

ANNEXS

A.1 Avantage du langage VHDL-AMS

Le langage VHDL-AMS présente pour le concepteur les avantages suivants :

A.1.1 Normalisation

Le langage VHDL-AMS est normalisé, donc il est non propriétaire et indépendant des fournisseurs de logiciels choisis. Il permet donc l'échange de données/modèles de manière simple entre les différents acteurs d'un projet.

A.1.2 Multidisciplinarité

L'approche multi-domaines native du langage VHDL-AMS permet une communication aisée entre les différents domaines scientifiques : un électronicien, un mécanicien ou même un chimiste peuvent dès lors modéliser la partie d'un dispositif qui le concerne directement sans problème de dialogue avec les autres parties.

A.1.3 Multi-niveaux (multi-abstractions)

VHDL-AMS propose des mécanismes permettant de gérer aussi bien les abstractions comportementales (fonction réalisée par le système qui est modélisé et non sa physique), que les abstractions structurelles (le système est divisé en sous-ensembles qui peuvent eux-mêmes être modélisés au moyen de différentes abstractions, ...) ou bien de type Work-Flow (enchaînement de blocs fonctionnels dont les entrées n'ont pas d'influence sur les sorties des blocs précédents). Les modèles créés avec VHDL-AMS peuvent donc aussi bien être descriptifs que prédictifs.

A.1.4 Simulation mixte

Le langage VHDL-AMS autorise de manière conjointe aussi bien les modélisations à temps continu (analogiques) qu'à événements discrets (logiques), ou mélangeant les deux. VHDL-AMS prévoit en outre la gestion des discontinuités et des nouvelles conditions initiales ainsi qu'une synchronisation explicite des noyaux temps continu et temps discret.

A.1.5 Traitement des équations implicites

Les équations implicites sont des équations pour lesquelles l'inconnue ne se trouve pas forcément dans le membre de gauche. Dans ce cas les quantités sont liées par une équation qui représente une contrainte à respecter au cours de la simulation et pas une affectation (comme dans Matlab par exemple). L'écriture des équations devient donc naturelle pour le concepteur (langage de technologie).

A.2 Comparaison entre VHDL-AMS et d'autres langages de simulation

Les langages multi domaines permettent de traiter indifféremment des modélisations analogiques, logiques ou mixtes. Ainsi, ces langages de simulation permettent de concevoir des fonctions pluridisciplinaires telles que mécanique, optique, chimique, biochimique, fluide. Des langages ont été créés dans ce sens, tels le VHD-LAMS. D'autres langages, comme MAST (issu de logiciel SABER), MODELICA permettent de décrire des fonctions mixtes et MATLAB qui permet de faire des calculs numérique. La modélisation des grandeurs optiques en VHDL-AMS est possible avec des résultats convergents. Les grandeurs optiques sont régulièrement beaucoup plus grandes qu'on trouve habituellement dans le domaine électronique par exemple la densité de porteurs et des photons.

La spécification multi-domaine n'existe pas dans le langage MAST, qu'est un langage propriétaire (non normalisé) développé en 1984 et complètement lié au simulateur SABER [A.1-A.2]. Ce langage propose des mécanismes de modélisation non adaptés pour la modélisation de système numérique. La modélisation VHDL-AMS, met en jeu le couplage de deux simulateurs distincts dédiés respectivement à la partie discrète et à la partie continue, qui vont se synchroniser à certains instants T (dépassement de seuil, évènements). Par contre le langage ouvert Modelica, Malgré qu'il permet de modéliser des systèmes complexes avec des modèles mathématiquement décrits, ce langage considère à tout instant l'ensemble des équations différentielles, algébriques et discrètes comme un seul et unique système d'équations c'est-à-dire qu'il n'y a pas une prise en charge adaptée du domaine discret comme le fait le VHDL-AMS. Modelica est un langage de modélisation orienté objet, développé par l'association Modelica en 1997 [A.3].

Les connexions VHDL-AMS sont soit de type conservatif (échange bidirectionnel de l'énergie), ou de type flot de signal. Par contre, le logiciel de calcul numérique MATLAB ne permet pas que l'utilisation des connexions de type flot de signal [A.4], dans ce cas les sorties ne sont pas influencées par les entrées (impédance de sortie nulle et d'entrée infinie). Matlab permet une manipulation simple des matrices, tracer des fonctions et des données, implémenter des algorithmes, et de s'interfacer facilement avec d'autres programmes. Dans [A.4], les chercheurs constatent que le VHDL-AMS, avec son type de connexion conservatif et son écriture des équations physiques, permet une modélisation plus aisée et plus directe nécessaires pour fabriquer des composants entièrement réutilisables. Les études effectuées par [A.4,A.5-A.6,A.7-A.13] présentent les principales caractéristiques de chaque langage. Nous résumons dans le tableau A.1, l'étude comparative entre les quatre langages. L'étude des

caractéristiques principales pour les différents langages de modélisation citée dans tableau A.1, permet d'apprécier que le langage VHDL-AMS (multi-domaine, multi-abstraction, multi-architecture) est bien adapté pour notre application. Ce langage de haut niveau permet d'écrire un cahier des charges abstrait et simulable. Une architecture de système peut être faite de manière fonctionnelle en associant des modèles existants ou à créer.

Tableau A.1 : étude comparative entre MTLAB, Modelica, MAST et VHDL-AMS

Languages	MATLAB	Modelica	MAST	VHDL-AMS
Characteristic				
Standard	No	No	No	IEEE 1076.1-2007
Language Presentation	digital calculation + Simulink	object-oriented modeling	Own language linked to SABER tool	Extension of VHDL (IEEE 1076)
Modularity	mono-architecture	mono- architecture	mono-architecture	multi-architecture
Algebraic and differential equations	Yes implicit forms order 1	Yes implicit forms order 1	Yes explicit forms order 1	Yes Explicit and implicit forms order n
Multi-discipline	No	Yes	Yes	Yes
Conservative/signal flow	No / yes	Yes / yes	Yes / yes	Yes / yes
Continuous time modeling	Non-conservative	Physical / block- oriented	Conservative/ non-conservative	Conservative / non- conservative
Discrete time modeling	-	event-driven, but no event queue supported	Event-driven	event-driven
Multi-abstraction	No	No	Yes	Yes
Discontinuity management	No	Yes	Yes Control Section	Yes Instruction 'Break'
Dependence on a tool	Yes	No	Yes	Yes
Library management	No	Yes	No	Yes
Hierarchical structure	No	Yes	Yes	Yes
Expressive semantic	No	No	No	Yes
Compatibility with tools Spice	No	No	No	Yes

A.3 Comparaison des simulateurs VHDL-AMS [A.14]

Le tableau A.2 compare les plus importants simulateurs disponibles actuellement sur le marché pouvant supporter le langage VHDL-AMS.

Tableau A.2 : Comparaison des simulateurs VHDL-AMS.

	ADVance-MS	SystemVision	Saber HDL	SMASH	Simplorer
Editeur	Mentor Graphics	Mentor Graphics	Synopsys (Avant-Analysis)	Dolphin Integration	ANSOFT
Plate-forme	<u>UNIX</u> Linux	<u>PC</u> Windows 2000 et XP	<u>PC & UNIX</u> -Sun Solaris 2.6.8 -Windows NT4.0/2000 -Redhat Linux7.2	<u>PC & UNIX</u> -Sun Solaris 7 -Linux RHL 7 to RHEL3 -Windows XP/vista	<u>PC</u> Windows 2000 et XP
Simulateur	Eldo	Eldo	Newton Calaveras	Single kernel	Simec
Interface graphique	NON	OUI	OUI	OUI	OUI
Bibliothèque VHDL-AMS	Eléments de base	Eléments de base	Eléments de base	NON	Eléments de base
Bibliothèques composants	Bibliothèques des modèles C/C++	Bibliothèques des modèles Spice	Bibliothèques des modèles MAST	Support des Modèles Spice	Bibliothèques des modèles SMI.

A.4 Réponse sinusoïdale des puissances optiques

Pour confirmer nos résultats, les Figures A.1, A.2 et A.3 représentent la réponse sinusoïdale des puissances optique à travers respectivement d'une FOGI, d'une FOSI et d'une FOMM. Nous constatons qu'on a le même profil et les mêmes remarques qu'avec une source en créneaux.

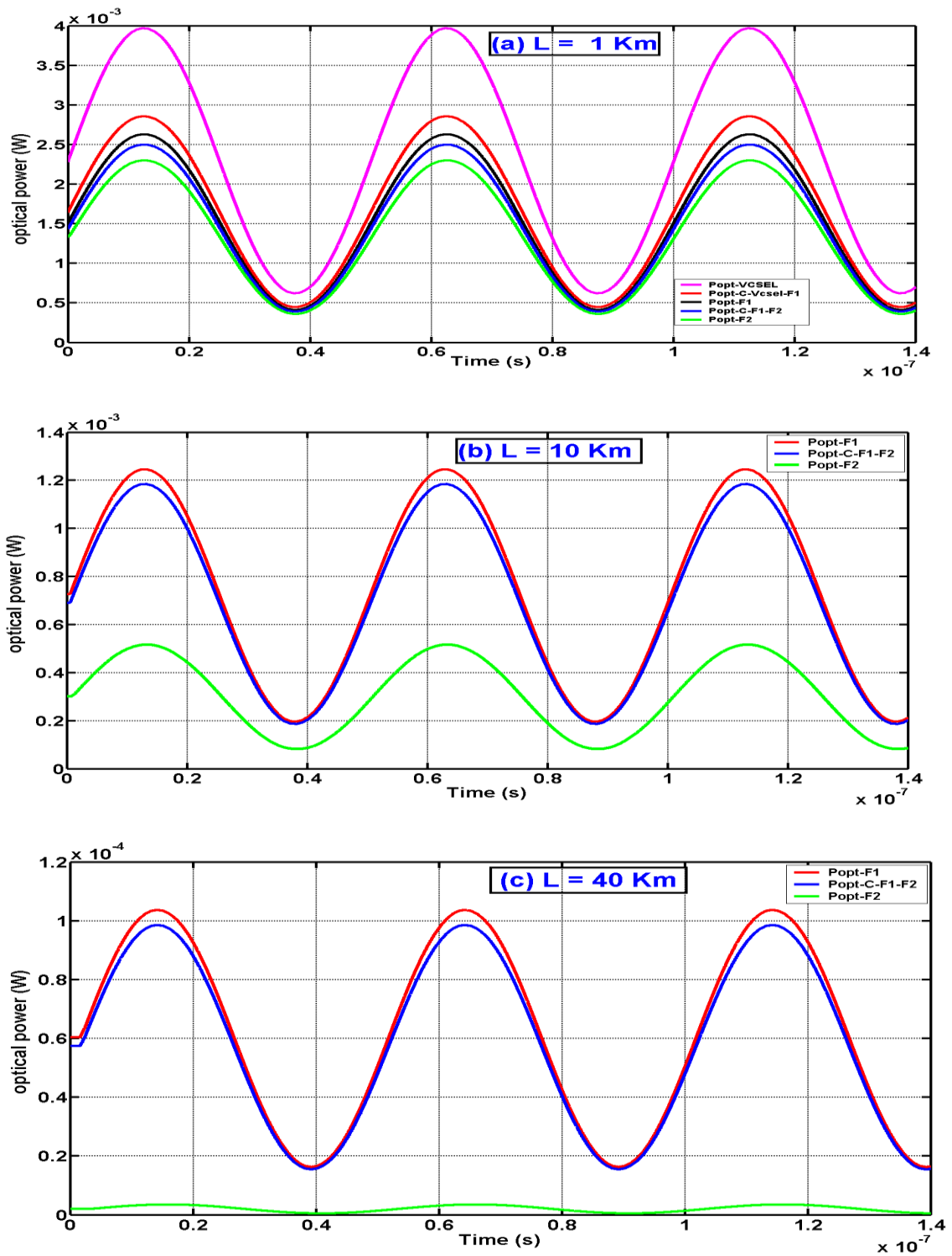


Figure A.1 Réponse sinusoïdale des puissances optique à travers une FOGI pour plusieurs longueurs : (a) $L = 1 \text{ Km}$; (b) $L = 10 \text{ Km}$; (c) $L = 40 \text{ Km}$.

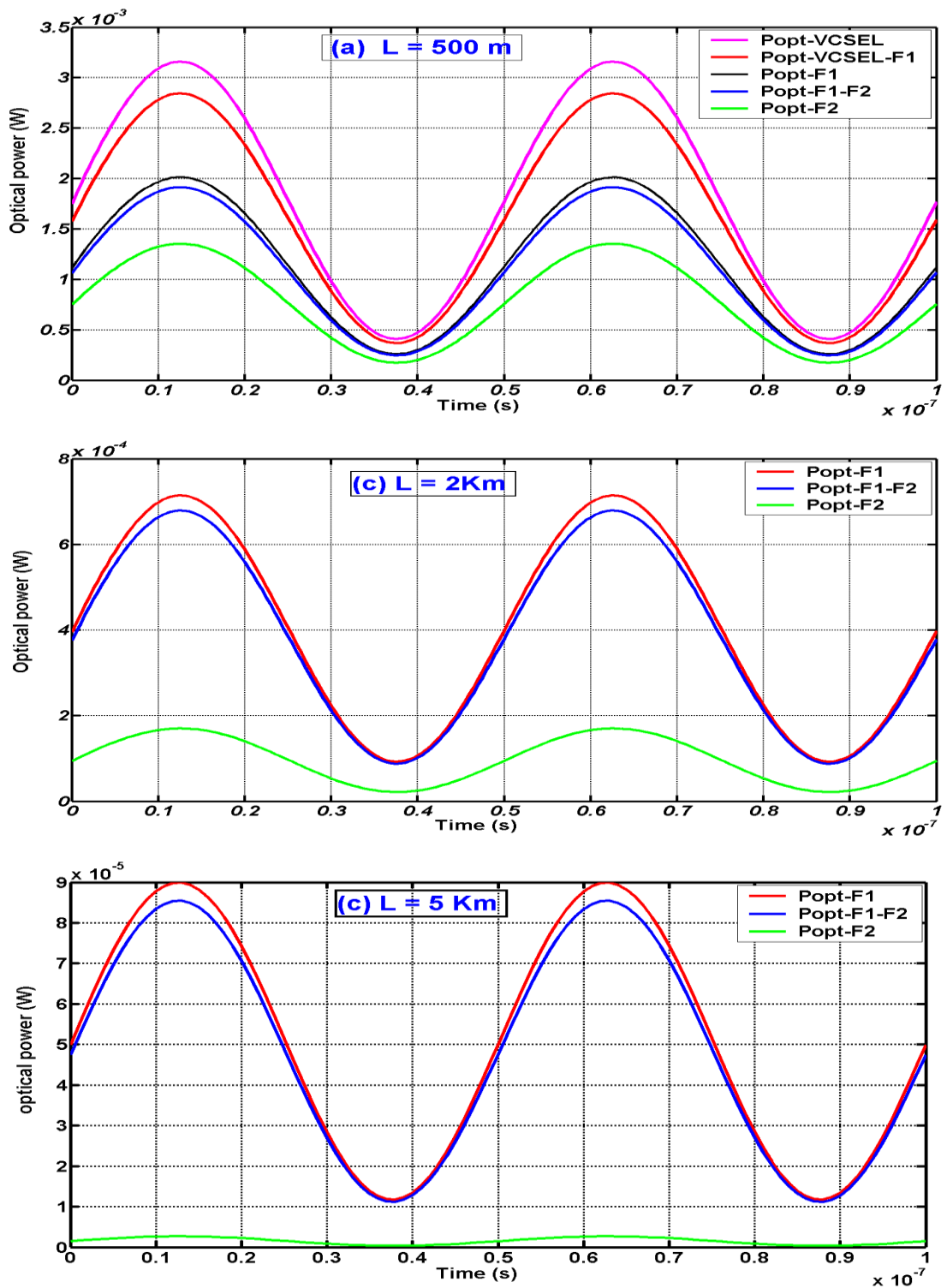


Figure A.2 Réponse sinusoïdale des puissances optique à travers une FOSI pour plusieurs longueurs : (a) $L = 500 \text{ m}$; (b) $L = 2 \text{ Km}$; (c) $L = 5 \text{ Km}$.

A.5 Références bibliographiques

- [A.1] R. Penaeguiluz, « Commande algorithmique d'un système mono-onduleur bimachine asynchrone destiné à la traction ferroviaire ». Ph.D. thesis; University of Toulouse; Order n° 1912, 2002.
- [A.2] Saber, MAST Language, Book 2. User Guide Version W-2004.12; DON: 00000-000 WA, 2004.
- [A.3] F. Broenink J, "Object-oriented modeling with bond graphs and Modelica". ICBGM'99 part of WMC'99 San Francisco; 2-6, 1999.
- [A.4] S. Guessab, J. Oudinot, "How can MATLAB/Simulink co-exist with VHDLAMS". Euro-Design Conference 2004.
- [a.5] Y. Hervé, « VHDL-AMS application et enjeux industriels. Dunod éditeur Paris 2002 »; ISBN 2100058886, 2002.
- [A.6] Standards Board. "IEEE Standard VHDL-AMS analog and mixed-signal extensions". IEEE SA 2007; ISBN 0-7381-5627-2 SH 95694, 2007.
- [A.7] C. Clauss, H. Elmqvist, S-E. Mattsson, M. Otter, P. Schwarz, "Mixed Domain Modeling in Modelica; Forum on Specification and Design Languages". FDL'02 2002 Marseille; 12 pp, 2002.
- [A.8] O. Enge-Rosenblatt, J. Haase, C. Clauß, "Important Characteristics of VHDLAMS and modelica with Respect to Model Exchange". 1st International Workshop on Equation-Based Object-Oriented Languages and Tools (EOOLT07) Berlin-Germany; 89-98, 2007.
- [A.9] F. Pêcheux, C. Lallement, A. Vachoux, « VHDL-AMS and Verilog-AMS as Alternative Hardware Description Languages for Efficient Modeling of Multidiscipline Systems". IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems: 24(2): 204-225, 2005.
- [A.10] C-K. Chuang, C-G. Harrison, "Analogue behavioural modelling and simulation using VHDL-AMS and Saber-MAST". IEEE Transaction on Power Electronic London 1994.
- [A.11] F. Pêcheux, B. Allard, C. Lallement, A. Vachoux, H. Morel, "Modeling and Simulation of Multi-Discipline systems Using Bond Graphs and VHDL-AMS". Proc. of International Conference Bond Graph and Modeling; 149-155, 2005.
- [A.12] B. Allard, H. Morel, « Utilisation des graphes de liens en électronique de puissance ». Techniques de l'Ingénieur 2005 Paris; D3064 : 23 pp, 2005.
- [A.13] O. Rolland, Y. Herve, "Complex systems mastering with VHDL-AMS and systems'Vip". copyright systems'Vip 2007.
- [A.14] I. Their, « contribution au développement de modèles pour l'électronique de puissance en VHDL-AMS », thèse de doctorat de l'Ecole doctorale : Electronique, Electrotechnique, Automatique (E.E.A), Lyon 2009.