is sufficient to prove that $(a)\implies(b)$ and $(c)\implies(d)$.

$(a)\implies(b)$: We prove that there exists $(k,\rho)\in\mathbb{N}^+\times(0,1)$ such that (b) is true for $j=0$. We have from Corollary 3 that there exist $(i,\varepsilon)\in\mathbb{N}^+\times(0,1)$ such that $\Phi(\hat{\mathcal{S}})\subseteq\varepsilon\hat{\mathcal{S}}$, where $\hat{\mathcal{S}}=\text{ch}\left(\bigcup_{j=0}^{i-1}\varepsilon^{-j}\Phi^j(\mathcal{S})\right)$. Then,

$$\hat{\Phi}(\hat{\mathcal{S}})=\text{ch}(\Phi(\hat{\mathcal{S}}))\subseteq\text{ch}(\varepsilon\hat{\mathcal{S}})=\varepsilon\hat{\mathcal{S}}. \tag{3.23}$$

Also $\mathcal{S}\in\mathcal{B}_0(\mathbb{R}^n)$ implies, from item (iii) of Assumption 2 that $\hat{\mathcal{S}}$ is bounded. Then, there exists $\overline{c}\in\mathbb{R}^+$ such that $\hat{\mathcal{S}}\subseteq\overline{c}\mathcal{S}$. Let us remark that $\mathcal{S}\subseteq\hat{\mathcal{S}}$, then, from (3.23) and items (i) and (ii) of Lemma 5, for all $k\in\mathbb{N}$,

$$\hat{\Phi}^k(\mathcal{S})\subseteq\hat{\Phi}^k(\hat{\mathcal{S}})\subseteq\varepsilon^k\hat{\mathcal{S}}\subseteq\overline{c}\varepsilon^k\mathcal{S}.$$

Since $\varepsilon\in(0,1)$, then for k sufficiently large it becomes true that $\overline{c}\varepsilon^k<1$ which allows us to conclude.

$(c)\implies(d)$: Let $\varepsilon=\rho^{\frac{1}{k}}$; since $\rho\in(0,1)$ then for all $j\in\mathbb{N}_{[0,k-1]}$, $\rho\leq\varepsilon^{k-j}$ and

$$\hat{\Phi}^k(\mathcal{S})\subseteq\rho\text{ch}\left(\bigcup_{j=0}^{k-1}\hat{\Phi}^j(\mathcal{S})\right)\subseteq\text{ch}\left(\bigcup_{j=0}^{k-1}\varepsilon^{k-j}\hat{\Phi}^j(\mathcal{S})\right). \tag{3.24}$$

Let $\hat{\mathcal{S}}'=\text{ch}(\bigcup_{j=0}^{k-1}\varepsilon^{-j}\hat{\Phi}^j(\mathcal{S}))$, then by item (iv) of Lemma 5, items (i) and (ii) of Assumption 2, we have

$$\begin{aligned}\hat{\Phi}(\hat{\mathcal{S}}') &= \hat{\Phi}\left(\bigcup_{j=0}^{k-1}\varepsilon^{-j}\hat{\Phi}^j(\mathcal{S})\right)=\text{ch}\left(\Phi\left(\bigcup_{j=0}^{k-1}\varepsilon^{-j}\hat{\Phi}^j(\mathcal{S})\right)\right)\subseteq\text{ch}\left(\bigcup_{j=0}^{k-1}\varepsilon^{-j}\Phi(\hat{\Phi}^j(\mathcal{S}))\right)\\ &\subseteq \text{ch}\left(\bigcup_{j=0}^{k-1}\varepsilon^{-j}\hat{\Phi}^{j+1}(\mathcal{S})\right)\subseteq\text{ch}\left(\left(\bigcup_{j=0}^{k-2}\varepsilon^{-j}\hat{\Phi}^{j+1}(\mathcal{S})\right)\cup\varepsilon^{-k+1}\hat{\Phi}^k(\mathcal{S})\right).\end{aligned}$$

Making a change of index in the union and using (3.24) yield

$$\hat{\Phi}(\hat{\mathcal{S}}')\subseteq\varepsilon\text{ch}\left(\bigcup_{j=0}^{k-1}\varepsilon^{-j}\hat{\Phi}^j(\mathcal{S})\right)=\varepsilon\hat{\mathcal{S}}'.$$

Let us remark that $\mathcal{S}\subseteq\hat{\mathcal{S}}'$, moreover, since $\mathcal{S}$ is bounded then from item (iii) of Lemma 5, $\hat{\mathcal{S}}'$ is bounded. It follows that there exist $\underline{c}'\in\mathbb{R}^+$, $\overline{c}'\in\mathbb{R}^+$ such that $\underline{c}'\mathcal{B}\subseteq\hat{\mathcal{S}}'\subseteq\overline{c}'\mathcal{B}$. Following the same steps after (3.21) of the proof of Theorem 17, one concludes that (3.22) is GES. □

Remark 6. *The results in this section can be applied to stability analysis of discrete-time switched linear systems of the form* $\xi_{k+1} = A_{i_k}\xi_k$ *where* $i_k \in \mathbb{N}_{[1,N]}$*, by defining the associated set-valued map* $\Phi(\mathcal{S}) = \bigcup_{i=1}^{N} A_i\mathcal{S}$*. In particular, by Theorem 18, we can recover the result in [LA09, Proposition 1] stating the equivalence between stability of the switched system and of the difference inclusion* $\xi_{k+1} \in ch(\{A_1, \ldots, A_n\})\xi_k$*. Also, the stability characterizations established in [AL14a, Theorem 1 and Corollary 2] for discrete-time switched linear systems can be obtained directly from Theorems 17 and 18, respectively.*

3.2.3 An algorithm for stability verification

In this section, we present an algorithm for verifying the stability of system (3.15).

3.2.3.1 A sufficient condition for stability

The maps Φ and $\hat{\Phi}$ involved in Theorems 17 and 18 can be impractical to compute exactly. This is the case for instance with linear impulsive systems, which involve the computation of the reachable set of a linear system on a time interval. In that case, we may use an over-approximation $\overline{\Phi} : 2^{\mathbb{R}^n} \to 2^{\mathbb{R}^n}$, which is easier to compute and satisfies the following assumption:

Assumption 4. *For all* $\mathcal{S} \subseteq \mathbb{R}^n$*, the following assertions hold:*

(i) $\Phi(\mathcal{S}) \subseteq \overline{\Phi}(\mathcal{S})$*;*

(ii) if $\mathcal{S}$ *is bounded then* $\overline{\Phi}(\mathcal{S})$ *is bounded.*

The iterates of $\overline{\Phi}$ are defined similarly to those of Φ. We now derive sufficient conditions for stability of system (3.15) based on $\overline{\Phi}$.

Corollary 4. *Under Assumptions 2 and 4, if there exist* $\mathcal{S} \in \mathcal{B}_0(\mathbb{R}^n)$ *and* $(k, i, \rho) \in \mathbb{N}^+ \times \mathbb{N}_{[0,k-1]} \times (0, 1)$ *such that* $\overline{\Phi}^k(\mathcal{S}) \subseteq \rho\overline{\Phi}^i(\mathcal{S})$*, then system (3.15) is GES.*

Proof: First, $\overline{\Phi}^k(\mathcal{S}) \subseteq \rho\overline{\Phi}^i(\mathcal{S}) \subseteq \rho\bigcup_{j=0}^{k-1}\overline{\Phi}^j(\mathcal{S}) \subseteq \bigcup_{j=0}^{k-1}\varepsilon^{k-j}\overline{\Phi}^j(\mathcal{S})$ where $\varepsilon = \rho^{\frac{1}{k}}$. Similar to the second part of the proof of Theorem 17, let $\mathcal{S}' = \bigcup_{j=0}^{k-1}\varepsilon^{-j}\overline{\Phi}^j(\mathcal{S})$. Then, by items (i) and (ii) of Assumption 2, and item (i) of Assumption 4, we have

$$\Phi(\mathcal{S}') = \Phi\left(\bigcup_{j=0}^{k-1}\varepsilon^{-j}\overline{\Phi}^j(\mathcal{S})\right) \subseteq \bigcup_{j=0}^{k-1}\varepsilon^{-j}\Phi(\overline{\Phi}^j(\mathcal{S})) \subseteq \bigcup_{j=0}^{k-1}\varepsilon^{-j}\overline{\Phi}(\overline{\Phi}^j(\mathcal{S})) = \bigcup_{j=0}^{k-1}\varepsilon^{-j}\overline{\Phi}^{j+1}(\mathcal{S}).$$

Then, following the same steps as in (3.21), we can show that $\Phi(\mathcal{S}') \subseteq \varepsilon\mathcal{S}'$. Following the same lines as in the proof of Theorem 17 after (3.21) and using item (ii) of Assumption 4, one concludes that (3.15) is GES. □

Let us remark that if the images of $\overline{\Phi}$ are convex sets, then $\Phi(\mathcal{S}) \subseteq \hat{\Phi}(\mathcal{S}) \subseteq \overline{\Phi}(\mathcal{S})$. In such a case, in regards of Theorem 18, the only conservatism introduced by Corollary 4 is due to the over-approximation of $\hat{\Phi}(\mathcal{S})$.

3.2.3.2 Algorithm

We propose a method to solve Problem 4 based on the sufficient condition given in Corollary 4. The stability verification algorithm consists of an initialization step and a main loop. In the initialization step, we compute an initial set $\mathcal{S}_0 \in \mathcal{B}_0(\mathbb{R}^n)$, which is then propagated in the main loop using the map $\overline{\Phi}$ to check the stability condition given by Corollary 4.

The choice of the initial set is important in order to try to minimize the value of the integer k such that the stability condition given by Corollary 4 holds. One approach to choose this set for the particular case of linear impulsive systems is given in Section 3.2.4.1. The function computing $\mathcal{S}_0$ is denoted by $\mathsf{init}(\Phi)$.

In the main loop, the initial set is propagated using the map $\overline{\Phi}$. The stability condition given by Corollary 4 is checked after each iteration. If the condition is verified then system (3.15) is proved GES and the algorithm returns true. We impose a maximum number of iterations k_{max}. If that number of iterations is reached then the algorithm fails to prove stability and returns unknown. The overall method is then summarized by the Algorithm 1:

Algorithm 1. *Stability verification*

function *is_GES(Φ)*

input: Φ

output: *true if system (3.15) is proved GES, unknown otherwise*

parameter: $k_{max} \in \mathbb{N}^+$

1: $\mathcal{S}_0$:=*init*(Φ); ▷ *compute initial set*

2: ***for*** $k = 1$ *to* k_{max} ***do***

3: $\mathcal{S}_k$:=$\overline{\Phi}(\mathcal{S}_{k-1})$; ▷ *set propagation*

4: ***if*** $\exists (i, \rho) \in \mathbb{N}_{[0,k-1]} \times (0, 1) : \mathcal{S}_k \subseteq \rho \mathcal{S}_i$ ***then*** ▷ *stability check*

5: *return true;*

6: ***end if***

7: ***end for***

8: *return unknown;*

The proposed approach above induces conservativeness due to the over-approximation of the map Φ and to the limited number of iterations. Consequently, it is possible that some stable systems (3.15) cannot be

verified by the algorithm. On the other hand, if the maps Φ or $\hat{\Phi}$ can be effectively computed then these can replace $\overline{\Phi}$ in Algorithm 1, and for any initial set $\mathcal{S}_0$ there exists a value for $k_{max} \in \mathbb{N}$ such that the algorithm returns true if and only if system (3.15) is GES.

3.2.4 Case of linear impulsive systems

In this section, we give the practical details regarding the implementation of Algorithm 1 for the linear impulsive system (2.2)-(3.8).

We use sets given by polytopes of $\mathbb{R}^n$, which can be defined as the intersection of a finite number of closed half-spaces, that is $\mathcal{S} = \{x \in \mathbb{R}^n : Hx \leq b\}$ where $H \in \mathbb{R}^{r\times n}$, $b \in \mathbb{R}^m$ and the vector of inequalities is interpreted component-wise.

3.2.4.1 Initial set computation

The choice of the initial set $\mathcal{S}_0$ is crucial as it may impact significantly the number of iterations of $\overline{\Phi}$ that are necessary to check the condition of Corollary 4. Intuitively, in order to minimize this number of iterations, $\mathcal{S}_0$ should be already close to an invariant set. Indeed, if $\overline{\Phi}(\mathcal{S}_0) \subseteq \mathcal{S}_0$, the stability condition holds after only one iterate of $\overline{\Phi}$. One way to choose $\mathcal{S}_0$ close to an invariant set is to define $\mathcal{S}_0$ as a common contracting polytope to $L \in \mathbb{N}^+$ linear discrete-time systems, such that

$$\forall j \in \mathbb{N}_{[1,L]},\ e^{h_j\beta I}e^{(h_j-\tau_j)A_c}A_a e^{\tau_j A_c}A_s\mathcal{S}_0 \subseteq \text{int}(\mathcal{S}_0),$$

where the couples (τ_j, h_j) satisfy timing contract (2.2) for all $j \in \mathbb{N}_{[1,L]}$. Then, $\mathcal{S}_0$ can be computed either using a backward iterative method as in [Bla91] and [FM16] or using a forward iterative method as in [AL14a]. We denote the function computing $\mathcal{S}_0$ by $\mathsf{init}(A_c, A_a, A_s, \underline{\tau}, \overline{\tau}, \underline{h}, \overline{h}, L)$. Then, $\mathcal{S}_0 = \{x \in \mathbb{R}^n : Hx \leq b_0\}$. The matrix H defining $\mathcal{S}_0$ is used in the main loop of the algorithm in the computation of the map $\overline{\Phi}$.

3.2.4.2 Main loop

The initial set is propagated using the map $\overline{\Phi}$ given by the following corollary which could be derived in the same way as the approximation scheme in Theorem 16.

Corollary 5. *Given $\mathcal{S} \in \mathcal{B}_0(\mathbb{R}^n)$, $N_1, N_2 \in \mathbb{N}^+$, and $H \in \mathbb{R}^{r\times n}$, such that $0 \in int(ch(\{H_1, \ldots, H_r\}))$, let $\overline{\Phi} : \mathcal{B}_0(\mathbb{R}^n) \to \mathcal{B}_0(\mathbb{R}^n)$ be given by*

$$\overline{\Phi}(\mathcal{S}) = \Gamma_H\left(\bigcup_{j_1=1}^{N_1}\bigcup_{j_2=1}^{n_2(j_1)} e^{(\theta(j_1)+(j_2-1)\delta_2)(A_c+\beta I_n)}\overline{\Phi}_{j_1}(\mathcal{S})\right) \tag{3.25}$$

where for $j_1 \in \mathbb{N}_{[1,N_1]}$,

$$\overline{\Phi}_{j_1}(\mathcal{S}) = \overline{\mathcal{R}}_{[0,\delta_2]}^{(A_c+\beta I_n)}\Big(A_a e^{(\underline{\tau}+(j_1-1)\delta_1)(A_c+\beta I_n)}\overline{\mathcal{R}}_{[0,\delta_1]}^{A_c}(A_s\mathcal{S})\Big)$$

with δ_1, δ_2, $\theta(j_1)$, $n_2(j_1)$ *given by (3.13) , and* $\overline{\mathcal{R}}_{[0,\delta_1]}^{(A_c+\beta I_n)}$, $\overline{\mathcal{R}}_{[0,\delta_2]}^{A_c}$ *computed as in (3.3). Then,* $\overline{\Phi}(\mathcal{S})$ *satisfies Assumption 4.*

Proof. The proof is straightforward as that in Theorem 16 and Corollary 2. □

Then if the stability condition given by Corollary 4 is verified, system (2.2)-(3.8) is β^*-stable as stated by Proposition 3 otherwise, if a maximum number of iterations is reached then the algorithm fails to prove stability.

Remark 7. *A normal issue arising after proposing an over-approximation* $\overline{\Phi}$ *is to know how far the sufficient condition proposed by Corollary 4 is from being necessary. In regards of Theorem 18, this is related to the distance between* $\hat{\Phi}(\mathcal{S})$ *and* $\Phi(\mathcal{S})$*. First of all, from Corollary 5, it appears that, by choosing the time steps* δ_1 *and* δ_2 *small enough,* $\Phi(\mathcal{S})$ *and* $\hat{\Phi}(\mathcal{S})$ *can be approximated arbitrarily close by* $\bigcup_{j_1=1}^{N_1}\bigcup_{j_2=1}^{n_2(j_1)} e^{(\theta(j_1)+(j_2-1)\delta_2)A_c}\overline{\Phi}_{j_1}(\mathcal{S})$ *and* $ch\big(\bigcup_{j_1=1}^{N_1}\bigcup_{j_2=1}^{n_2(j_1)} e^{(\theta(j_1)+(j_2-1)\delta_2)(A_c+\beta I_n)}\overline{\Phi}_{j_1}(\mathcal{S})\big)$*, respectively.*

Then, the set $ch\big(\bigcup_{j_1=1}^{N_1}\bigcup_{j_2=1}^{n_2(j_1)} e^{(\theta(j_1)+(j_2-1)\delta_2)(A_c+\beta I_n)}\overline{\Phi}_{j_1}(\mathcal{S})\big)$ *can be approximated arbitrarily close by* $\Gamma_H\big(ch(\bigcup_{j_1=1}^{N_1}\bigcup_{j_2=1}^{n_2(j_1)} e^{(\theta(j_1)+(j_2-1)\delta_2)(A_c+\beta I_n)}\overline{\Phi}_{j_1}(\mathcal{S}))\big)$ *by considering a sufficient number of approximation directions* H_i*. Thus, it follows that, by choosing appropriately the time steps* δ_1 *and* δ_2 *in addition to the matrix* H*,* $\hat{\Phi}(\mathcal{S})$ *can be approximated arbitrarily close by* $\overline{\Phi}(\mathcal{S})$*.*

Remark 8. *In case of NPILS (3.4)-(2.24) we define the map* Φ *by*

$$\Phi(\mathcal{S}) = \bigcup_{h\in[\underline{h},\overline{h}]} e^{h(A_c'+\beta h)}A_s'\mathcal{S} \tag{3.26}$$

and which could be over-approximated as the following

Corollary 6. *Let the matrix* $H \in \mathbb{R}^{r\times n'}$*, such that* 0 *is in the interior of* $ch(\{H_1,\dots,H_r\})$*. Let* $\overline{\Phi}$ *be given by*

$$\overline{\Phi}(\mathcal{S}) = \Gamma_H\left(ch\big(\overline{\mathcal{R}}_{[0,\overline{h}-\underline{h}]}^{(A_c'+\beta I_{n'})}(e^{\underline{h}(A_c'+\beta I_{n'})}A_s'\mathcal{S})\big)\right), \tag{3.27}$$

where $\overline{\mathcal{R}}_{[0,\overline{h}-\underline{h}]}^{(A_c'+\beta I_{n'})}(e^{\underline{h}(A_c'+\beta I_{n'})}A_s'\mathcal{S})$ *is computed as in Theorem 15. Then,* $\overline{\Phi}$ *satisfies Assumption 4.*

Proof: *The result follows from Corollary 2 after defining the continuous dynamics with the matrix* $A_c' + \beta I_{n'}$*.* □

Then we can directly prove that Φ given by (3.26) satisfies Assumption 2 and 3. Also, $\overline{\Phi}$ given by (3.27) satisfies Assumption 4.

3.3 Applications and numerical results

We implement Algorithm 1 in Matlab using the Multi-Parametric Toolbox [HKJM13] to verify stability for the special case of NPILS (3.4)-(2.24) and then for systems, given by (3.8), under timing contracts given in Section 2.1.1.

3.3.1 Nearly periodic impulsive linear systems

3.3.1.1 An academic example

Example 1. *The following example is taken from [HDTP13] proposing an LMI based approach to verify stability of a linear impulsive system. In order to compare our results with existing ones in literature we set $\beta = 0$ and search for verifying β^*-stability which is equivalent to the global uniform exponential stability (GUES) in this case (see [AKGD16b]). Consider system (3.4)-(2.24) with*

$$A_c' = \begin{pmatrix} 0 & -3 & 1 \\ 1.4 & -2.6 & 0.6 \\ 8.4 & -18.6 & 4.6 \end{pmatrix}, \quad A_s' = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{3.28}$$

As noted in [HDTP13], the matrix $\prod_{i\in\mathbb{N}_{[1,5]}}(e^{h_i A_c'}A_s')$ has eigenvalues outside the unit circle for $h_1 = 0.515$ and $h_i = 0.1$, for $i \in \mathbb{N}_{[2,5]}$. As a result, we can consider that if $\underline{h} = 0.1$, the value 0.515 is an upper bound for admissible values of $\overline{h}$. For $\underline{h} = 0.1$, stability could be proven up to $\overline{h} = 0.3$ following the LMI approach in [HDTP13], and up to $\overline{h} = 0.375$ following the set based approach in [FM16]. Results obtained

Table 3.1: Results of Algorithm 1 on system (3.28) for several values of parameters L (number of subsystems chosen to find the initial set $\mathcal{S}_0$) and k_{max} (maximum number of iterations of Algorithm 1) with $N = 100$ (number steps used in reachability analysis): find for $\underline{h} = 0.1$ the maximum value of $\overline{h}$ for which stability could be proved; T_{CPU} is the computation time in seconds; i, k are the index values for which the stability condition $\mathcal{S}_k \subseteq \text{int}(\mathcal{S}_i)$ is verified; r is such that $H = H^0 \in \mathbb{R}^{r\times 3}$ in computing (3.27).

Parameter setup	$\overline{h}$	$T_{CPU}(s)$	i	k	r
A ($L = 1$, $k_{max} = 1$)	0.11	0.2	0	1	32
B ($L = 1$, $k_{max} = 100$)	0.5	0.4	3	7	32
C ($L = 2$, $k_{max} = 1$)	0.5	1.1	0	1	30
D ($L = 2$, $k_{max} = 100$)	0.514	2.0	27	32	26

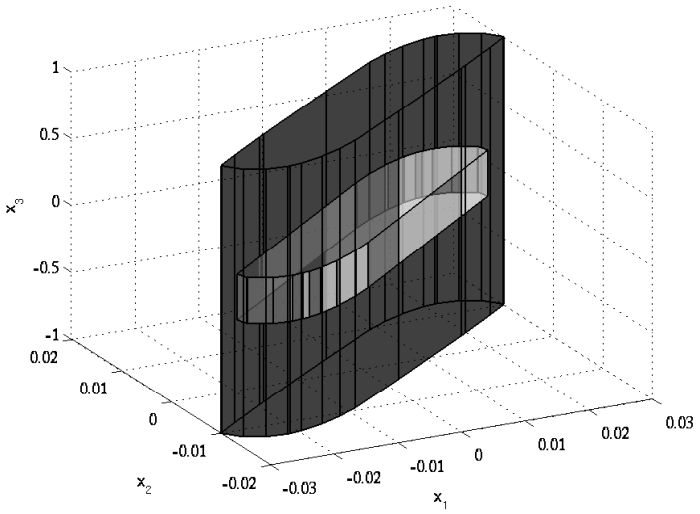


Figure 3.2: Polytopes $\mathcal{S}_0$ and $\mathcal{S}_1$ computed by Algorithm 1 using parameter setup C for system (3.28) with $\underline{h} = 0.1$ and $\overline{h} = 0.5$; $\mathcal{S}_1$ is strictly included in $\mathcal{S}_0$.

using Algorithm 1 with several parameter setups are reported in Table 3.5. In this example, parameter setups B and D lead to less conservative results than the mentioned approaches since stability is verified at least up to $\overline{h} = 0.5$. *Moreover, with parameter setup D, the verified value* $\overline{h} = 0.514$ *is tight, since it is very close to the known upper-bound* 0.515. *Figure 3.2 shows the polytopes* $\mathcal{S}_0$ *and* $\mathcal{S}_1$ *computed by Algorithm 1 using parameter setup C for* $\underline{h} = 0.1$ *and* $\overline{h} = 0.5$. *The inclusion of* $\mathcal{S}_1$ *in* $\mathcal{S}_0$ *proves the stability of the linear impulsive system.*

3.3.1.2 Sampled-data systems

Example 2. *This sampled data system is taken from [Bri13], which compares results of LMI or SOS based approaches for stability analysis of linear impulsive systems. Consider the state space plant model given by (2.9a)-(2.9b) with*

$$A = \begin{pmatrix} 0 & 1 \\ 0 & -0.1 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ 0.1 \end{pmatrix}, \quad K = \begin{pmatrix} -3.75 & -11.5 \end{pmatrix}. \tag{3.29}$$

After rewriting the problem in the form of (3.4)-(2.24), we set $\underline{h} = 10^{-5}$. *For this system, we can check numerically that the matrix* $e^{hA_c'}A_s'$ *is Schur for* $h \in]0, 1.7294]$ *and has eigenvalues outside the unit circle for larger values of* h. *Thus, we know that* 1.7294 *is an upper bound for the maximal value of* $\overline{h}$ *guaranteeing* β^**-stability with* $\beta = 0$. *Table 3.2 reports the maximum value of* $\overline{h}$, *for which* β^**-stability could be verified by*

our approach, for different values of β, and by other existing methods, as reported in [Bri13] for the case of $\beta = 0$. The results obtained by our approach are similar to the least conservative result reported in [Bri13], which was obtained by the method presented in [SP13]. More precisely for the case of $\beta = 0$, β^*-stability could be proven up to $\overline{h} = 1.7294$ using Algorithm 1 with parameters $L = 2$, $k_{max} = 1$ and $N = 1011$. This shows the tightness of our approach since we know that the system becomes unstable for $\overline{h} > 1.7294$ Note that a matrix $H = H^0 \in \mathbb{R}^{10\times 3}$ is used in (3.27) and the computation time was 0.1207 seconds. As for the case of $\beta = 0.06$, we can prove β^*-stability up to $\overline{h} = 0.28981$ using the same parameter setup as that used in the case of $\beta = 0$. The stability condition $\mathcal{S}_k \subseteq int(\mathcal{S}_i)$ was verified for $k = 30$ and $i = 29$. Also, a matrix $H = H^0 \in \mathbb{R}^{10\times 3}$ is used in (3.27) and the computation time was 3.92 seconds.

Example 3. *The second sampled-data control system is also taken from [Bri13], with:*

$$A = \begin{pmatrix} 0 & 1 \\ -2 & 0.1 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad K = \begin{pmatrix} 1 & 0 \end{pmatrix}. \tag{3.30}$$

We set $\underline{h} = 0.4$ and $\beta = 0$. After rewriting the system in the impulsive form (3.4)-(2.24) we remark that the system becomes unstable for $\overline{h} = 1.889$ since the matrix $\prod_{i\in\mathbb{N}_{[1,2]}}(e^{h(A_c'+\beta I_{n'}}A_s')$ has eigenvalues outside the unit circle for $h_1 = 0.4$ and $h_2 = 1.889$. Results obtained by our approach and by several others are also reported in Table 3.2. Our approach has better results than the existing ones since it was able to verify stability for the system up to $\overline{h} = 1.888$, instead of $\overline{h} = 1.828$ for the method presented in [SP13]. Again, our approach appears to be quite tight since the maximal value of $\overline{h}$ for which stability was verified is very close to the known upper bound 1.889. Algorithm 1 was used with parameters $L = 2$, $k_{max} = 30$ and the number of time steps used for the over-approximation of the reachable set is $N = 100$. The stability condition

Table 3.2: Maximum value of $\overline{h}$ for which β^*-stability of systems (3.29) and (3.30) could be proved by our approach and several existing methods, as reported in [Bri13] where $\beta = 0$.

	System (3.29)		**System** (3.30)	
	$\underline{h}$	$\overline{h}$	$\underline{h}$	$\overline{h}$
[Bri13]	10^{-5}	1.7279	0.4	1.827
[FSR04]	10^{-5}	0.869	–	–
[NHT08]	10^{-5}	1.113	–	–
[Fri10]	10^{-5}	1.695	–	–
[LSF10]	10^{-5}	1.695	–	–
[Seu12]	10^{-5}	1.723	0.4	1.251
[SP13]	10^{-5}	1.7294	0.4	1.828
Algorithm 1($\beta = 0$)	10^{-5}	1.72941	0.4	1.888
Algorithm 1($\beta = 0.06$)	10^{-5}	0.28981	0.4	0.709

$\mathcal{S}_k \subseteq int(\mathcal{S}_i)$ *was verified for* $k = 14$ *and* $i = 12$. *Also, a matrix* $H = H^0 \in \mathbb{R}^{18\times 3}$ *is used in (3.27) and the computation time was* 0.824 *seconds.*

Now we set $\beta = 0.04$ *and search for the largest* $\overline{h}$ *such that* β^**-stability is guaranteed. Results are reported in Table 3.2 where stability is guaranteed for* $\overline{h} = 0.309$ *using the same parameter setup as that for the case of* $\beta = 0$. *The stability condition in Algorithm 1,* $\mathcal{S}_k \subseteq int(\mathcal{S}_i)$, *is verified for* $k = 16$ *and* $i = 9$. *Also, a matrix* $H = H^0 \in \mathbb{R}^{30\times 3}$ *is used in (3.27) and the computation time is* 1.76 *seconds.*

Example 4. *We consider the state space plant model of a batch reactor [DHVDWH11], with a static feedback, given by (2.9a-2.9b) with*

$$A = \begin{pmatrix} 1.38 & -0.2077 & 6.715 & -5.676 \\ -0.5814 & -4.29 & 0 & 0.675 \\ 1.067 & 4.273 & -6.654 & 5.893 \\ 0.048 & 4.273 & 1.343 & -2.104 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 0 \\ 5.679 & 0 \\ 1.136 & -3.146 \\ 1.136 & 0 \end{pmatrix}, \tag{3.31}$$

and one of the two feedback gains

$$K_1 = \begin{pmatrix} 0.13 & 0.02 & 0.07 & -0.18 \\ 1.21 & 0.28 & 0.48 & -0.06 \end{pmatrix}, \; K_2 = \begin{pmatrix} 0.41 & -0.45 & 0.38 & -0.59 \\ 1.65 & -0.2 & 0.91 & -0.59 \end{pmatrix}.$$

We rewrite the problem in the form of (3.4)-(2.24). Then, we set $\beta = 0$ *and apply Algorithm 1 to check stability of the impulsive system. We compare our results to those obtained using the NCS toolbox [BvLD*$^+$*12] in Table 3.3. After setting* $\underline{h} = 0.01$ *and* $\underline{\tau} = \overline{\tau} = 0$, *we report the maximal value of* $\overline{h}$ *for which stability has been verified. Note that we conducted an extra experiment labeled "(exp1)" to compare the results in terms of CPU time after fixing the same values of* $\overline{h}$ *in Algorithm 1 and the NCS toolbox.*

The NCS toolbox uses three different approximation methods to embed the timing uncertainty (Jordan Normal Form (JNF), Cayley Hamilton, and Gridding and Norm Bounding (GNB)), so the experiments are conducted as follows: we search for the maximum value of $\overline{h}$ *that guarantees stability by running experiments using the three approximation methods. Then we report the computation time for the experiment in which we obtained this maximum value. In case the maximum bound could be obtained by more than one experiment, we report the CPU time corresponding to the fastest in terms of computation. Stability for system (3.31) with the feedback gains* K_1 *and* K_2 *is guaranteed using the GNB approximation with* 65 *gridpoints. Parameter setups used by Algorithm 1 are summarized by Table 3.6. It is clear, for the systems at hand, that our method is competitive with the NCS toolbox in terms of CPU time and tightness, since Algorithm 1 yields better results for system (3.31) with the feedback gain* K_1 *and has quite similar results for the same system with controller* K_2. *Notice that in this example the dimension of the problem increased where matrices* $H = H^0 \in \mathbb{R}^{88\times 6}$ *and* $H = H^0 \in \mathbb{R}^{80\times 6}$ *were used in computing (3.27) for the former and latter results*

Table 3.3: Results of Algorithm 1 for system (3.31) with feedback gains K_1 and K_2.

		$\underline{h}$	$\overline{h}$	$T_{CPU}(s)$
K_1	NCS toolbox	0.01	0.75	22.8
	Algorithm 1(exp1)	0.01	0.75	9.2
	Algorithm 1(exp2)	0.01	0.80	23.0
K_2	Algorithm 1	0.01	0.582	10.8
	NCS toolbox(exp1)	0.01	0.582	18.1
	NCS toolbox(exp2)	0.01	0.583	19.1

respectively. For gain K_2, we could not obtain a better value for $\overline{h}$ since the Matlab implementation of Algorithm 1 ran into numerical problems when increasing the parameters L or k_{max} or N.

Table 3.4: Parameter setup for Algorithm 1 for system (3.31) with feedback gains K_1 and K_2.

		L	N	k_{max}
K_1	Algorithm 1(exp1)	3	200	100
	Algorithm 1(exp2)	5	100	100
K_2	Algorithm 1	4	100	100

3.3.2 Systems under different timing contract

In this section we study the stability verification problem for systems under the DET contracts given in Section 2.1.1 and the general timing contract (2.2). And for the sake of performing comparisons with existing results in literature we just consider the case of $\beta = 0$.

Example 5. *Given again systems (3.29) and (3.30) we consider the stability verification problem for these two 2-dimensional systems. First, we write the systems into 4-dimensional impulsive systems (3.8). Then, we apply Algorithm 1 to check stability of the impulsive system under several timing contracts. We compare our results to those obtained using the NCS toolbox [BvLD+12] in Table 3.5. For the DET timing contract ($\underline{\tau} = 0$, $\underline{h} = \overline{h} = h$), we fix parameter h and report the maximal value of $\overline{\tau}$ for which stability has been verified. Second, for the general timing contract given by (2.2), we fix parameters $\underline{\tau}$, $\overline{\tau}$, $\underline{h}$ and report the maximal value of $\overline{h}$ for which stability has been verified. Note that we conducted extra experiments labeled "Algorithm 1 (exp1)" to compare the results in terms of CPU time after fixing the same parameters as those used with the NCS toolbox.*

The experiments conducted using the NCS toolbox are done in a particular manner as illustrated previously in Example 4. Stability for system (3.29) is guaranteed using the GNB approximation for the DET and general contracts, with 50, 35, and 50 gridpoints respectively. As for system (3.30), stability is guaranteed using the JNF approximation for all three contracts. Parameter setups used by Algorithm 1, for the different

Table 3.5: Results of Algorithm 1 for systems (3.29) and (3.30) under two timing contracts. T_{CPU} is the computation time in seconds.

		DET ($\underline{\tau}=0$, $\underline{h}=\overline{h}=h$)			General contract (2.2)				
		$\overline{\tau}$	h	T_{CPU}	$\underline{\tau}$	$\overline{\tau}$	$\underline{h}$	$\overline{h}$	T_{CPU}
System (3.29)	NCS toolbox (GNB)	0.63	1	3.42	0	0.4	0.2	1.13	9.17
	Algorithm 1(exp1)	0.63	1	0.18	0	0.4	0.2	1.13	4.49
	Algorithm 1(exp2)	0.67	1	1.16	0	0.4	0.2	1.23	9.95
System (3.30)	NCS toolbox (JNF)	0.78	1	2.07	0	0.1	0.4	0.44	3.62
	Algorithm 1(exp1)	0.78	1	0.41	0	0.1	0.4	0.44	1.13
	Algorithm 1(exp2)	1	1	2.97	0	0.1	0.4	1.71	5.15

Table 3.6: Parameter setup for Algorithm 1 for systems (3.29) and (3.30) under timing contracts.

	DET ($\underline{\tau}=0$, $\underline{h}=\overline{h}=h$)				General contract (2.2)			
	k_{max}	N_1	N_2	L	k_{max}	N_1	N_2	L
System (3.29)(exp1)	30	30	1	2	30	10	10	4
System (3.29)(exp2)	30	100	1	2	30	20	50	4
System (3.30)(exp1)	30	15	1	2	30	10	1	4
System (3.30)(exp2)	30	150	1	2	30	20	60	4

experiments, are summarized by Table 3.6. Note that for the NPILS contract, the parameter N_1 has no effect. It is clear, for the two systems at hand, that our method gives better results than the NCS toolbox in terms of CPU time and tightness.

In the next two sections, we extend the work presented in this chapter to handle two other problems, for the special case of NPILS (3.4)-(2.24), which are the self-triggered control problem and the stability verification problem under stochastic timing contracts. The reader should notice that these two extensions are not directly related to the main scope of the thesis, which is to solve Problems 1, 2, and 3, but only discuss two interesting problems that could be solved based on the stability verification approach presented in this chapter.

3.4 Extension 1: Self-triggered control

This section analyzes and designs the behavior of a sampler in a sampled-data system where the instants at which sampling occurs strongly influence the stability and performance of the overall system. Given the dynamics of the system and the control law, the simplest strategy for a sampler to work is to sample periodically with a fixed sampling period (time-triggered sampling). Alternatively, this period could vary so that sampling occurs only when needed. In fact, implementing sampled-data systems using variable sampling periods is proved to be more efficient in terms of performance and resource utilization [Tab07, DH12, FHPR12]. In literature, two frameworks define the latter strategy: Event-triggered [Tab07] and Self-triggered [MAT09, FHPR12]. The first control strategy requires dedicated hardware to continuously monitor the state of the plant and calls for sampling whenever it is necessary. On the other hand, the second strategy emulates the first one but requires to know the state just at the sampling instants and thus results in less intensive on-line computations.

In the following, we propose a self-triggered control strategy, obtained using reachability analysis, in order to define the sampling period as a function of the state. In other words, we define, using off-line computations, a fixed set of sampling periods as well as some associated regions of the state space. Then in real-time and at each sampling instant, the next sampling period is chosen from the fixed pool depending on the position of the state with respect to the predefined regions.

Our contribution is mainly based on the work presented in Section 3.2.2. Therein, we rely on reachability tools to compute contracting sets [Bla99, Bla91] for a class of difference inclusions. In case of sampled-data systems that could be modeled in the latter formulation the existence of such contracting sets assures stability for all sampling periods defined within a lower and upper-bound. Therefore, we benefit of these sets and design a map from the state-space to a set of sampling periods in order to enlarge the upper-bound found earlier while guaranteeing stability and satisfying, in terms of performance, a specific decay rate.

Let us first formulate the self-triggered control problem and then establish the main result before discussing some applications on sampled-data control systems and comparisons with existing work in the literature.

3.4.1 Problem formulation

The system at hand is the impulsive system (3.4)

$$\begin{aligned} \dot{x}(t) &= A_c' x(t) \quad t \neq t_k^s \\ x(t_k^{s+}) &= A_s' x(t_k^s), \end{aligned} \tag{3.4 revisited}$$

where the inter-sampling delay is:

$$h_k = t^s_{k+1} - t^s_k \in [\underline{h}, \theta(x(t^s_k))],\ \forall k \in \mathbb{N}, \tag{3.32}$$

with $\theta : \mathbb{R}^{n'} \to \mathbb{R}^+$ and $\underline{h} \in \mathbb{R}^+$ a given lower bound on h_k to avoid Zeno phenomena. The notion of stability studied for (3.4)-(3.32) is the β^*-stability given by Definition 8.

The following section provides a solution to the self-triggered problem defined by:

Problem 5 (Self-triggered control)**.** *Given $A'_c, A'_s \in \mathbb{R}^{n'\times n'}$, $\underline{h} \in \mathbb{R}^+$, and $\beta \in \mathbb{R}^+$ as a performance measure, define a strategy (3.32) that renders (3.4) β^*-stable while enlarging $\theta(x(t^s_k))$, for all $k \in \mathbb{N}$.*

3.4.2 Self-triggered control synthesis

In this section we propose an approach to solve Problem 5. Our approach is divided into two distinct parts. Primarily, we fix the value of $\theta(x)$, for all $x \in \mathbb{R}^{n'}$, to a given value $\overline{h} > \underline{h}$ and compute a contracting polytope that ensures β^*-stability of (3.4)-(3.32) with $\theta(x(t^s_k)) = \overline{h}$, for all $k \in \mathbb{N}$. Next, we use the computed set to enlarge $\theta(x)$, in (3.32), based on the position of x in the state space.

3.4.2.1 Finding the contracting set

Let us define the map $\Phi : 2^{\mathbb{R}^{n'}} \to 2^{\mathbb{R}^{n'}}$, given for all $\mathcal{S} \subseteq \mathbb{R}^{n'}$ and $h, h' \in \mathbb{R}^+$ with $h \leq h'$ by

$$\Phi_{[h,h']}(\mathcal{S}) = \bigcup_{\tau\in[h,h']} e^{\tau(A'_c+\beta I_{n'})} A'_s \mathcal{S}. \tag{3.33}$$

Notice that Φ in this case is equivalent to (3.26) for $[h, h'] = [\underline{h}, \overline{h}]$. Thus Φ in (3.33) satisfies Assumptions 2 and 3 and consequently all the properties given by Lemma 5. Therefore, the following result holds:

Corollary 7. *Let $\mathcal{S} \in \mathcal{B}_0(\mathbb{R}^{n'})$, $\beta \in \mathbb{R}^+$, $\underline{h} \in \mathbb{R}^+$, and $\overline{h} \in \mathbb{R}^+$. System (3.4)-(3.32) is β^*-stable, with $\theta(x(t^s_k)) = \overline{h}$ for all $k \in \mathbb{N}$, if and only if there exist $l \in \mathbb{N}^+$ and $\epsilon \in (0, 1)$ such that*

$$\Phi_{[\underline{h},\overline{h}]}(\mathcal{P}) \subseteq \epsilon\mathcal{P}, \tag{3.34}$$

where $\mathcal{P} = ch\left(\bigcup_{j=0}^{l-1} \epsilon^{-j} \Phi^j_{[\underline{h},\overline{h}]}(\mathcal{S})\right)$.

Proof: The Corollary is a direct consequence of Proposition 3 and Theorem 17. □

The idea to synthesize a contracting polytope

$$\mathcal{P} = \{x \in \mathbb{R}^{n'} : Hx \leq 1\}, H \in \mathbb{R}^{r \times n'}, \tag{3.35}$$

for (3.4)-(3.32), with $\theta(x(t_k^s)) = \overline{h}$ for all $k \in \mathbb{N}$, is inspired from Corollary 7 which defines in theory an explicit form of $\mathcal{P}$ whenever the system is β^*-stable.

It is often impossible to exactly compute Φ. Thus we use as in Section 3.2.3 an over-approximation $\overline{\Phi} : \mathcal{B}_0(\mathbb{R}^{n'}) \to \mathcal{B}_0(\mathbb{R}^{n'})$ satisfying Assumption 4 where the iterates of the map $\overline{\Phi}$ are defined similarly to those of Φ. In addition, we rely on the effective computation of the over-approximation $\overline{\Phi}$ given by (3.27) and which could be rewritten as:

$$\overline{\Phi}_{[h,h']}(\mathcal{S}) = \Gamma_H \left(\mathrm{ch}\big(\overline{\mathcal{R}}_{[0,h'-h]}^{(A_c'+\beta I_{n'})}(e^{h(A_c'+\beta I_{n'})} A_s' \mathcal{S})\big)\right). \tag{3.36}$$

We refer the reader to Section 3.2.4.1 for an efficient computation of the initial set $\mathcal{S}$ in which the set $\overline{\Phi}_{[h,h']}(\mathcal{S})$ is indeed a polytope for any $\mathcal{S} \in \mathcal{B}_0(\mathbb{R}^{n'})$.

In the following result, we synthesize the contracting set based on the map $\overline{\Phi}$.

Corollary 8. *Let $\beta \in \mathbb{R}^+$, $\underline{h} \in \mathbb{R}^+$, and $\overline{h} \in \mathbb{R}^+$. Under Assumption 4, if there exist $\mathcal{S} \in \mathcal{B}_0(\mathbb{R}^{n'})$ and $(k, i, \rho) \in \mathbb{N}^+ \times \mathbb{N}_{[0,k-1]} \times (0,1)$ such that*

$$\overline{\Phi}^k(\mathcal{S}) \subseteq \rho \overline{\Phi}^i(\mathcal{S}), \tag{3.37}$$

then

(a) System (3.4)-(3.32) is β^*-stable, with $\theta(x(t_k^s)) = \overline{h}$ for all $k \in \mathbb{N}$.

(b) There exists $\epsilon \in (0,1)$ such that $\Phi_{[\underline{h},\overline{h}]}(\mathcal{P}) \subseteq \epsilon \mathcal{P}$,

where

$$\mathcal{P} = ch\left(\bigcup_{j=0}^{k-1} \epsilon^{-j} \overline{\Phi}_{[\underline{h},\overline{h}]}^j(\mathcal{S}) \right). \tag{3.38}$$

Proof: Let $\mathcal{S}' = \bigcup_{j=0}^{k-1} \epsilon^{-j} \overline{\Phi}^j(\mathcal{S})$ with $\epsilon = \rho^{\frac{1}{k}}$ then $\mathcal{P}$ in (3.38) is equal to $ch(\mathcal{S}')$.

(a) It follows directly from Corollary 4 and Proposition 3.

(b) It follows from the proof of Corollary 4 that

$$\Phi_{[\underline{h},\overline{h}]}(\mathcal{S}') \subseteq \epsilon \mathcal{S}'. \tag{3.39}$$

Now using (3.39) and the fact that Φ satisfies Assumption 3 we have

$$\begin{aligned}\Phi_{[\underline{h},\overline{h}]}(\mathcal{P}) &\subseteq \Phi_{[\underline{h},\overline{h}]}(ch(\mathcal{S}')) \subseteq ch(\Phi_{[\underline{h},\overline{h}]}(\mathcal{S}')) \\ &\subseteq ch(\epsilon\mathcal{S}') = \epsilon\mathcal{P}. \end{aligned} \tag{3.40}$$

□

Practically, we compute the contracting set (3.35) as the following: we start iterating forward from a chosen set $\mathcal{S}$ by computing, at each iteration k, $\overline{\Phi}^k_{[\underline{h},\overline{h}]}(\mathcal{S})$ until condition (3.37) is satisfied. Consequently, Corollary 8 allows us to set $\mathcal{P}$ as given by (3.38).

3.4.2.2 Sampling strategy design

Suppose that for a given $\overline{h} \in \mathbb{R}^+$ we have a contracting set $\mathcal{P}$ for (3.4-3.32), with $\theta(x(t^s_k)) = \overline{h}$ for all $k \in \mathbb{N}$. We intend further to increase the upper-bound on sampling, i.e. $\overline{h}$, for some regions in the state space while conserving β^*-stability.

We consider a polytopic covering of $q \in \mathbb{N}$ polytopes $\{\mathcal{P}_s : s \in \mathbb{N}_{[1,q]}\}$, such that

$$\mathcal{P} = \bigcup_{s=1}^{q} \mathcal{P}_s, \tag{3.41}$$

and a set of sampling periods $\{h_s \geq \overline{h} : s \in \mathbb{N}_{[1,q]}\}$, such that

$$\overline{\Phi}_{[\underline{h},h_s]}(\mathcal{P}_s) \subseteq \text{int}(\mathcal{P}). \tag{3.42}$$

Two coverings are suggested in the next section: the first relies on the facets of the contracting polytope $\mathcal{P}$ and the second on the discrete-time behavior of the system. In fact, the latter is inspired by [FHPR12]; therein conic coverings are computed instead of polytopic ones. Now we define a sampling strategy as (3.32) with

$$\theta(x) = \max\{h_s \in \{h_1, \ldots, h_q\} : x \in \gamma(x)\mathcal{P}_s\}, \tag{3.43}$$

where

$$\gamma(x) = \min\{\gamma \in \mathbb{R}^+ : x \in \gamma\mathcal{P}\}. \tag{3.44}$$

Eventually, the following instumental result solves Problem 5.

Theorem 19. *Given a contracting set $\mathcal{P}$ by (3.35), a set of polytopic coverings $\{\mathcal{P}_s : s \in \mathbb{N}_{[1,q]}\}$ satisfying (3.41), a set of sampling periods $\{h_s \geq \overline{h} : s \in \mathbb{N}_{[1,q]}\}$ satisfying (3.42), and a performance measure $\beta \in \mathbb{R}^+$, then under Assumption 4 (3.4)-(3.32) is β^*-stable with θ given by (3.43).*

Proof: The state of (3.4-3.32) at any sampling instant t^s_{k+1}, $k \in \mathbb{N}$ is given by

$$\begin{aligned} x_{k+1} &= e^{A_c\Delta_k}A_r x_k \\ &= e^{-\Delta_k\beta}e^{(A_c+\beta I)\Delta_k}A_r x_k \quad \forall \Delta_k \in [\underline{h}, \theta(x_k)]. \end{aligned}$$

Then there exist $h_s = \theta(x_k)$ and $\mathcal{P}_s$ such that

$$x_{k+1} \in e^{-\Delta_k\beta}\Phi_{[\underline{h},h_s]}(\gamma(x_k)\mathcal{P}_s) \quad \forall \Delta_k \in [\underline{h}, \theta(x_k)]. \tag{3.45}$$

Using (3.42), properties of Φ in Lemma 5, and Assumption 4 we get

$$\begin{aligned} x_{k+1} &\in e^{-\Delta_k\beta}\gamma(x_k)\Phi_{[\underline{h},h_s]}(\mathcal{P}_s) \subseteq e^{-\Delta_k\beta}\gamma(x_k)\overline{\Phi}_{[\underline{h},h_s]}(\mathcal{P}_s) \\ &\subseteq \text{int}(e^{-\Delta_k\beta}\gamma(x_k)\mathcal{P}) \quad \forall \Delta_k \in [\underline{h}, \theta(x_k)]. \end{aligned}$$

In other words there exists $\epsilon \in (0, 1)$ such that

$$x_{k+1} \in \epsilon e^{-\Delta_k\beta}\gamma(x_k)\mathcal{P} \quad \forall \Delta_k \in [\underline{h}, \theta(x_k)]. \tag{3.46}$$

The definition of γ gives

$$x_{k+1} \in \gamma(x_{k+1})\mathcal{P} \subseteq \epsilon e^{-\Delta_k\beta}\gamma(x_k)\mathcal{P} \quad \forall \Delta_k \in [\underline{h}, \theta(x_k)]. \tag{3.47}$$

This implies that for all $\Delta_i \in [\underline{h}, \theta(x_i)]$

$$\gamma(x_{k+1}) \leq \epsilon e^{-\Delta_k\beta}\gamma(x_k) \leq \cdots \leq \epsilon^{k+1}e^{-\sum_{i=0}^{k}\Delta_i\beta}\gamma(x_0). \tag{3.48}$$

Since $\mathcal{P} \in \mathcal{B}_0(\mathbb{R}^{n'})$, then there exist $\underline{c} \in \mathbb{R}^+$, $\overline{c} \in \mathbb{R}^+$ such that $\underline{c}\mathcal{B} \subseteq \mathcal{P} \subseteq \overline{c}\mathcal{B}$. Thus for any $x \in \mathbb{R}^n$:

$$\frac{|x|}{\overline{c}} \leq \gamma(x) \leq \frac{|x|}{\underline{c}}. \tag{3.49}$$

Using (3.47), (3.48), (3.49), and $t_k^s = \sum_{i=0}^{k-1} \Delta_i$ for all $\Delta_i \in [\underline{h}, \theta(x_i)]$ yields

$$|x_k| \leq \epsilon^i e^{-t_k^s \beta} \frac{\overline{c}}{\underline{c}} |x_0|. \tag{3.50}$$

Now, let $t \in \mathbb{R}^+$ and $k \in \mathbb{N}$ be such that $t \in (t_k^s, t_{k+1}^s]$, then $t - t_k^s \leq h_{max}$ and $k \geq t/\overline{h} - 1$, with $h_{max} = \max_{i \in \{1,\ldots,q\}} h_i$. Moreover,

$$\begin{aligned} |x(t)| &\leq \epsilon^i e^{|A_c| h_{max}} |A_r| e^{-t_k \beta} \frac{\overline{c}}{\underline{c}} |x_0| \\ &\leq \frac{e^{(|A_c|+\beta) h_{max}} |A_r| \frac{\overline{c}}{\underline{c}}}{\epsilon} e^{-t(\beta - \frac{ln \epsilon}{h_{max}})} |x_0|, \end{aligned}$$

which finishes the proof since $\epsilon \in (0, 1)$.

□

Note that Theorem 19 guarantees robustness of the sampling strategy in the sense that at any $t_k^s \in \mathbb{R}^+$, the next sampling instant t_{k+1}^s can take any value within $[t_k^s + \underline{h}, t_k^s + \theta(x(t_k^s))]$, while guaranteeing β^*-stability. Furthermore, a consequence of Theorem 19 is that the map γ, given by (3.44), is a set-induced Lyapunov function [BM07] for (3.4)-(3.32) that obviously decreases at the sampling times t_k^s for all $k \in \mathbb{N}$.

3.4.2.3 Polytopic covering

We propose two different methods to compute a polytopic covering $\{\mathcal{P}_s : s \in \mathbb{N}_{[1,q]}\}$ satisfying (3.41).

- **Method 1: Using the facets of the contracting polytope**

  Let the contracting set $\mathcal{P}$ be given in the form (3.35), where $H \in \mathbb{R}^{r \times n}$, then $\mathcal{P}_s$ are defined for all $s \in \mathbb{N}_{[1,q]}$ by

  $$\mathcal{P}_s = \{x \in \mathbb{R}^{n'} : H_s x \leq 1, (H_i - H_s)x \leq 0 \ \ \forall i \neq s\}, \tag{3.51}$$

  with $q = r$ as the number of facets of $\mathcal{P}$ and H_s as the $s-$th row of H.

  Note that with this method no additional off-line computations are required after we compute $\mathcal{P}$. As for the online computations, given the state at a sampling instant, i.e. $x(t_k^s)$, the latest next sampling is defined as

  $$t_{k+1}^s = t_k^s + \mathsf{max}\{h_k : k = \mathsf{argmax}_{s \in \mathbb{N}_{[1,q]}} H_s x(t_k^s)\},$$

  which requires only q multiplications of $n'-$dimensional vectors and one argmax operation.

- **Method 2: Using the discrete-time behavior of the system**

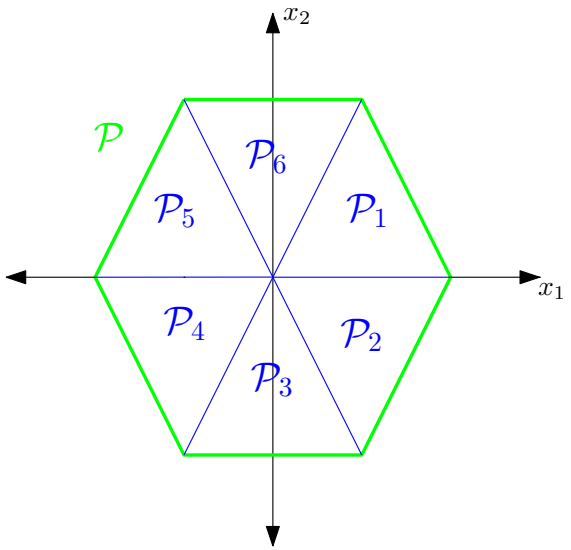


Figure 3.3: Covering the contracting polytope $\mathcal{P}$ of dimension 2 with $q = 6$ polytopic regions $\mathcal{P}_s$ using Method 1.

Given a scalar $\sigma > \overline{h}$, $q \in \mathbb{N}$, and $H \in \mathbb{R}^{r \times n}$ as the matrix defining the contracting set $\mathcal{P}$ in (3.35), we define a set of sampling times $\{T_s = \overline{h} + (s-1)\frac{\sigma - \overline{h}}{q-1} : s \in \mathbb{N}_{[1,q]}\}$. Then $\mathcal{P}_s$ are defined for all $s \in \mathbb{N}_{[1,q]}$ by

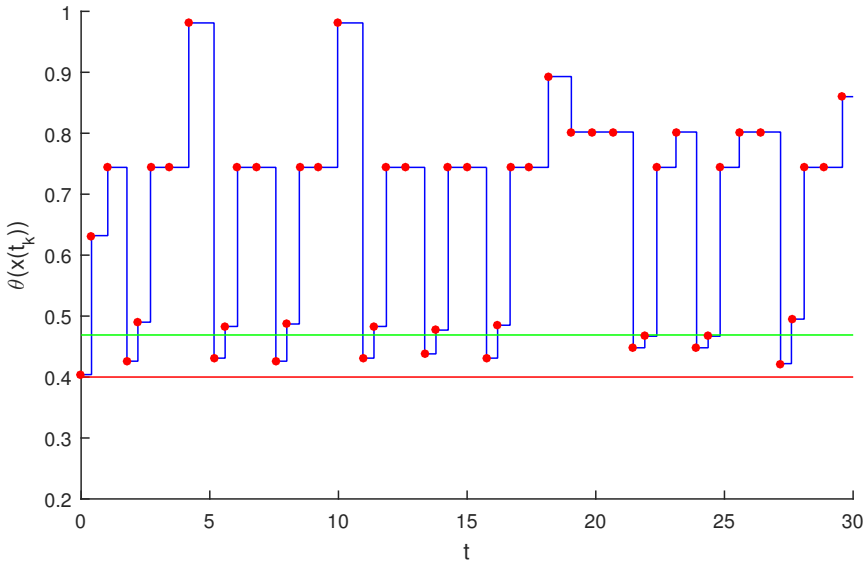
$$\mathcal{P}_s = \{x \in \mathbb{R}^{n'} : \begin{pmatrix} H \\ He^{T_1(A'_c + \beta I_{n'})}A_s \\ \vdots \\ He^{T_s(A'_c + \beta I_{n'})}A_s \end{pmatrix} x \leq 1\}. \tag{3.52}$$

In this case the additional off-line computations required, after finding $\mathcal{P}$, are those that correspond for computing (3.52) for all $s \in \mathbb{N}_{[1,q]}$. Also this method is more complex than Method 1 for on-line computations since at each state $x(t_k^s)$ the latest next sampling instant is given by

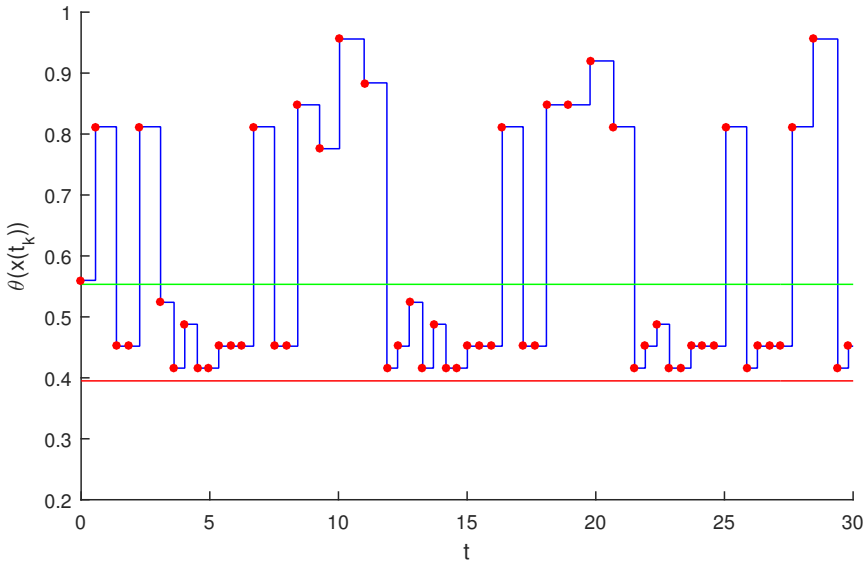
$$t_{k+1}^s = t_k^s + \mathsf{max}\{h_s \in \{1, \ldots, q\} : x(t_k^s) \in \mathcal{P}_s\},$$

which requires at most one max operation, $\sum_{s=1}^{q}(r \times s)$ multiplications of $n'-$dimensional vectors, and the same latter number of inequality checks.

We remark that in fact the on-line computations are reduced by half in both methods since the contracting set $\mathcal{P}$ is practically projected on the first $p = \frac{n'}{2}$ dimensions and hence all vectors' dimensions will be reduced by half. This results from the fact that the deleted dimensions correspond to the error e which is null for (3.4) – (3.32) at all sampling instants t_k^s.



(a)



(b)

Figure 3.4: Inter-execution times $\theta(x(t_k^s))$, in Example 6, for a decay rate (a) $\beta = 0$ using Method 1 for covering $\mathcal{P}$ and (b) $\beta = 0.05$ using Method 2.

3.4.3 Numerical results

We conduct several experiments to validate the efficiency of our proposed self-triggered control approach. In the sequel, we also compare our results with existing approaches in literature. Our implementation relies on the matlab Mpt toolbox [HKJM13].

Example 6. *Consider the following system from [FHPR12] given by (2.9a-2.9b) with:*

$$A = \begin{pmatrix} -0.5 & 0 \\ 0 & 3.5 \end{pmatrix}, B = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, K = \begin{pmatrix} 1.02 & -5.62 \end{pmatrix}.$$

Reformulating the problem into an impulsive linear system (3.4)-(3.32), the contracting set $\mathcal{P}$ is computed with $h_k \in [0.01, 0.4]$. This implies that the maximum value of h in (2.2) is at least $\overline{h} = 0.4s$. After covering $\mathcal{P}$ with $q = 272$ polytopes using Method 1 in Section 3.4.2.3, sampling intervals $h_1, \ldots, h_{272}$ are defined such that (3.42) holds for $\beta = 0$. For a constant sampling greater than $T_{max} = 0.469s$ the system becomes unstable. Whereas, we can go with our approach beyond the limit T_{max} for some regions of the state space (up to $0.981s$).

We fix $\beta = 0$ and run simulations from 1000 different initial positions, uniformly distributed on the unit circle, with a duration corresponding to 30 resets for each one. If we sample each time with $h_k = \theta(x(t_k^s)), \forall k \in \mathbb{N}$, the resulting average inter-sampling time is $T_{av} = 0.676s > T_{max}$. Considering Method 2, we cover $\mathcal{P}$ with $q = 16$ polytopes after setting $\sigma = 0.96$. Then we get an average sampling interval of $T_{av} = 0.77$ for the same previous experiment.

Now we set $\beta = 0.05$ and compute a contracting set $\mathcal{P}$ with $h_k \in [0.01, 0.38]$. After covering $\mathcal{P}$ with $q = 16$ polytopes, using Method 2 with $\sigma = 0.96$, we rerun the simulations from 1000 different initial positions as done previously to get an average inter-sampling interval of $T_{av} = 0.5913s > T_{max}$. For the two cases of $\beta = 0$ and $\beta = 0.05$, Figure 3.4 shows, for a random initial state, the sampling intervals (blue/piecewise constant curve), with the lower-bound of the off-line computed state dependent sampling function (red/lower-horizontal line), and the limit T_{max} of the periodic case (green/upper horizontal line). The sampling times are represented by the red dots assuming that we are always sampling with $h_k = \theta(x(t_k^s)), \forall k \in \mathbb{N}$.

Example 7. *We cite another example with higher dimension ($p = 4$) from [FHPR12] given by (2.9a-2.9b) with:*

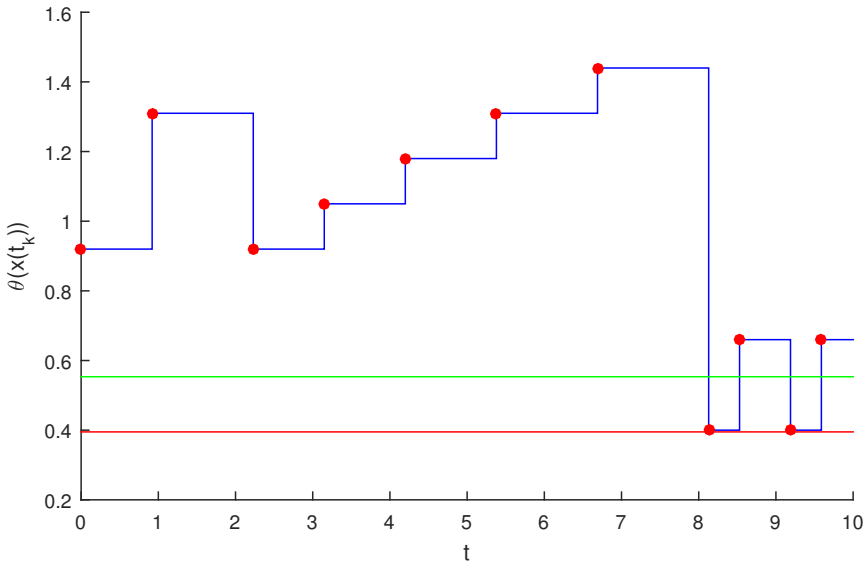


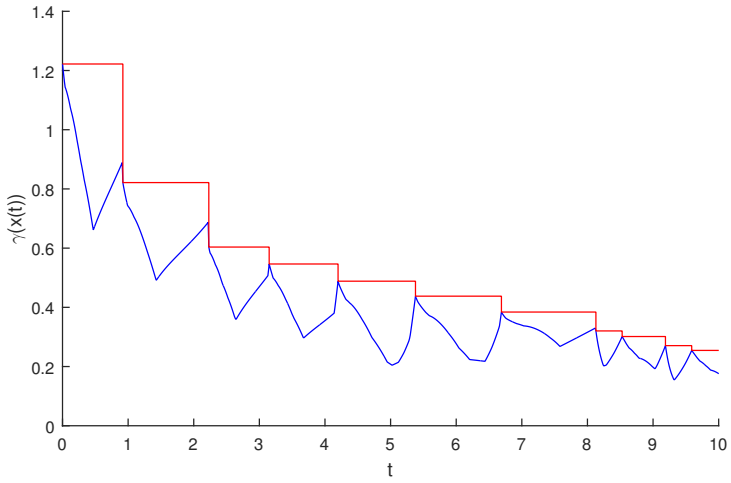
Figure 3.5: Inter-execution times $\theta(x(t_k^s))$, in Example 7, for a decay rate $\beta = 0$.

$$A = \begin{pmatrix} 1.38 & -0.2 & 6.71 & -5.67 \\ -0.58 & -4.29 & 0 & 0.67 \\ 1.06 & 4.27 & -6.65 & 5.89 \\ 0.04 & 4.27 & 1.34 & -2.1 \end{pmatrix}, B = \begin{pmatrix} 0 & 0 \\ 5.67 & 0 \\ 1.13 & -3.14 \\ 1.13 & 0 \end{pmatrix},$$

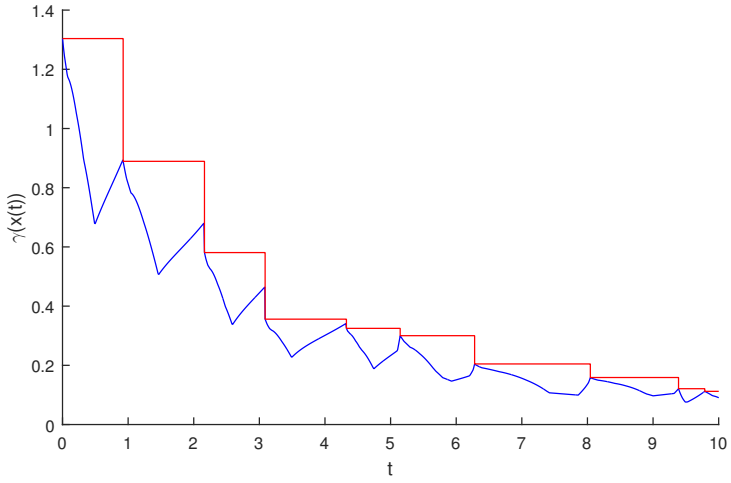
$$K = \begin{pmatrix} -0.1006 & 0.2469 & 0.0952 & 0.2447 \\ -1.4099 & 0.1966 & -0.0139 & -0.0823 \end{pmatrix}.$$

Reformulating the problem into an impulsive linear system (3.4)-(3.32), we compute the contracting set $\mathcal{P}$ *for* $h_k \in [0.05, 0.4]$.

We cover $\mathcal{P}$ *with* $q = 13$ *polytopes using Method 2 in Section 3.4.2.3 after setting* $\sigma = 2.1$. *Correspondingly we have* 13 *different sampling intervals given by* $\{h_1, \ldots, h_{13}\} = \{0.4, \ldots, 2.1\}$. *We check then that (3.42) holds for* $\beta = 0$ *and run a simulation for* 10s *to validate our results. Albeit for a constant sampling greater than* $T_{max} = 0.553s$ *the system becomes unstable we can go with our approach up to* 2.1s *for some regions of the state space and sample in average by* $T_{av} = 0.746s$. *These results are comparable with those in literature where for the first 10s, [MAT09] actuated* 32 *times in the best mentioned case, [FHPR12] sampled* 17 *times, and as Figure 3.5 shows only* 11 *samplings were required using our method.*



(a)



(b)

Figure 3.6: The Lyapunov function $\gamma(x(t))$, in Example 7, for (a) $\beta = 0$ and (b) $\beta = 0.06$.

Now, we take $\beta = 0.06$ *and an arbitrary initial position. Using a similar covering as in the previous case, Figure 3.6 shows the set-induced Lyapunov function* $\gamma(x)$ *which obviously decreases at the sampling instants* t_k^s *ensuring* β^**-stability for the two cases* $\beta = 0$ *and* $\beta = 0.06$.

3.5 Extension 2: Stability verification under stochastic timing contracts

In this section, we extend the stability verification approach on NPILS to stochastic systems. *Stochastic impulsive linear systems* (SILS), considered in this section, take the same form as (3.4)

$$\begin{aligned} \dot{x}(t) &= A_c' x(t) \quad t \neq t_k^s \\ x(t_k^{s+}) &= A_s' x(t_k^s), \end{aligned} \tag{3.4 revisited}$$

with independent and identically distributed (i.i.d.) random durations between resets:

$$t_0 = 0,\ t_{k+1}^s - t_k^s = \underline{\tau} + \delta_k,\ \delta_k \sim \mathcal{U}([0, \Delta]), \text{i.i.d. } k \in \mathbb{N}, \tag{3.53}$$

where $\Delta \in \mathbb{R}^+$ and $\mathcal{U}([0, \Delta])$ is the uniform distribution over $[0, \Delta]$. Let us remark that the method presented in this section can be easily extended to other types of probability distributions with compact support. We consider the following notion of stability for stochastic systems:

Definition 10 (GMES)**.** *The SILS (3.4)-(3.53) is globally mean exponentially stable (GMES) if there exist* $\lambda \in \mathbb{R}^+$ *and* $C \in \mathbb{R}^+$ *such that the solutions of (3.4)-(3.53) verify:*

$$\mathbb{E}[|x(t)|] \leq Ce^{-\lambda t} |x(0)|,\ \forall t \in \mathbb{R}^+.$$

3.5.1 Sufficient stability condition

Let $\mathcal{S} \in \mathcal{B}_0(\mathbb{R}^{n'})$, in the following we provide a sufficient condition for GMES based on a map $\rho_{\mathcal{S}} : [0, \Delta] \to \mathbb{R}^+$ satisfying the following assumption:

Assumption 5. *Let* $\mathcal{S} \in \mathcal{B}_0(\mathbb{R}^{n'})$*, for all* $\tau \in [0, \Delta]$*,* $e^{(\underline{\tau}+\delta)A_c'} A_s' \mathcal{S} \subseteq \rho_{\mathcal{S}}(\delta)\mathcal{S}$*.*

Then, we can state the following stability condition:

Proposition 5. *Under Assumption 5, if there exists a set* $\mathcal{S} \in \mathcal{B}_0(\mathbb{R}^{n'})$ *such that*

$$\rho_{\mathcal{S}}^* = \mathbb{E}[\rho_{\mathcal{S}}(\delta)] < 1 \text{ where } \delta \sim \mathcal{U}([0, \Delta]),$$

then SILS (3.4)-(3.53) is GMES.

Proof: Since $\mathcal{S} \in \mathcal{B}_0(\mathbb{R}^{n'})$, then there exist $\underline{c} \in \mathbb{R}^+$, $\overline{c} \in \mathbb{R}^+$ such that $\underline{c}\mathcal{B} \subseteq \mathcal{S} \subseteq \overline{c}\mathcal{B}$. Now consider a trajectory x of SILS (3.4)-(3.53), then $x(0) \in |x(0)|\mathcal{B} \subseteq \frac{|x(0)|}{\underline{c}}\mathcal{S}$ and for all $i \in \mathbb{N}$

$$x(t_i) \in \left(\prod_{k=0}^{i-1}(e^{(\underline{\tau}+\delta_k)A_c'}A_s')\right)\frac{|x(0)|}{\underline{c}}\mathcal{S}.$$

Then, Assumption 5 yields

$$x(t_i) \in \left(\prod_{k=0}^{i-1}\rho_{\mathcal{S}}(\delta_k)\right)\frac{|x(0)|}{\underline{c}}\mathcal{S}$$

and therefore

$$|x(t_i)| \leq \left(\prod_{k=0}^{i-1}\rho_{\mathcal{S}}(\delta_k)\right)\frac{\overline{c}}{\underline{c}}|x(0)|.$$

Then, we have

$$\mathbb{E}[|x(t_i)|] \leq (\rho_{\mathcal{S}}^*)^i\frac{\overline{c}}{\underline{c}}|x(0)|. \tag{3.54}$$

Let $t \in \mathbb{R}^+$ and $i(t) \in \mathbb{N}$ be such that $t \in (t_{i(t)}, t_{i(t)+1}]$, then $t - t_{i(t)} \leq \underline{\tau} + \Delta$, and it follows

$$|x(t)| \leq e^{|A_c'|(\underline{\tau}+\Delta)}|A_s'||x(t_{i(t)})|.$$

Then,

$$\mathbb{E}[|x(t)|] \leq e^{|A_c'|(\underline{\tau}+\Delta)}|A_s'|\mathbb{E}[|x(t_{i(t)})|]. \tag{3.55}$$

Let us remark that $t/(\underline{\tau}+\Delta) \leq i(t) \leq t/\underline{\tau}$ which yields

$$\begin{aligned}\mathbb{E}[|x(t_{i(t)})|] &= \sum_{t/(\underline{\tau}+\Delta)\leq j\leq t/\underline{\tau}}\mathbb{E}[|x(t_j)|]P(i(t)=j)\\ &\leq \max_{t/(\underline{\tau}+\Delta)\leq j\leq t/\underline{\tau}}\mathbb{E}[|x(t_j)|]\end{aligned} \tag{3.56}$$

Then, $\rho_{\mathcal{S}}^* < 1$, (3.54), (3.55) and (3.56) give

$$\begin{aligned}\mathbb{E}[|x(t)|] &\leq e^{|A_c'|(\underline{\tau}+\Delta)}|A_s'|(\rho_{\mathcal{S}}^*)^{t/(\underline{\tau}+\Delta)}\frac{\overline{c}}{\underline{c}}|x(0)|\\ &\leq \frac{e^{|A_c'|(\underline{\tau}+\Delta)}|A_s'|\overline{c}'}{\underline{c}'}e^{\frac{\ln(\rho_{\mathcal{S}}^*)}{\underline{\tau}+\Delta}t}|x(0)|.\end{aligned}$$

Since $\rho_{\mathcal{S}}^* < 1$, SILS (3.4)-(3.53) is GMES. □

3.5.2 Stability verification

We now present an approach based on reachability analysis for computing a function $\rho_{\mathcal{S}}$ satisfying Assumption 5.

Let us consider a polytope $\mathcal{P} = \{x \in \mathbb{R}^{n'} : Hx \leq b\}$ where the matrix $H \in \mathbb{R}^{r \times n'}$ is such that $0 \in \text{int}(\text{ch}(\{H_1, \ldots, H_r\}))$ and $b_i \geq 0$ for all $i \in \mathbb{N}_{[1,m]}$. Then, $\mathcal{P} \in \mathcal{B}_0(\mathbb{R}^{n'})$.

Proposition 6. *Let $\rho_{\mathcal{P}} : [0, \Delta] \to \mathbb{R}^+$ be given by*

$$\rho_{\mathcal{P}}(\delta) = \rho_i, \text{ if } \delta \in [(i-1)h, ih], i \in \mathbb{N}_{[1,N]}$$

where $N \in \mathbb{N}^+$, $h = \Delta/N$ is the time step, and ρ_i satisfies for $i \in \mathbb{N}_{[1,N]}$

$$\overline{\mathcal{R}}^{A'_c}_{[(i-1)h,ih]}(e^{A'_c \underline{\tau}} A'_s \mathcal{P}) \subseteq \rho_i \mathcal{P},$$

with $\overline{\mathcal{R}}^{A'_c}_{[(i-1)h,ih]}(e^{A'_c \underline{\tau}} A'_s \mathcal{P})$ computed as in Theorem 15. Then, $\rho_{\mathcal{P}}$ satisfies Assumption 5.

Proof: Let $\delta \in [0, \Delta]$, let $i \in \mathbb{N}_{[1,N]}$ such that $\delta \in [(i-1)h, ih]$. Then, from Theorem 15

$$\begin{aligned} e^{(\underline{\tau}+\delta)A'_c} A'_s \mathcal{P} &\subseteq \mathcal{R}^{A'_c}_{[(i-1)h,ih]}(e^{A'_c \underline{\delta}} A'_s \mathcal{P}) \\ &\subseteq \overline{\mathcal{R}}^{A'_c}_{[(i-1)h,ih]}(e^{A'_c \underline{\tau}} A'_s \mathcal{P}) \subseteq \rho_i \mathcal{P} = \rho_{\mathcal{P}}(\delta)\mathcal{P}. \end{aligned}$$

□

Then, stability can be effectively verified using the following result:

Corollary 9. *Let ρ_i, $i \in \mathbb{N}_{[1,N]}$ be computed as in Proposition 6, if*

$$\sum_{i=1}^{N} \rho_i < N$$

then SILS (3.4)-(3.53) is GMES.

Proof: For $\delta \sim \mathcal{U}([0, \Delta])$, we have

$$\begin{aligned} \mathbb{E}[\rho_{\mathcal{P}}(\delta)] &= \frac{1}{\Delta} \int_0^{\Delta} \rho_{\mathcal{S}}(\delta) d\delta = \frac{1}{\Delta} \sum_{i=1}^{N} \int_{(i-1)h}^{ih} \rho_i d\delta \\ &= \frac{1}{N} \sum_{i=1}^{N} \rho_i. \end{aligned}$$

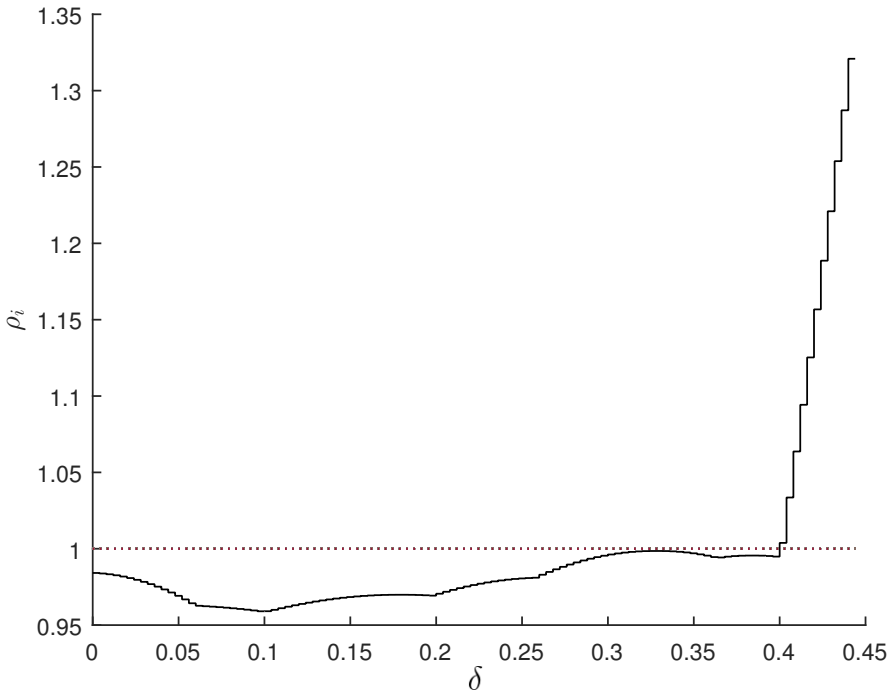


Figure 3.7: $\rho_{\mathcal{P}}(\delta)$ for the polytope $\mathcal{P}$ given by the initial polytope $\mathcal{P}_0$ computed as in Example 1 by Algorithm 1 with parameter setup C.

Hence $\sum_{i=1}^{N} \rho_i < N$ is equivalent to $\mathbb{E}[\rho_{\mathcal{P}}(\delta)] < 1$. Then, Proposition 5 allows us to conclude. □

3.5.3 Numerical results

Example 8. *We consider, as in Example 1, the system (3.4) with matrices*

$$A_c' = \begin{pmatrix} 0 & -3 & 1 \\ 1.4 & -2.6 & 0.6 \\ 8.4 & -18.6 & 4.6 \end{pmatrix}, \quad A_s' = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \qquad \text{(3.28 revisited)}$$

where the inter-sampling delay satisfies (3.53). Figure 3.7 *illustrates the result on stochastic impulsive linear systems of Section 3.5. The figure shows the graph of the function $\rho_{\mathcal{P}}(\delta)$ defined in Proposition 6 for the polytope $\mathcal{P}$ given by the initial polytope $\mathcal{P}_0$ computed in Example 1 by Algorithm 1 with parameter setup C. One can check that $\rho_{\mathcal{P}}(\delta) < 1$ for $\delta \in [0, 0.4]$ which shows that the NPILS is GUES for $\delta = 0.4$. Then, we can check that the condition given by Proposition 5 for GMES of SILS is verified for $\delta = 0.444$. The computation time was* 2.16 *seconds.*

Chapter 4

Scheduling of embedded controllers under timing contracts

Abstract

We adopt in this chapter the point of view of the real-time engineer who has to implement several controllers, each subject to a timing contract on a given number of shared CPUs. In other words, we have to solve the scheduling problem, i.e. Problem 2. Given a set of controllers, each of which is subject to a timing contract, and best and worst case execution times for each control task on each CPU, we synthesize a dynamic scheduling policy, which guarantees that each timing contract is satisfied and that each of the shared CPUs are allocated to at most one embedded controller at any time. The approach is based on a timed game formulation that allows us to write the scheduling problem as a timed safety game. Then using the tool UPPAAL-TIGA [BCD+07], solutions to the safety game provides a suitable scheduling policy. In addition, we provide a novel necessary and sufficient condition for schedulability of the control tasks based on a simplified timed game automaton. Most results of this chapter are published in [AKGD17a] for the case of a single shared CPU.

The real-time engineering point of view is considered in this chapter while implementing several controllers, each subject to a timing contract, on a number of shared CPUs. Given best and worst case execution times for each control task on each CPU, we synthesize a dynamic scheduling policy, which guarantees that each timing contract is satisfied and that each of the shared CPU is allocated to at most one controller at any time. Our approach is based on the use of timed automata (the reader is referred to Section 4.1.1 for defining timed and timed game automata) where we show that the scheduling problem can be formulated as a times safety game, which can be solved by the tool UPPAAL-TIGA, and whose solution provides a suitable scheduling policy.

4.1 Scheduling using Timed Game Automata (TGA)

In this section, we propose a solution to Problem 2 based on a reformulation using timed game automata, which we introduce next.

4.1.1 Timed game automata and safety games

This section is intended to briefly introduce timed automata [AD94], timed game automata [MPS95], and safety games.

4.1.1.1 Timed and timed game automata

Let C be a finite set of real-valued variables called clocks. We denote by $\mathcal{B}(C)$ the set of conjunctions of clock constraints of the form $c \sim \alpha$ where $\alpha \in \mathbb{R}_0^+$, $c \in C$ and $\sim \in \{<, \leq, =, >, \geq\}$. We define a timed automaton (TA) and a timed game automaton (TGA) as in [CDF+05]:

Definition 11. *A timed automaton is a sextuple (L, l_0, Act, C, E, I) where*

- *L is a finite set of locations;*
- *$l_0 \in L$ is the initial location;*
- *Act is a set of actions;*
- *C is a finite set of real-valued clocks;*
- *$E \subseteq L \times \mathcal{B}(C) \times Act \times 2^C \times L$ is the set of edges;*
- *$Inv : L \to \mathcal{B}(C)$ is a function that assigns invariants to locations.*

Definition 12. *A timed game automaton is a septuple $(L, l_0, Act_c, Act_u, C, E, I)$ such that $(L, l_0, Act_c \cup Act_u, C, E, I)$ is a timed automaton and $Act_c \cap Act_u = \emptyset$, where Act_c defines a set of controllable actions and Act_u defines a set of uncontrollable actions.*

Formal semantics of TA and TGA are restated in Appendix A from [CDF+05]. Informally, semantics of a TA is described by a transition system whose state consists of the current location and value of the clocks. Then, the execution of a TA can be described by two types of transitions defined as follows:

- time progress: the current location $l \in L$ is maintained and the value of the clocks grow at unitary rate; these transitions are enabled as long as the value of the clocks satisfies $Inv(l)$.
- discrete transition: an instantaneous transition from the current location $l \in L$ to a new location $l' \in L$ labelled by an action $a \in Act$ is triggered; these transitions are enabled if there is an edge

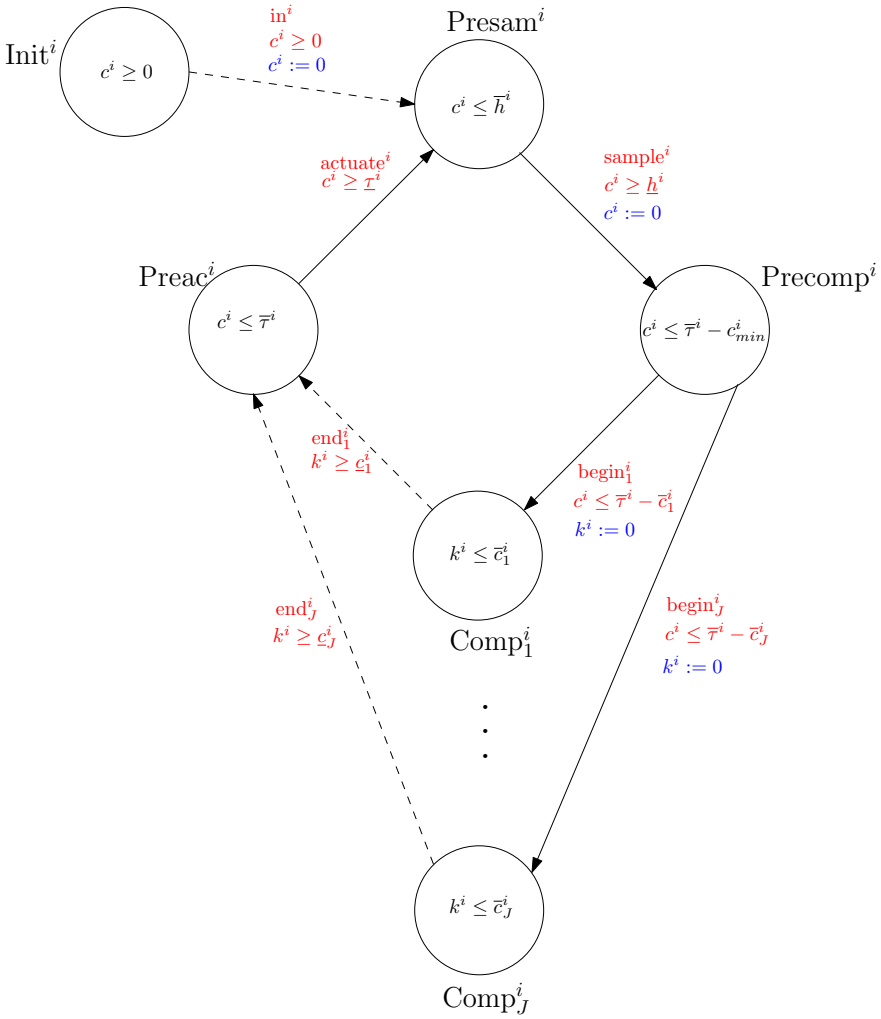


Figure 4.1: TGA_i, where plain and dashed edges correspond to controllable and uncontrollable actions respectively.

$(l, G, a, C', l') \in E$, such that the value of the clocks satisfies G; in that case, the value of the clocks belonging to C' resets to zero.

The semantics of TGA is similar to that of TA with the specificity that discrete transitions labelled by a controllable action (i.e. $a \in Act_c$) are triggered by a controller, while discrete transitions labelled by an uncontrollable action (i.e. $a \in Act_u$) are triggered by the environment/opponent.

4.1.1.2 Safety games

Safety games (see e.g. [CDF+05]) are defined by a timed game automaton and a set of unsafe locations $L_u \subseteq L$. A solution to the safety game is given by a winning strategy for the controller such that under any behavior of the environment/opponent, the set of unsafe locations is avoided by all executions of the TGA.

4.1.2 Reformulation into TGA

We first associate to each control task and timing contract a timed game automaton.

Definition 13. *Let $i \in \mathbb{N}_{[1,N]}$, the timed game automaton generated by control task $T_i = \big((\underline{c}_1^i, \overline{c}_1^i), \ldots, (\underline{c}_J^i, \overline{c}_J^i)\big)$ and timing contract $\theta(\underline{\tau}^i, \overline{\tau}^i, \underline{h}^i, \overline{h}^i)$ is displayed in Figure 4.1 and is formally defined by*

$$\mathsf{TGA}_i = (L^i, l_0^i, Act_c^i, Act_u^i, C^i, E^i, Inv^i)$$

where

- $L^i = \{Init^i, Presam^i, Precomp^i, Preac^i, Comp_1^i, \ldots, Comp_J^i\}$;
- $l_0^i = Init^i$;
- $Act_c^i = \{sample^i, begin_1^i, \ldots, begin_J^i, actuate^i\}$;
- $Act_u^i = \{end_1^i, \ldots, end_J^i, in^i\}$;
- $C^i = \{c^i, k^i\}$;
- $E^i = \{(Init^i, c^i \geq 0, in^i, \{c^i\}, Presam^i),$
  $(Presam^i, c^i \geq \underline{h}^i, sample^i, \{c^i\}, Precomp^i),$
  $(Precomp^i, c^i \leq \overline{\tau}^i - \overline{c}_1^i, begin_1^i, \{k^i\}, Comp_1^i), \ldots, (Precomp^i, c^i \geq 0, begin_J^i, \{k^i\}, Comp_J^i),$
  $(Comp_1^i, k^i \geq \underline{c}_1^i, end_1^i, \emptyset, Preac^i), \ldots, (Comp_J^i, k^i \geq \underline{c}_J^i, end_J^i, \emptyset, Preac^i),$
  $(Preac^i, c^i \geq \underline{\tau}^i, actuate^i, \emptyset, Presam^i)\}$;
- $Inv^i(Init^i) = \{c^i \geq 0\},$
  $Inv^i(Presam^i) = \{c^i \leq \overline{h}^i\},$
  $Inv^i(Precomp^i) = \{c^i \leq \overline{\tau}^i - c_{min}^i\}, with\ \ c_{min}^i = \min_{j \in \mathbb{N}_{[1,J]}}(\overline{c}_j^i),$
  $Inv^i(Comp_1^i) = \{k^i \leq \overline{c}_1^i\}, \ldots, Inv^i(Comp_J^i) = \{k^i \leq \overline{c}_J^i\},$
  $Inv^i(Preac^i) = \{c^i \leq \overline{\tau}^i\}.$

Intuitively, the set of locations L^i denotes all the possible "situations" that a control task T_i may be in and E^i denotes all the possible transitions between locations. If we assume that the control loop has not started yet then this is realized by the location $Init^i$. After that the control loop starts at a certain time that is determined by the environment and thus an uncontrollable transition $(Init^i, c^i \geq 0, in^i, \{c^i\}, Presam^i)$ takes place, where the task has to wait until sampling could occur. The latter is realized by the location $Presam^i$. Then whenever possible, a controller (which is the scheduler as we will see in the next section) has to decide when sampling must occur. When sampling takes place, the control task will be waiting until a CPU is assigned to compute its control input. This "waiting situation" is realized by the $Precomp^i$ location. The mission of assigning a CPU for task T_i is that of the scheduler, thus a possible controllable transition

occurs when the assignment of CPU_j takes place declaring that the task is in a new situation realized in TGA_i by the location $Comp_J^i$. The task rests in this situation until its execution on the CPU finishes which means that this duration is decided by the environment (which is the CPU and not the scheduler) and thus an uncontrollable transition from $Comp_j^i$ to a new location $Preac^i$ means that the execution has terminated and the control task is in the situation where actuation is to happen next. The latter decision is took by the scheduler, and thus is controllable, where the control input is fed to the plant and the control task is back again in the pre-sampling situation realized as before by the $Presam^i$ location. In such a case, the control loop is closed and the "behavior" of the control task is repeated infinitely. Note that all the executions of TGA_i explained informally above must respect the semantics of the timed game automata introduced in Section 4.1.1.1.

Now let the sequences $(t_k^{s_i})$, $(t_k^{a_i})$, $(t_k^{b_i})$ and $(t_k^{e_i})$ be given by the instants of the discrete transitions labeled by actions $sample^i$, $actuate^i$, $begin^i$ and end^i, respectively. It is easy to see that these sequences satisfy the constraints given by (2.4). Conversely, one can check that all sequences satisfying (2.4) can be generated by executions of TGA_i.

Moreover, let us restate that the controllable actions are $sample^i$, $actuate^i$, $begin^i$, which means that the scheduler determines the instants when sampling and actuation occur and when computation begins. However, end^i is uncontrollable, which means that the execution time, and thus the instant at which computation ends is determined by the environment.

Finally, CPU j is used by system $\mathcal{S}_i$ if the current location of TGA_i is $Comp_j^i$, with $j \in \mathbb{N}_{[1,J]}$. To take into account the constraint given by (2.5), stating that two systems cannot access any of the J CPUs at the same time, we need to define the composition of the timed game automata defined above:

Definition 14. *The timed game automaton generated by the set of control tasks $\mathcal{T} = \{T_1, \ldots, T_N\}$, with $T_i = \big((\underline{c}_1^i, \overline{c}_1^i), \ldots, (\underline{c}_J^i, \overline{c}_J^i)\big)$ for all $i \in \mathbb{N}_{[1,N]}$, and timing contracts $\Theta = \{\theta(\underline{\tau}^1, \overline{\tau}^1, \underline{h}^1, \overline{h}^1), \ldots, \theta(\underline{\tau}^N, \overline{\tau}^N, \underline{h}^N, \overline{h}^N)\}$ is given by $\mathsf{TGA} = (\overline{L}, \bar{l}_0, \overline{Act}_c, \overline{Act}_u, \overline{C}, \overline{E}, \overline{Inv})$ where*

- $\overline{L} = L^1 \times \cdots \times L^N$, *thus* $l = (l^1, \ldots, l^N) \in L$ *denotes the location of* TGA;
- $\bar{l}_0 = (Init^1, \ldots, Init^N)$;
- $\overline{Act}_c = \bigcup_{i=1}^N Act_c^i$;
- $\overline{Act}_u = \bigcup_{i=1}^N Act_u^i$;
- $\overline{C} = \bigcup_{i=1}^N C^i$;
- $\overline{E} = \{(l_m, \lambda, act, C', l_n) \in \overline{L} \times \mathcal{B}(\overline{C}) \times (\overline{Act}_c \cup \overline{Act}_u) \times \overline{L} : \exists i \in \mathbb{N}_{[1,N]}, l_m^j = l_n^j \ \ \forall j \neq i \text{ and } (l_m^i, \lambda, act, C', l_n^i) \in E^i\}$;
- $Inv(l) = \bigwedge_{i=1}^N Inv^i(l^i)$, $i \in \mathbb{N}_{[1,N]}$.

TGA describes the parallel evolution of the $\mathsf{TGA}_1, \ldots, \mathsf{TGA}_N$ and thus models the concurrent execution of the control tasks $T_1, \ldots, T_N$.

4.1.3 Scheduling as a safety game

In our setting, we denote the safety game by (TGA,$\overline{L}_u$), where the set of locations corresponding to conflicting accesses to the CPUs $\overline{L}_u \subseteq \overline{L}$ is defined by:

$$\begin{aligned} \overline{L}_u = \{l \in \overline{L} : \exists (m,n,j) \in \mathbb{N}^2_{[1,N]} \times \mathbb{N}_{[1,J]}, m \neq n, \\ (l^m = Comp^m_j) \wedge (l^n = Comp^n_j)\}. \end{aligned} \tag{4.1}$$

From the previous discussions, we define the following property:

Definition 15. *(schedulability) $\mathcal{T}$ is schedulable under timing contracts Θ if and only if there is a winning strategy to (TGA,$\overline{L}_u$).*

From the practical point of view, the safety game, and thus Problem 2, can be solved using the tool UPPAAL-TIGA [BCD$^+$07]. The latter synthesizes also a winning strategy when it exists, which provides us with a dynamic scheduling policy for generating the sequences $(t^{s_i}_k)_{k\in\mathbb{N}}$, $(t^{b_i}_k)_{k\in\mathbb{N}}$, $(t^{e_i}_k)_{k\in\mathbb{N}}$, and $(t^{a_i}_k)_{k\in\mathbb{N}}$ satisfying (2.4-2.5), for all $i \in \mathbb{N}_{[1,N]}$.

4.1.4 A simplified scheduling condition

In fact, a simpler scheduling condition could be obtained using a simplified timed game automaton. But before we introduce a simple useful lemma.

Lemma 6. *If $\underline{h} \leq \underline{\tau}$ then timing contract $\theta(\underline{\tau}, \overline{\tau}, \underline{h}, \overline{h})$ is equivalent to $\theta(\underline{\tau}, \overline{\tau}, \underline{\tau}, \overline{h})$.*

Proof: We have $t^s_k \leq t^a_k \leq t^s_{k+1}$, for all $k \in \mathbb{N}$. This implies that, given that $\underline{h} \leq \underline{\tau}$, the least inter-sampling time that could occur for all $k \in \mathbb{N}$ is when $h_k = \max(\underline{h}, \min_{m\in\mathbb{N}}(t^a_m - t^s_m)) = \underline{\tau}$. This implies that $\theta(\underline{\tau}, \overline{\tau}, \underline{h}, \overline{h}) = \theta(\underline{\tau}, \overline{\tau}, \underline{\tau}, \overline{h})$. □

The previous lemma allows us without adding any restrictions to study timing contracts for the case when $\underline{h} \geq \underline{\tau}$. Let us define now the following two compositions of TGA:

Definition 16. *We define a timed game automaton generated by the set of control tasks $\mathcal{T} = \{T_1, \ldots, T_N\}$, with $T_i = \big((\overline{c}^i_1, \overline{c}^i_1), \ldots, (\overline{c}^i_J, \overline{c}^i_J)\big)$ for all $i \in \mathbb{N}_{[1,N]}$, and timing contracts $\Theta = \{\theta(\underline{\tau}^1, \overline{\tau}^1, \underline{h}^1, \overline{h}^1), \ldots,$ $\theta(\underline{\tau}^N, \overline{\tau}^N, \underline{h}^N, \overline{h}^N)\}$ as $\mathsf{TGA}' = (\overline{L}, \overline{l}_0, \overline{Act}_c, \overline{Act}_u, \overline{C}, \overline{E}', \overline{Inv})$ where all elements of TGA' are the same as TGA except for $\overline{E}'$ which is given by*

- $\overline{E}' = \{(l_m, \lambda, act, C', l_n) \in \overline{L} \times \mathcal{B}(\overline{C}) \times (\overline{Act}_c \cup \overline{Act}_u) \times \overline{L} : \exists i \in \mathbb{N}_{[1,N]}, l_m^j = l_n^j \ \ \forall j \neq i \text{ and } (l_m^i, \lambda, act, C', l_n^i) \in E_*^i\}$;

with

- $E_*^i = \{$
  $(Init^i, c^i \geq 0, in^i, \{c^i\}, Presam^i)$,
  $(Presam^i, c^i \geq \underline{h}^i, sample^i, \{c^i\}, Precomp^i)$,
  $(Precomp^i, c^i \leq \overline{\tau}^i - \overline{c}_1^i, begin_1^i, \{k^i\}, Comp_1^i), \ldots, (Precomp^i, c^i \geq 0, begin_J^i, \{k^i\}, Comp_J^i)$,
  $(Comp_1^i, k^i = \overline{c}_1^i, end_1^i, \emptyset, Preac^i), \ldots, (Comp_J^i, k^i = \overline{c}_J^i, end_J^i, \emptyset, Preac^i)$,
  $\begin{cases} (Preac^i, c^i \geq \underline{h}^i, actuate^i, \emptyset, Presam^i) & \text{if } \overline{\tau}^i \geq \underline{h}^i \\ (Preac^i, c^i = \overline{\tau}^i, actuate^i, \emptyset, Presam^i)\} & \text{if } \overline{\tau}^i \leq \underline{h}^i \end{cases}$
  $\};$

TGA' is actually a network of TGA which is defined similarly to TGA with two differences. The first one is that the transition from $Preac^i$ state to the $Presam^i$ state is only allowed in TGA' when $c^i \geq \underline{h}^i$ if $\overline{\tau}^i \geq \underline{h}^i$ and when $c^i = \overline{\tau}^i$ elsewhere with $i \in \mathbb{N}_{[1,N]}$. The second is that the transition from $Comp_1^i$ to $Preac^i$ happens when $k^i = \overline{c}_1^i$ for all $i \in \mathbb{N}_{[1,N]}$ and $j \in \mathbb{N}_{[1,J]}$. Then using TGA' and Lemma 6 we can conclude on the schedulability of the task-set $\mathcal{T}$.

Proposition 7. *$\mathcal{T}$ is schedulable under timing contracts Θ if and only if there is a winning strategy to (TGA',$\overline{L}_u$).*

The reader could find the proof of the Proposition 7 in Appendix B. In words, TGA' suggests to always consider the worst case execution time (WCET) when computing the control input in each of its timed game automaton, and then obviously scheduling is guaranteed for this special case if it is guaranteed when considering the execution time to be possible for any duration given in between the best case execution time (BCET) and WCET. In the other way around, if scheduling is guaranteed for the former case then it is also guaranteed for the latter, since when computation of the control input on a certain processor finishes in one of the N timed game automata with the corresponding clock (computing the execution time) less than the WCET then this particular timed game automaton could wait without taking any transition. Next when the clock reaches the WCET the timed automaton could execute again under the supervision of the schedule defined for TGA'. In addition, it is easy to see that a winning strategy for TGA' is also a winning strategy for the same timed game automata if the choice for the controlled action $actuate^i$, $i = 1, \ldots, N$, is given a wider range as in TGA. In the other way around, any schedule for TGA could be modified so that it could still serve as a schedule by delaying the actuation action in each of the timed game automaton, when

possible, up till the time when the earliest sampling action could be taken as defined by the transitions from the $Preac^i$ location to the $Presam^i$ location in E^i_* of Definition 16.

4.2 Illustrative example

In this section, we are interested in synthesizing schedules for a given number N of sampled-data systems who are subject to timing contracts and whose control input is computed by J shared CPUs, with $J < N$. Indeed, the schedule should guarantee the stability of each system. We implemented the scheduling approach presented in Section 4.1 using UPPAAL-TIGA [BCD+07], and used Algorithm 1 to verify stability.

4.2.1 One processor

Example 9. *We take $N = 2$ where the two systems $\mathcal{S}_1 = (A_1, B_1, K_1)$ and $\mathcal{S}_2 = (A_2, B_2, K_2)$ are taken from [Bri13] and given by the following matrices:*

$$A_1 = \begin{pmatrix} 0 & 1 \\ 0 & -0.1 \end{pmatrix},\ B_1 = \begin{pmatrix} 0 \\ 0.1 \end{pmatrix},\ K_1 = \begin{pmatrix} -3.75 & -11.5 \end{pmatrix}. \tag{4.2}$$

$$A_2 = \begin{pmatrix} 0 & 1 \\ -2 & 0.1 \end{pmatrix},\ B_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix},\ K_2 = \begin{pmatrix} 1 & 0 \end{pmatrix}. \tag{4.3}$$

4.2.1.1 Stability verification

After setting $\beta = 0$, we use Algorithm 1 in Section 3.2.3.2 to verify that systems $\mathcal{S}_1$ and $\mathcal{S}_2$ are β'-stable under timing contracts $\theta(0.1, 0.35, 0.3, 0.85)$ and $\theta(0.2, 0.6, 0.8, 1.15)$ respectively. This means obviously that each of the two systems with any synthesized scheduling policy on a shared CPU, respecting the above timing contracts, is guaranteed to be stable. The computation times required for stability verification are 1.96 *seconds and* 1.5 *seconds, respectively.*

4.2.1.2 Scheduling

Now, we consider the set of control tasks $\mathcal{T} = \{T_1, T_2\}$ running on a single processor, or $J = 1$. After setting the best and worst case execution times for each task as $\underline{c}^1_1 = 0.12$, $\overline{c}^1_1 = 0.35$, $\underline{c}^2 = 0.04$, and $\overline{c}^2 = 0.12$ we define task $T_1 = ((\underline{c}^1_1, \overline{c}^1_1))$, task $T_2 = ((\underline{c}^2_1, \overline{c}^2_1))$, and the set of timing contracts $\Theta = \{\theta(0.1, 0.35, 0.3, 0.85), \theta(0.2, 0.6, 0.8, 1.15)\}$ which contain the same contracts as in the previous section.

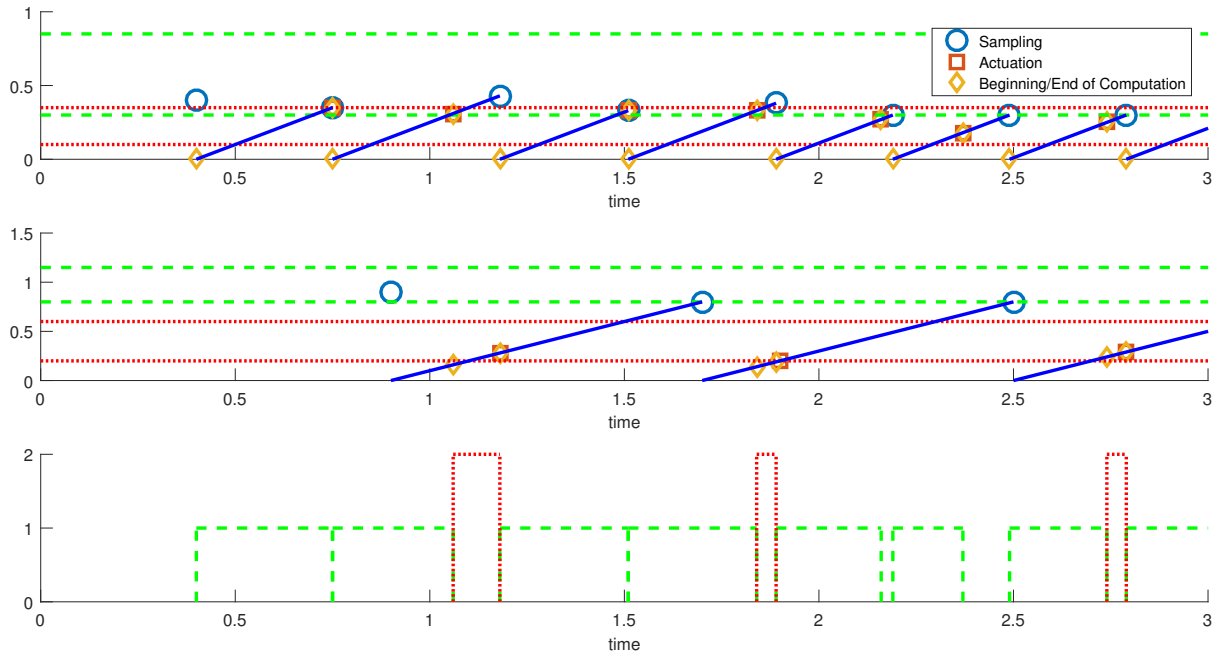


Figure 4.2: Timing of events (sampling, beginning/end of computation, and actuation) for systems $\mathcal{S}_1$ (first plot) and $\mathcal{S}_2$ (second plot) during the first 3 seconds; dotted lines represent constraints on actuation instants, while dashed lines represent constraints on sampling instants. In the third plot, the dotted line represents $\mathsf{Com}(\mathcal{S}_2, t)$ (less frequent) and the dashed line represents $\mathsf{Com}(\mathcal{S}_1, t)$ (more frequent).

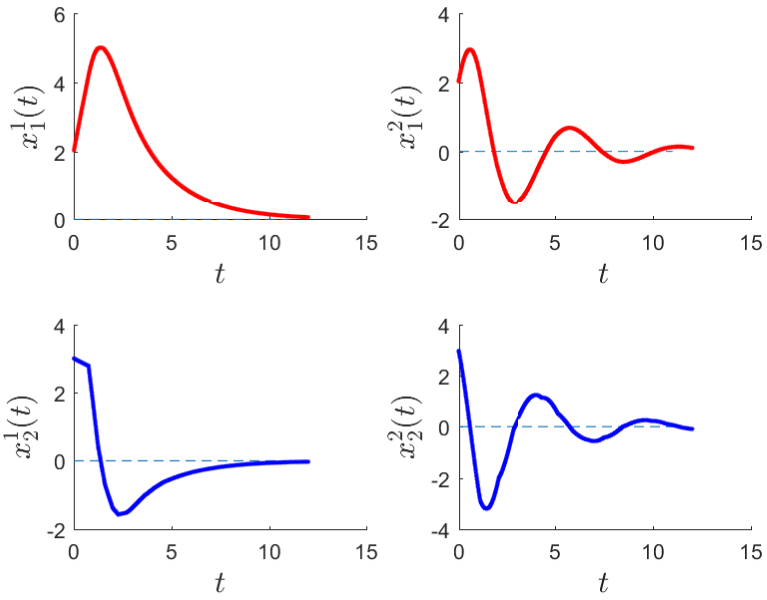


Figure 4.3: Trajectories for systems $\mathcal{S}_1$ (left) and $\mathcal{S}_2$ (right) using the synthesized scheduling policy.

In order to solve the scheduling problem, we associate to $\mathcal{T}$ *the timed game automaton* TGA *as given in Definition 14. Following the approach in Section* 4.1*, we solve the safety game on* TGA *to find a strategy (if it exists) for the triggering of controllable actions that occur at* $(t_k^{s_i})_{k\in\mathbb{N}}$*,* $(t_k^{b_i})_{k\in\mathbb{N}}$*, and* $(t_k^{a_i})_{k\in\mathbb{N}}$*, with* $i \in \mathbb{N}_{[1,2]}$*, guaranteeing that the set of bad states* $\overline{L}_u$ *of the system, given by (4.1), is never reached regardless of when uncontrollable actions occurring at* $(t_k^{e_i})_{k\in\mathbb{N}}$*,* $i \in \mathbb{N}_{[1,2]}$*, are exactly taken.*

Using UPPAAL-TIGA, we successfully prove that $\mathcal{T}$ *is schedulable under timing contracts* Θ*, and thus a scheduling policy was found. The computation time required to solve the game was* 1.37 *seconds.*

Figure 4.2 shows the timing of events resulting from this scheduling policy. The first and second plots show that the timing contracts $\theta(0.1, 0.35, 0.3, 0.85)$ *and* $\theta(0.2, 0.6, 0.8, 1.15)$ *are respected for both systems* $\mathcal{S}_1$ *and* $\mathcal{S}_2$ *respectively. The third plot shows that only one of the two systems gains access to the shared processor at a time since it appears clearly that*

$$\forall (m, n) \in \mathbb{N}^2_{[1,2]} \text{ with } m \neq n,$$
$$\mathsf{Com}(\mathcal{S}_m, 1) \cap \mathsf{Com}(\mathcal{S}_n, 1) = \emptyset.$$

One can notice that in the first three control cycles of $\mathcal{S}_2$*, the beginning of the computation has to be delayed until the CPU is released by* $\mathcal{S}_1$*.*

Using this scheduling policy, Figure 4.3 shows results of simulating $\mathcal{S}_1$ *and* $\mathcal{S}_2$*, when they share a single processor to compute the value of their control inputs, for the initial states* $z_0^1 = \binom{2}{3}$ *and* $z_0^2 = \binom{2}{3}$ *with* $t_0^{s_1} = 0.4$ *and* $t_0^{s_2} = 0.9$*. As shown, trajectories of both systems converge to zero and therefore the scheduling policy in this case guarantees the exponential stability of each system.*

4.2.2 Two processors

Example 10. *We take* $N = 3$*, where we have two identical systems* $\mathcal{S}_1$ *and* $\mathcal{S}_2$ *whose matrices are given by (4.2) and another system* $\mathcal{S}_3$ *with matrices given by (4.3). First we consider a single processor to compute the control input of the three systems (i.e.* $J = 1$*), the control tasks* T_1, T_2*, and* T_3 *with* $T_1 = T_2 = (0.12, 0.25)$ *and* $T_3 = (0.04, 0.1)$*, and timing contracts* $\Theta_a = \{\theta(0.1, 0.35, 0.1, 0.35), \theta(0.1, 0.35, 0.1, 0.35)\}$*,* $\Theta_b = \{\theta(0.1, 0.35, 0.1, 0.35), \theta(0.1, 0.2, 0.1, 0.2)\}$*, and* $\Theta_c = \{\theta(0.1, 0.35, 0.1, 0.35), \theta(0.1, 0.35, 0.1, 0.35),$ $\theta(0.1, 0.2, 0.1, 0.2)\}$*. Following the approach in Section* 4.1 *we can prove that each of the task-set* $\{T_1, T_2\}$*, the task-set* $\{T_2, T_3\}$*, and obviously the task-set* $\{T_1, T_2, T_3\}$ *is not schedulable under timing contracts* Θ_a*,* Θ_b*, and* Θ_c *respectively. On the other hand, this doesn't mean that systems* $\mathcal{S}_1$*,* $\mathcal{S}_2$*, and* $\mathcal{S}_3$ *cannot share two processors to compute their control input.*

Now, we consider two CPUs, or $J = 2$*, and define the task-set* $\mathcal{T} = \{T_4, T_5, T_6\}$ *with* $T_4 = T_5 = ((0.12, 0.25), (0.12, 0.25)$ *and* $T_6 = ((0.04, 0.01), (0.04, 0.01))$*. Then we associate to* $\mathcal{T}$ *the* TGA *as given in Definition 14 and solve the safety game on* TGA *to find a strategy (if it exists) for the triggering of controllable actions that occur at* $(t_k^{s_i})_{k\in\mathbb{N}}$*,* $(t_k^{b_i})_{k\in\mathbb{N}}$*, and* $(t_k^{a_i})_{k\in\mathbb{N}}$*, with* $i \in \mathbb{N}_{[1,3]}$*, guaranteeing that the set of bad states* $\overline{L}_u$ *of the system is never reached regardless of when uncontrollable actions occurring at* $(t_k^{e_i})_{k\in\mathbb{N}}$*,* $i \in \mathbb{N}_{[1,3]}$*, are exactly taken.*

Using UPPAAL-TIGA, we successfully prove that $\mathcal{T}$ *is schedulable under timing contracts* Θ_c*, and thus a scheduling policy was found. The computation time required to solve the game and output the scheduling policy was* 10 *seconds.*

Figure 4.5 shows the timing of events resulting from this scheduling policy. The first three plots show that the timing contracts $\theta(0.1, 0.35, 0.1, 0.35)$*,* $\theta(0.1, 0.35, 0.1, 0.35)$*, and* $\theta(0.1, 0.2, 0.1, 0.2)$ *are respected for systems* $\mathcal{S}_1$*,* $\mathcal{S}_2$*, and* $\mathcal{S}_3$ *respectively. The fourth and fifth plots show that only one of the three systems gains access to each of the two shared processors at a time since it appears clearly that*

$$\forall (m, n, j) \in \mathbb{N}^2_{[1,3]} \times \mathbb{N}_{[1,2]} \text{ with } m \neq n, \mathsf{Com}(\mathcal{S}_m, j) \cap \mathsf{Com}(\mathcal{S}_n, j) = \emptyset.$$

At this point we shall mention that we followed the stability verification approach we suggested earlier in Chapter 3 and prove that for $\beta = 0$*,* β'*-stability is guaranteed for systems* $\mathcal{S}_1$*,* $\mathcal{S}_2$*, and* $\mathcal{S}_3$ *under timing contracts* $\theta(0.1, 0.35, 0.1, 0.35)$*,* $\theta(0.1, 0.35, 0.1, 0.35)$*, and* $\theta(0.1, 0.2, 0.1, 0.2)$ *respectively. Using this scheduling policy, Figure 4.4 shows results of simulating* $\mathcal{S}_1$*,* $\mathcal{S}_2$*, and* $\mathcal{S}_3$ *when they share two processors to compute the value of their control inputs, for the initial states* $z_0^1 = z_0^2 = z_0^3 = \binom{2}{3}$ *with* $t_0^{s_1} = t_0^{s_2} = t_0^{s_3} = 0.01s$*. As shown, trajectories of the three systems converge to zero and therefore the scheduling policy in this case guarantees the exponential stability of each system.*

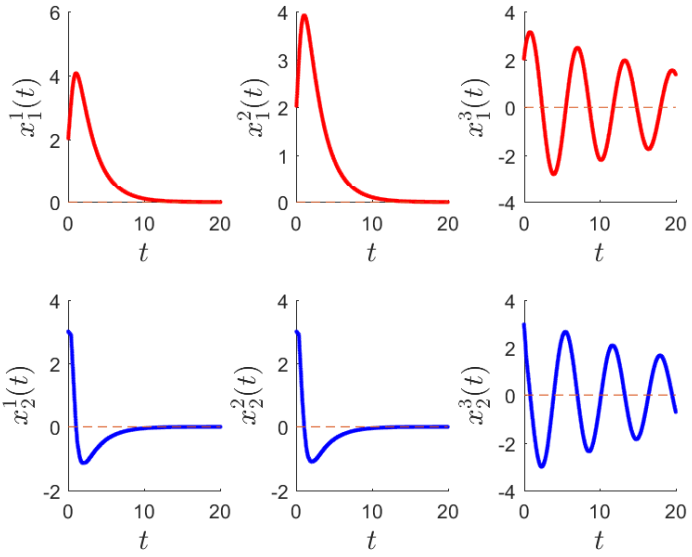


Figure 4.4: Trajectories for systems $\mathcal{S}_1$ (left), $\mathcal{S}_2$ (middle), and $\mathcal{S}_3$ (right) using the synthesized scheduling policy.

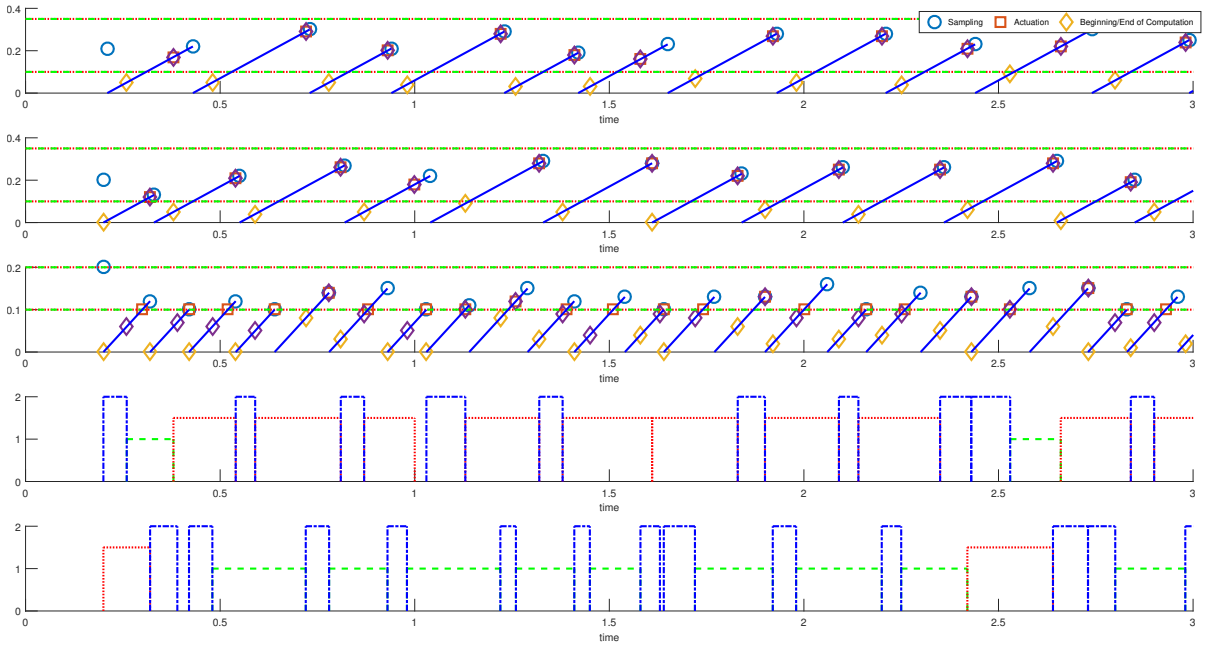


Figure 4.5: Timing of events (sampling, beginning/end of computation, and actuation) for systems $\mathcal{S}_1$ (first plot), $\mathcal{S}_2$ (second plot), $\mathcal{S}_3$ (third plot) during the first 3 seconds; dotted lines represent constraints on actuation instants, while dashed lines represent constraints on sampling instants. In the fourth and fifth plot, the dashed line (magnitude 1) represents $\mathsf{Com}(\mathcal{S}_1, t)$, dotted line (magnitude 1.5) represents $\mathsf{Com}(\mathcal{S}_1, t)$, and the dotted-dashed line (magnitude 2) represents $\mathsf{Com}(\mathcal{S}_3, t)$.

Chapter 5

Parameter synthesis

Abstract

Our aim here is to provide a solution to Problem 3, whereby we synthesize a set of timing contracts that guarantees at the same time the schedulability and the stability of the embedded controllers. For the proposed synthesis procedure, we propose an interesting and novel re-parameterization of the timing contract parameters so as to make them monotonic. This enables a sampling-based procedure for synthesizing timing contracts, guaranteeing stability and schedulability, by keeping track of an under- and an over-approximation of the parameter space and repeatedly sampling from the unexplored parameter set until the distance between over- and under-approximation is smaller than a given threshold. An experimental evaluation section is given at the end considering two 2-dimensional systems and a uniprocessor. Results of this chapter are published in [AKGD16a, AKGD17a].

In Chapters 3 and 4 we fix the parameters $\underline{\tau}$, $\overline{\tau}$, $\underline{h}$, and $\overline{h}$ of the timing contracts (2.2) and (2.4). Then we solve the stability verification problem (Problem 1) in Chapter 3 and the scheduling problem (Problem 2) in Chapter 4. In this chapter, requirement engineers are addressed. For several sampled-data systems of the form (2.1-2.2) sharing a set of computational resources to compute their control inputs, the engineer is interested in finding a set of timing contracts for these systems such that stability and schedulability are guaranteed for each single system and for the control tasks respectively. We present in this chapter our approach, that is based on a re-parameterization of contracts, which provides some monotonicity property to the problem and allows us to develop an effective synthesis method based on guided sampling of the timing contract parameter space. Note that a similar parameter synthesis approach, using guided sampling in monotone sets, is used in [KAS16] to identify the set of admissible disturbance signals and initial states that generate trajectories satisfying a given temporal logic specification for a dynamical system.

Now given a collection of systems $\{\mathcal{S}_1, \ldots, \mathcal{S}_N\}$, a set of convergence rates $\{\beta_i\}_{i \in \mathbb{N}_{[1,N]}}$, J computational units, a set of control tasks $\mathcal{T} = \{T_1, \ldots, T_N\}$ with $T_i = \big((\underline{c}_1^i, \overline{c}_1^i), \ldots, (\underline{c}_J^i, \overline{c}_J^i)\big)$ and $0 \leq \underline{c}_j^i \leq \overline{c}_j^i$ for all $i \in \mathbb{N}_{[1,N]}$ and $j \in \mathbb{N}_{[1,J]}$, and parameter sets $\mathcal{D}_1, \ldots, \mathcal{D}_N$, we use a monotonicity property to design an

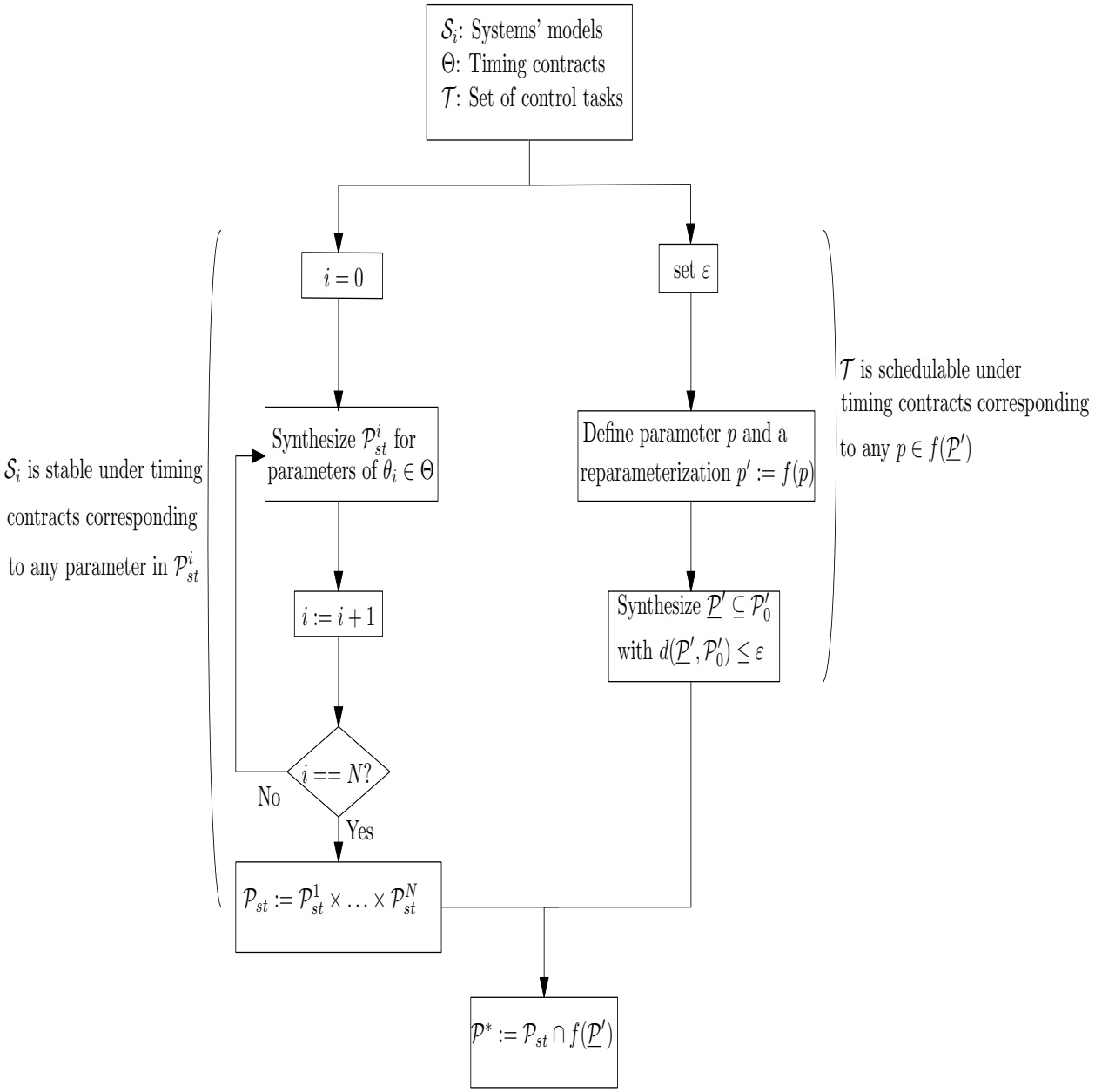


Figure 5.1: Workflow of the proposed approach.

algorithm that synthesizes a set of timing contracts ensuring $\beta_i'-$stability of each system $\mathcal{S}_i$ and schedulability of $\mathcal{T}$.

The workflow of the proposed approach is depicted in Figure 5.1. The problem is divided into two parts. First, we synthesize timing contract giving a guarantee on stability for every system $\mathcal{S}_i$, $i \in \mathbb{N}_{[1,N]}$, where a set $\mathcal{P}_{st}$ is synthesized as explained in Section 5.1. Also in this part, when it is necessary we could use any method from Table 2.1 Section 2.2.5 for checking stability. In the second part, we follow the steps illustrated in Section 5.2 to synthesize a set $\mathcal{P}_{sched} = f(\underline{\mathcal{P}'})$ giving a guarantee on schedulability. Notice that

the method proposed in Section 4.1 is used for checking schedulability of control tasks. As a consequence, a solution to Problem 3 is given by $\mathcal{P}^*$.

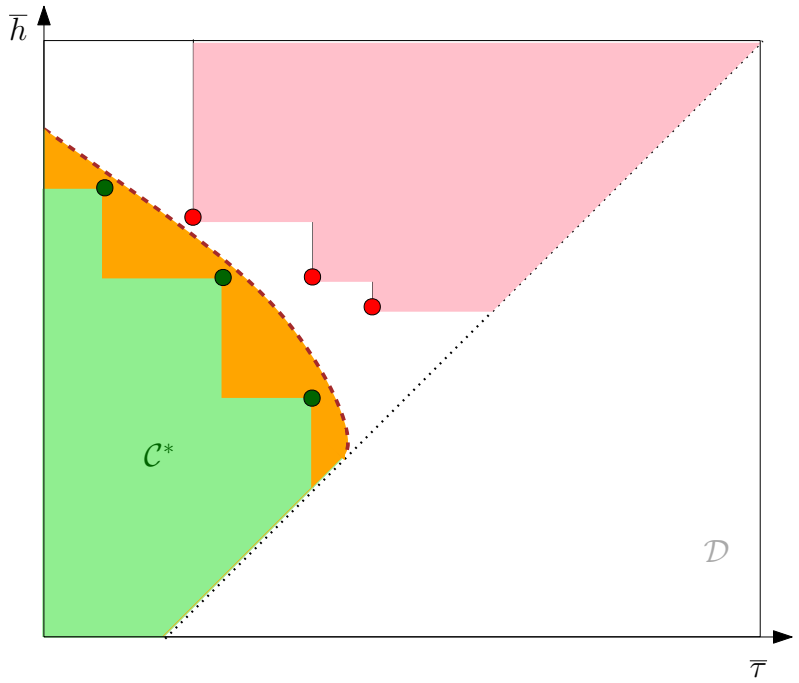


Figure 5.2: Sampling based algorithm for parameter synthesis: If a point is in a lower (respectively higher) set, then we can extrapolate that all points, in the feasible region above the dotted line, below (respectively above) it are also contained in that set. Note that all points in $\mathcal{C}$, which are necessarily above the dotted line, and below the dashed surface defines the desired set $\mathcal{C}_0$.

5.1 Guarantee on stability

In this section, the setup is given by N systems the form (2.1) where each system $\mathcal{S}_i = (A_i, B_i, K_i)$, is subject to a timing contract $\theta(\underline{\tau}^i, \overline{\tau}^i, \underline{h}^i, \overline{h}^i)$ of the form (2.2), with parameters $(\underline{\tau}^i, \overline{\tau}^i, \underline{h}^i, \overline{h}^i) \in \mathcal{C}$, $i \in \mathbb{N}_{[1,N]}$. Given a set $\{\beta_i\}_{i \in \mathbb{N}_{[1,N]}}$, we synthesize a set of parameters $\mathcal{P}^i_{st}$ for every $i \in \mathbb{N}_{[1,N]}$ such that for all $(\underline{\tau}^i, \overline{\tau}^i, \underline{h}^i, \overline{h}^i) \in \mathcal{P}^i_{st}$ $\mathcal{S}_i$ under timing contract $\theta(\underline{\tau}^i, \overline{\tau}^i, \underline{h}^i, \overline{h}^i)$ is β'_i-stable. For simplicity we drop the $i, i \in \mathbb{N}_{[1,N]}$, in this section since the same procedure is followed to synthesize all the sets $\mathcal{P}^i_{st}$. Then the subproblem we solve here is:

Problem 6 (Timing contract synthesis)**.** *Given $\beta \in \mathbb{R}^+$, $A \in \mathbb{R}^{p \times p}$, $B \in \mathbb{R}^{p \times m}$, $K \in \mathbb{R}^{m \times p}$, and $\mathcal{D} \subset \mathbb{R}^4$, synthesize a set $\mathcal{C}^* \subseteq \mathcal{C} \cap \mathcal{D}$ such that for all $(\underline{\tau}, \overline{\tau}, \underline{h}, \overline{h}) \in \mathcal{C}^*$,(2.1-2.2) is β'-stable.*

We first define a re-parametrization of the timing contract such that stability of system (2.1-2.2) becomes monotone with respect to the new parameters. Monotonicity is a very attractive property for designing efficient heuristics for timing contract synthesis since stability is preserved when the parameter values increase. This allows us to tackle the timing contract synthesis by sampling the parameter space.

5.1.1 Re-parametrization

Figure 5.2 clearly shows that stability is not monotonic with respect to the default parameters of the timing contract in a rectangular parameter space $\mathcal{D}$. The reason is that the constraint set $\mathcal{C}$, specifically the constraint $\overline{\tau}^i \leq \overline{h}^i$, is not monotonic with respect to parameters. This hindrance motivates the re-parametrization of the timing contract Given the bounds on the parameters $0 \leq \underline{\tau}_{min} \leq \underline{\tau}_{max} \leq \overline{\tau}_{max}$, $\underline{\tau}_{min} \leq \overline{\tau}_{min} \leq \overline{\tau}_{max}$, $0 < \underline{h}_{min} \leq \underline{h}_{max} \leq \overline{h}_{max}$, $\underline{h}_{min} < \overline{h}_{min} \leq \overline{h}_{max}$, with $\underline{\tau}_{min} \leq \underline{h}_{min}$, $\overline{\tau}_{min} \leq \overline{h}_{min}$, $\underline{\tau}_{max} \leq \underline{h}_{max}$, $\overline{\tau}_{max} \leq \overline{h}_{max}$, let

$$\mathcal{D} = [\underline{\tau}_{min}, \underline{\tau}_{max}] \times [\overline{\tau}_{min}, \overline{\tau}_{max}] \times [\underline{h}_{min}, \underline{h}_{max}] \times [\overline{h}_{min}, \overline{h}_{max}]. \tag{5.1}$$

We also denote the vector of timing contract parameters $\alpha = (\underline{\tau}, \overline{\tau}, \underline{h}, \overline{h}) \in \mathcal{D}$. For $\alpha \in \mathcal{C} \cap \mathcal{D}$ we denote the property:

$$\mathsf{Stab}(\alpha) \equiv \text{(2.1-2.2) is } \beta'\text{-stable with parameters } \alpha.$$

Solving Problem 3 is equivalent to computing (a subset of) the set $\mathcal{C}_o$ defined by

$$\mathcal{C}_o = \{\alpha \in \mathcal{C} \cap \mathcal{D} : \mathsf{Stab}(\alpha)\}.$$

Let us define a new parameter $\eta = (\eta_1, \eta_2, \eta_3, \eta_4) \in \mathcal{D}'$ where $\mathcal{D}' = [\tau_{min}, \tau_{max}] \times [-\tau_{max}, -\tau_{min}] \times [h_{min}, h_{max}] \times [-h_{max}, -h_{min}]$ and the map $f : \mathcal{D}' \to \mathcal{D}$ such that $f(\eta) = \alpha = (\underline{\tau}, \overline{\tau}, \underline{h}, \overline{h})$ where

$$\underline{\tau} = \eta_1,\ \overline{\tau} = \min(-\eta_2, -\eta_4),\ \underline{h} = \eta_3,\ \overline{h} = -\eta_4.$$

We define the following constraint set for the parameter η:

$$\mathcal{C}' = \left\{ \eta \in \mathbb{R}_0^+ \times \mathbb{R}_0^- \times \mathbb{R}^+ \times \mathbb{R}^- : \begin{array}{l} \eta_1 \leq \min(-\eta_2, -\eta_4) \\ \eta_3 \leq -\eta_4 \end{array} \right\}. \tag{5.2}$$

The following result holds:

Lemma 7. *Let $\mathcal{C}'_o$ be given by*

$$\mathcal{C}'_o = \{\eta \in \mathcal{C}' \cap \mathcal{D}' : \mathit{Stab}(f(\eta))\}.$$

Then, $f(\mathcal{C}' \cap \mathcal{D}') = \mathcal{C} \cap \mathcal{D}$ and $f(\mathcal{C}'_o) = \mathcal{C}_o$.

Proof. Let us first show that $f(\mathcal{C}' \cap \mathcal{D}') \subseteq \mathcal{C} \cap \mathcal{D}$ and $f(\mathcal{C}'_o) \subseteq \mathcal{C}_o$. Let $\eta \in \mathcal{C}' \cap \mathcal{D}'$ and $\alpha = f(\eta) = (\underline{\tau}, \overline{\tau}, \underline{h}, \overline{h})$. Then, $\eta \in \mathcal{D}'$ implies that $\alpha \in \mathcal{D}$, using the fact that $\tau_{min} \leq h_{min}$, $\tau_{max} \leq h_{max}$. Also, $\eta \in \mathcal{C}'$ implies that $\underline{\tau} \leq \overline{\tau}$ and $\underline{h} \leq \overline{h}$. Moreover, $\overline{\tau} = \min(-\eta_2, -\eta_4) \leq -\eta_4 = \overline{h}$. Hence, $\alpha \in \mathcal{C}$. Thus, $\alpha \in \mathcal{C} \cap \mathcal{D}$. Moreover, if $\eta \in \mathcal{C}'_o$ then $\eta \in \mathcal{C}' \cap \mathcal{D}'$ and $\mathsf{Stab}(f(\eta))$ gives $\alpha \in \mathcal{C} \cap \mathcal{D}$ and $\mathsf{Stab}(\alpha)$. Thus, $\alpha \in \mathcal{C}_o$. We now show that $\mathcal{C} \cap \mathcal{D} \subseteq f(\mathcal{C}' \cap \mathcal{D}')$ and $\mathcal{C}_o \subseteq f(\mathcal{C}'_o)$. Let $\alpha = (\underline{\tau}, \overline{\tau}, \underline{h}, \overline{h}) \in \mathcal{C} \cap \mathcal{D}$ and let $\eta = (\underline{\tau}, -\overline{\tau}, \underline{h}, -\overline{h})$. Then, $f(\eta) = (\underline{\tau}, \min(\overline{\tau}, \overline{h}), \underline{h}, \overline{h})$. Since $\alpha \in \mathcal{C}$, it follows that $\min(\overline{\tau}, \overline{h}) = \overline{\tau}$ and $f(\eta) = \alpha$. Moreover, it is straightforward to verify that $\alpha \in \mathcal{C} \cap \mathcal{D}$ implies $\eta \in \mathcal{C}' \cap \mathcal{D}'$ and that $\alpha \in \mathcal{C}_o$ implies $\eta \in \mathcal{C}'_o$. □

The previous result has two important implications. The first one is that the proposed re-parametrization does not introduce any conservatism in the solution to Problem 6 since the set $\mathcal{C}_o$ of admissible parameters α can be obtained by computing the set $\mathcal{C}'_o$ of admissible parameters η, despite the fact that the map f is not injective nor surjective. The second one is stated in the following lemma:

Lemma 8. *Let $\mathcal{C}'^* \subseteq \mathcal{C}'_o$, then $\mathcal{C}^* = f(\mathcal{C}'^*)$ is a solution to Problem 3.*

Proof. It holds that $\mathcal{C}^* = f(\mathcal{C}'^*) \subseteq f(\mathcal{C}'_o) = \mathcal{C}_o$. □

We further define the following set

$$\mathcal{E}'_o = \{\eta \in \mathcal{D}' : (\eta \notin \mathcal{C}') \vee ((\eta \in \mathcal{C}') \wedge \mathsf{Stab}(f(\eta)))\}.$$

One can easily check that the following relation holds:

$$\mathcal{C}'_o = \mathcal{C}' \cap \mathcal{E}'_o. \tag{5.3}$$

Hence, from the previous equality and Lemma 8, we can solve Problem 3 by computing (a subset of) the set $\mathcal{E}'_o$. Moreover, $\mathcal{E}'_o$ satisfies the following monotonicity property:

Proposition 8. *For all $\eta, \eta' \in \mathcal{D}'$, the following implications hold:*

$$((\eta \leq \eta') \wedge (\eta \in \mathcal{E}'_o)) \implies \eta' \in \mathcal{E}'_o.$$

$$((\eta \leq \eta') \wedge (\eta' \notin \mathcal{E}'_o)) \implies \eta \notin \mathcal{E}'_o.$$

Proof. Let us assume $\eta \leq \eta'$ and $\eta \in \mathcal{E}'_o$. There are two cases:

1. If $\eta \notin \mathcal{C}'$, then either $-\eta'_4 \leq -\eta_4 < \eta_3 \leq \eta'_3$, or $-\eta'_2 \leq -\eta_2 < \eta_1 \leq \eta'_1$, or $-\eta'_4 \leq -\eta_4 < \eta_1 \leq \eta'_1$. In all three cases $\eta' \notin \mathcal{C}'$ and therefore $\eta' \in \mathcal{E}'_o$.

2. If $\eta \in \mathcal{C}'$ and $\mathsf{Stab}(f(\eta))$, then either $\eta' \notin \mathcal{C}'$ which implies $\eta' \in \mathcal{E}'_o$, or $\eta' \in \mathcal{C}'$. In this latter case, $\alpha = (\underline{\tau}, \overline{\tau}, \underline{h}, \overline{h}) = f(\eta)$ and $\alpha' = (\underline{\tau}', \overline{\tau}', \underline{h}', \overline{h}') = f(\eta')$ satisfy $\alpha \in \mathcal{C}$, $\alpha' \in \mathcal{C}$ and

$$\underline{\tau}' \geq \underline{\tau},\ \overline{\tau}' \leq \overline{\tau},\ \underline{h}' \geq \underline{h},\ \overline{h}' \leq \overline{h}. \tag{5.4}$$

It is straightforward to check that if (2.1-2.2) is β'-stable for $(\underline{\tau}, \overline{\tau}, \underline{h}, \overline{h}) \in \mathcal{C}$ then (2.1-2.2) is β'-stable for all $(\underline{\tau}', \overline{\tau}', \underline{h}', \overline{h}') \in \mathcal{C}$ satisfying (5.4). Thus, $\mathsf{Stab}(f(\eta'))$ holds and $\eta' \in \mathcal{E}'_o$.

This proves the first implication. For the second implication, it is sufficient to check that

$$\begin{aligned} & ((\eta \leq \eta') \wedge (\eta \in \mathcal{E}'_o)) \implies \eta' \in \mathcal{E}'_o \\ \equiv\ & \neg(\eta \leq \eta') \vee (\eta \notin \mathcal{E}'_o) \vee (\eta' \in \mathcal{E}'_o) \\ \equiv\ & ((\eta \leq \eta') \wedge (\eta' \notin \mathcal{E}'_o)) \implies \eta \notin \mathcal{E}'_o. \end{aligned}$$

□

The previous property is instrumental for computing a subset of $\mathcal{E}'_o$ since it allows us to state the following theorem:

Theorem 20. *Let $\underline{\eta}^1, \ldots, \underline{\eta}^{M_1} \in \mathcal{E}'_o$, and $\overline{\eta}^1, \ldots, \overline{\eta}^{M_2} \in \mathcal{D}' \setminus \mathcal{E}'_o$ and let*

$$\underline{\mathcal{E}}' = \bigcup_{j=1}^{M_1} \{\eta \in \mathcal{D}' : \underline{\eta}^j \leq \eta\},\ \overline{\mathcal{E}}' = D' \setminus \bigcup_{j=1}^{M_2} \{\eta \in \mathcal{D}' : \eta \leq \overline{\eta}^j\}.$$

Then, $\underline{\mathcal{E}}' \subseteq \mathcal{E}'_o \subseteq \overline{\mathcal{E}}'$. Moreover, $\mathcal{C}^* = f\left(\mathcal{C}' \cap \underline{\mathcal{E}}'\right)$ is a solution to Problem 6 and $\mathcal{C}_o \subseteq f(\mathcal{C}' \cap \overline{\mathcal{E}}')$.

Proof. $\underline{\mathcal{E}}' \subseteq \mathcal{E}'_o \subseteq \overline{\mathcal{E}}'$ is a direct consequence of Proposition 8. Then, from (5.6) and Lemmas 7 and 8, it follows that $\mathcal{C}^*$ is a solution to Problem 6 and $\mathcal{C}_o \subseteq f(\mathcal{C}' \cap \overline{\mathcal{E}}')$. □

5.1.2 Timing contract synthesis algorithm with stability guarantee

The previous theorem shows that it is possible to compute under and over-approximations of the set $\mathcal{E}'_o$ by sampling the parameter space $\mathcal{D}'$. In this section, we use this property to design a synthesis algorithm. Similar algorithms have been used in [LLGCM10, Ten14] for computing an approximation of the Pareto front of a monotone multi-criteria optimization problem. Indeed, this latter problem can be tackled by computing an under and over-approximation of a set satisfying a monotonicity property similar to that of Proposition 8.

Algorithm 2. *Timing contract synthesis*

function *TC_Synth(A,B,K,$\mathcal{D}$,β)*

input: $A \in \mathbb{R}^{p\times p}$, $B \in \mathbb{R}^{p\times m}$, $K \in \mathbb{R}^{m\times p}$, $\mathcal{D} = [\tau_{min}, \tau_{max}]^2 \times [h_{min}, h_{max}]^2$, $\beta \in \mathbb{R}^+$

output: $\mathcal{C}^* \subseteq \mathcal{C} \cap \mathcal{D}$ *such that for all* $(\underline{\tau}, \overline{\tau}, \underline{h}, \overline{h}) \in \mathcal{C}^*$, *(2.2-3.8) is* β^**-stable.*

parameter: $\varepsilon \in \mathbb{R}^+$

1: ***if*** $\eta_{min} = (\tau_{min}, -\tau_{max}, h_{min}, -h_{max}) \in \mathcal{E}'_o$ ***then***

2: *return* $\mathcal{C} \cap \mathcal{D}$;

3: ***else*** $\overline{\mathcal{E}}' := \mathcal{D}' \setminus \{\eta_{min}\}$;

4: ***end if***

5: ***if*** $\eta_{max} = (\tau_{max}, -\tau_{min}, h_{max}, -h_{min}) \notin \mathcal{E}'_o$ ***then***

6: *return* $\emptyset$;

7: ***else*** $\underline{\mathcal{E}}' := \{\eta_{max}\}$;

8: ***end if***

9: ***while*** $d(\underline{\mathcal{E}}', \overline{\mathcal{E}}') > \varepsilon$ ***do*** ▷ *main loop*

10: *Pick* $\eta \in \overline{\mathcal{E}}' \setminus \underline{\mathcal{E}}'$; ▷ *select next sample*

11: ***if*** $\eta \in \mathcal{E}'_o$ ***then*** $\underline{\mathcal{E}}' := \underline{\mathcal{E}}' \cup \{\eta' \in \mathcal{D}' : \eta \leq \eta'\}$;

12: ***else*** $\overline{\mathcal{E}}' := \overline{\mathcal{E}}' \setminus \{\eta' \in \mathcal{D}' : \eta' \leq \eta\}$;

13: ***end if***

14: ***end while***

15: *return* $f(\mathcal{C}' \cap \underline{\mathcal{E}}')$;

Algorithm 2 computes an under-approximation $\underline{\mathcal{E}}'$ and an over-approximation $\overline{\mathcal{E}}'$ of the set $\mathcal{E}'_o$ by sampling iteratively the parameter space $\mathcal{D}'$.

Lines 1 to 8 correspond to the initialization of these approximations by testing the lower bound η_{min} and the upper bound η_{max} of the set $\mathcal{D}'$. If $\eta_{min} \in \mathcal{E}'_o$, then by Theorem 20, $f(\mathcal{C}' \cap \mathcal{D}') = \mathcal{C} \cap \mathcal{D}$ is a solution to Problem 6. Note that in that case, all timing-contract parameters in $\mathcal{C} \cap \mathcal{D}$ guarantee the stability of (2.1-2.2). If $\eta_{min} \notin \mathcal{E}'_o$, then $\mathcal{D}' \setminus \{\eta_{min}\}$ is an over-approximation of $\mathcal{E}'_o$. Similarly, if $\eta_{max} \notin \mathcal{E}'_o$, then by Theorem 20, $\mathcal{E}'_o = \emptyset$. Note that in that case, no timing-contract parameters in $\mathcal{C} \cap \mathcal{D}$ can guarantee the stability of (2.1-2.2). If $\eta_{max} \in \mathcal{E}'_o$, then $\{\eta_{max}\}$ is an under-approximation of $\mathcal{E}'_o$.

Lines 9 to 14 describe the main loop of the timing contract synthesis algorithm. At any time of the execution, $\underline{\mathcal{E}}' \subseteq \mathcal{E}'_o \subseteq \overline{\mathcal{E}}'$ holds. We pick a sample $\eta \in \overline{\mathcal{E}}' \setminus \underline{\mathcal{E}}'$ which is the unexplored parameter region lying in the over-approximation of $\mathcal{E}'_o$ but not in its under-approximation. If $\eta \in \mathcal{E}'_o$ (or if $\eta \notin \mathcal{E}'_o$), then we update the under-approximation $\underline{\mathcal{E}}'$ (or the over-approximation $\overline{\mathcal{E}}'$) according to Theorem 20. The algorithm

stops when the Hausdorff distance between the $\underline{\mathcal{E}}'$ and $\overline{\mathcal{E}}'$ becomes smaller than ε. Of course, the choice of the sample $\eta \in \overline{\mathcal{E}}' \setminus \underline{\mathcal{E}}'$, at line 10, is crucial for the efficiency of the algorithm. In our implementation of the algorithm, we use the selection criteria proposed in [LLGCM10] which consists in choosing the sample that will produce the fastest decrease of the Hausdorff distance $d(\underline{\mathcal{E}}', \overline{\mathcal{E}}')$. In [Ten14] an alternative selection criteria based on multi-scale grid exploration was proposed.

Finally, it is important to note that Algorithm 2 needs testing if the samples $\eta \in \mathcal{E}'_o$ which require checking the condition $\mathsf{Stab}(f(\eta))$. In our implementation, this is done using Algorithm 1. If it returns true, then we can consider that $\mathsf{Stab}(f(\eta))$ holds. If it returns $\mathsf{unknown}$, we treat the sample as if $\mathsf{Stab}(f(\eta))$ is false. As a consequence, in practice it may be the case that $\overline{\mathcal{E}}'$ is not an over-approximation of $\mathcal{E}'_o$. However, it always holds that $\underline{\mathcal{E}}' \subseteq \mathcal{E}'_o$ and therefore the set returned by Algorithm 2 is always a valid solution to Problem 6. Note that the property $\mathsf{Stab}(f(\eta))$ need not be checked using Algorithm 1 but one can use any of the algorithms mentioned in Table 2.1.

Practically, we run Algorithm 2, N times where in each iteration $i \in \mathbb{N}_{[1,N]}$, the function $\mathsf{TC\_Synth\_Stab}(A_c^i, A_a^i, A_s^i, \mathcal{D}_i, \beta_i)$ returns the set $\mathcal{P}_{st}^i$ that guarantees β_i'-stability for system $\mathcal{S}_i$. Then at the end of this section we compute the set $\mathcal{P}_{st} = \mathcal{P}_{st}^1 \times \cdots \times \mathcal{P}_{st}^N$.

5.2 Guarantee on schedulability

In this section, given the set $\mathcal{P}_{st}$, defined in the previous section, we solve Problem 3 by synthesizing a set $\mathcal{P}^* \subseteq \mathcal{P}_{st}$ such that for all $(\underline{\tau}^1, \overline{\tau}^1, \underline{h}^1, \overline{h}^1, \ldots, \underline{\tau}^N, \overline{\tau}^N, \underline{h}^N, \overline{h}^N) \in \mathcal{P}^*$, the set of control tasks $\mathcal{T}$ is schedulable under timing contracts $\Theta = \{\theta(\underline{\tau}^1, \overline{\tau}^1, \underline{h}^1, \overline{h}^1), \ldots, \theta(\underline{\tau}^N, \overline{\tau}^N, \underline{h}^N, \overline{h}^N)\}$.

The timing contract parameters are given by the vector $p = (\underline{\tau}^1, \overline{\tau}^1, \underline{h}^1, \overline{h}^1, \ldots, \underline{\tau}^N, \overline{\tau}^N, \underline{h}^N, \overline{h}^N)$. Then, given the parameter sets $\mathcal{D}_i$ for all $i \in \mathbb{N}_{[1,N]}$ by (5.1), we define the following Boolean function for $p \in \mathcal{C}^N \cap (\mathcal{D}_1 \times \cdots \times \mathcal{D}_N)$:

$$\mathsf{Sched}(p) \equiv \mathcal{T} \text{ is } \textit{schedulable} \text{ under timing contracts } \Theta.$$

In order to solve Problem 3 we need to compute (a subset of) the set $\mathcal{P}_0$ defined by

$$\mathcal{P}_0 = \{p \in \mathcal{C}^N \cap (\mathcal{D}_1 \times \cdots \times \mathcal{D}_N) : \mathsf{Sched}(p)\}.$$

Re-parametrization

We define a new parameter $p' \in \mathcal{D}'_1 \times \cdots \times \mathcal{D}'_N$ with

$$\mathcal{D}'_i = \quad [\underline{\tau}^i_{min}, \underline{\tau}^i_{max}] \times [-\overline{\tau}^i_{max}, -\overline{\tau}^i_{min}] \times [h^i_{min}, h^i_{max}] \times \\ [-h^i_{max}, -h^i_{min}], \;\; i \in \mathbb{N}_{[1,N]}, \tag{5.5}$$

such that $p' = (\eta^1_1, \eta^1_2, \eta^1_3, \eta^1_4, \ldots, \eta^N_1, \eta^N_2, \eta^N_3, \eta^N_4)$. We further define the map $f : (\mathcal{D}'_1 \times \cdots \times \mathcal{D}'_N) \to (\mathcal{D}_1 \times \cdots \times \mathcal{D}_N)$ such that $f(p') = p = (\underline{\tau}^1, \overline{\tau}^1, \underline{h}^1, \overline{h}^1, \ldots, \underline{\tau}^N, \overline{\tau}^N, \underline{h}^N, \overline{h}^N)$, where for all $i \in \mathbb{N}_{[1,N]}$

$$\underline{\tau}^i = \eta^i_1, \; \overline{\tau}^i = \min(-\eta^i_2, -\eta^i_4), \; \underline{h}^i = \eta^i_3, \; \overline{h}^i = -\eta^i_4.$$

We associate to the parameter p' a constraint set $(\mathcal{C}')^N$ where $\mathcal{C}'$ is given by (5.2). Last we define the set $\mathcal{P}'_o$ by :

$$\mathcal{P}'_o = \left\{ p' \in \mathcal{D}'_1 \times \cdots \times \mathcal{D}'_N : ((p' \in (\mathcal{C}')^N) \wedge \mathsf{Sched}(f(p'))) \right\}.$$

One can check that the following relation holds:

$$\mathcal{P}_o = f(\mathcal{P}'_o). \tag{5.6}$$

We can then show that $\mathcal{P}'_o$ satisfies the following monotonicity property:

Proposition 9. *For all $p'_1, p'_2 \in \mathcal{D}'_1 \times \cdots \times \mathcal{D}'_N$, the following implications hold:*

$$((p'_2 \leq p'_1) \wedge (p'_1 \in \mathcal{P}'_o)) \implies p'_2 \in \mathcal{P}'_o.$$

$$((p'_2 \leq p'_1) \wedge (p'_2 \notin \mathcal{P}'_o)) \implies p'_1 \notin \mathcal{P}'_o.$$

Proof. Let $p'_1 = (\eta^1_1, \eta^1_2, \eta^1_3, \eta^1_4, \ldots, \eta^N_1, \eta^N_2, \eta^N_3, \eta^N_4)$ and $p'_2 = (\alpha^1_1, \alpha^1_2, \alpha^1_3, \alpha^1_4, \ldots, \alpha^N_1, \alpha^N_2, \alpha^N_3, \alpha^N_4)$. We assume $p'_2 \leq p'_1$ and $p'_1 \in \mathcal{P}'_o$. Then $p'_1 \in (\mathcal{C}')^N$ which implies $\alpha^i_1 \leq \eta^i_1 \leq -\eta^i_2 \leq \alpha^i_2$, $\alpha^i_1 \leq \eta^i_1 \leq -\eta^i_4 \leq -\alpha^i_4$, and $\alpha^i_3 \leq \eta^i_3 \leq -\eta^i_4 \leq -\alpha^i_4$ for all $i \in \mathbb{N}_{[1,N]}$. Thus $p'_2 \in (\mathcal{C}')^N$. We also have $\mathsf{Sched}(f(p'_1))$. In this case, $p_1 = f(p'_1) = (\underline{\tau}^1_1, \overline{\tau}^1_1, \underline{h}^1_1, \overline{h}^1_1, \ldots, \underline{\tau}^N_1, \overline{\tau}^N_1, \underline{h}^N_1, \overline{h}^N_1)$ and $p_2 = f(p'_2) = (\underline{\tau}^1_2, \overline{\tau}^1_2, \underline{h}^1_2, \overline{h}^1_2, \ldots, \underline{\tau}^N_2, \overline{\tau}^N_2, \underline{h}^N_2, \overline{h}^N_2)$ satisfy $p_1 \in \mathcal{C}^N$, $p_2 \in \mathcal{C}^N$ and for all $i \in \mathbb{N}_{[1,N]}$

$$\underline{\tau}^i_2 \leq \underline{\tau}^i_1, \; \overline{\tau}^i_2 \geq \overline{\tau}^i_1, \; \underline{h}^i_2 \leq \underline{h}^i_1, \; \overline{h}^i_2 \geq \overline{h}^i_1. \tag{5.7}$$

It is easy to check that if $\mathcal{T}$ is schedulable under timing contracts $\Theta_1 = \{\theta(\underline{\tau}_1^1, \overline{\tau}_1^1, \underline{h}_1^1, \overline{h}_1^1), \ldots, \theta(\underline{\tau}_1^N, \overline{\tau}_1^N, \underline{h}_1^N, \overline{h}_1^N)\}$ then $\mathcal{T}$ is schedulable under timing contracts $\Theta_2 = \{\theta(\underline{\tau}_2^1, \overline{\tau}_2^1, \underline{h}_2^1, \overline{h}_2^1), \ldots, \theta(\underline{\tau}_2^N, \overline{\tau}_2^N, \underline{h}_2^N, \overline{h}_2^N)\}$ for all $p_2 \in \mathcal{C}$ satisfying (5.7). Thus, $\mathsf{Sched}(f(p_2'))$ holds and $p_2' \in \mathcal{P}_o'$.

This proves the first implication. For the second implication, it is sufficient to check that

$$\begin{aligned} & ((p_2' \leq p_1') \wedge (p_1' \in \mathcal{P}_o')) \implies p_2' \in \mathcal{P}_o' \\ \equiv \quad & \neg(p_2' \leq p_1') \vee (p_1' \notin \mathcal{P}_o') \vee (p_2' \in \mathcal{P}_o') \\ \equiv \quad & ((p_2' \leq p_1') \wedge (p_2' \notin \mathcal{P}_o')) \implies p_1' \notin \mathcal{P}_o'. \end{aligned}$$

□

Now using Proposition 9 and the set $\mathcal{P}_{st}$ obtained in Section 5.1 we can sample the parameter space to solve Problem 3.

Theorem 21. *Let $\underline{p}^1, \ldots, \underline{p}^{M_1} \in \mathcal{P}_o'$, and $\overline{p}^1, \ldots, \overline{p}^{M_2} \in \mathcal{D}_1' \times \cdots \times \mathcal{D}_N' \setminus \mathcal{P}_o'$ and let*

$$\begin{aligned} \underline{\mathcal{P}}' &= \bigcup_{j=1}^{M_1} \{p' \in \mathcal{D}_1' \times \cdots \times \mathcal{D}_N' : \underline{p}^j \geq p'\}, \\ \overline{\mathcal{P}}' &= (\mathcal{D}_1' \times \cdots \times \mathcal{D}_N') \setminus \bigcup_{j=1}^{M_2} \{p' \in \mathcal{D}_1' \times \cdots \times \mathcal{D}_N' : p' \geq \overline{p}^j\}. \end{aligned}$$

Then, $\underline{\mathcal{P}}' \subseteq \mathcal{P}_o' \subseteq \overline{\mathcal{P}}'$. Moreover, $\mathcal{P}^* = f\left(\underline{\mathcal{P}}'\right) \cap \mathcal{P}_{st}$ is a solution to Problem 3.

Proof. $\underline{\mathcal{P}}' \subseteq \mathcal{P}_o' \subseteq \overline{\mathcal{P}}'$ is a direct consequence of Proposition 9. Then, it follows that $\mathcal{P}^*$ is a solution to Problem 3. □

5.3 Algorithm for timing contract synthesis

Theorem 21 shows that it is possible to compute under and over-approximations of the set $\mathcal{P}_o'$ by sampling the parameter space $\mathcal{D}_1' \times \cdots \times \mathcal{D}_N'$ as done in Section 5.1.2.

Algorithm 3. *Timing contract synthesis*

function *TC_Synth($\mathcal{T}$,$\{\mathcal{D}_1, \ldots, \mathcal{D}_N\}$,$\mathcal{P}_{st}$)*

input: $\mathcal{T}$, $\mathcal{D}_i = [\tau_{min}^i, \tau_{max}^i]^2 \times [h_{min}^i, h_{max}^i]^2, i \in \mathbb{N}_{[1,N]}$, $\mathcal{P}_{st} \subseteq \mathcal{C}^N \cap (\mathcal{D}_1 \times \cdots \times \mathcal{D}_N)$,

output: $\mathcal{P}^* \subseteq \mathcal{C}^N \cap (\mathcal{D}_1 \times \cdots \times \mathcal{D}_N)$

parameter: $\varepsilon \in \mathbb{R}^+$

1: ***if*** $p'_{max} \in \mathcal{P}'_o$ ***then***

2: *return* $(\mathcal{D}_1 \times \cdots \times \mathcal{D}_N) \cap \mathcal{P}_{st}$;

3: ***else*** $\overline{\mathcal{P}}' := (\mathcal{D}'_1 \times \cdots \times \mathcal{D}'_N) \setminus \{p'_{max}\}$;

4: ***end if***

5: ***if*** $p'_{min} \notin \mathcal{P}'_o$ ***then***

6: *return* $\emptyset$;

7: ***else*** $\underline{\mathcal{P}}' := \{p'_{min}\}$;

8: ***end if***

9: ***while*** $d(\underline{\mathcal{P}}', \overline{\mathcal{P}}') > \varepsilon$ ***do*** ▷ *main loop*

10: *Pick* $p' \in \overline{\mathcal{P}}' \setminus \underline{\mathcal{P}}'$; ▷ *select next sample*

11: ***if*** $p' \in \mathcal{P}'_o$ ***then***

12: $\underline{\mathcal{P}}' := \underline{\mathcal{P}}' \cup \{p'_* \in (\mathcal{D}'_1 \times \cdots \times \mathcal{D}'_N) : p'_* \leq p'\}$;

13: ***else*** $\overline{\mathcal{P}}' := \overline{\mathcal{P}}' \setminus \{p'_* \in (\mathcal{D}'_1 \times \cdots \times \mathcal{D}'_N) : p' \leq p'_*\}$;

14: ***end if***

15: ***end while***

16: *return* $f\left(\underline{\mathcal{P}}'\right) \cap \mathcal{P}_{st}$;

Algorithm 3 computes an under-approximation $\underline{\mathcal{P}}'$ and an over-approximation $\overline{\mathcal{P}}'$ of the set $\mathcal{P}'_o$ by sampling iteratively the parameter space $\mathcal{D}'_1 \times \cdots \times \mathcal{D}'_N$.

Lines 1 to 8 initialize these approximations by testing both the lower bound $p'_{min} = (\tau^1_{min}, -\tau^1_{max}, h^1_{min}, -h^1_{max}, \ldots, \tau^N_{min}, -\tau^N_{max}, h^N_{min}, -h^N_{max})$ and the upper bound $p'_{max} = (\tau^1_{max}, -\tau^1_{min}, h^1_{max}, -h^1_{min}, \ldots, \tau^N_{max}, -\tau^N_{min}, h^N_{max}, -h^N_{min})$ of the set $\mathcal{D}'_1 \times \cdots \times \mathcal{D}'_N$. If $p'_{max} \in \mathcal{P}'_o$, then by Theorem 21, $f(\mathcal{D}'_1 \times \cdots \times \mathcal{D}'_N) \cap \mathcal{P}_{st} = (\mathcal{D}_1 \times \cdots \times \mathcal{D}_N) \cap \mathcal{P}_{st}$ is a solution to Problem 3. Note that in that case, all timing-contract parameters, $(\underline{\tau}^1_1, \overline{\tau}^1_1, \underline{h}^1_1, \overline{h}^1_1, \ldots, \underline{\tau}^N_1, \overline{\tau}^N_1, \underline{h}^N_1, \overline{h}^N_1) \in \mathcal{D}_1 \times \cdots \times \mathcal{D}_N$ guarantee the schedulability of $\mathcal{T}$ under timing contracts $\Theta = \{\theta(\underline{\tau}^1, \overline{\tau}^1, \underline{h}^1, \overline{h}^1), \ldots, \theta(\underline{\tau}^N, \overline{\tau}^N, \underline{h}^N, \overline{h}^N)\}$. If $p_{max} \notin \mathcal{P}'_o$, then $(\mathcal{D}'_1 \times \cdots \times \mathcal{D}'_N) \setminus \{p'_{max}\}$ is an over-approximation of $\mathcal{P}'_o$. Similarly, if $p'_{min} \notin \mathcal{P}'_o$, then by Theorem 21, $\mathcal{P}'_o = \emptyset$. Note that in that case, no timing-contracts can guarantee the schedulability of $\mathcal{T}$. If $p'_{min} \in \mathcal{P}'_o$, then $\{p'_{min}\}$ is an under-approximation of $\mathcal{P}'_o$.

Lines 9 to 14 describe the main loop of the timing contract synthesis algorithm. At any time of the execution, $\underline{\mathcal{P}}' \subseteq \mathcal{P}'_o \subseteq \overline{\mathcal{P}}'$ holds. We pick a sample $p' \in \overline{\mathcal{P}}' \setminus \underline{\mathcal{P}}'$ which is the unexplored parameter region lying in the over-approximation of $\mathcal{P}'_o$ but not in its under-approximation. If $p' \in \mathcal{P}'_o$ (or if $p' \notin \mathcal{P}'_o$), then we update the under-approximation $\underline{\mathcal{P}}'$ (or the over-approximation $\overline{\mathcal{P}}'$) according to Theorem 21. The algorithm stops when the Hausdorff distance between the $\underline{\mathcal{P}}'$ and $\overline{\mathcal{P}}'$ becomes smaller than ε. One rising

issue is that the choice of the sample $p' \in \overline{\mathcal{P}}' \setminus \underline{\mathcal{P}}'$, at line 10, is crucial for the efficiency of the algorithm. In our implementation, we use the same selection criteria as that in section which consists in choosing the sample that will produce the fastest decrease of the Hausdorff distance $d(\underline{\mathcal{P}}', \overline{\mathcal{P}}')$.

Furthermore, it is important to note that Algorithm 3 needs testing if the samples $p' \in \mathcal{P}'_o$, which requires checking the condition $\mathsf{Sched}(f(p'))$. In our implementation, this is done using the method proposed in Section 4.1, which assures us that the set $f(\underline{\mathcal{P}}')$ tends to $\mathcal{P}_0$ as $\varepsilon \to 0$, where $\mathcal{P}_0$ is the set of all solutions $(\underline{\tau}^1, \overline{\tau}^1, \underline{h}^1, \overline{h}^1, \ldots, \underline{\tau}^N, \overline{\tau}^N, \underline{h}^N, \overline{h}^N)$ such that $\mathcal{T}$ is schedulable under timing contracts $\Theta = \{\theta(\underline{\tau}^1, \overline{\tau}^1, \underline{h}^1, \overline{h}^1), \ldots, \theta(\underline{\tau}^N, \overline{\tau}^N, \underline{h}^N, \overline{h}^N)\}$.

Finally a solution to Problem 3 is computed as $\mathcal{P}^* = \mathsf{TC\_Synth}\ (\mathcal{T}, \{\mathcal{D}_1, \ldots, \mathcal{D}_N\}, \mathcal{P}_{st})$.

5.4 Illustrative example

In this section, we show an application of the timing contract synthesis algorithm on two systems sharing a common computational resource.

We implemented the scheduling approach presented in Section 4.1 using UPPAAL-TIGA [BCD+07] and Algorithm 1 in Matlab.

Example 11. *We reconsider the two systems $\mathcal{S}_1 = (A_1, B_1, K_1)$ and $\mathcal{S}_2 = (A_2, B_2, K_2)$, taken from [Bri13], given by matrices (4.2) and (4.3) respectively.*

Furthermore, we set the best and worst case execution times for each task as $\underline{c}^1 = 0.12$, $\overline{c}^1 = 0.35$, $\underline{c}^2 = 0.04$, and $\overline{c}^2 = 0.12$.

We now consider the timing contract synthesis problem for systems $\mathcal{S}_1$ and $\mathcal{S}_2$ and the set of control tasks $\mathcal{T} = \{(\underline{c}^1, \overline{c}^1), (\underline{c}^2, \overline{c}^2)\}$. We fix $\beta_1 = \beta_2 = 0$, $\underline{\tau}^1 = 0.1$, $\underline{h}^1 = 0.3$, $\underline{\tau}^2 = 0.2$, and $\underline{h}^2 = 0.8$ and consider the following bounds on parameters $\mathcal{D}_1 = [0.1, 0.1] \times [0.1, 0.76] \times [0.3, 0.3] \times [0.3, 1.72]$ and $\mathcal{D}_2 = [0.2, 0.2] \times [0.2, 1.16] \times [0.8, 0.8] \times [0.8, 2.02]$.

Using Algorithm 2, we synthesize the set $\mathcal{P}_{st} = \mathcal{P}^1_{st} \times \mathcal{P}^2_{st} \subseteq \mathcal{C}^2 \cap (\mathcal{D}_1 \times \mathcal{D}_2)$ such that for all $(\underline{\tau}^1, \overline{\tau}^1, \underline{h}^1, \overline{h}^1, \underline{\tau}^2, \overline{\tau}^2, \underline{h}^2, \overline{h}^2) \in \mathcal{P}_{st}$, system $\mathcal{S}_i = (A_i, B_i, K_i)$ is β'_i-stable under timing contract $\theta(\underline{\tau}_i, \overline{\tau}_i, \underline{h}_i, \overline{h}_i)$, for all $i \in \mathbb{N}_{[1,2]}$. The sets $\mathcal{P}^1_{st}$ and $\mathcal{P}^2_{st}$, in the $(\overline{\tau}^1, \overline{h}^1)$ plane and $(\overline{\tau}^2, \overline{h}^2)$ plane respectively, are shown by Figure 5.3.

Then, we search for a set $\mathcal{P}^* \subseteq \mathcal{P}_{st}$ such that for all $(\underline{\tau}^1, \overline{\tau}^1, \underline{h}^1, \overline{h}^1, \underline{\tau}^2, \overline{\tau}^2, \underline{h}^2, \overline{h}^2) \in \mathcal{P}^*$, the set of control tasks $\mathcal{T}$ is schedulable under timing contracts $\Theta = \{\theta(\underline{\tau}^1, \overline{\tau}^1, \underline{h}^1, \overline{h}^1), \theta(\underline{\tau}^2, \overline{\tau}^2, \underline{h}^2, \overline{h}^2)\}$. We set the parameter $\varepsilon = 0.04$, and apply Algorithm 3 to compute the set $\mathcal{P}^*$. The algorithm tested 944 pa-

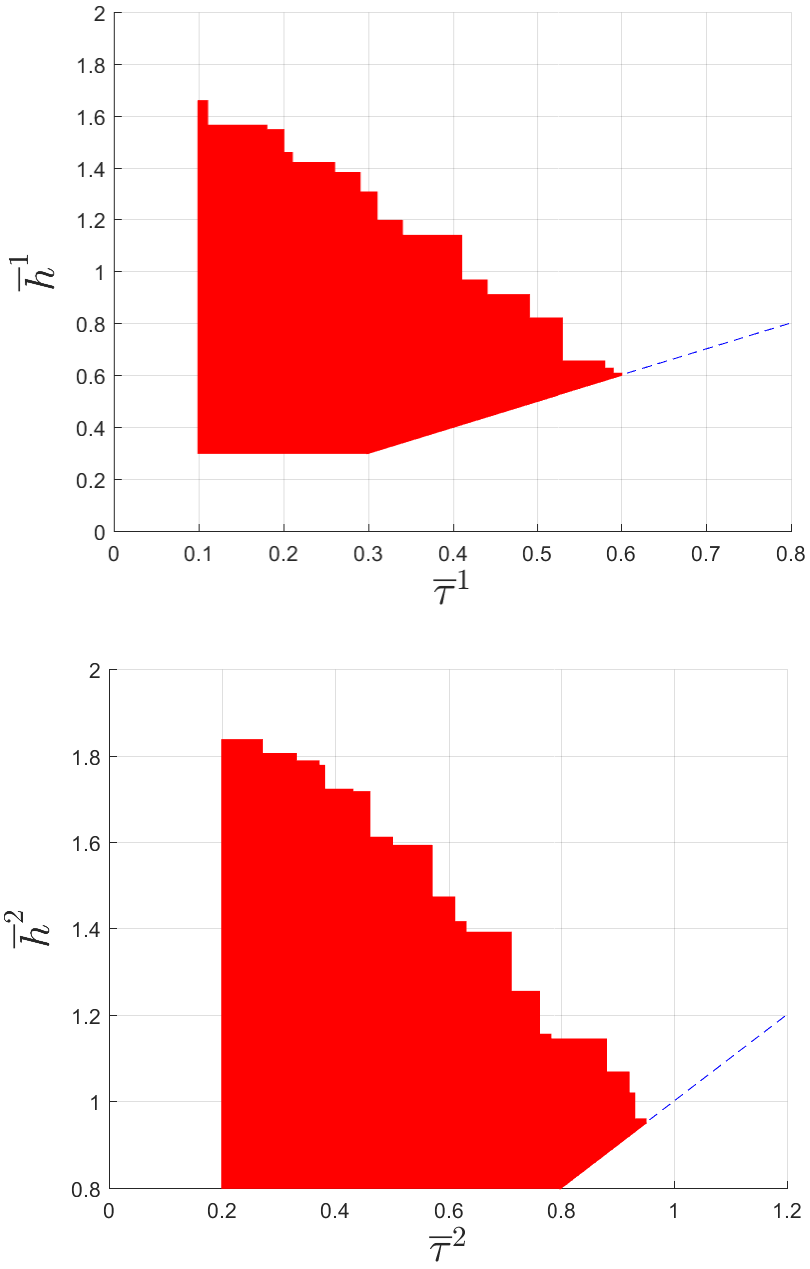


Figure 5.3: Timing contract parameters that guarantee stability for each system $\mathcal{S}_1$ and $\mathcal{S}_2$: $\mathcal{P}^1_{st}$ (top) and $\mathcal{P}^2_{st}$ (bottom).

rameter samples and the computation time was 43.4 *minutes. A section of the sets* $f(\underline{\mathcal{P}}')$ *and* $\mathcal{P}^*$ *in the* $(0.1, \overline{\tau}^1, 0.3, \overline{h}^1, 0.6, 0.6, 1.15, 1.15)$ *domain is shown in Figure 5.4.*

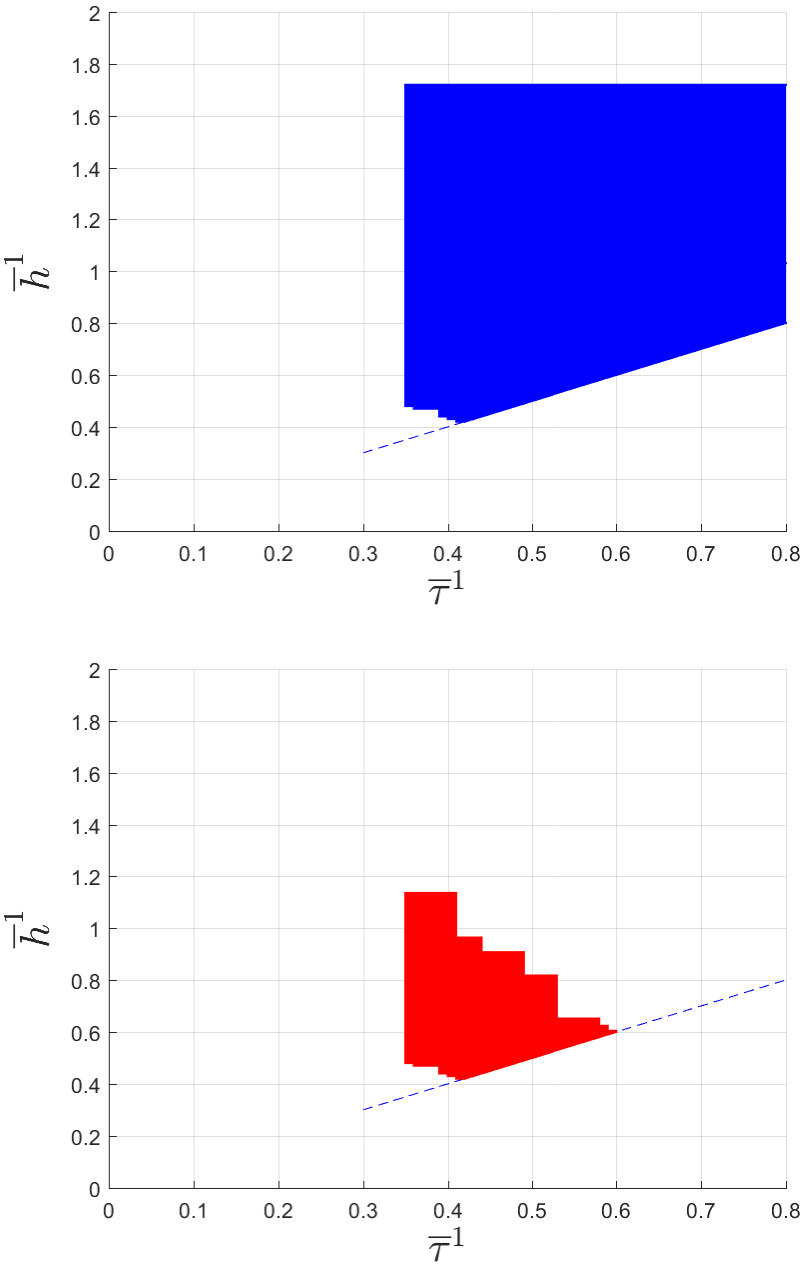


Figure 5.4: A 2D section of $f(\underline{\mathcal{P}'})$ (top) and $\mathcal{P}^*$ (bottom) in the $(0.1, \overline{\tau}^1, 0.3, \overline{h}^1, 0.6, 0.6, 1.15, 1.15)$ domain.

Chapter 6

Conclusion and perspectives

6.1 Summary

In this thesis, we handle three different problems that arises in CPSs and specifically for embedded control systems. For the stability verification problem we proposed in the well known modeling framework of differential inclusions a novel approach based on reachability analysis. Results of this approach allowed us to extend our work to other two problems which are self-triggered control problem and stability verification under stochastic timing contracts. For the scheduling problem a novel approach based on timed safety games is proposed allowing us to synthesize a scheduling strategy guaranteeing that each processor is at most accessed by one control loop at a time and that all timing contracts are satisfied. Also, a simplified condition for schedulability is given reducing the computational complexity of the approach. For the last problem, which is the parameter synthesis problem, we suggested a re-parameterization of the timing contracts allowing us to explore the parameter space using monotonic sets. The synthesized timing contracts then guarantee the schedulability of the control tasks and the stability of each control loop.

Throughout the thesis, results are illustrated with numerical simulations and comparisons with existing results in the literature, where our stability verification algorithm and self-triggered strategy have promising and competitive results.

Next, we provide numerous directions for future developments.

6.2 Future work

We describe below the most promising directions to develop further the results presented in this thesis.

6.2.1 Stability verification

The stability verification approach presented in Chapter 3 provides just a sufficient stability condition which is given by Corollary 4. Nonetheless if the over-approximation $\overline{\Phi}$ satisfies additional assumptions, necessity

could be established as well. Such assumptions are related to the tightness of the over-approximation schemes used to compute $\overline{\Phi}$ where if such schemes do not suffer from the wrapping effect, or the effect of over-approximating the error, the first necessary and sufficient practical conditions could be established in the literature. Another direction to improve the results is by improving the self-triggered strategy synthesized in Section 3.4. The strategy at hand is not optimal in the sense that there may exist other strategies that lead to less samplings in a given time window. Given a polytopic covering (3.41), we may improve the condition given in (3.42), and which associate a fixed sampling period for each set in this covering, by proposing a dynamic assignment of sampling periods or by choosing a sampling period for each set of the covering that leads to the least number of samplings in a given time window. Another problem that the existing algorithms in literature suffer from is the computational tractability where most of the results are evaluated on simple low dimensional systems. Although we have no clear ideas to handle this, we may propose in the long term to develop reachability computations that relies on an effective set representation to compute the reachable set Φ given by (3.17).

6.2.2 Scheduling

In Chapter 4, we synthesize scheduling strategies using the tool UPPAAL Tiga. However, the synthesized strategy is given just by text and therefore could not be imported easily to matlab in order to simulate the supervised systems. As a result, some programming effort should be done to overcome this hindrance such as translating UPPAAL models using stateflow in matlab as done in [PLMS11]. Other directions are to find optimal schedules in the sense that the control loop is to be closed as soon as possible for each task. Tools that uses priced time automata like UPPAAL Cora [BLR05] could be used for this purpose. In the long term, scheduling algorithms for preemptive scheduling under timing contracts shall be handled unlike the policies synthesized in Chapter 4. At this point, timed automata is not believed capable of handling this problem since once a task is preempted the time in the task should stop which could not be obviously translated with a timed automata as the one in Figure 4.1.

6.2.3 Parameter synthesis

The choice of the next sample in the timing contract synthesis algorithm, or Algorithm 3, is not quite obvious and requires the use of existing methods as the one in [LLGCM10]. The latter's complexity is exponentially increasing with respect to the number of systems, and consequently to the dimension of the parameter space, which requires future work to come out with other methods to keep track of the distance between the under-

and over-approximation of the parameter space and to choose repeatedly the sample from the unexplored set.

6.2.4 Others

Throughout the thesis the controller is assumed to be given. Therefore, for systems under timing contracts it would be interesting to design the controller using reachability analysis as done in [AL14b]. Therein, the authors used contractive sets to derive new synthesis methods for constrained stabilization of linear systems. Other long term interesting problems for systems under timing contracts are the co-design of schedules and controllers, co-design of controllers and timing contracts, design of controllers under temporal logic specifications, analysis and design of systems with stochastic timing contracts, and analysis and design of nonlinear plants.

Appendices

Appendix A

Formal definitions of semantics of TA and strategies

This appendix, taken mainly from [CDF$^+$05], is intended to give the formal semantics of a timed automata [AD94] given by Definition 11 in Chapter 4. Note that the semantics in the sequel are used mainly in this manuscript for the proof of Proposition 7 as in Appendix B.

Let us start with some notations and definitions: Let X be a finite set of real-valued clocks. We note $\mathcal{C}(X)$ the set of constraints Λ generated by the grammar: $\Lambda ::= x \sim k | x-y \sim k | \Lambda \wedge \Lambda$, where $k \in \mathbb{Z}$, $x, y \in X$, and $\sim \in \{<, \leq, =, >, \geq\}$; then we can say $\mathcal{B}(X) \subset \mathcal{C}(X)$. A *valuation* of the variables in X is a mapping $X \to \mathbb{R}_0^+$ (thus $\mathbb{R}_{\geq 0}^X$). We refer to $\overline{0}$ for the valuation that assigns 0 to each cl0ck. For $Y \subseteq X$, we denote by $v[Y]$ the valuation assigning 0 (respectively $v(x)$) for any $x \in Y$ (respectively $x \in X \setminus Y$). We write $v + \delta$ for $\delta \in \mathbb{R}_0^+$ the valuation such that for all $x \in X$, $(v+\delta)(x) = v(x) + \delta$. For $g \in \mathcal{C}(X)$ and $v \in \mathbb{R}_{\geq 0}^X$, we write $v \models g$ if v satisfies g and $[g]$ denotes the set of valuations $\{v \in \mathbb{R}_{\geq 0}^X : v \models g\}$.

A state of a TA is a pair $(l, v) \in L \times \mathbb{R}_{\geq 0}^X$ that consists of a discrete part and a valuation of the clocks. From a state (l, v) such that $v \models Inv(l)$, a TA can either let time progress or do a discrete transition and reach a new state. This is defined by the transition relation $\to$ built as follows: for $a \in Act$, $(l, v) \xrightarrow{a} (l', v')$ if there exists a transition $(l, \lambda, act, C', l') \in E$ such that $v \models \lambda$, $v' = v[C']$, and $v' \models Inv(l')$; for $\delta \geq 0$, $(l, v) \xrightarrow{\delta} (l, v')$ if $v' = v + \delta$ and $v, v' \in [Inv(l)]$. Then we can define the semantics of a timed automaton TA as a labeled transition system $S_{\mathsf{TA}} = (Q_{\mathsf{TA}}, q_0, \to_{S_{\mathsf{TA}}})$ where $Q_{\mathsf{TA}} = L \times \mathbb{R}_{\geq 0}^X$, $q_0 = (l_0, \overline{0})$, and the set of labels is $Act \cup \mathbb{R}_0^+$. A *run* of a timed automaton TA is a sequence of alternating time and discrete transitions in S_{TA}. $Runs((l_0, \overline{0}), \mathsf{TA})$ denotes the set of runs that start in (l, v). Also we write $Runs(\mathsf{TA})$ for $Runs((l_0, \overline{0}), \mathsf{TA})$. If ρ is a finite run we denote by $last(\rho)$ the last state of the run.

We use next the notion of a run to define memoryless strategies for TGA. In a safety game, or a *safety control problem*, we ask for the strategy f such that a given TGA supervised by f constantly avoid a set of undesired locations $\overline{L}_u$. Formally, we can define a strategy as a function that suggests to the scheduler to either "do a particular controllable action" or "just wait" which will be denoted by the special symbol λ.

Definition 17. *(Memoryless strategy) Given* $\mathsf{TGA} = (L, l_0, Act_c, Act_u, C, E, I)$. *A memoryless strategy* f *over* TGA *is a partial function from* Q_{TGA} *to* $Act_c \cup \{\lambda\}$ *such that for every* $q \in Q_{\mathsf{TGA}}$ *if* $f(q) \in Act_c$ *then* $q \xrightarrow{f(q)}_{S_{\mathsf{TGA}}} (l', v')$ *for some* (l', v').

Consequently, under the supervision of a winning strategy f any finite or infinite (omitting runs with an infinite number of consecutive time transitions of duration 0) run $\rho = (l_0, v_0) \xrightarrow{e_0}_{S_{\mathsf{TGA}}} \cdots \xrightarrow{e_n}_{S_{\mathsf{TGA}}} (l_{n+1}, v_{n+1}) \cdots \in Runs(G)$ is *winning* and thus satisfies the property that for all $k \in \mathbb{N}$, $(l_k, v_k) \notin \overline{L}_u$.

The restricted behavior of a timed automaton, under the supervision of a strategy, is formally defined using the notion of an outcome.

Definition 18. *(Outcome) Given* $\mathsf{TGA} = (L, l_0, Act_c, Act_u, C, E, I)$ *and a memoryless strategy* f *over* TGA, *the outcome* $\mathsf{Outcome}_{\mathsf{TGA}}(q, f)$ *of* f *from* q *in* S_{TGA} *is the subset of* $Runs(q, \mathsf{TGA})$ *defined by:*

- $q \in \mathsf{Outcome}_{\mathsf{TGA}}(q, f)$;
- *if* $\rho \in \mathsf{Outcome}_{\mathsf{TGA}}(q, f)$ *then* $\rho' = \rho \xrightarrow{e}_{S_{\mathsf{TGA}}} q' \in \mathsf{Outcome}_{\mathsf{TGA}}(q, f)$ *if* $\rho' \in Runs(q, \mathsf{TGA})$ *and one of the following three conditions hold:*
  1. $e \in Act_u$;
  2. $e \in Act_c$ *and* $e = f(last(\rho))$;
  3. $e \in \mathbb{R}_0^+$ *and for all* $0 \leq e' < e$, *there exists* $q'' \in Q_{\mathsf{TGA}}$ *such that* $last(\rho) \xrightarrow{e'}_{S_{\mathsf{TGA}}} q'' \wedge f(q'') = \lambda$.
- *for an infinite run* ρ, $\rho \in \mathsf{Outcome}_{\mathsf{TGA}}(q, f)$ *if all the finite prefixes of* ρ *are in* $\mathsf{Outcome}_{\mathsf{TGA}}(q, f)$.

Appendix B

Proof of Proposition 7

The proof of Proposition 7 is moved to this appendix since it uses the formal semantics of a timed automaton, which are given in Appendix A. In such a way a sketch of the proof could be followed easier as in the lines below.

If there exists a strategy f' for $(\mathsf{TGA}',\overline{L}_u)$ then we can synthesize a winning strategy f for $(\mathsf{TGA},\overline{L}_u)$ where:

- $f(q) = f'(q)$ for all $q \in dom(f')$.
- As for every $q = (l, v) \in Q_{\mathsf{TGA}}$ such that

$$\exists i \in \mathbb{N}_{[1,N]} : (l^i = Preac^i) \wedge (q \notin dom(f')),$$

  with $l = (l^1, \ldots, l^N)$, then $f(q) = \lambda$.

Since a strategy f exists for $(\mathsf{TGA}',\overline{L}_u)$ then $\mathcal{T}$ is schedulable and the sufficient condition is fulfilled.

Now the schedulability of $\mathcal{T}$ under Θ implies the existence of a winning strategy f for $(\mathsf{TGA},\overline{L}_u)$. Then we can synthesize a winning strategy f' for $(\mathsf{TGA},\overline{L}_u)$, and therefore prove the necessary condition, such that $f'(q) = f(q)$ for all $q = (l, v) \in Q_{\mathsf{TGA}}$, where $l = (l^1, \ldots, l^N)$, except those in the following two conditions:

- for each $i \in \mathbb{N}_{[1,N]}$ such that $\overline{\tau}^i \geq \underline{h}^i$: if furthermore $\underline{\tau}^i \geq \underline{h}^i$ then using Lemma 6 it is direct that $\mathcal{T}$ is schedulable under $\Theta = \{\theta(\underline{\tau}^1, \overline{\tau}^1, \underline{h}^1, \overline{h}^1), \ldots, \theta(\underline{\tau}^i, \overline{\tau}^i, \underline{h}^i, \overline{h}^i), \ldots, \theta(\underline{\tau}^N, \overline{\tau}^N, \underline{h}^N, \overline{h}^N)\}$ if and only if it is schedulable under $\Theta* = \{\theta(\underline{\tau}^1, \overline{\tau}^1, \underline{h}^1, \overline{h}^1), \ldots, \theta(\underline{\tau}^i, \overline{\tau}^i, \underline{h}^i_*, \overline{h}^i), \ldots, \theta(\underline{\tau}^N, \overline{\tau}^N, \underline{h}^N, \overline{h}^N)\}$ where $\underline{h}^i_* = \underline{\tau}^i$. Therefore, we only need to take the case when $\underline{\tau}^i \leq \underline{h}^i \leq \overline{\tau}^i$. In the latter, when $f(q) = actuate^i$ and

$$\exists i \in \mathbb{N}_{[1,N]} : (l^i = Preac^i) \wedge \left(v \models (c^i \geq \underline{\tau}^i) \wedge v \models (c^i < \underline{h}^i)\right).$$

  In such an exception, we just set $f'(q) = \lambda$ and take the controllable action $actuate^i$ when $c^i = \underline{h}^i$ i.e. we do not take the controllable action $actuate^i$ as long as $c^i < \underline{h}^i$. By doing so we are sure that there

exists (l_1, v_1), and (l_2, v_2) such that $q \xrightarrow{f(q)=\lambda}_{S_{\mathsf{TGA}}} (l_1, v_1) \xrightarrow{e_0}_{S_{\mathsf{TGA}}} \ldots \xrightarrow{e_n} (l_2, v_2)$ with $l_1^i = l_2^i = Presam^i$, $(v_1 \models (c_1^i < \underline{h}^i)) \wedge (v_1 \models (c_1^i \geq \underline{\tau}^i))$, and $v_2 \models (c_2^i = \underline{h}^i)$

- for each $i \in \mathbb{N}_{[1,N]}$ such that $\overline{\tau}^i \leq \underline{h}^i$: When $f(q) = actuate^i$ and

$$\exists i \in \mathbb{N}_{[1,N]} : (l^i = Preac^i) \wedge \left(v \models (c^i < \overline{\tau}^i)\right).$$

  In such an exception, we just set $f'(q) = \lambda$ and take the controllable action $actuate^i$ when $c^i = \overline{\tau}^i$ i.e. we do not take the controllable action $actuate^i$ as long as $c^i < \overline{\tau}^i$. Similar to the previous case, by doing so we are also sure that there exists (l_1, v_1), and (l_2, v_2) such that $q \xrightarrow{f(q)=\lambda}_{S_{\mathsf{TGA}}} (l_1, v_1) \xrightarrow{e_0}_{S_{\mathsf{TGA}}} \ldots \xrightarrow{e_n} (l_2, v_2)$ with $l_1^i = l_2^i = Presam^i$, $(v_1 \models (c_1^i < \overline{\tau}^i))$, and $v_2 \models (c_2^i = \overline{\tau}^i)$.

In this manner, and upon assigning the proper actions in the previous run to the strategy f' we successfully handle the exceptions illustrated above with f' and thus we conclude that f' is a winning strategy for the safety game (TGA,$\overline{L}_u$) and also by construction for (TGA',$\overline{L}_u$).

References

[AAM+06] Yasmina Abdeddaı, Eugene Asarin, Oded Maler, et al. Scheduling with timed automata. *Theoretical Computer Science*, 354(2):272–300, 2006.

[ABRW91] Neil C Audsley, Alan Burns, Mike F Richardson, and Andy J Wellings. Real-time scheduling: the deadline-monotonic approach. In *in Proc. IEEE Workshop on Real-Time Operating Systems and Software*. Citeseer, 1991.

[AD94] Rajeev Alur and David L Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[AdNLR17] Björn Andersson, Mark Klein de Niz, John Lehoczky, and Ragunathan (Raj) Rajkumar. Real-time scheduling for cyber-physical systems. *Cyber-Physical Systems*, 2017.

[AJPR14] Amir Ali Ahmadi, Raphaël M Jungers, Pablo A Parrilo, and Mardavij Roozbehani. Joint spectral radius and path-complete graph lyapunov functions. *SIAM Journal on Control and Optimization*, 52(1):687–717, 2014.

[AKGD15] Mohammad Al Khatib, Antoine Girard, and Thao Dang. Stability verification of nearly periodic impulsive linear systems using reachability analysis. In *IFAC Conference on Analysis and Design of Hybrid Systems*, pages 358–363, 2015.

[AKGD16a] Mohammad Al Khatib, Antoine Girard, and Thao Dang. Stability verification and timing contract synthesis for linear impulsive systems using reachability analysis. *Nonlinear Analysis: Hybrid Systems*, 2016.

[AKGD16b] Mohammad Al Khatib, Antoine Girard, and Thao Dang. Verification and synthesis of timing contracts for embedded controllers. In *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*, pages 115–124. ACM, 2016.

[AKGD17a] Mohammad Al Khatib, Antoine Girard, and Thao Dang. Scheduling of embedded controllers under timing contracts. In *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control*, pages 131–140. ACM, 2017.

[AKGD17b] Mohammad Al Khatib, Antoine Girard, and Thao Dang. Self-triggered control for sampled-data systems using reachability analysis. In *IFAC World Congress*, 2017.

[AL14a] Nikolaos Athanasopoulos and Mircea Lazar. Alternative stability conditions for switched discrete time linear systems. In *IFAC World Congress*, pages 6007–6012, 2014.

[AL14b] Nikolaos Athanasopoulos and Mircea Lazar. On controlled-invariance and stabilization of time-delay systems. In *Control Conference (ECC), 2014 European*, pages 778–783. IEEE, 2014.

[And08] Björn Andersson. Global static-priority preemptive multiprocessor scheduling with utilization bound 38%. In *International Conference on Principles of Distributed Systems*, pages 73–88. Springer, 2008.

[AP08] Amir Ali Ahmadi and Pablo A Parrilo. Non-monotonic lyapunov functions for stability of discrete time nonlinear and switched systems. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 614–621. IEEE, 2008.

[ASB10] Matthias Althoff, Olaf Stursberg, and Martin Buss. Computing reachable sets of hybrid systems using a combination of zonotopes and polytopes. *Nonlinear Analysis: Hybrid Systems*, 4(2):233–249, 2010.

[AT06] Björn Andersson and Eduardo Tovar. Multiprocessor scheduling with few preemptions. In *Embedded and Real-Time Computing Systems and Applications, 2006. Proceedings. 12th IEEE International Conference on*, pages 322–334. IEEE, 2006.

[BCD+07] Gerd Behrmann, Agnes Cougnard, Alexandre David, Emmanuel Fleury, Kim G Larsen, and Didier Lime. UPPAAL-TIGA: Time for playing games! In *International Conference on Computer Aided Verification*, pages 121–125. Springer, 2007.

[Bla91] Franco Blanchini. Ultimate boundedness control for uncertain discrete-time systems via set-induced lyapunov functions. In *Conference on Decision and Control*, pages 1755–1760, 1991.

[Bla99] Franco Blanchini. Survey paper: Set invariance in control. *Automatica*, 35(11):1747–1767, 1999.

[BLR05] Gerd Behrmann, Kim G Larsen, and Jacob I Rasmussen. Optimal scheduling using priced timed automata. *ACM SIGMETRICS Performance Evaluation Review*, 32(4):34–40, 2005.

[BM07] Franco Blanchini and Stefano Miani. *Set-theoretic methods in control*. Springer, 2007.

[BMH12] Nicolas William Bauer, Paul J H Maas, and W P M H Heemels. Stability analysis of networked control systems: A sum of squares approach. *Automatica*, 48(8):1514–1524, 2012.

[Bri13] Corentin Briat. Convex conditions for robust stability analysis and stabilization of linear aperiodic impulsive and sampled-data systems under dwell-time constraints. *Automatica*, 49(11):3449–3457, 2013.

[BS12] Corentin Briat and Alexandre Seuret. Convex dwell-time characterizations for uncertain linear impulsive systems. *IEEE Transactions on Automatic Control*, 57(12):3241–3246, 2012.

[BT00] Oleg Botchkarev and Stavros Tripakis. Verification of hybrid systems with linear differential inclusions using ellipsoidal approximations. In *Hybrid Systems: Computation and Control*, pages 73–88. Springer, 2000.

[BvLD+12] N. W. Bauer, S. J. L. M. van Loon, M. C. F. Donkers, N van de Wouw, and W. P. M. H. Heemels. Networked control systems toolbox: Robust stability analysis made easy. In *IFAC Workshop on Distributed Estimation and Control in Networked Systems*, pages 55–60, 2012.

[CDF+05] Franck Cassez, Alexandre David, Emmanuel Fleury, Kim G Larsen, and Didier Lime. Efficient on-the-fly algorithms for the analysis of timed games. In *CONCUR 2005-Concurrency Theory*, pages 66–80. Springer-Verlag, 2005.

[CHVDW+10] Marieke B G Cloosterman, Laurentiu Hetel, Nathan Van De Wouw, W P M H Heemels, Jamal Daafouz, and Henk Nijmeijer. Controller synthesis for networked control systems. *Automatica*, 46(10):1584–1594, 2010.

[CTN07] Daniele Carnevale, Andrew R Teel, and Dragan Nesic. A lyapunov proof of an improved maximum allowable transfer interval for networked control systems. *IEEE Transactions on Automatic Control*, 52(5):892–897, 2007.

[DB01] Jamal Daafouz and Jacques Bernussou. Parameter dependent lyapunov functions for discrete time systems with time varying parametric uncertainties. *Systems & control letters*, 43(5):355–359, 2001.

[DB11] Robert I Davis and Alan Burns. A survey of hard real-time scheduling for multiprocessor systems. *ACM computing surveys (CSUR)*, 43(4):35, 2011.

[DH12] MCF Donkers and WPMH Heemels. Output-based event-triggered control with guaranteed-gain and improved and decentralized event-triggering. *Automatic Control, IEEE Transactions on*, 57(6):1362–1376, 2012.

[DHVDWH11] M C F Donkers, W P M H Heemels, Nathan Van De Wouw, and Laurentiu Hetel. Stability analysis of networked control systems using a switched linear systems approach. *IEEE Transactions on Automatic Control*, 56(9):2101–2115, 2011.

[DILS09] Alexandre David, Jacob Illum, Kim G Larsen, and Arne Skou. Model-based framework for schedulability analysis using uppaal 4.1. *Model-based design for embedded systems*, 1(1):93–119, 2009.

[DLG10] Shi-Lu Dai, Hai Lin, and Shuzhi Sam Ge. Scheduling-and-control codesign for a collection of networked control systems with uncertain delays. *IEEE Transactions on Control Systems Technology*, 18(1):66–78, 2010.

[DLTT13] Patricia Derler, Edward A Lee, Stavros Tripakis, and Martin Törngren. Cyber-physical system design contracts. In *International Conference on Cyber-Physical Systems*, pages 109–118, 2013.

[Feh99] Ansgar Fehnker. *Scheduling a steel plant with timed automata*. Computing Science Institute Nijmegen, Faculty of Mathematics and Informatics, Catholic University of Nijmegen, 1999.

[FGNZ14] Fulvio Forni, Sergio Galeani, Dragan Nešić, and Luca Zaccarian. Event-triggered transmission for linear control over communication channels. *Automatica*, 50(2):490–498, 2014.

[FHPR12] Christophe Fiter, Laurentiu Hetel, Wilfrid Perruquetti, and Jean-Pierre Richard. A state dependent sampling for linear state feedback. *Automatica*, 48(8):1860–1867, 2012.

[FLGD+11] Goran Frehse, Colas Le Guernic, Alexandre Donzé, Scott Cotton, Rajarshi Ray, Olivier Lebeltel, Rodolfo Ripado, Antoine Girard, Thao Dang, and Oded Maler. SpaceEx: Scalable verification of hybrid systems. In *Computer Aided Verification*, pages 379–395. Springer, 2011.

[FM14] Mirko Fiacchini and Irinel-Constantin Morarescu. Set theory conditions for stability of linear impulsive systems. In *Conference on Decision and Control*, 2014.

[FM16] M. Fiacchini and I.-C. Morărescu. Constructive necessary and sufficient condition for the stability of quasi-periodic linear impulsive systems. *IEEE Transactions on Automatic Control*, 2016.

[Fri10] Emilia Fridman. A refined input delay approach to sampled-data control. *Automatica*, 46(2):421–427, 2010.

[FSR04] Emilia Fridman, Alexandre Seuret, and Jean-Pierre Richard. Robust sampled-data stabilization of linear systems: an input delay approach. *Automatica*, 40(8):1441–1446, 2004.

[Fuj09] Hisaya Fujioka. Stability analysis of systems with aperiodic sample-and-hold devices. *Automatica*, 45(3):771–775, 2009.

[GCK03] Keqin Gu, Jie Chen, and Vladimir L Kharitonov. *Stability of time-delay systems*. Springer Science & Business Media, 2003.

[Gir05] Antoine Girard. Reachability of uncertain linear systems using zonotopes. In *Hybrid Systems: Computation and Control*, pages 291–305. Springer, 2005.

[GMCL10] Huijun Gao, Xiangyu Meng, Tongwen Chen, and James Lam. Stabilization of networked control systems via dynamic output-feedback controllers. *SIAM Journal on Control and Optimization*, 48(5):3643–3658, 2010.

[GST09] Rafal Goebel, Ricardo G Sanfelice, and Andrew R Teel. Hybrid dynamical systems. *IEEE Control Systems*, 29(2):28–93, 2009.

[GST12] Rafal Goebel, Ricardo G Sanfelice, and Andrew R Teel. *Hybrid Dynamical Systems: modeling, stability, and robustness*. Princeton University Press, 2012.

[HB10] Tingshu Hu and Franco Blanchini. Non-conservative matrix inequality conditions for stability/stabilizability of linear differential inclusions. *Automatica*, 46(1):190–196, 2010.

[HDI06] Laurentiu Hetel, Jamal Daafouz, and Claude Iung. Stabilization of arbitrary switched linear systems with unknown time-varying delays. *IEEE Transactions on Automatic Control*, 51(10):1668–1674, 2006.

[HDTP13] Laurentiu Hetel, Jamal Daafouz, Sophie Tarbouriech, and Christophe Prieur. Stabilization of linear impulsive systems through a nearly-periodic reset. *Nonlinear Analysis: Hybrid Systems*, 7(1):4–15, 2013.

[HFO+17] Laurentiu Hetel, Christophe Fiter, Hassan Omran, Alexandre Seuret, Emilia Fridman, Jean-Pierre Richard, and Silviu Iulian Niculescu. Recent developments on the stability of systems with aperiodic sampling: an overview. *Automatica*, 76:309–335, 2017.

[HKJM13] M. Herceg, M. Kvasnica, C.N. Jones, and M. Morari. Multi-Parametric Toolbox 3.0. In *European Control Conference*, pages 502–510, July 17–19 2013.

[HKPR11] Laurentiu Hetel, Alexandre Kruszewski, Wilfrid Perruquetti, and Jean-Pierre Richard. Discrete and intersample analysis of systems with aperiodic sampling. *IEEE Transactions on Automatic Control*, 56(7):1696–1701, 2011.

[HLCS03] Li-Sheng Hu, James Lam, Yong-Yan Cao, and Hui-He Shao. A linear matrix inequality (lmi) approach to robust h/sub 2/sampled-data control for linear uncertain systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 33(1):149–155, 2003.

[HTVdWN10] W P Maurice H Heemels, Andrew R Teel, Nathan Van de Wouw, and Dragan Nešić. Networked control systems with communication constraints: Tradeoffs between transmission intervals, delays and performance. *IEEE Transactions on Automatic Control*, 55(8):1781–1796, 2010.

[HVDWG+10] W PMH Heemels, Nathan Van De Wouw, Rob H Gielen, MCF Donkers, Laurentiu Hetel, Sorin Olaru, Mircea Lazar, Jamal Daafouz, and Silviu Niculescu. Comparison of overapproximation methods for stability analysis of networked control systems. In *International Conference on Hybrid systems: computation and control*, pages 181–190, 2010.

[JP86] Mathai Joseph and Paritosh Pandya. Finding response times in a real-time system. *The Computer Journal*, 29(5):390–395, 1986.

[KAS16] Eric S Kim, Murat Arcak, and Sanjit A Seshia. Directed specifications and assumption mining for monotone dynamical systems. In *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*, pages 21–30. ACM, 2016.

[KK84] E Kamen and P Khargonekar. On the control of linear systems whose coefficients are functions of parameters. *IEEE Transactions on Automatic Control*, 29(1):25–33, 1984.

[KLR12] Junsung Kim, Karthik Lakshmanan, and Ragunathan Raj Rajkumar. Rhythmic tasks: A new task model with continually varying periods for cyber-physical systems. In *Proceedings of the 2012 IEEE/ACM Third International Conference on Cyber-Physical Systems*, pages 55–64. IEEE Computer Society, 2012.

[Kra63] Nikolaj N Krasovskij. *Stability of motion: applications of Lyapunov's second method to differential systems and equations with delay*. Stanford university press, 1963.

[KV00] Alexander B Kurzhanski and Pravin Varaiya. Ellipsoidal techniques for reachability analysis: internal approximation. *Systems & Control Letters*, 41(3):201–211, 2000.

[KW14] Chung-Yao Kao and Dian-Rong Wu. On robust stability of aperiodic sampled-data systems-an integral quadratic constraint approach. In *American Control Conference*, pages 4871–4876. IEEE, 2014.

[LA09] H. Lin and P.J. Antsaklis. Stability and stabilizability of switched linear systems: a survey of recent results. *IEEE Transactions on Automatic Control*, 54(2):308–322, 2009.

[LF12] Kun Liu and Emilia Fridman. Wirtinger's inequality and lyapunov-based sampled-data stabilization. *Automatica*, 48(1):102–108, 2012.

[LFH15] Kun Liu, Emilia Fridman, and Laurentiu Hetel. Networked control systems in the presence of scheduling protocols and communication delays. *SIAM Journal on Control and Optimization*, 53(4):1768–1788, 2015.

[LG09] Colas Le Guernic. *Reachability analysis of hybrid systems with linear continuous dynamics*. PhD thesis, Université Joseph-Fourier-Grenoble I, 2009.

[LGG10] Colas Le Guernic and Antoine Girard. Reachability analysis of linear systems using support functions. *Nonlinear Analysis: Hybrid Systems*, 4(2):250–262, 2010.

[Lio66] ML Liou. A novel method of evaluating transient response. *Proceedings of the IEEE*, 54(1):20–23, 1966.

[LL73] Chung Laung Liu and James W Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the ACM (JACM)*, 20(1):46–61, 1973.

[LLGCM10] Julien Legriel, Colas Le Guernic, Scott Cotton, and Oded Maler. Approximating the pareto front of multi-criteria optimization problems. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 69–83. Springer, 2010.

[LPJ+14] Tae H Lee, Ju H Park, Ho Youl Jung, OM Kwon, and SM Lee. Improved results on stability of time-delay systems using wirtinger-based inequality. *IFAC Proceedings Volumes*, 47(3):6826–6830, 2014.

[LSF10] Kun Liu, Vladimir Suplin, and Emilia Fridman. Stability of linear systems with general sawtooth delay. *IMA Journal of Mathematical Control and Information*, 27(4):419–436, 2010.

[MAT09] Manuel Mazo, Adolfo Anta, and Paulo Tabuada. On self-triggered control for linear systems: Guarantees and complexity. In *Control Conference (ECC), 2009 European*, pages 3767–3772. IEEE, 2009.

[Mir07] Leonid Mirkin. Some remarks on the use of time-varying delay to model sample-and-hold circuits. *IEEE Transactions on Automatic Control*, 52(6):1109–1112, 2007.

[MMD13] Frederic Mazenc, Michael Malisoff, and Thach N Dinh. Robustness of nonlinear systems with respect to delay and sampling of the controls. *Automatica*, 49(6):1925–1931, 2013.

[MPS95] Oded Maler, Amir Pnueli, and Joseph Sifakis. On the synthesis of discrete controllers for timed systems. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 229–242. Springer, 1995.

[MR97] Alexandre Megretski and Anders Rantzer. System analysis via integral quadratic constraints. *IEEE Transactions on Automatic Control*, 42(6):819–830, 1997.

[NHT08] Payam Naghshtabrizi, Joao P Hespanha, and Andrew R Teel. Exponential stability of impulsive systems with application to uncertain sampled-data systems. *Systems & Control Letters*, 57(5):378–385, 2008.

[NT04] Dragan Nesic and Andrew R Teel. Input-output stability properties of networked control systems. *IEEE Transactions on Automatic Control*, 49(10):1650–1667, 2004.

[NTC09] Dragan Nesic, Andrew R Teel, and Daniele Carnevale. Explicit computation of the sampling period in emulation of controllers for nonlinear sampled-data systems. *IEEE Transactions on Automatic Control*, 54(3):619–624, 2009.

[PHYT11] Chen Peng, Qing-Long Han, Dong Yue, and Engang Tian. Sampled-data robust h∞ control for t–s fuzzy systems with time delay and uncertainties. *Fuzzy Sets and Systems*, 179(1):20–33, 2011.

[PKJ11] PooGyeon Park, Jeong Wan Ko, and Changki Jeong. Reciprocally convex approach to stability of systems with time-varying delays. *Automatica*, 47(1):235–238, 2011.

[PLMS11] Miroslav Pajic, Insup Lee, Rahul Mangharam, and Oleg Sokolsky. Upp2sf: Translating uppaal models to simulink. *University of Pennsylvania, Tech. Rep*, 2011.

[PPP02] Stephen Prajna, Antonis Papachristodoulou, and Pablo A Parrilo. Introducing sostools: A general purpose sum of squares programming solver. In *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, volume 1, pages 741–746. IEEE, 2002.

[Raz56] BS Razumikhin. On the stability of systems with a delay. *Prikl. Mat. Mekh*, 20(4):500–512, 1956.

[RLM+11] Paul Regnier, George Lima, Ernesto Massa, Greg Levin, and Scott Brandt. Run: Optimal multiprocessor real-time scheduling via reduction to uniprocessor. In *Real-Time Systems Symposium (RTSS), 2011 IEEE 32nd*, pages 104–115. IEEE, 2011.

[SA02] Anand Srinivasan and James H Anderson. Optimal rate-based scheduling on multiprocessors. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 189–198. ACM, 2002.

[Seu12] Alexandre Seuret. A novel stability analysis of linear systems under asynchronous samplings. *Automatica*, 48(1):177–182, 2012.

[SG13] Alexandre Seuret and Frédéric Gouaisbaut. Wirtinger-based integral inequality: application to time-delay systems. *Automatica*, 49(9):2860–2866, 2013.

[SG14] Alexandre Seuret and Frédéric Gouaisbaut. Complete quadratic lyapunov functionals using bessel-legendre inequality. In *Control Conference (ECC), 2014 European*, pages 448–453. IEEE, 2014.

[SLCR10] Jian Sun, GP Liu, Jie Chen, and David Rees. Improved delay-range-dependent stability criteria for linear systems with time-varying delays. *Automatica*, 46(2):466–470, 2010.

[SP13] Alexandre Seuret and M Peet. Stability analysis of sampled-data systems using sum of squares. *IEEE Transactions on Automatic Control*, 58(6):1620–1625, 2013.

[Tab07] Paulo Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *Automatic Control, IEEE Transactions on*, 52(9):1680–1685, 2007.

[Ten14] Pranav Tendulkar. *Mapping and Scheduling on Multi-core Processors using SMT Solvers.* PhD thesis, Universite de Grenoble I-Joseph Fourier, 2014.

[YMH98] Hui Ye, Anthony N Michel, and Ling Hou. Stability theory for hybrid dynamical systems. *IEEE transactions on automatic control*, 43(4):461–474, 1998.

[ZDG+96] Kemin Zhou, John Comstock Doyle, Keith Glover, et al. *Robust and optimal control*, volume 40. Prentice hall New Jersey, 1996.