République Algérienne démocratique et populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université Dr Yahia Farès de Médéa

Faculté des sciences et de la technologie
Département du génie électrique et de l'informatique

MEMOIRE DE MAGISTER

Spécialité : Electronique
Option : Instrumentation et micro-électronique

Présenté par : Djalila ACHOURI

THEME

Conception et étude comparative entre une DPLL et une SPLL à base d'un DSP

Présenté devant le jury composé de :

Pr. Salam ABUDURA	Professeur, U. Médéa	Président
Pr. Abderrezak GUESSOUM	Professeur, U. Blida	Examinateur
Dr. Mohamed OULD ZMIRLI	MCA, U. Médéa	Examinateur
Dr. Noureddine BATEL	MCA, U. Médéa	Encadrant

Décembre 2015

REMERCIEMENTS

Mes remerciements vont tout d'abord à ALLAH qui a insufflé en moi la vie et m'a comblée de ses grâces.

Je tiens à remercier mon promoteur Mr. Noureddine BATEL Maitre de conférences à l'université Yahia Farès de Médéa pour m'avoir introduite dans ce domaine passionnant, pour m'avoir orientée et conseillée.

Mes remerciements les plus sincères vont aussi à Mr. Salam ABUDURA Professeur à l'université Yahia Farès de Médéa pour l'honneur qu'il me fait en présidant le jury.

Je tiens à remercier également Mr. Abderrezak GUESSOUM Professeur à l'université Saad Dahlab de Blida et Mr. Mohamed OULD ZMIRLI Maitre de conférences à l'université Yahia Farès de Médéa pour avoir eu la gentillesse d'accepter de faire partie du jury.

À titre plus personnel, je remercie toutes les personnes qui m'ont soutenue et qui ont fait beaucoup de sacrifices pour me voir arriver à ce stade, en particulier mes parents et mon mari.

RÉSUMÉS

ملخص:

دوائر قفل الطور هي دوائر رئيسية في أي فرع من فروع الإلكترونيك: الأجهزة والاتصالات والتحكم، الخ. المعرفة النظرية والعملية لهذه الدوائر تسمح بالتحكم في التطبيقات المستهدفة. في هذا العمل، نجري دراسة وتصميم وبناء نوعين من حلقات قفل الطور : حلقة قفل الطور شبه رقمية و حلقة قفل الطور مبرمجة قائمة على DSP.
مقارنة نظرية وعملية من حيث الأداء والدينامية ستجرى بين النوعين السابقين من الحلقات مقفلة الطور.

الكلمات المفتاحية: حلقات قفل الطور، حلقة قفل الطور شبه رقمية، حلقة قفل الطور مبرمجة.

Résumé :

Les circuits à verrouillage de phase sont des circuits clés dans toute branche d'électronique : instrumentation, communications, automatique, etc. Leur connaissance théorique et pratique permet de les maîtriser au niveau des applications cibles. Dans ce travail, nous effectuons une étude, une conception et une réalisation de deux types de boucles à verrouillage de phase ; une PLL semi-numérique et une PLL software à base d'un DSP.
Une comparaison théorique et pratique en termes de performances statiques et dynamiques est menée entre les deux types de PLL.

Mots-clés : Boucles à verrouillage de phase, PLL semi-numérique, PLL software, DSP.

Abstract :

Phase locked circuits are key circuits in any electronic branch : instrumentation, communications, control, etc. Their theoretical and practical knowledge allows to control in target applications. In this work, we carry out a study, design and construction of two phase locked loops types; an hybrid PLL and a software PLL based on a DSP.
A theoretical and practical comparison in terms of static and dynamic performance is made between the two types of PLL.

Key words : Phase locked loops, hybrid PLL, software PLL, DSP.

LISTE DES FIGURES

Figure I.1	Schéma-bloc d'une PLL.	4
Figure I.2	Récupération de porteuse.	9
Figure I.3	Récupération d'horloge.	9
Figure I.4	Filtre suiveur.	10
Figure I.5	Démodulation de fréquence.	10
Figure I.6	Démodulation de phase.	11
Figure I.7	Modulation de phase.	11
Figure I.8	Synthèse de fréquence.	12
Figure II.1	Schéma-bloc d'une DPLL.	14
Figure II.2	Caractéristiques des détecteurs de phase.	16
Figure II.3	Détecteur fréquence-phase (DFP).	17
Figure II.4	Diagramme d'état du DFP.	18
Figure II.5	Comparaison de phase du DFP pour un écart nul (a), écart positif (b), écart négatif (c).	19
Figure II.6	Caractéristiques de transconductance des MOSFETs complémentaires.	21
Figure II.7	Réponse de la pompe de charge à une fréquence d'entrée supérieure à celle du VCO.	21
Figure II.8	Réponse de la pompe de charge à une fréquence d'entrée inférieure à celle du VCO.	22
Figure II.9	Schéma de principe du VCO.	23
Figure II.10	Trigger de Schmitt à transistors bipolaires.	24
Figure II.11	Fenêtre d'hystérésis du Trigger bipolaire.	24
Figure II.12	Fenêtre d'hystérésis du Trigger de Schmitt intégré de type 7414.	25
Figure II.13	Modèle de la source de courant I_0 contrôlé par tension V_C.	25
Figure II.14	Miroir de courant simple.	26
Figure II.15	La source de courant contrôlée par tension.	27
Figure II.16	Circuit VCO.	27
Figure II.17	Courant dans la capacité C_1 (en haut) et tensions en sortie du Trigger et aux bornes de C_1 (en bas).	28
Figure II.18	Tensions de sortie et du collecteur de Q_9 (en haut), tensions de contrôle et son image (en bas).	29
Figure II.19	Courant collecteur du transistor Q_9 (en haut) et sa tension-crête V_{CE} (en bas).	30
Figure II.20	Caractéristique tension-fréquence du VCO.	31

Figure II.21	Courants de consigne et de charge (en haut), impédance de sortie de la source contrôlée (en bas).	32
Figure II.22	Source de courant contrôlée par tension, améliorée.	34
Figure II.23	Courants de consigne et de charge (en haut), impédance de sortie de la source contrôlée (en bas).	34
Figure II.24	VCO finalisé.	35
Figure II.25	Tension $V(R_2)$ image du courant contrôlé et tension V_{CE} du transistor Q_9.	36
Figure II.26	Courant collecteur du transistor Q_9 (en haut) et courant dans la capacité C_1 (en bas).	37
Figure II.27	Caractéristique tension-fréquence du VCO amélioré.	37
Figure II.28	Filtre passe-bas de boucle et son diagramme d'amplitude.	38
Figure II.29	Erreur de phase ou tension moyenne du DFP (au milieu) et tension de filtre ou fréquence du VCO (en bas) en réponse à un échelon de fréquence (en haut).	39
Figure II.30	Schéma-bloc fonctionnel de la DPLL	40
Figure II.31	Tension moyenne du DFP (en haut) et fréquence de VCO (en bas).	43
Figure II.32	Tension moyenne du DFP (en haut) et fréquence du VCO (en bas).	45
Figure II.33	Tensions d'entrées et de sortie du DFP avec tension équivalente et tension de filtre.	46
Figure II.34	Processus de capture lente avec détecteur de phase à porte XOR (a), à bascule JK (b), et à DFP présentant un temps de capture similaire mais pour un pas d'entrée plus grand (c).	50
Figure II.35	Signal U_1 d'information, signal U_{1b} à bande réduite par le canal, signal U_{1n} superposé de bruit, et signal U_{1r} remis en forme.	52
Figure II.36	DPLL munie d'un pré-filtre d'entrée.	53
Figure II.37	DPLL avec sa PSD d'entrée (a), sa réponse impulsionnelle (b), et sa PSD de sortie (c).	55
Figure II.38	Filtre de boucle.	58
Figure II.39	Schéma bloc de la DPLL simulée.	59
Figure II.40	Tracés de tension du détecteur de phase (rouge) et du filtre (vert), pour les fréquences d'entrée 100kHz (a), 200kHz (b), et 625kHz (c).	62
Figure II.41	Réponses du filtre de boucle pour trois différentes fréquences d'entrée	63
Figure II.42	Réponse du filtre à la fréquence d'entrée de 100kHz.	64

Figure II.43	Réponse du filtre à la fréquence d'entrée de 200kHz.	65
Figure II.44	Réponse du filtre à la fréquence d'entrée de 625kHz.	65
Figure II.45	Réponse du filtre à la fréquence d'entrée de 289kHz.	66
Figure II.46	Réponses réunies du filtre amélioré pour quatre différentes fréquences d'entrée.	67
Figure III.1	Schéma-bloc des opérations d'une SPLL opérant en tant qu'une DPLL.	73
Figure III.2	Tracé des signaux calculés par l'algorithme SPLL.	75
Figure III.3	Organigramme des opérations effectuées par l'algorithme SPLL.	77
Figure III.4	Organigramme de croisement du zéro-phase pour $\varphi_2(t)$.	79
Figure III.5	Calcul géométrique de t_{cross}.	79
Figure III.6	Organigramme du DFP.	80
Figure III.7	Vérification des cas de figures III.2 et III.8.	81
Figure III.8	Cas de tracés complémentaires de la Figure III.2.	82
Figure III.9	Organigramme du filtre digital.	84
Figure III.10	Signal d'entrée et sa forme décalée, appliqués aux pins INT5 et INT4.	88
Figure III.11	Mesure des durées de séparation entre niveaux du signal à 100kHz.	89
Figure III.12	Tensions de filtre à c^{stes} de temps modifiées en réponse aux quatre fréquences de test.	91
Figure III.13	Tension de filtre en réponse à 100kHz d'entrée.	92
Figure III.14	Tension de filtre en réponse à 200kHz d'entrée.	92
Figure III.15	Tension de filtre en réponse à 289kHz d'entrée.	93
Figure III.16	Tension de filtre en réponse à 625kHz d'entrée.	94
Figure III.17	Tensions de filtre à c^{stes} de temps optimales comme valeurs moyennes de celles à c^{stes} de temps modifiées.	94
Figure III.18	Tensions de filtre superposées de la DPLL et la SPLL en réponse aux quatre fréquences de test.	95

SOMMAIRE

RESUMES

REMERCIEMENTS

LISTE DES FIGURES

SOMMAIRE

INTRODUCTION GENERALE 1

CHAPITRE I : GENERALITES SUR LES PLLs

I.1 La boucle à verrouillage de phase 3

I.1.1 Principe de base du verrouillage de phase 3

I.1.2 Les blocs d'une PLL 4

I.1.2.1 L'oscillateur contrôlé en tension (VCO) 4

I.2.2.2 Le détecteur de phase (PD) 5

I.2.2.3 Le filtre de boucle 6

I.2 Classification des PLLs 6

I.3 Historique et applications des PLLs 7

I.4 Etat de l'art 12

CHAPITRE II : ETUDE ET CONCEPTION D'UNE DPLL

II.1 Blocs constituants la DPLL avec choix et dimensionnement 14

II.1.1 Le détecteur ou comparateur de phase 15

II.1.2 L'oscillateur contrôlé par tension (VCO) 22

II.1.3 Le filtre de boucle 38

II.2 Fonctionnement d'une PLL 39

II.2.1 Performance dynamique d'une DPLL en dehors de bruit 40
II.2.2 Plages de fonctionnement d'une DPLL 41
II.2.2.1 La plage de maintien 41
II.2.2.2 La plage de verrouillage (ou plage de capture rapide) 42
II.2.2.3 La plage de capture lente (pull in) 44
II.2.2.4 La plage de poursuite (tracking) (pull out) 50
II.2.3 Performance dynamique d'une DPLL en présence de bruit 52
II.3 Conception de la DPLL 57

CHAPITRE III : CONCEPTION D'UNE SPLL

III.1 Compromis soft-Hard 68
III.2 Faisabilité d'une conception de SPLL 69
III.3 Support hard et soft du DSP 70
III.3.1 Interruptions 70
III.3.2 Logiciels d'accompagnement 70
III.3 Fonctionnement d'une SPLL comme DPLL 73
III.3.1 Algorithme SPLL 74
III.3.2 Sous-programmes fonctionnels de la SPLL 78
III.5 Implémentation de l'algorithme SPLL sur DSP 85
III.6 Test d'acquisition en temps réel du signal d'entrée 88
III.6 Test d'acquisition en temps différé du signal d'entrée 89

CONCLUSION GENERALE 96
BIBLIOGRAPHIE 99

INTRODUCTION GÉNÉRALE

La boucle à verrouillage de phase, appelée plus communément PLL (en anglais Phase-locked loop), est l'un des composants électroniques les plus répandus. Présentes dans presque tous les appareils électroniques des téléviseurs aux téléphones mobiles, les PLLs possèdent une large gamme d'applications, y compris la synthèse de fréquence, la récupération de données d'horloge, la démodulation AM et FM, le contrôle de la vitesse de moteurs, les décodeurs FSK [1], et la robotique pour n'en citer que quelques-unes. L'omniprésence et la popularité des PLLs sont dues aussi à la nature robuste et la pureté spectrale de leur signal de sortie, choses qui sont impossibles à réaliser sans leur utilisation.

Il existe diverses architectures de boucles à verrouillage de phase et le choix de la structure à adopter dépend aussi bien de l'application dans laquelle elle est impliquée que de la nature analogique et/ou numérique des signaux qu'elle met en œuvre.

Dans les années 80, la dominance en matière de circuits à asservissement de phase était celle des PLLs analogiques ou linéaires (LPLL) ainsi que les PLLs semi-numériques (DPLL). Peu de temps après, deux différents types de PLL ont gagné d'importance : les PLLs numériques (ADPLL) avec les avantages qu'elles présentent et les PLLs softwares (SPLL) suite à la disponibilité des processeurs du traitement numérique du signal.

En effet, l'existence actuelle de DSP rapides, a ouvert grandement le champ aux applications SPLL. Ces dernières peuvent être programmées pour agir telle une LPLL, une DPLL, ou une ADPLL. Ainsi, la SPLL est le type le plus universel des PLLs, malgré qu'elle doit remplir une condition nécessaire : l'algorithme de calcul remplissant la fonction d'une PLL doit impérativement être exécuté au moins une fois à chaque période du signal d'entrée, limitant ainsi sévèrement la rangée couverte en fréquence.

Tout récemment [2], la PLL neuronale (NPLL) est en train de se développer. Dans ce nouveau type de PLL, tous ses constituants sont à base de réseaux de neurones, y compris son oscillateur contrôlé par débit (RCO). La NPLL est utilisée dans la poursuite et décodage des basses modulations fréquentielles (< 1kHz), comme celles qui se produisent dans les capteurs actifs biologiques chez les mammifères [2].

On note cependant que les différents types de PLL qu'on vient de citer, actent différemment et qu'il n'y a pas une théorie commune qui les couvre tous. Par conséquent, on est obligé de limiter notre travail à l'étude de deux types d'entre elles uniquement, DPLL et SPLL.

Notre mémoire est composé de trois chapitres :

Le premier chapitre, consacré aux généralités sur les PLLs, est composé de quatre parties. On commence par élucider le principe de base du verrouillage de phase avec un aperçu rapide sur les trois blocs constituant une PLL. Nous donnons ensuite une classification des types les plus connus des boucles de verrouillage de phase. Un bref historique de ces dernières est donné dans la troisième partie, suivi de la présentation de plusieurs applications des PLLs dans différents domaines. En dernier, on a jugé utile de faire un court état de l'art.

Dans le chapitre deux, on se focalise sur les boucles à verrouillage de phase dites semi-numériques ou DPLLs. En effet, sachant que les LPLLs et les DPLLs possèdent des propriétés communes, une présentation plus approfondie du fonctionnement des PLLs a été omise du premier pour faire partie de ce chapitre. Ayant aussi comme objectif de concevoir et de réaliser une DPLL, le choix et le dimensionnement des blocs la constituant, ainsi que l'étude de ses performances dynamiques en absence et en présence de bruit sont détaillés dans ce chapitre avec la présentation des différents résultats obtenus lors des simulations établies sur Orcad Pspice. Ceci a abouti au choix d'une DPLL remplissant les conditions de notre cahier de charges.

L'implémentation et l'analyse de la DPLL à base de processeur de traitement numérique du signal (DSP) est effectuée dans le troisième chapitre. On commence par mettre en évidence l'intérêt de ce type de PLL. Nous présentons ensuite l'algorithme et les organigrammes proposés, avec un bref aperçu sur les supports hard et soft du DSP C6713 de Texas instruments choisi pour implémenter notre SPLL. Une autre partie de ce chapitre est consacrée à l'implémentation de l'algorithme SPLL avec une présentation détaillée de la mise en œuvre de l'application conçue. En dernier, les résultats obtenus pour notre SPLL sont présentés, interprétés et comparés avec ceux de la DPLL.

Nous clôturons notre mémoire avec une conclusion générale dans laquelle on fait le tour du travail accompli.

CHAPITRE I : GÉNÉRALITÉS SUR LES PLLs

La boucle de verrouillage de phase représente l'un des sujets les plus actifs dans le domaine de l'électronique. Une PLL sert en fait à synchroniser la phase et la fréquence de sortie d'un oscillateur contrôlé pour assortir la phase et la fréquence de sortie d'un oscillateur de référence. L'importance de ce principe a fait que les PLLs jouissent d'un rôle important dans différents domaines comme le filtrage, la synthèse de fréquence, la modulation de fréquence, la démodulation, la détection des signaux, le suivi de fréquence, etc.

Notre objectif dans ce premier chapitre est de faire une présentation générale des PLLs. Nous commencerons par donner une définition formelle d'une PLL et par expliquer son principe de fonctionnement. Les différents blocs constituant une PLL seront présentés ensuite. Une autre section sera consacrée à un bref historique de l'étude des PLLs qui sera suivi d'une présentation de quelque unes de leurs applications par lesquelles nous voulons mettre en évidence leur importance dans différents domaines. Nous ne manquerons pas non plus de faire un bref état de l'art de l'étude des PLLs dans lequel seront données des références importantes pouvant guider le lecteur voulant explorer ce domaine.

I.1 La boucle à verrouillage de phase

I.1.1 Principe de base du verrouillage de phase

La boucle à verrouillage de phase est un système à contre-réaction qui a pour but de synchroniser en phase et en fréquence deux signaux dont l'un est la référence et l'autre est issu d'un oscillateur local. Dans l'état synchronisé ou de verrouillage (en anglais locked state), l'erreur de phase entre le signal de sortie de l'oscillateur et le signal de référence est égal à zéro, ou elle reste constante.

Si une erreur de phase s'accumule, un mécanisme de contrôle agit sur l'oscillateur de telle sorte que l'erreur de phase est à nouveau réduite au minimum. Dans un tel système

de commande, la phase du signal de sortie est effectivement verrouillée sur la phase du signal de référence. C'est pourquoi ce système est appelé boucle à verrouillage de phase.

Bien qu'il y ait de nombreuses manières de réaliser une PLL, sa structure globale présentée dans la figure I.1 reste la même. Une PLL est constituée des trois blocs suivants :

- Un détecteur de phase.
- Un filtre de boucle.
- Un oscillateur contrôlé en tension.

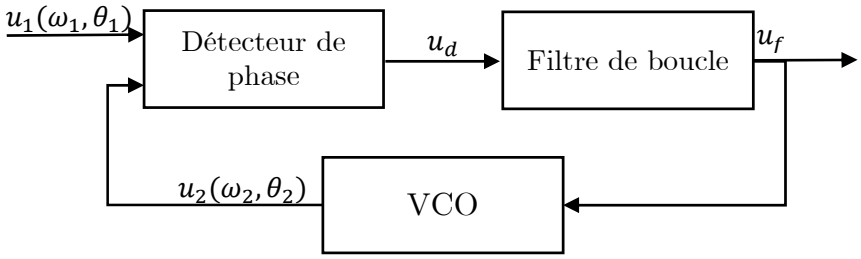


Figure I.1 Schéma-bloc d'une PLL.

Le fonctionnement de base d'une PLL peut être divisé en 3 étapes :

1. Le détecteur de phase détecte la différence de phase entre deux entrées et génère un signal d'erreur u_d dont la valeur moyenne est linéairement proportionnelle à la différence de phase.

2. Le filtre de boucle est ensuite utilisé pour supprimer les composantes haute-fréquence de la sortie du détecteur de phase, permettant à la valeur moyenne de contrôler la fréquence de VCO.

3. L'oscillateur génère un signal de fréquence instantanée directement proportionnelle à la tension de contrôle générée par le filtre de boucle. Le signal généré est injecté à l'entrée du détecteur de phase et une autre comparaison de phase est démarrée, jusqu'à ce que la différence de phase atteigne une relation fixe [1] et [3].

I.1.2 Les blocs d'une PLL

I.1.2.1 L'oscillateur contrôlé en tension (VCO)

Les oscillateurs contrôlés en tension (VCO) utilisés dans les PLLs ne sont pas différents de ceux utilisés pour d'autres applications, telles que la modulation et la commande automatique de fréquence. Les principales exigences pour le VCO sont :

- La stabilité de phase,

- Une grande déviation de fréquence,
- Une grande sensibilité de modulation K_o,
- La linéarité de la fréquence en fonction de la tension de commande.
- La capacité d'accepter la modulation à large bande.

La stabilité de phase, qui en réalité en oppositions avec les quatre autres exigences, peut être améliorée par un certain nombre de façons:

- En utilisant des cristaux et des circuits à High-Q.
- Le maintien d'un faible bruit dans la partie de l'amplificateur, sachant que La plus grande partie de la gigue de phase d'un oscillateur est produite du bruit dans l'amplificateur associé.
- La stabilisation de la température.
- Le maintien de la stabilité mécanique.

Les VCOs sont généralement de la forme d'un oscillateur en anneau, d'un oscillateur à relaxation ou d'un oscillateur de résonance. L'oscillateur en anneau prend la forme d'un nombre impair d'inverseurs connectés en une boucle de rétroaction. L'oscillateur de relaxation utilise un trigger de Schmitt pour générer une onde carrée stable [4]. L'oscillateur de résonnance met un circuit résonnant dans le trajet positif de rétroaction de la tension à un amplificateur de courant.

D'autres formes de VCOs, tels que les oscillateurs à quartz et les oscillateurs YIG opèrent essentiellement sur le même principe, mais modifient le circuit résonant [5, 6, 7]. Pour les PLLs entièrement numériques et les software PLLs qui vont être présentées plus tard, le VCO est remplacé par un oscillateur numérique ou à commande numérique (DCO/NCO). Dans ce cas, la tension d'entrée est remplacée par une valeur numérique, et la sortie est une forme d'onde numérique oscillante [8].

I.1.2.2 Le détecteur de phase (PD)

Le détecteur de phase (comparateur de phase) a comme tâche de détecter la différence de phase entre le signal d'entrée et le signal bouclé. Le fonctionnement exact du détecteur de phase diffère selon son type ; il peut être un détecteur de phase mélangeur (multiplieur), une porte XOR, une pompe de charge qui détecte la fréquence et la phase, ou une bascule JK [9]. Le détecteur de phase multiplieur est utilisé pour déterminer la différence de phase entre deux signaux analogiques, tandis que les autres sont généralement utilisés pour la comparaison de signaux numériques.

Cependant, aucun comparateur de phase n'est idéal pour toutes les applications. Néanmoins, le comparateur de phase à pompe de charge (DPF) qui va être considéré dans ce mémoire est le plus utilisé.

Les différents détecteurs de phase sont classés dans le tableau I.1 [10].

Type	Classe	Gain (K_d)	Plage linéaire	Applications
Multiplieur	Analogique	Non linéaire et proportionnel à l'amplitude du signal d'entrée	$-\pi/2$ à $\pi/2$	Modulation et démodulation de fréquence
XOR	Numérique	$(U_h - U_l)/\pi$	$-\pi/2$ à $\pi/2$	Récupération d'horloge et de données
Bascule JK	Numérique	$(U_h - U_l)/2\pi$	$-\pi$ à π	Redressement
Détecteur de phase fréquence (PFD)	Numérique	$(U_h - U_l)/4\pi$	-2π à 2π	Synchronisation d'horloge et synthèse de fréquence

Tableau I.1 Classes de détecteur de phase.

I.1.2.3 Le filtre de boucle

Le filtre de boucle est un filtre passe-bas qui est utilisé pour réduire le bruit et les composantes de signal à haute fréquence du détecteur de phase. Il fournit un signal de commande à l'oscillateur qui est proportionnel à la différence de phase entre la référence et la sortie de la PLL.

Il est généralement de faible ordre (premier ou deuxième). Ses performances, du point de vue bruit, peuvent être améliorées en augmentant son ordre; Mais cela a pour conséquence négative d'accroitre la complexité de la boucle et potentiellement réduire la stabilité [11].

Les paramètres du filtre de boucle ainsi que ses différentes topologies peuvent avoir une grande influence sur les performances dynamiques de la boucle [10].

I.2 Classification des PLLs

Les premières boucles à verrouillage de phase ont été construites à partir de composants discrets, y compris les tubes électroniques et, plus tard, les transistors discrets. Tous ces circuits sont des circuits linéaires. Les premiers circuits intégrés PLL sont apparus vers 1965, ils étaient également purement analogiques. Un multiplieur analogique a été utilisé comme détecteur de phase, le filtre de boucle a été construit à partir d'un filtre actif ou passif RC, et le VCO a été utilisé pour générer le signal de sortie de la PLL. Ce type de PLL est désigné aujourd'hui sous le nom de PLL linéaire (LPLL). Dans les

années qui ont suivi, la PLL avançait lentement mais sûrement dans le territoire numérique. La première PLL numérique (DPLL), qui est apparu vers 1970, était un système hybride : seul le détecteur de phase a été construit à partir d'un circuit numérique (par exemple, d'une porte XOR ou une bascule JK), mais les blocs restants étaient toujours analogiques. Quelques années plus tard, la "tout-numérique" PLL (ADPLL) a été inventée. La boucle ADPLL est exclusivement construite à partir de blocs fonctionnels numériques ; par conséquent, elle ne contient pas de composants passifs tels que des résistances et des condensateurs.

Analogiquement avec les filtres, les PLLs peuvent également être implémentées par software. Dans ce cas, la fonction de la PLL n'est plus effectuée par des composants électroniques, mais plutôt par un programme informatique. Ce dernier type de PLL est appelé SPLL. La SPLL est implémentée sous une plate-forme matérielle telle qu'un microcontrôleur ou un processeur de signal numérique (DSP). La fonction PLL est donc réalisée par software. Ceci offre une plus grande flexibilité, car un grand nombre de différents algorithmes peuvent être développés. Par exemple, une SPLL peut être programmée pour se comporter comme une LPLL, une DPLL, ou une ADPLL.

Cette classification [1] et [12] est récapitulée dans le tableau I.2

	PLL	**Détecteur de phase**	**Filtre de boucle**	**Oscillateur**
01	PLL linéaire (LPLL)	Analogique	Analogique	Oscillateur contrôlé par tension (VCO) à sortie sinusoïdale
02	PLL semi-numérique (DPLL)	Numérique	Analogique	VCO
03	PLL numérique (ADPLL)	Numérique	Numérique	Oscillateur Contrôlé numériquement (DCO)
04	Software PLL (SPLL)	Software	Software	Software

Tableau I.2 Classification des PLLs.

I.3 Historique et applications des PLLs

Bien que les PLLs sont souvent considérées comme de nouvelles structures, la littérature du concept remonte au début du XXème siècle [13]. Les premières études systématiques avec une réalisation électronique d'un asservissement de phase semblent être celles de Appleton en 1922 [14] et de Van Der Pol en 1927 [15], qui ont montré que l'on pouvait asservir la phase d'un oscillateur à triodes au moyen d'un signal de fréquence légèrement différente. La description connue d'une PLL par rétroaction a été publiée par De Bellescize [16] en 1932, à propos de la réception synchrone de signaux

radio. Il proposait un asservissement de phase dans le but de reconstruire la porteuse d'un signal modulé en amplitude pour opérer à la réception de ce signal.

Ce principe de réception dit hétérodyne a été en un premier temps délaissé, car trop complexe, au profit de la réception synchrone pendant quelques années. La réception hétérodyne est ensuite devenue incontournable avec le besoin d'accroître les performances et avec la réduction des coûts apportée par l'électronique.

Même quand l'introduction des boucles à verrouillage de phase dans un récepteur de radio a été considérée comme coûteuse pour l'utilisation massive, il y avait encore de nombreuses utilisations pour les PLLs. Un exemple, qui a permis aux PLLs d'atteindre leur première utilisation à grande échelle a été la synchronisation horizontale et verticale des balayages des postes de télévision [17]. Une autre application antérieure de la PLL était la démodulation FM [18].

L'histoire des boucles PLLs devient difficile d'écrire après 1960 en raison de l'intérêt explosif et au grand nombre de publications dans le domaine. La disponibilité des circuits imprimés (CI) de PLLs en 1965 a facilité l'introduction rapide des boucles à verrouillage de phase dans les produits destinés aux consommateurs. Vers 1970, les PLLs numériques ou échantillonnées, ont commencé à apparaître dans la littérature et mêmes dans les produits de consommation. L'article de Gupta [19] de 1975 fournit une partie de l'histoire des PLLs analogiques, et l'article de Lindsey et Chie [20] de 1981 fournit un aperçu de l'évolution des PLLs numériques.

Aujourd'hui, les PLLs jouent un rôle important dans les appareils de communication modernes. Elles sont considérées comme étant les moyens les plus robustes de génération d'un signal de référence de faible bruit et le meilleur moyen de détermination du signal d'horloge provenant d'une source bruitée. Il y a eu beaucoup de recherches dans ce domaine et de nouvelles applications et avancées architecturales continuent à être trouvées. L'omniprésence de la PLL est certainement due à sa large gamme d'applications [21] comme :

Récupération de porteuse

Figure I.2 montre un signal reçu v_i composé d'impulsions d'une sinusoïde. Ceci est similaire aux "salves couleur" dans un signal de télévision. La fréquence et la phase d'un oscillateur dans le récepteur de télévision doivent être verrouillées à celles des salves. Ce signal v_0 de l'oscillateur est ensuite utilisé pour démoduler les informations de la couleur dans le signal de télévision. Quand une salve est produite, le détecteur de phase a une chance de comparer la phase de v_0 avec celle de v_i. Toute erreur produit une tension v_d qui est appliquée au VCO afin de corriger la phase. (Une petite variation dans la fréquence change la phase).

Un spectre de l'entrée indique que le signal d'entrée v_i comporte une composante de fréquence ω_i, qui est celle de la sinusoïde lors de la salve (voir Fig. 1-2). Mais le spectre comporte aussi beaucoup d autres composantes à plus ou moins 10% d'écart de ω_i.

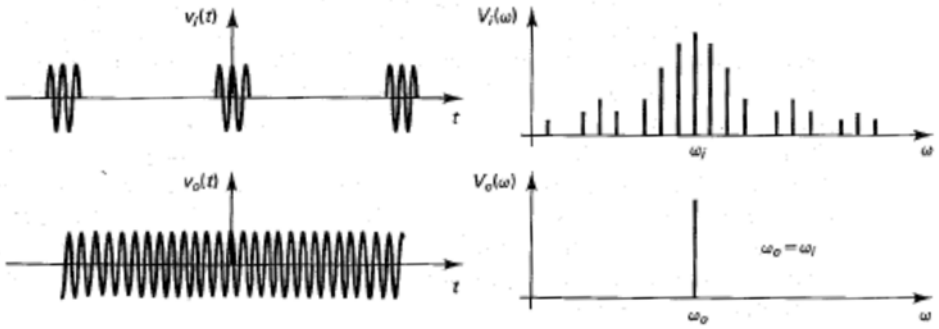


Figure I.2 Récupération de porteuse.

Récupération d'horloge

Dans cette application, un signal d'horloge v_0 doit être synchronisé à un signal de données numérique v_i. Pour l'exemple de la Figure I.3, v_i représente un état logique "1" par une impulsion et un "0" par l'absence d'une impulsion. La séquence de données ici est 1,0,1,1,0,1,1,1,0,0,1. Une analyse du spectre de ce signal de données indique qu il y a un composante de fréquence ω_i, où $2\pi/\omega_i$ est l'espacement entre les symboles logiques. Il ya aussi une large densité spectrale de fond en raison des vides aléatoires représentant les "0" dans les données. Une PLL peut être utilisé pour verrouiller la fréquence ω_0 de l'oscillateur à la composante ω_i, produisant ainsi le signal d'horloge v_0 représenté.

L'horloge aurait pu être récupérée avec un filtre à bande étroite au lieu d'une PLL. Toutefois, la densité spectrale de fond au voisinage de ω_i passerait également à travers le filtre, ce qui va corrompre l'horloge.

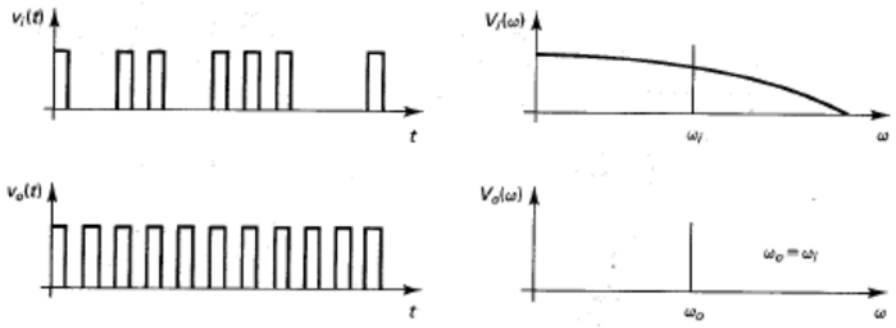


Figure I.3 Récupération d'horloge.

Filtre suiveur

Un avantage d'une PLL sur un filtre à bande étroite est sa capacité à suivre une fréquence d'entrée ω_i se dérivant avec le temps. La Figure I.4 montre une fréquence ω_i qui baisse avec le temps, ceci est peut être dû à l'effet Doppler. La PLL suit la composante à la fréquence ω_i et continue de récupérer l'horloge. Si un filtre à bande étroite avait été utilisé au lieu d'une PLL pour récupérer la fréquence ω_i, cette dernière dériverait rapidement hors de la bande passante étroite. Dans cette application, la PLL agit en quelque sorte comme un filtre à bande étroite dont la fréquence centrale peut se déplacer.

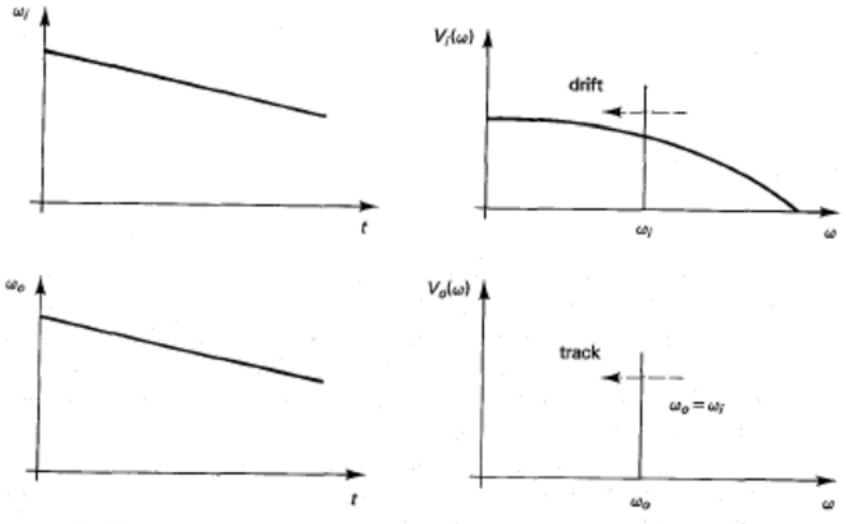


Figure I.4 Filtre suiveur.

Démodulation de fréquence

La majorité des récepteurs FM utilisent une PLL pour la démodulation de fréquence. Dans cette application, la fréquence de sortie ω_0 de la PLL suit la fréquence d'entrée ω_i qui varie en fonction de la modulation (voir Figure I.5). Si la tension de contrôle v_c du VCO est proportionnelle à ω_0, elle est également proportionnelle à ω_i. Par conséquent, v_c est le signal démodulé. Il est à noter que la bande passante de la PLL doit être suffisamment large pour avoir la vitesse nécessaire pour de suivre les variations de ω_i.

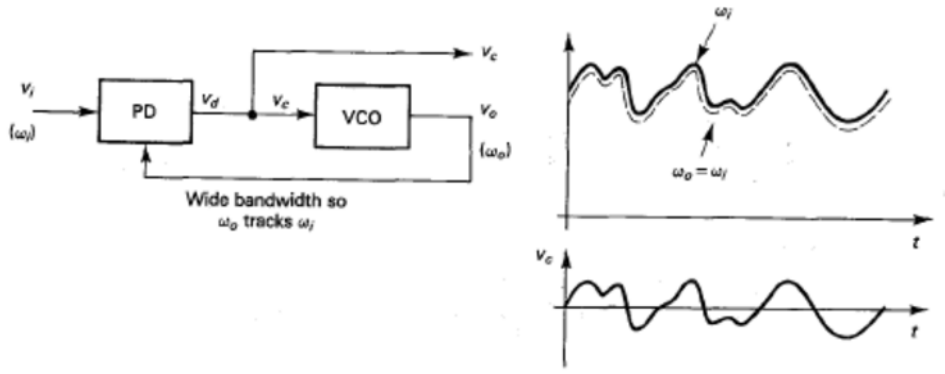


Figure I.5 Démodulation de fréquence.

Démodulation de phase

Dans une application similaire, une PLL peut être utilisée pour la démodulation de phase. Ici, le signal reçu v_i est un transporteur dont la phase modulée θ_i transmet l'information (voir Figure I.6). Dans cette application, la bande passante de la PLL est si étroite (la PLL est donc "lente") que θ_0 reste à la moyenne de θ_i plutôt que de la suivre. La phase de sortie θ_0 est presque constante; c'est à dire, v_0 est la porteuse non modulée récupérée. Ceci sert de référence pour le détecteur de phase pour démoduler θ_i. Si le détecteur de phase a une caractéristique linéaire, sa sortie v_d est proportionnelle à θ_i, et v_d est la sortie démodulée.

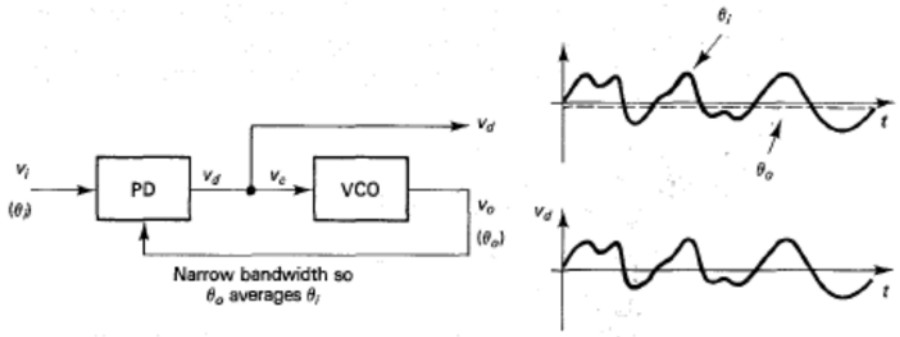


Figure I.6 Démodulation de phase.

Modulation de phase

Dans la Figure I.7, une PLL est modifiée en additionnant un signal de modulation v_m dans le circuit pour moduler la phase θ_0. La tension v_m tente de changer la fréquence du VCO. Mais si la bande passante de la PLL est assez large, cette dernière peut réagir rapidement et ajuster v_d pour annuler l'effet de v_m. Ainsi, $v_d \approx -v_m$, et $v_c = v_d + v_m$ reste essentiellement constante. Si la phase d'entrée θ_i est constante (zéro), v_d est proportionnelle à $-\theta_0$, et θ_0 est donc proportionnelle à v_m. Ainsi, le signal v_m module la phase θ_0 du VCO.

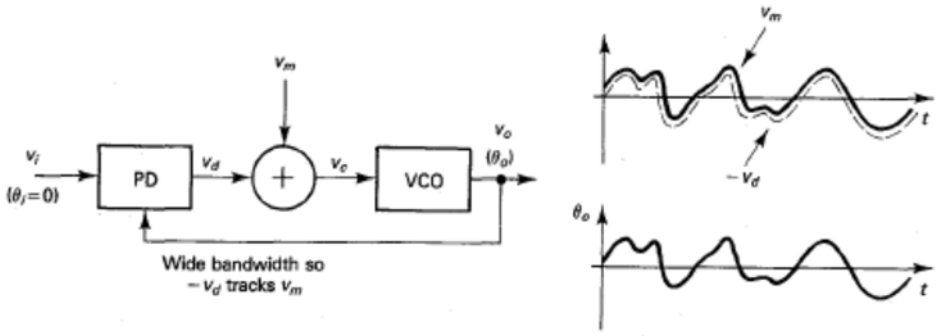


Figure I.7 Modulation de phase.

Synthèse de fréquence

Un synthétiseur de fréquence génère des multiples de la fréquence de référence ω_i. Par exemple, si $\omega_i = 1$ krad/s, alors le synthétiseur peut générer 100, 101, ..., 200 krad/s. Autrement dit, $\omega_0 = N\omega_i$, où N varie de 100 à 200. Un tel multiplieur de fréquence peut être réalisé avec une PLL, comme le montre la Figure I.8. Dans cette application, un diviseur de fréquence est inclus dans le chemin de contre-réaction de la PLL. L'entier N par lequel ω_0 est divisée peut être sélectionné par l'utilisateur. En état de verrouillage, la PLL assure que les deux fréquences ω_i et w_0/N à l'entrée du détecteur de phase (PD) soient égales. Ensuite, $\omega_0 = N\omega_i$, comme prévu.

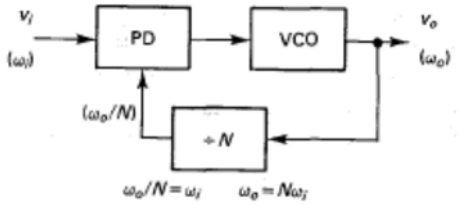


Figure I.8 Synthèse de fréquence.

I.4 Etat de l'art

Les PLLs ont été surtout étudiées à partir de leur première utilisation intensive dans les postes de télévision. De 160 articles publiés en 1966, le nombre a franchi les 1000 articles en 1980 et a progressé avec la diversité des applications. De nos jours, une recherche sur internet avec la phrase « phase locked loop » atteint quelque millions de réponses.

La PLL analogique est la première et la plus étudiée des boucles à verrouillage de phase. Comme ses composants sont tous analogiques, elle peut être modélisée par un système d´équations différentielles ordinaires ; cette simplicité d'analyse est en fait une base de référence pour l'étude des autres types d'architecture. De très bons livres [3] et [22-24], deux numéros spéciaux IEEE [25-26], et un tutoriel [19] sont consacrés à la théorie et aux applications des LPLL.

Le progrès permanent dans l'amélioration de la performance, la vitesse, la fiabilité et la réduction simultanée de la taille et du coût des circuits intégrés (LSI et VLSI) a donné lieu à un grand intérêt pour la mise en œuvre des PLLs dans le domaine numérique. Mis à part les avantages évidents associés aux systèmes numériques, la version numérique de la PLL atténue certains des problèmes associés à son homologue analogique.

Vers la fin des années soixante, la technologie des LPLLs avait déjà atteint un stade avancé de maturité. Un vif intérêt est apparu pour la mise en œuvre et la conception des DPLLs développées en raison de la popularité et la disponibilité constante des intégrateurs numériques à grande échelle (LSI). Les efforts initiaux dans la conception

des DPLLs ont été principalement concentrés sur le remplacement partiel de composants analogiques par des composants numériques, d'où leurs autres appellations : PLLs semi analogiques ou PLLs hybrides. La première publication dans ce sens a été celle de Westlake [27]. Il a introduit un circuit d'échantillonnage et de maintien à la sortie du filtre de boucle afin de profiter des améliorations offertes par un VCO numérique. Le détecteur de phase basée à bascule a été introduit par Byrne [28] en 1962. En 1968 Gupta [29] a utilisé un filtre numérique à la place d'un filtre de boucle analogique, et analysé la performance des PLLs résultantes.

Nous avons vu qu'une ADPLL est un système construit avec des composants numériques uniquement. En 1981, Lindsey et al. [20] ont présenté une étude globale du travail théorique/expérimentale accompli dans le domaine des ADPLLs pendant la période allant de 1960 à 1980.

Dans les années 90, l'implémentation de PLLs numériques avait principalement des fins de synchronisation [30-36]. Un oscillateur numérique commandé (DCO) utilise une source à haute fréquence pour fournir une référence. La plage de fonctionnement est limitée par la référence à haute fréquence. Dans le XXIe siècle, le thème principal de la recherche dans le domaine des PLLs numériques consiste à des applications de communication. Récemment, plusieurs DPLLs et ADPLLs, pour différentes applications (y compris les multi-gigahertz), ont été développées [37-38]. Ceci montre la capacité d'une mise en œuvre numérique d'atteindre les performances des PLLs analogiques et même de les surpasser.

Avec la technologie de pointe des microcontrôleurs et des processeurs de traitement numérique du signal (DSP), toutes les fonctions de la PLL classique ont été mis en œuvre par logiciel. En effet, la PLL software est devenue une alternative compétitive à la PLL matérielle. De plus, La conception des software PLLs offre plus de degrés de liberté par rapport à la conception de PLLs classiques, et ce parce qu'une SPLL peut être adaptée pour exercer une fonction, sans modifier le matériel qui n'est pas le cas de la PLL matérielle [1].

CHAPITRE II : ETUDE ET CONCEPTION D'UNE DPLL

Chaque système DPLL à mettre en œuvre présente une solution optimale pour sa conception. Par exemple, il y a des applications de DPLL dans les communications où le système est utilisé pour extraire le signal d'horloge à partir d'un signal d'information éventuellement bruité. Dans une telle application, la suppression de bruit est capitale. Une application différente est la synthèse de fréquence. Ici, l'effet de bruit n'est pas d'importance, mais le synthétiseur doit être capable de commuter rapidement d'une fréquence à l'autre, mettant en exergue le temps de capture (lente) comme l'ultime paramètre. Pour ces raisons, notre conception de DPLL ne présente pas un caractère universel à suivre mais sert plutôt comme un guide de conception. Dans plusieurs cas, la conception de DPLL est un processus itératif dans lequel certaines étapes sont à refaire jusqu'à ce que les paramètres-clé (par exemple, le temps de capture) soient à l'intérieur des rangées planifiées [1].

II.1 Blocs constituants la DPLL avec choix et dimensionnement

La DPLL classique est un système hybride construit à partir de blocs analogiques et numériques. Le diagramme-bloc est montré en figure II.1.

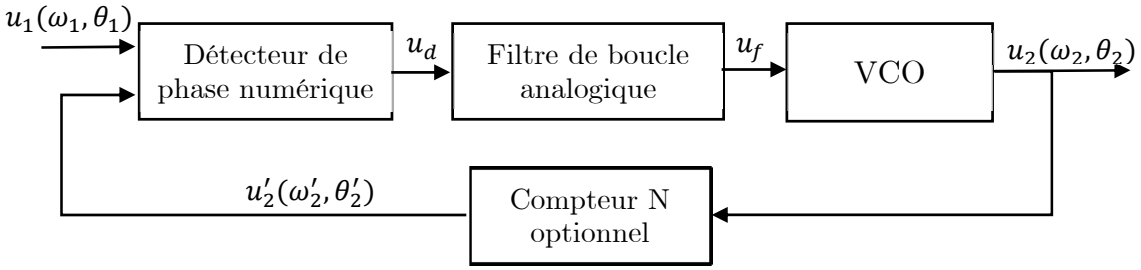


Figure II.1 : Schéma-bloc d'une DPLL.

Il consiste en trois blocs fonctionnels qui sont, le détecteur de phase, le filtre de boucle, et l'oscillateur contrôlé en tension (VCO). La seule partie vraiment numérique est le détecteur de phase. Eventuellement, une autre partie numérique peut être rajoutée,

c'est le compteur diviseur de fréquence assurant un rapport N entre la fréquence de sortie et celle d'entrée , $\omega_2 = N\omega_1$.

Dans certaines applications, les fréquences d'entrée et de sortie de la DPLL peuvent être variables. Dans la variante des synthétiseurs de fréquence, la fréquence d'entrée est toujours la même, mais la fréquence de sortie est variable. Lorsqu'une DPLL est conçue en tant qu'un synthétiseur de fréquence, le rapport entre la fréquence de sortie et celle d'entrée est variable telle qu'une rangée de N doit être définie $(N_{min} \leq N \leq N_{max})$. Lorsque N est variable, la fréquence naturelle ω_n et le facteur d'amortissement ε de la DPLL varient avec $1/\sqrt{N}$ (voir plus tard, l'équation de la fonction de transfert de la DPLL en boucle fermée, numéro II.21). Par conséquent, nous obtenons les rangées $\omega_{nmin} \leq \omega_n \leq \omega_{nmax}$ et $\varepsilon_{min} \leq \varepsilon_n \leq \varepsilon_{max}$ dont on obtient respectivement :

$$\frac{\omega_{nmax}}{\omega_{nmin}} = \sqrt{\frac{N_{max}}{N_{min}}} = \frac{\varepsilon_{max}}{\varepsilon_{min}} \qquad \text{(II.1)}$$

Si N varie d'un facteur 10, ε varie d'un facteur d'environ 3 qui peut être toléré sachant que le facteur d'amortissement optimal est pris entre 0.5 et 1. De plus grandes variations doivent être écartées, car la boucle devient trop oscillatoire ou trop amortie. Ainsi, pour de grandes valeurs pour N (1 à 100, par exemple), on définit souvent plusieurs rangées fréquentielles pour la DPLL dont on y accède par commutation. Dans le cas de N variable, les paramètres de la DPLL sont spécifiés tel que ε devient optimal $(\varepsilon = 0.7)$ pour un rapport de division N_{moyen} qui est donné par la moyenne géométrique [1]:

$$N_{moy} = \sqrt{N_{min}N_{max}} \qquad \text{(II.2)}$$

Dans notre application, il faut spécifier les fréquences d'entrée et de sortie de la DPLL qu'on prendra égales sur deux décades dans la rangée de $f_{min} = 10kHz$ à $f_{max} = 1MHz$. Aucun diviseur de fréquence n'est donc nécessaire $(N = 1)$.

II.1.1 Le détecteur ou comparateur de phase

Un détecteur de phase analogique qui n'est autre qu'un multiplieur, présente à sa sortie lorsque la PLL est verrouillée, une composante haute-fréquence qui sera éliminée par le filtre de boucle et une composante utile basse fréquence dépendant uniquement de la phase d'erreur $\theta_e(t)$ au verrouillage :

$$u_d(t) = K_d sin\theta_e \qquad \text{(II.3)}$$

Avec K_d, la sensibilité ou gain du détecteur. La caractéristique d'un tel détecteur de phase est donc sinusoïdale [39], Figure II.2.

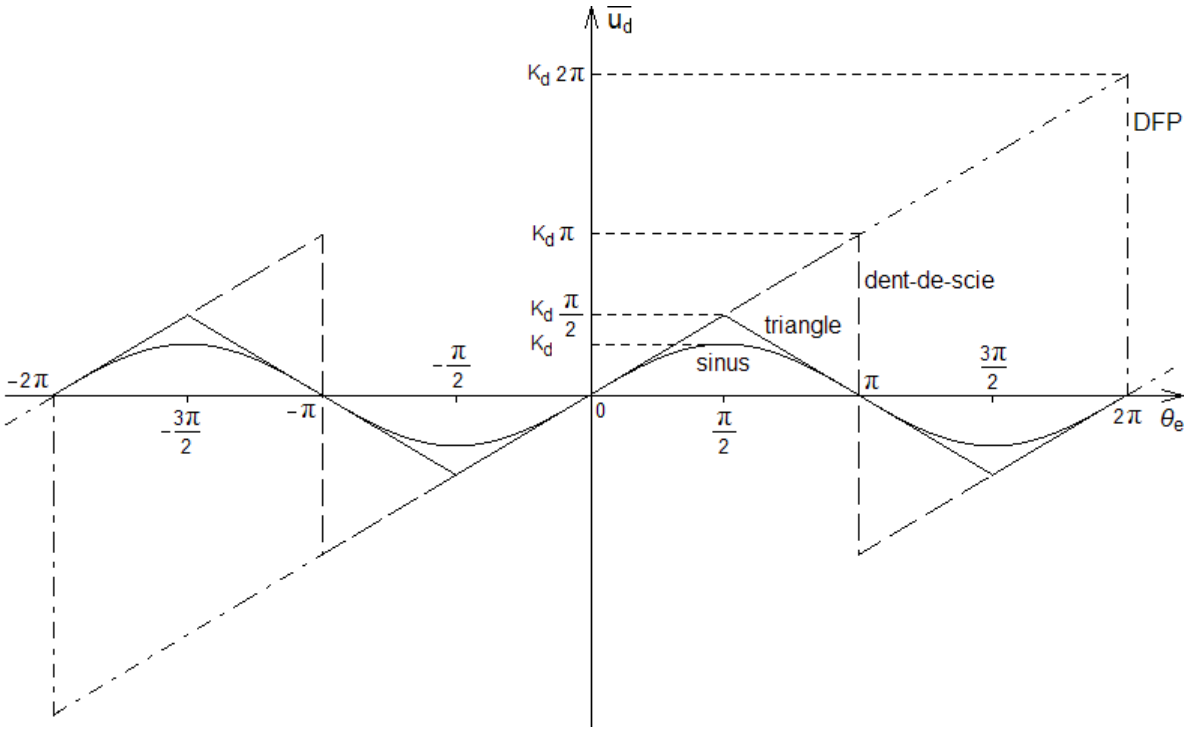


Figure II.2 : Caractéristiques des détecteurs de phase.

A l'encontre du détecteur de phase analogique, le détecteur de phase numérique présente une caractéristique linéaire par morceau. L'étendue du morceau linéaire de la caractéristique dépend du type de détecteur dont on distingue dans la littérature technique, trois types [1] et [39] :

- La porte XOR,
- La bascule JK (déclenchable par front),
- La pompe de charge ou détecteur fréquence-phase DFP.

La caractéristique du comparateur de phase à porte XOR est triangulaire, représentant la valeur moyenne $\overline{U_d}$ à la sortie du XOR en fonction de l'erreur de phase θ_e.
$\overline{U_d}$ est exactement proportionnelle à θ_e pour la rangée utile $-\pi/2 < \theta_e < \pi/2$:

$$\overline{U_d} = K_d\theta_e \qquad \text{(II.4)}$$

avec $K_d = U_{alim}/\pi,$ qui représente aussi la fonction de transfert du détecteur de phase.

La performance du XOR devient sévèrement altérée lorsque les signaux d'entrées U_1 et U_2 deviennent rectangulaires. La valeur moyenne de U_d s'écrête alors à certains niveaux et la plage de verrouillage de la PLL devient réduite.

Le rapport cyclique des signaux d'entrées U_1 et U_2 devient non important lorsqu'une bascule JK est utilisée comme détecteur de phase. La bascule est sensible aux transitions plutôt qu'aux niveaux des signaux d'entrées, tel qu'un front montant

appliqué à l'entrée J la fait basculer à l'état haut ($Q = 1$) et un front montant appliqué à l'entrée K la fait basculer à l'état bas ($Q = 0$). Ceci, permet d'étendre la rangée utile à $-\pi < \theta_e < \pi$ pour une caractéristique en dent de scie, Figure II.2. Dans ce cas, $\overline{U_d}$ est proportionnelle à θ_e (II.4), avec un gain $K_d = U_{alim}/2\pi$.

L'étendue de la rangée utile pour θ_e est encore meilleure pour le comparateur de phase à pompe de charge ou détecteur fréquence-phase (DFP), Figure II.3, qui présente comme son nom l'indique, un signal à sa sortie qui ne dépend pas uniquement de l'erreur de phase mais aussi de l'erreur en fréquence $\Delta\omega = \omega_1 - \omega_2$ lorsque la boucle n'est pas encore verrouillée. Le DFP est conçu à partir de deux bascules D dont les sorties sont notées « UP » et « DN », respectivement. Par une additionnelle porte AND supprimant l'état de sortie $UP = DN = 1$ en agissant sur les entrées CLEAR des deux bascules, le DFP peut se trouver dans l'un des trois états qui restent (dénotés $-1, 0, 1$) :

$UP = 0, DN = 1$: état -1,
$UP = 0, DN = 0$: état 0,
$UP = 1, DN = 0$: état $+1$.

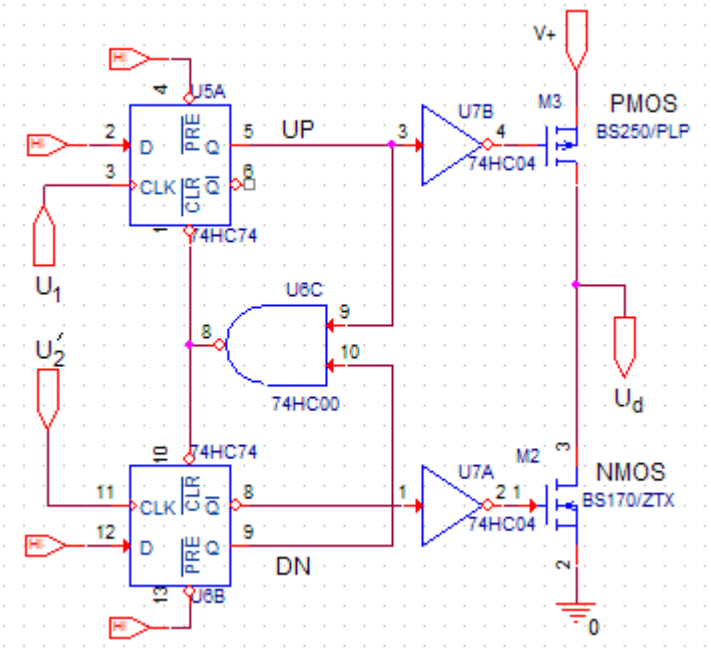


Figure II.3 : Détecteur fréquence-phase (DFP).

L'état actuel du DFP est déterminé par les transitions positives des signaux d'entrées U_1 et U_2 comme expliqué sur le diagramme d'état de la Figure II.4. Ainsi, un front montant de U_1 force le DFP à aller vers le prochain état plus élevé, à moins qu'il se trouve déjà à l'état $+1$. Par analogie, un front montant de U_2 force le DFP à aller vers le prochain état moins élevé, sauf s'il se trouve déjà à l'état -1.

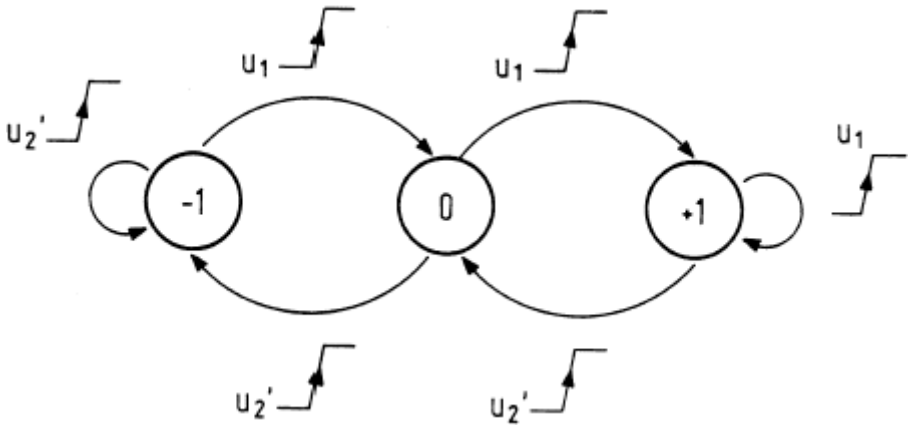


Figure II.4 : Diagramme d'état du DFP.

Le signal de sortie U_d est une fonction logique de l'état du DFP. Lorsque le DFP est à l'état $+1$, U_d doit être positive, lorsque le DFP est à l'état -1, U_d doit être négative, et lorsque le DFP est à l'état 0, U_d doit être nulle. Ce signal ternaire peut être interprété comme un signal logique à trois états, dont l'état $U_d = 0$ lui correspond l'état à haute impédance.

La circuiterie de la Figure II.3 montre comment que le signal U_d est généré sous une alimentation positive. Lorsque le signal UP est à l'état haut, le transistor PMOS se sature et U_d est égale à $+V_{alim}$. Lorsque le signal DN est à l'état haut, le transistor NMOS se sature et U_d est nulle. Si aucun signal n'est à l'état haut, les deux transistors sont bloqués et la sortie U_d flotte suite à son état en haute impédance.

Afin de comprendre le fonctionnement du DFP dans un système DPLL réel, les formes d'ondes de la Figure II.5 sont considérées. La Figure II.5a montre le cas où l'erreur de phase est nulle. Les signaux d'entrées U_1 et U_2 sont ici exactement en phase, leurs transitions positives coïncident aux mêmes moments dont l'effet est supprimé. Le DFP stagne dans cet état indéfiniment. La Figure II.5b montre le cas où U_1 dépasse U_2. Le DFP commute maintenant entre les états 0 et $+1$. Si U_1 recule devant U_2 comme montré en Figure II.5c, le DFP commute alors entre les états -1 et 0. Il devient aisé à partir des figures I.5b et I.5c, de constater que U_d croît et s'approche de $360°$ quand l'erreur de phase est positive, alors qu'elle décroît et s'approche de $-360°$ quand l'erreur de phase est négative.

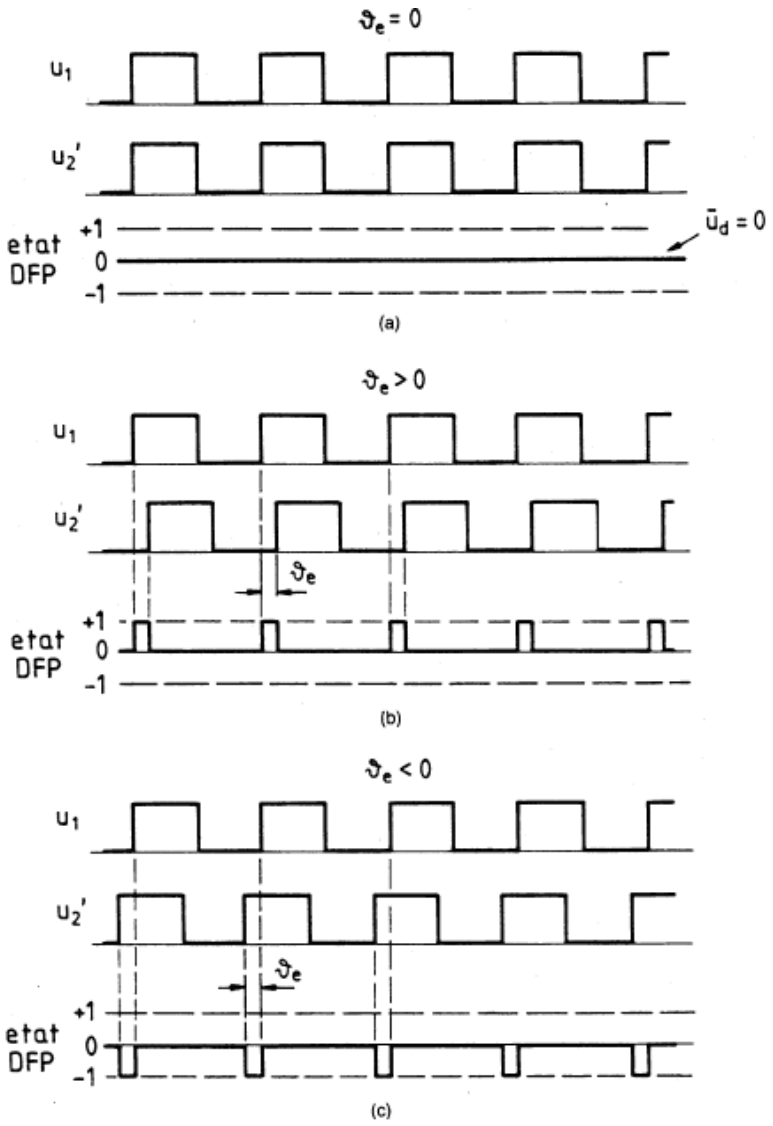


Figure II.5 : Comparaison de phase du DFP pour un écart nul (a), écart positif (b), écart négatif (c).

La caractéristique de la tension moyenne de U_d en fonction de l'erreur de phase est une dent scie, Figure II.2, dont la rangée s'étend entre $\pm 2\pi$. Lorsque l'erreur de phase θ_e excède $+2\pi$, le DFP se comporte comme si l'erreur de phase a recyclé vers zéro accusant ainsi une périodicité de 2π de la courbe caractéristique. Un raisonnement analogue peut être effectué pour des erreurs de phase plus faibles que -2π. Dans la rangée utile de $\pm 2\pi$, la tension moyenne de U_d est proportionnelle à l'erreur de phase θ_e, expression (II.4), avec un gain $K_d = U_{alim}/4\pi$, qui représente aussi la fonction de transfert du détecteur de phase.

Les différents détecteurs de phase se comportent différemment envers le bruit. Dans la situation où les fronts du signal d'entrée u_1 de référence sont tous affectés de bruit et pas de façon sélective, le meilleur choix sera un DFP [1] vu ses caractéristiques plus étendues que celles des autres détecteurs qui le rend préféré. Le DFP étant retenu pour notre application, son gain sous une alimentation de 5V, devient $K_d = 0.4\,V/rd$.

Pour apprécier l'avantage énorme offert par le DFP, nous devons supposer que la DPLL est déverrouillée initialement. En outre, nous faisons l'hypothèse que la fréquence de référence ω_1 est plus élevée que la fréquence de sortie ω_2. Le signal U_1 génère alors plus de transitions positives par unité de temps que le signal U_2. En consultant la Figure II.4, on s'aperçoit que sous cette condition, le DFP ne peut commuter qu'entre les états 0 et $+1$ et ne va jamais à l'état -1. Si ω_1 est beaucoup plus élevée que ω_2, le DFP sera à l'état $+1$ durant la majeure partie du temps. Cependant, lorsque ω_1 est plus faible que ω_2, le DFP commute entre les états -1 et 0, et si ω_1 est beaucoup plus faible que ω_2, le DFP sera à l'état -1 pendant la majorité du temps. On conclue donc que lorsque la DPLL se trouve hors verrouillage, la moyenne du signal de sortie U_d du DFP varie en parfaite correspondance avec l'erreur de fréquence $\Delta\omega = \omega_1 - \omega_2$. Ceci conduit au terme, détecteur fréquence-phase.

Les MOSFETs équipant le DFP de la Figure II.3 sont choisis parmi ceux existant dans la bibliothèque de Pspice : BS170/ZTX pour le NMOS et BS250/PLP pour le PMOS.

Le courant drain de chaque transistor est donné par l'équation quadratique simplifiée suivante :

$$I_D = I_0\left(1 + \frac{V_{GS} - 2V_0}{V_0}\right)^2 \qquad \text{(II.5)}$$

Où les différentes tensions sont positives dans le cas du NMOS, et négatives pour le PMOS. V_0 désigne la tension de seuil à partir de laquelle le courant drain prend naissance, tandis que I_0 est la valeur du courant drain prise pour V_{GS} arrivée au double du seuil V_0.

Le modèle Pspice de chaque transistor indique sa tension-seuil, dont on lit : 1.8V pour NMOS et -2.4V pour PMOS. Ces valeurs sont confirmées lors du tracé de leurs caractéristiques de transconductance $I_D = f(V_{GS})$, Figure II.6.

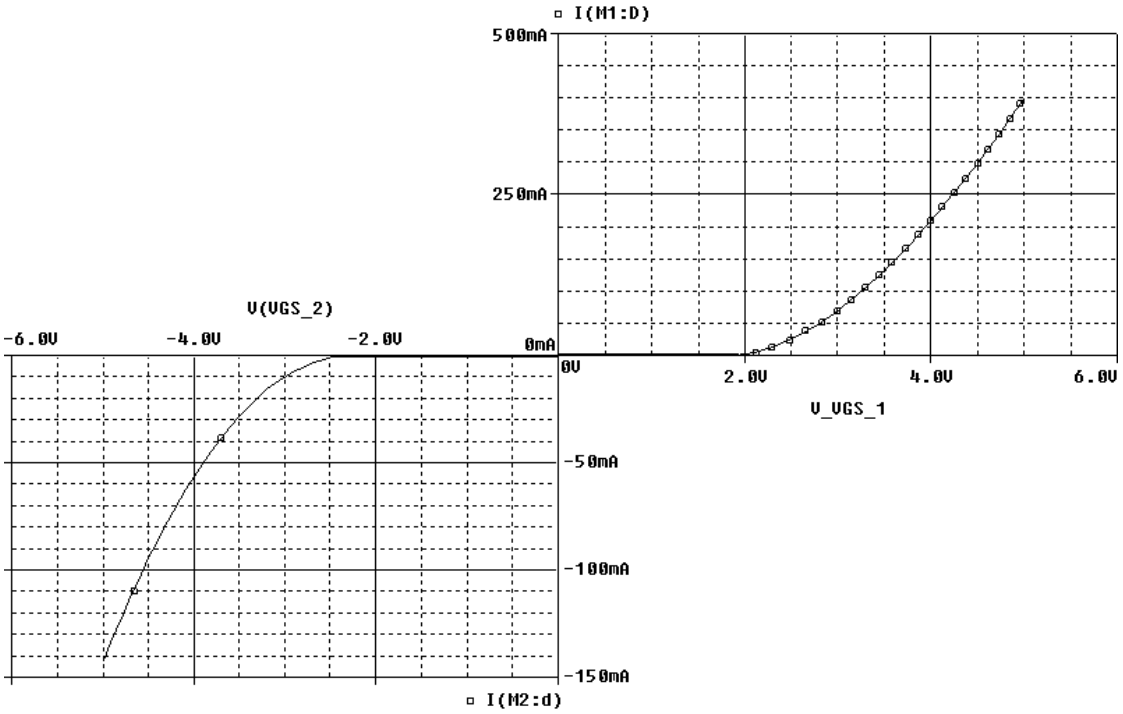


Figure II.6 : Caractéristiques de transconductance des MOSFETs complémentaires.

Suivi d'un filtre intégrateur passif du premier ordre, le DFP de la Figure II.3 lui est appliqué à ses entrées, des signaux carrés de période 1µs pour U_1 et 1.2µs pour U_2'. La fréquence d'entrée U_1 étant supérieure à celle du VCO, U_2', le tracé de la Figure II.7 montre un signal croissant issu du filtre.

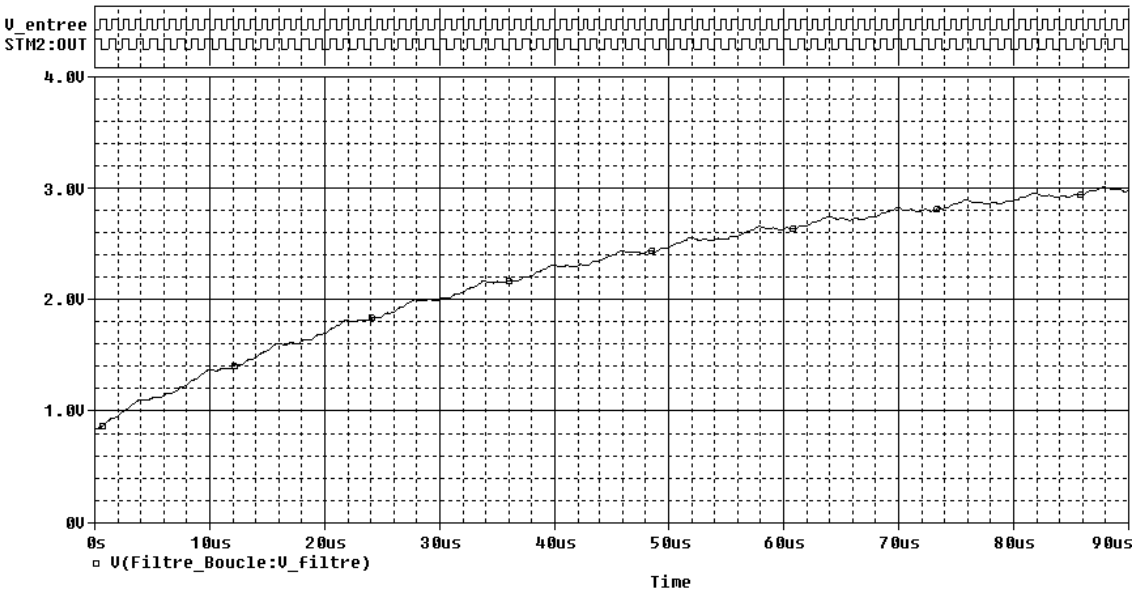


Figure II.7 : Réponse de la pompe de charge à une fréquence d'entrée supérieure à celle du VCO.

Par contre, lorsque les deux entrées du DFP sont permutées entre elles de façon à ce que la fréquence d'entrée U_1 soit inférieure à celle du VCO, U_2', le tracé de la Figure II.8 accuse un signal décroissant issu du filtre.

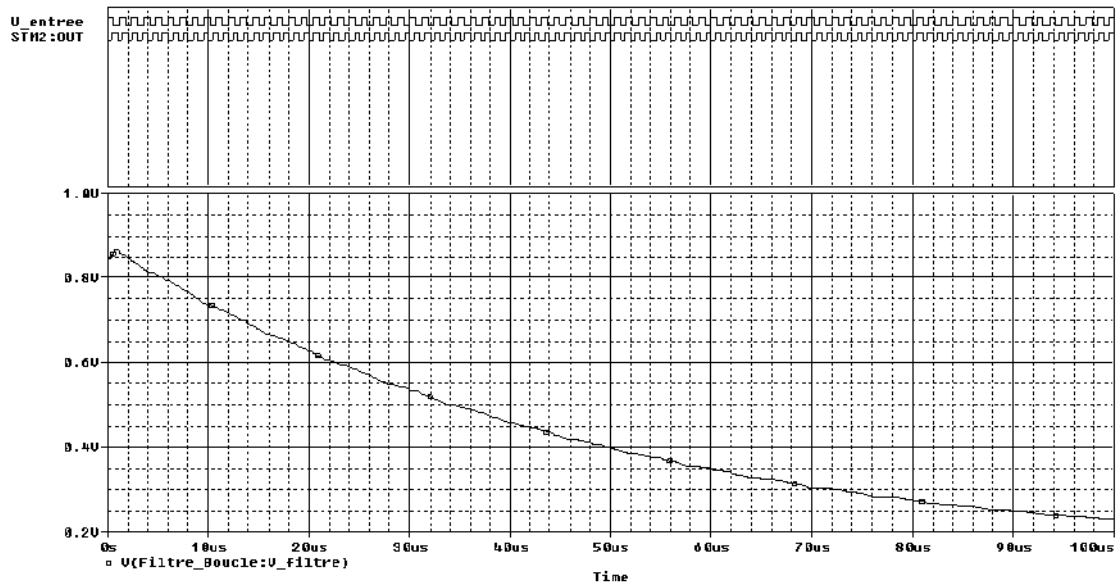


Figure II.8 : Réponse de la pompe de charge à une fréquence d'entrée inférieure à celle du VCO.

II.1.2 L'oscillateur contrôlé par tension (VCO)

Le VCO utilisé en Figure II.1, est la deuxième partie analogique de la DPLL après le filtre de boucle. Le VCO oscille à la fréquence angulaire ω_2 qui est déterminée par le signal de sortie U_f du filtre de boucle :

$$\omega_2(t) = \omega_0 + K_0 u_f(t) \tag{II.6}$$

Où ω_0 est la fréquence centrale du VCO et K_0 est la sensibilité ou gain du VCO, en $(SV)^{-1}$. Lorsque le modèle du VCO utilise la phase de sortie θ_2 et non la fréquence de sortie ω_2, la fonction de transfert du VCO utilisant la transformée de Laplace s'écrit :

$$\frac{\theta_2(P)}{u_f(P)} = \frac{K_0}{P} \tag{II.7}$$

Pour les signaux de phase, un VCO représente simplement un intégrateur, tandis que pour les signaux de fréquence, un détecteur de phase devient simplement un intégrateur.

La caractéristique du VCO est à déterminer à partir de la fréquence centrale ω_0 ou sa rangée qui doivent être connues. Dans la plupart des cas, le VCO fait partie de la DPLL qui est typiquement un circuit intégré. Le data-sheet indique la rangée $u_{fmin} \leq u_f \leq u_{fmax}$ utilisée pour la tension de contrôle (typiquement, 1 à 4V pour une alimentation de 5V). La sensibilité K_0 du VCO est calculée comme:

$$K_0 = \frac{\omega_{max} - \omega_{min}}{u_{fmax} - u_{fmin}}$$

Les composants externes du VCO peuvent aussi être déterminés, généralement, par indication du data-sheet.

Nous avons préféré procéder à une conception de notre VCO selon le schéma de principe [39] de la Figure II.9.

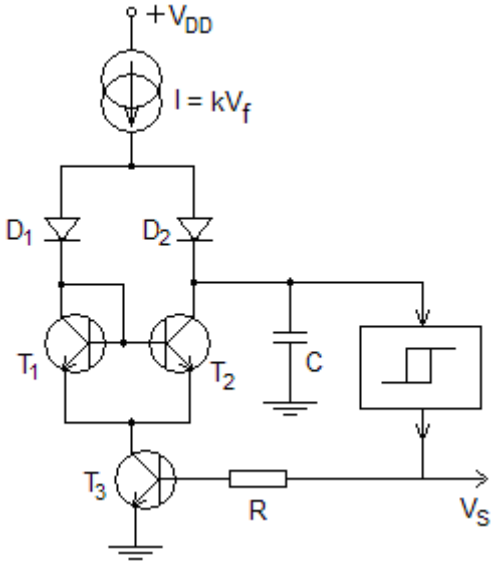


Figure II.9 : Schéma de principe du VCO.

Un relaxateur à rampes linéaires est articulé autour d'un condensateur C parcouru par un courant de charge-décharge I, issu d'une source de courant contrôlée par la tension de commande $V_c = V_f$ (fournie par le filtre de boucle de la DPLL). La charge s'effectue à travers la diode rapide D_2. La tension en rampe ascendante aux bornes de C est stoppée une fois arrivée au premier seuil du Trigger de Schmitt dont la sortie V_S passe à l'état haut, ce qui rend conducteur le transistor T_3. Ce dernier a comme charge active à son collecteur, un miroir de courant composé des transistors T_1 et T_2, qui deviennent à leur tour actionnés. La circulation du courant de la source contrôlée devient maintenant confiée à T_1 à travers la diode D_1, tandis que le courant miroité par T_2 et toujours égal à I, assure la décharge de C. La diode D_2 devient cependant, bloquée.

La tension en rampe descendante aux bornes de C est arrêtée lorsque le deuxième seuil du Trigger est atteint. Sa tension de sortie V_s passant à l'état bas, inhibe l'action du miroir de courant et laisse passer à nouveau le courant de charge à partir de la source contrôlée.

Les courants de charge et décharge de la capacité C étant égaux, un signal triangulaire évoluant dans la fenêtre d'hystérésis V_H du Trigger est récupéré aux bornes de celle-ci, avec une période :

$$V_H = \frac{1}{C}\int_0^{T/2} I\,dt = \frac{IT}{2c} \implies T = \frac{2C}{I}V_H \qquad \text{(II.8)}$$

Un premier Trigger à base de transistors bipolaires a été calculé, Figure II.10, et ce pour une fenêtre d'Hystérésis autour de 1V. Concernant le dimensionnement du Trigger, la valeur de la résistance R_{16} est initialisée à 3 kΩ puis ajustée par simulation à 1.5 kΩ pour des courants aux collecteurs, $I_{11} = 0.5\text{mA}$ et $I_{12} = 1\text{mA}$.

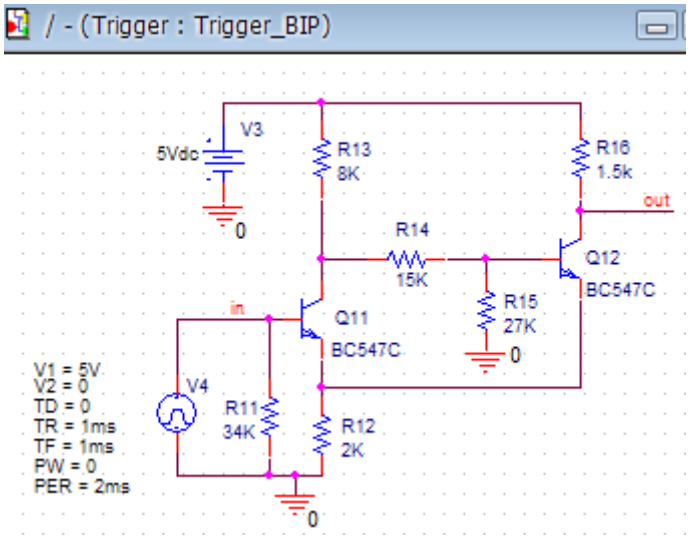


Figure II.10 : Trigger de Schmitt à transistors bipolaires.

Les seuils mesurés du Trigger sont 1.32V et 2.52V, soit une fenêtre d'hystérésis de 1.2V. La sortie du trigger bascule cependant entre 3.5V et 5V, Figure II.11.

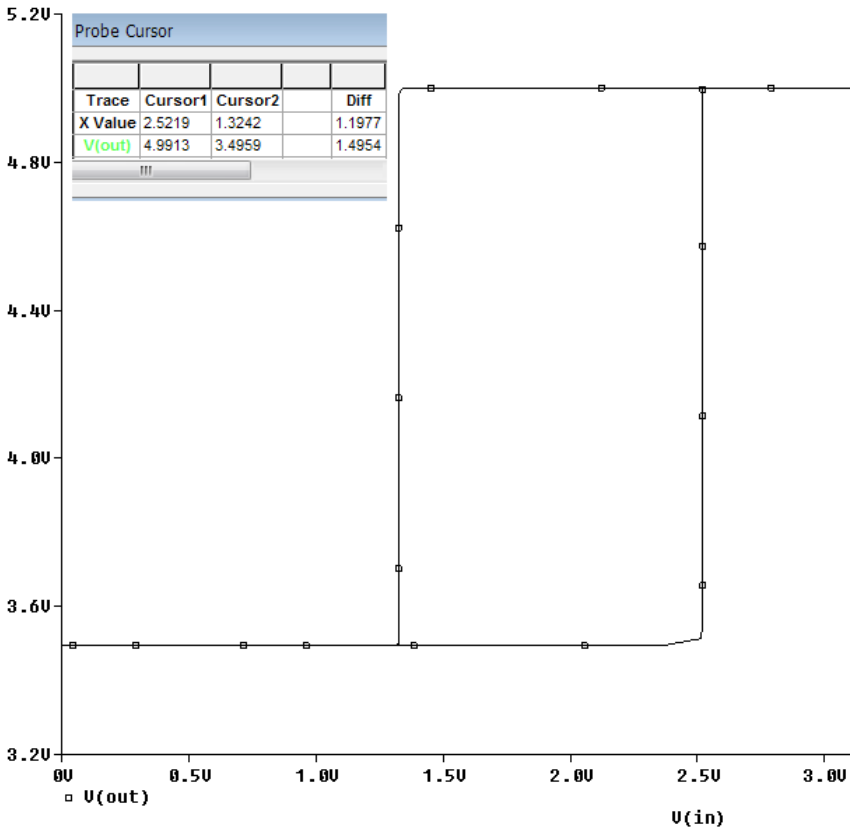


Figure II.11 : Fenêtre d'hystérésis du Trigger bipolaire.

Nous avons songé à utiliser un Trigger intégré de la famille 7414, dont les seuils mesurés sont : 0.9V et 1.7V, soit une fenêtre d'hystérésis V_H = 0.8V. La sortie inversée du trigger bascule cependant entre 0.09V et 3.1V, Figure II.12.

Ainsi, pour une fenêtre d'hystérésis équivalente à celle du trigger bipolaire, le saut de tension est double. Ce trigger est retenu pour notre application et nécessite un inverseur associé.

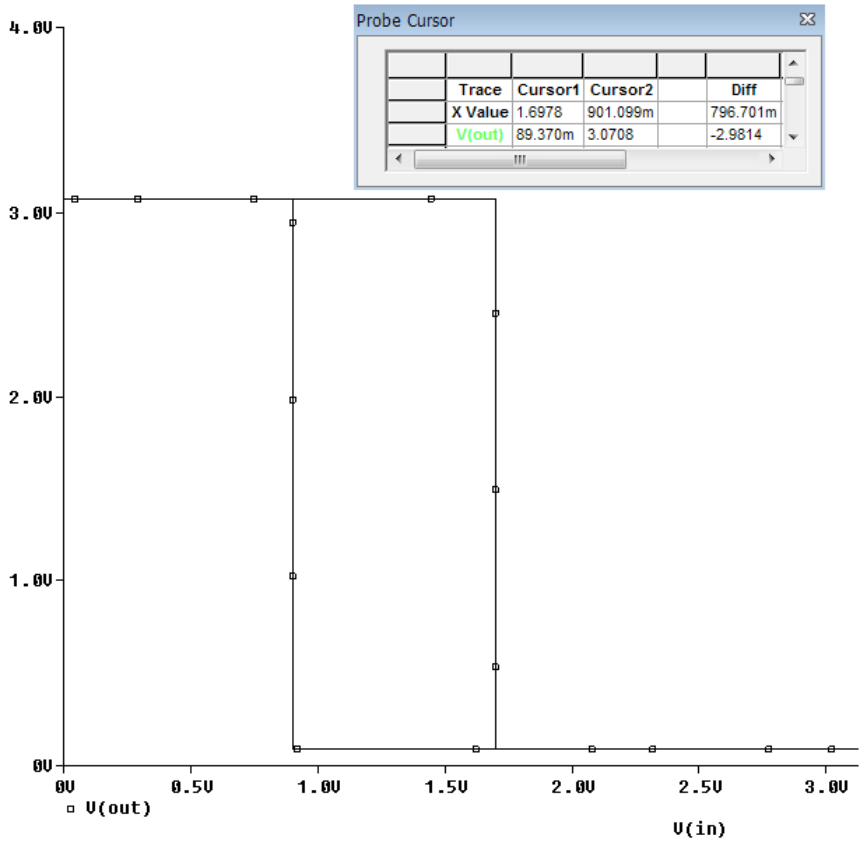


Figure II.12 : Fenêtre d'hystérésis du Trigger de Schmitt intégré de type 7414.

La source de courant contrôlée par la tension de commande V_C issue du filtre de boucle de la DPLL dans la Figure II.9, est réalisée selon le modèle de la Figure II.13 [40].

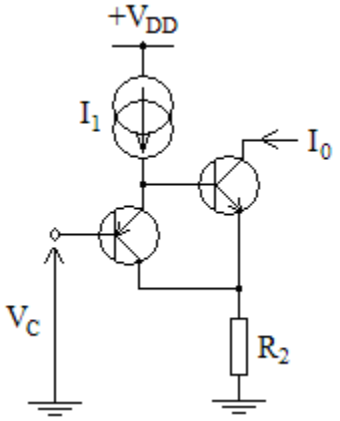


Figure II.13 : Modèle de la source de courant I_0 contrôlé par tension V_C.

Dans ce modèle, il y a indépendance du courant de sortie envers V_{BE} par compensation mutuelle des V_{BE} des deux transistors complémentaires et appariés, BC547C et BC557C :

$$I_0 = \frac{V_C}{R_2} - I_1 \qquad \text{(II.9)}$$

Le contrôle du courant I_0 se fait directement par la tension de commande V_C, sauf que ce courant est appelé par la source et non fourni.

Les transistors étant polarisés par la source de courant constant I_1, celle-ci est réalisée par un simple miroir de courant à transistors PNP [41], Figure II.14.

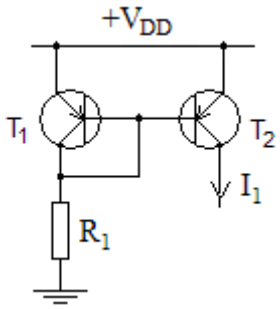


Figure II.14 : Miroir de courant simple.

Le courant débité par ce miroir sert à polariser la source de courant de la Figure II.13, avec une intensité :

$$I_1 = \frac{V_{DD} - V_{BE}}{R_1} \qquad \text{(II.10)}$$

Un autre miroir semblable à celui de la Figure II.14 et dont R_1 est substituée par le circuit de la Figure II.13, permet de rétablir le sens de circulation au courant I_0 dans la direction sortante de la source de courant contrôlée par tension, comme en Figure II.9. La Figure II.15 montre l'association de ces circuits.

En choisissant arbitrairement une valeur de $40\mu A$ pour ce courant, l'expression (II.10) permet sous une tension d'alimentation de $5V$, de tirer la valeur de la résistance R_1=$110k\Omega$ qu'on fixe à $100k\Omega$.

Pour une mi-tension de commande $V_C = 2.5V$ appliquée au circuit de la Figure II.13, celle-ci assure un courant image selon l'expression (II.9) avec la valeur de $0.5mA$ pour une résistance R_2=$4.7k\Omega$.

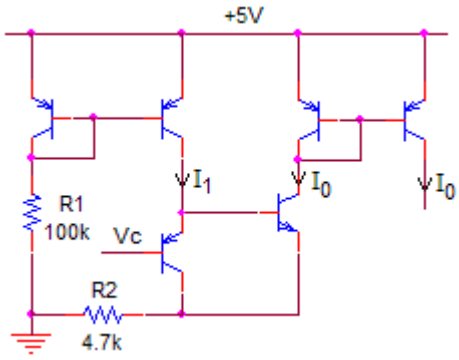


Figure II.15 : La source de courant contrôlée par tension.

Dans Pspice, cette source a pu alimenter le reste du circuit de la Figure II.9 suivant le montage de la Figure II.16. La valeur de 100pF est attribuée au condensateur C_1 afin que la fréquence maximale de relaxation du montage atteint le MHz. Un MOSFET à source commune joue le rôle d'adaptateur inverseur servant à tamponner la tension aux bornes du condensateur C_1 grâce à sa forte impédance d'entrée. Ses résistances de drain et de source sont choisies pour assurer un gain en tension de 10, jugé très suffisant pour que les rampes de charge et décharge de C_1 traversent la fenêtre d'hystérésis du Trigger.

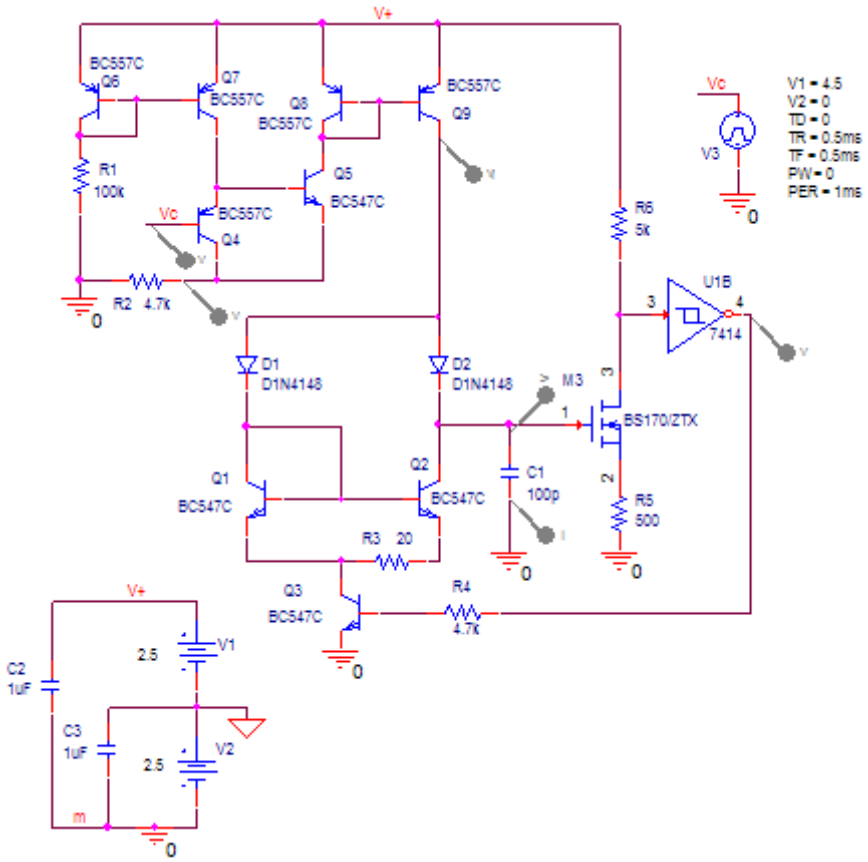


Figure II.16 : Circuit VCO.

Pour une tension de commande constante $V_C = 2.5$V, un signal de forme rectangulaire est relevé en sortie du Trigger, Figure II.17. Cette forme est due aux durées inégales

des courants de charge et décharge de C_1, qui sont débités par la source contrôlée autour des valeurs $I_0 = I_{C_1} = \pm 0.5\text{mA}$.

D'autre part, le signal aux bornes de C_1 présente une chute de tension conséquente avant chaque rampe descendante et qui coïncide avec les fronts montants du signal de sortie du Trigger, Figure II.17. Cela s'explique par le fait qu'à chaque front montant du Trigger, le transistor Q_3 se sature tout en mettant en service le miroir de courant Q_1/Q_2. Le front descendant résultant au collecteur de Q_1, se répercute à l'anode de la diode D_2, la rendant bloquée sur le champ. Sa capacité de jonction de 4pF apparaissant à cet état de blocage, laisse passer une partie du front descendant émanent de la tension collecteur de Q_1 qui se superpose à la tension aux bornes de C_1 causant ainsi sa chute quasi instantanée avant le régime de décharge. Le courant de décharge de C_1 se voit alors accompagné de pics de courant mesurés à -3mA pour chaque front montant du signal de sortie du Trigger, Figure II.17.

Ces phénomènes se réduisent notablement en introduisant une faible résistance R_3 à l'émetteur du transistor Q_2. Pour une valeur expérimentale $R_3 = 20\Omega$, les pics de courant s'atténuent à -2.2mA avec des intensités de charge-décharge qui deviennent de durées presque égales, ramenant ainsi une forme quasi triangulaire aux bornes de C_1 (faible chute de tension au début de chaque période de décharge) avec un signal du Trigger presque aussi carré aux fréquences élevées de la plage couverte. Tandis qu'aux fréquences les plus basses, leur rapport cyclique reste de 25%.

La fréquence de relaxation théorique prévue par l'expression (II.8) est approximative et vaut 3MHz, tandis que la simulation en donne une valeur de moitié, 1.4 MHz.

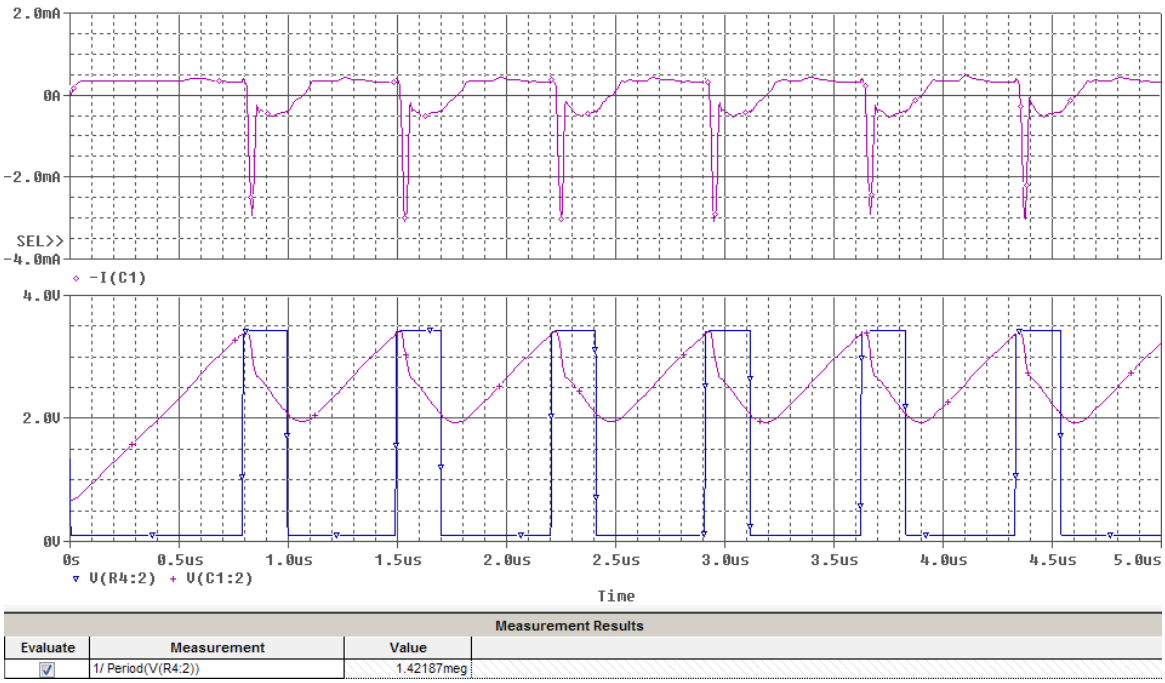
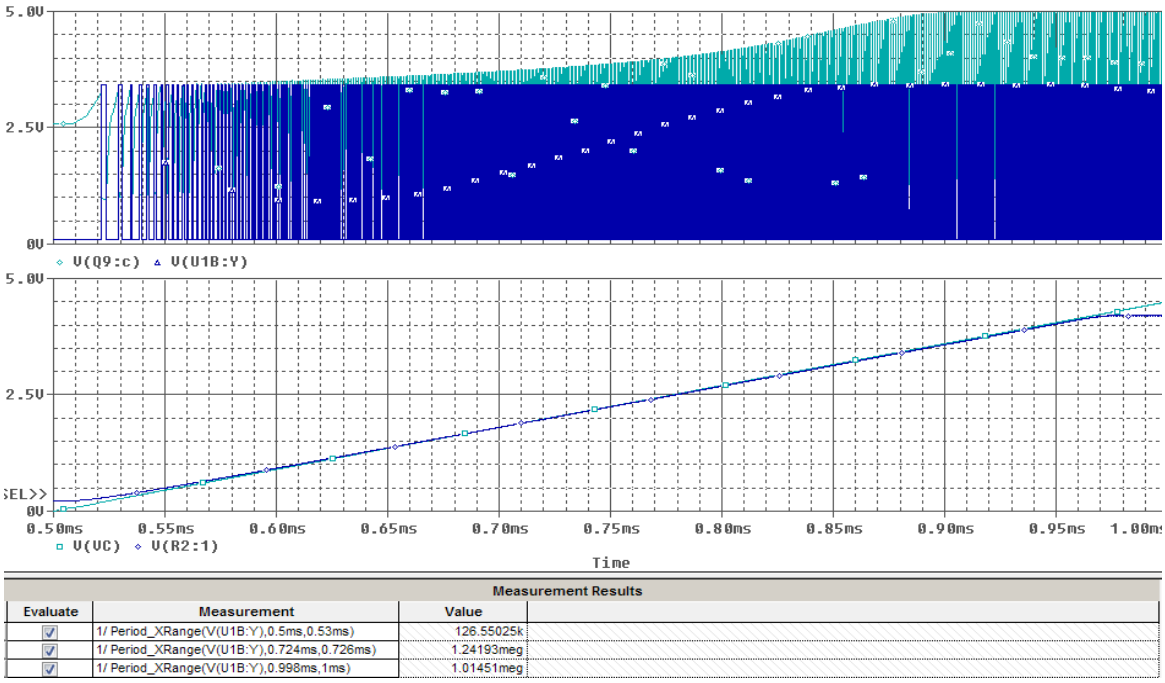


Figure II.17 : Courant dans la capacité C_1 (en haut) et tensions en sortie du Trigger et aux bornes de C_1 (en bas).

Utilisant cette fois-ci, une tension de commande assurée par une source indépendante délivrant des rampes linéaires en tension et allant de 0 à 4.5V, nous nous sommes intéressé aux caractéristiques techniques du circuit de la Figure II.16. La Figure II.18 rassemble certaines caractéristiques.



Measurement Results

Evaluate	Measurement	Value
☑	1/ Period_XRange(V(U1B:Y),0.5ms,0.53ms)	126.55025k
☑	1/ Period_XRange(V(U1B:Y),0.724ms,0.726ms)	1.24193meg
☑	1/ Period_XRange(V(U1B:Y),0.998ms,1ms)	1.01451meg

Figure II.18 : Tensions de sortie et du collecteur de Q_9 (en haut), tensions de contrôle et son image (en bas).

La tension de commande d'entrée V_C ainsi que la tension $V(R_2)$ image du courant contrôlé et circulant dans R_2, sont représentées dans le graphe du bas de la Figure II.18. Les deux droites issues de ces deux tensions ne se superposent pas aux extrémités. La linéarité de la commande s'étend de $V_C = 0.45$V à 4.23V et lui correspond $V(R_2) = 0.498$V à 4.18V, avec un écart relatif $\frac{|V_C - V(R_2)|}{V_C}$ d'environ 10% aux basses tensions et 1% aux hautes tensions. La courbure de la droite de commande $V(R_2)$ se manifestant au-delà de 4.18V, est due à l'action simultanée de saturation du transistor Q_7 et blocage du transistor Q_4, Figure II.16.

La variation en fréquence du signal de sortie du Trigger (U1B) ainsi que l'évolution de la tension au collecteur de Q_9 représentant le dernier étage du miroir de sortie de la source de courant contrôlée par tension, figurent dans le graphe du haut de la Figure II.18. La wobulation causée par la tension de commande, couvre une plage fréquentielle non uniforme du signal de sortie du Trigger. Pour une rampe de commande évoluant de 0.5ms à 1ms, la plage de fréquence couverte s'étend sur une décade, de 126kHz à 1.24MHz, mais la fréquence maximale est atteinte à mi-course à l'instant 724µs. A 1ms, la fréquence redescend à 1MHz, Figure II.18.

Une autre constatation se reporte à la saturation du transistor Q_9, qui se manifeste à 0.89ms lorsque la tension collecteur du PNP arrive aux 5V de l'alimentation pour une tension de commande de $V_C = 3.5$V ou $V(R_2) = 3.49$V.

La Figure II.19 donne l'allure du courant fourni par la source contrôlée par tension, par le biais de son transistor Q_9. A part les impulsions parasites accompagnant ce courant, on distingue l'accroissement régulier de son amplitude (élargissement de la bande foncée) en réponse à la rampe de commande $V(R_2)$ de la Figure II.18. Cet accroissement proportionnel cesse dès que Q_9 se sature.

La décroissance linéaire de la valeur moyenne du courant de source reflète celle de la ddp, $5 - V(Q9)$, entre émetteur et collecteur du transistor Q9 et dont le graphe du bas de la Figure II.19 ne représente qu'une partie de sa tension crête.

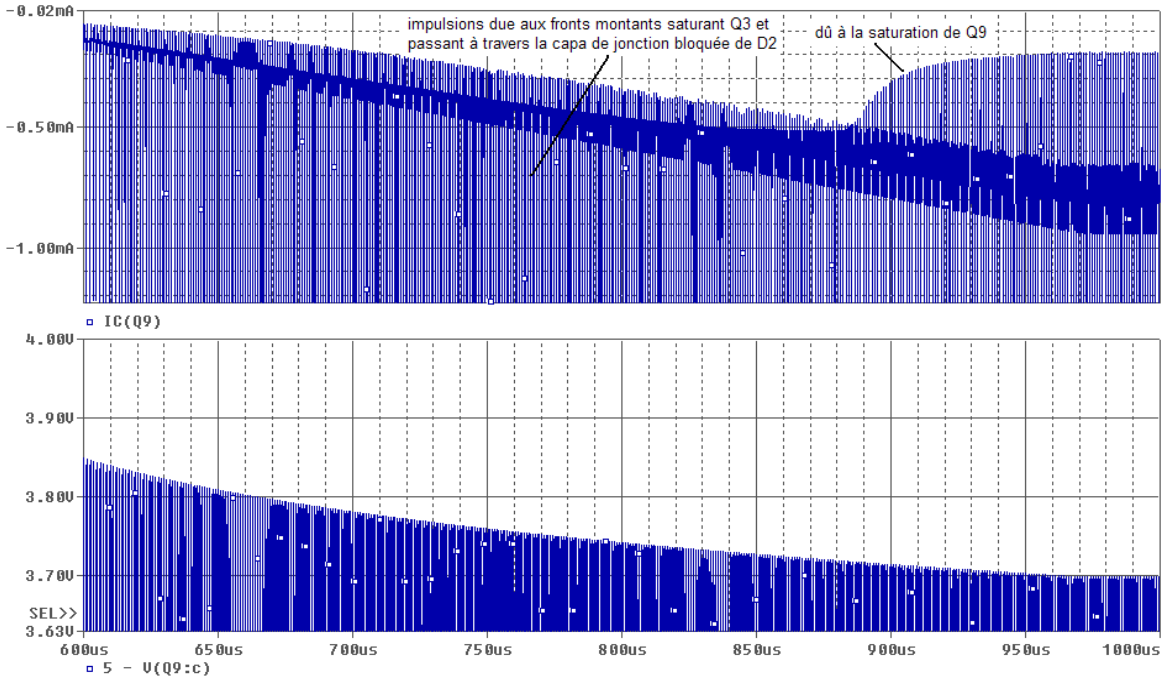


Figure II.19 : Courant collecteur du transistor Q_9 (en haut) et sa tension-crête V_{CE} (en bas).

Dans un domaine temporel court, entre 0.2ms et 0.3ms, nous utilisons un profil de simulation de Pspice basé sur un balayage de la tension de commande (issue du filtre dans une PLL), $V_f = 0.1\text{V}$ à 4.5V par pas de 0.1V.

Au niveau de la fenêtre des mesures (fenêtre Probe), on clique sur l'icône "Performance Analysis" pour changer l'axe temporel par celui de V_f. On clique ensuite sur l'icône "Add Trace" afin de visualiser la fréquence du signal de sortie du Trigger (U1B). Nous obtenons ainsi en Figure II.20, la caractéristique du gain (ou sensibilité K_0) du VCO de la Figure II.16.

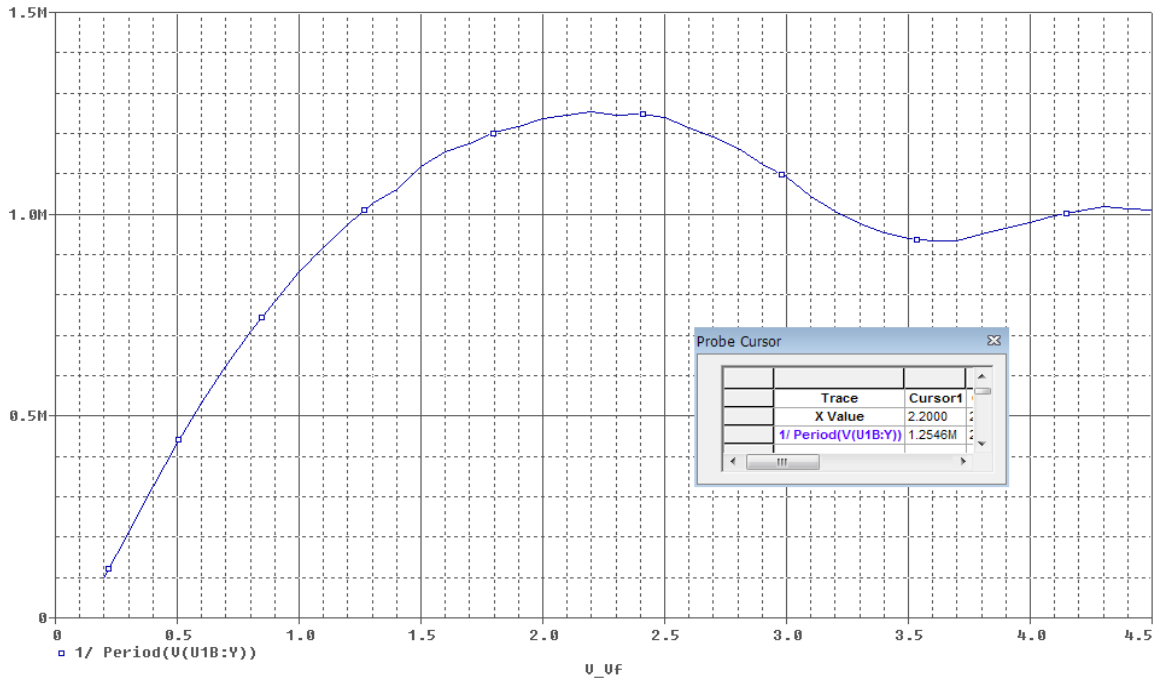


Figure II.20 : Caractéristique tension-fréquence du VCO.

La caractéristique du VCO est quasi linéaire sur une faible plage de V_f ne dépassant pas 1.5V. Un maximum de fréquence est repéré pour $V_f = 2.2$V, et vaut 1.25MHz. Cette valeur est pratiquement la même que celle mesurée dans la Figure II.18 à l'instant 724µs. La non linéarité frappante de la caractéristique du VCO en Figure II.20 est sûrement due à une mauvaise impédance de sortie de la source de courant contrôlé, utilisée.

Afin qu'on puisse remédier aux problèmes soulevés par le VCO de la Figure II.16 à travers l'étude faites sur les mesures des figures II.18 et II.19, nous avons isolé la source de courant contrôlée par tension pour la charger simplement par une résistance $R_3 = 1\text{k}\Omega$, Figure II.15.

La rampe de commande ($V_C = 0$ à 4.5V) de la Figure II.18 restant fonctionnelle, la Figure II.21 illustre les rampes du courant $I(R_2)$ contrôlé par la tension de commande d'entrée (courant de consigne) et du courant $I(R_3)$ généré par la source contrôlée (courant de charge). Présentant des pentes légèrement différentes, les deux rampes de courant se coupent à l'instant 724.9µs. A cet instant précis, l'impédance de sortie de notre source de courant contrôlé arrive à sa valeur maximale de 4.57MΩ, voir graphe du bas de la Figure II.21. Cet impédance interne Zs est évaluée comme le rapport entre la ddp aux bornes de la source (donc, aux bornes de Zs) et la différence entre le courant de consigne et celui de charge (le reste parcourt nécessairement Zs). L'énormité de la valeur maximale de Zs revient au très faible écart constaté au voisinage du point d'intersection des deux rampes de courant, $I(R_2) = 429.76\mu$A et $I(R_3) = 430.76\mu$A, Figure II.21. De part et d'autre de ce maximum, Zs s'effondre entre 103k et 88k aux deux extrémités de la courbe. La constance de cette impédance est donc très altérée.

En revenant aux mesures de la Figure II.18, nous remarquons que l'instant d'apparition de la fréquence maximale de relaxation est pratiquement le même que celui où Zs développe sa valeur maximale, aux environs de $724\mu s$.

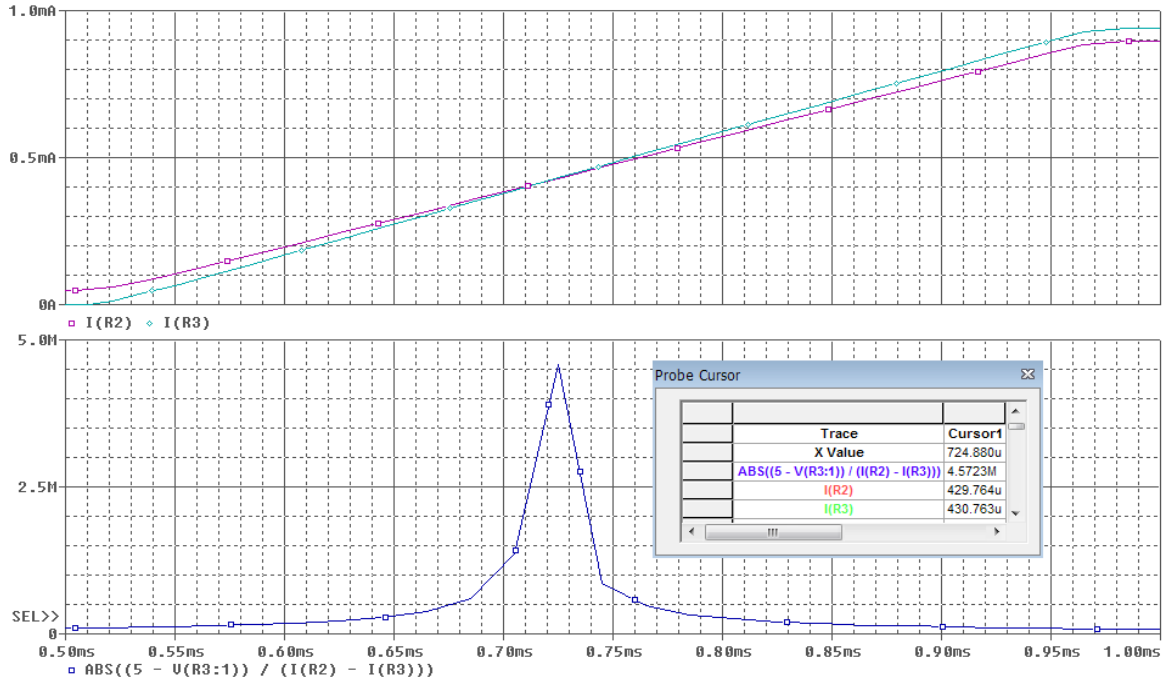


Figure II.21 : Courants de consigne et de charge (en haut), impédance de sortie de la source contrôlée (en bas).

En définitive, la solution consiste à améliorer l'impédance de sortie de la source contrôlée tout en évitant l'état de saturation de son étage de sortie. Cet état de saturation, revient au nombre de jonctions des semiconducteurs (transistors ou diodes) mises dans une même branche sous $5V$ seulement (3 en charge, et 5 en décharge). L'élévation de la tension d'alimentation pourra remédier à ce problème mais dans notre cas, on se contente d'une alimentation monotension de $5V$ sous laquelle, la plage de fréquence doit être couverte sans saturation de Q_9. Dans ce contexte, on mesure depuis la Figure II.18 le seuil de saturation du circuit de la Figure II.16. Une tension de commande $V(R_2) = 3.17V$ accuse un $V_{CEsat} = 0.4V$ du transistor Q_9 (cette valeur est courante chez les bipolaires).

Le courant de polarisation I_P de la source est donné par l'expression (II.10) qui devient :

$$I_P = \frac{4.4}{100k} = 44\mu A \qquad \text{(II.11)}$$

De l'expression (II.9), il devient aisé de tirer la valeur du courant de source I_{0sat} pour laquelle Q_9 entre en saturation :

$$I_{0sat} = \frac{3.17}{4.7k} - 44 = 630\mu A \qquad \text{(II.12)}$$

En fixant le courant de source minimal un peu plus grand que celui de polarisation, et en assumant la totalité de la plage de commande disponible, $V_C = 0.1\text{V}$ à 4.5V, les courants extrêmes de la source deviennent :

$$\begin{cases} I_{0min} = \dfrac{0.1}{R_2} - I_P = 1.5I_P \Longrightarrow \dfrac{1}{R_2} = 25I_P & \text{(II.13)} \\ I_{0max} = \dfrac{4.5}{R_2} - I_P & \text{(II.14)} \end{cases}$$

En substituant l'expression (II.13) dans celle de (II.14), on obtient :

$$I_{0max} = 111I_P \tag{II.15}$$

Le courant de source étant celui chargeant et déchargeant C_1, l'expression (II.8) permet de tirer un même rapport entre les courants extrêmes de source et celui des fréquences extrêmes atteintes :

$$\Delta = \frac{I_{0max}}{I_{0min}} = \frac{f_{max}}{f_{min}} \tag{II.16}$$

Le dimensionnement de la nouvelle source contrôlée a été confié à un programme en matlab utilisant l'expression (II.10) et les expressions établies de (II.12) à (II.16). Les valeurs estimées pour C_1=100pF fixée et courant de source maximal ne dépassant pas $630\mu\text{A}$, sont :

- courant de polarisation $I_P = 5.7\ \mu\text{A}$,
- résistance de polarisation $R_1 = 775\ \text{k}\Omega$,
- résistance de conversion (V en I) $R_2 = 7\ \text{k}\Omega$,
- courant de source minimal $I_{min} = 8.5\ \mu\text{A}$,
- delta = rapport de courant = rapport de fréquence =74,
- fréquence minimale du VCO $f_{min} = 53$ kHz.

Pspice confirme le non dépassement du seuil de saturation, $V_{CE} \geq 0.4\text{V}$, le long des fréquences balayées. Mais malgré que la caractéristique de la Figure II.20 est devenue monotone, sa courbure restait prononcée. Les valeurs des résistances ont été encore ajustées par tâtonnement jusqu'à obtention de la meilleure linéarité possible de la caractéristique du VCO (voir plus loin, Figure II.27), sans pour autant rendre saturé le transistor Q_9 durant toute la plage balayée. Cette amélioration a été obtenue au prix d'une baisse de la fréquence maximale atteinte (de 1.25MHz à 0.97MHz en Figure II.24). Les nouvelles valeurs sont donc : $R_1 = 470\text{k}$, $R_2 = 15\text{k}$.

Dans le souci d'améliorer aussi l'impédance de sortie de la source contrôlée qui vient d'être calculée et ajustée, le simple miroir de courant constituant le dernier étage du circuit de la Figure II.15, est remplacé par une source de Wilson [41]. Il s'agit d'un miroir de courant parmi les plus fiables à cause de la compensation mutuelle de ses courants de base et la délivrance d'un courant de source sous une meilleure impédance de sortie, Figure II.22.

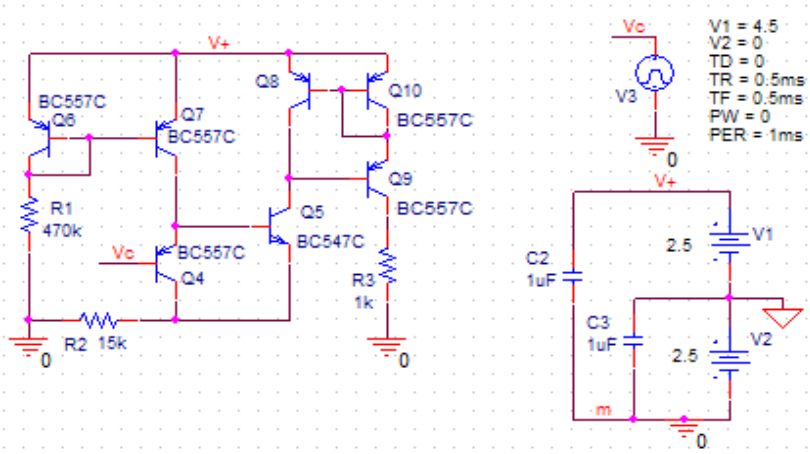


Figure II.22 : Source de courant contrôlée par tension, améliorée.

Du fait de l'absence de croisement des deux rampes de courant de consigne $I(R_2)$ et de charge $I(R_3)$, la Figure II.23 montre une courbe monotone de Zs, sans maxima.
On mesure sur la partie linéaire de la courbe :

$Zs = 485\text{k}\Omega$ à $t = 0.52\text{ms}$, correspondant à $I(R_2) = 15\mu\text{A}$ et $I(R_3) = 5\mu\text{A}$,
$Zs = 337\text{k}\Omega$ à $t = 0.9\text{ms}$, correspondant à $I(R_2) = 236\mu\text{A}$ et $I(R_3) = 222\mu\text{A}$.

soit $\Delta Zs = 148\text{k}\Omega$, alors que celle constatée lors des premières valeurs de R_1 et R_2 avant retouche, est $\Delta Zs = 462\text{k}\Omega$ sur la même plage de fréquence balayée. Ceci, reflète une constance de l'impédance de sortie améliorée de $(462 - 148) / 462 = 68\%$.

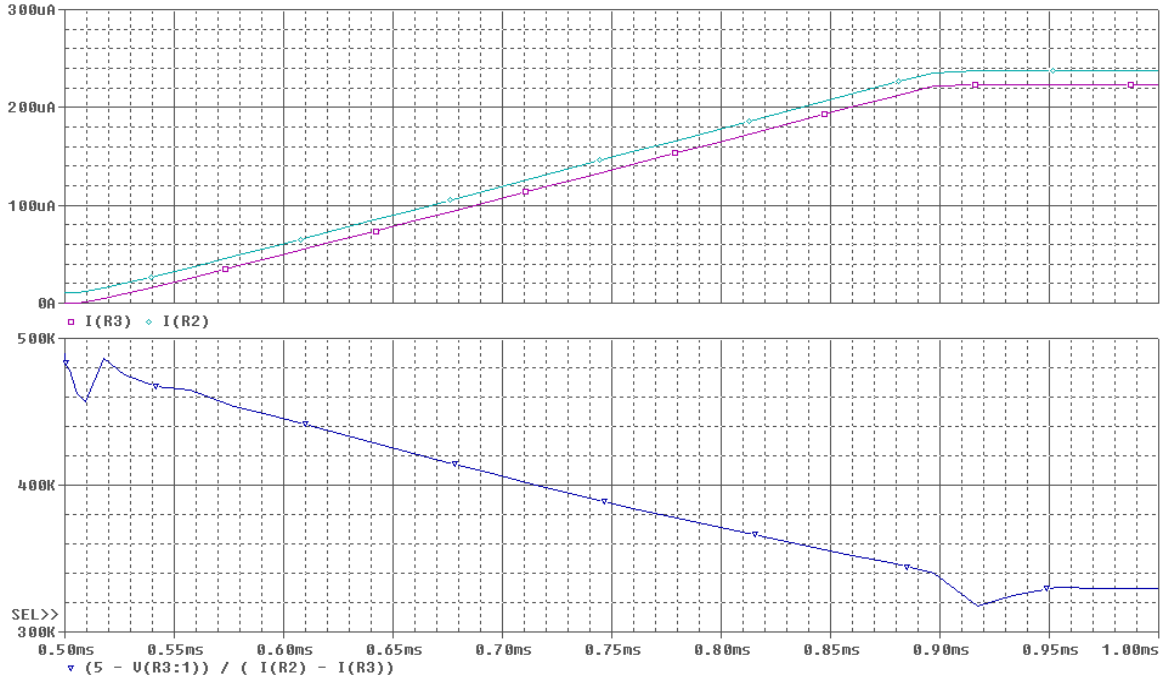


Figure II.23 : Courants de consigne et de charge (en haut), impédance de sortie de la source contrôlée (en bas).

Dans le but de poursuivre d'autres mesures intéressantes liées à notre nouvelle source de courant contrôlée par tension, la charge résistive R_3 de la Figure II.22 est remplacée par la véritable charge que constitue le VCO, Figure II.24. La commande linéaire en tension se fait toujours par une source indépendante de tension en rampe.

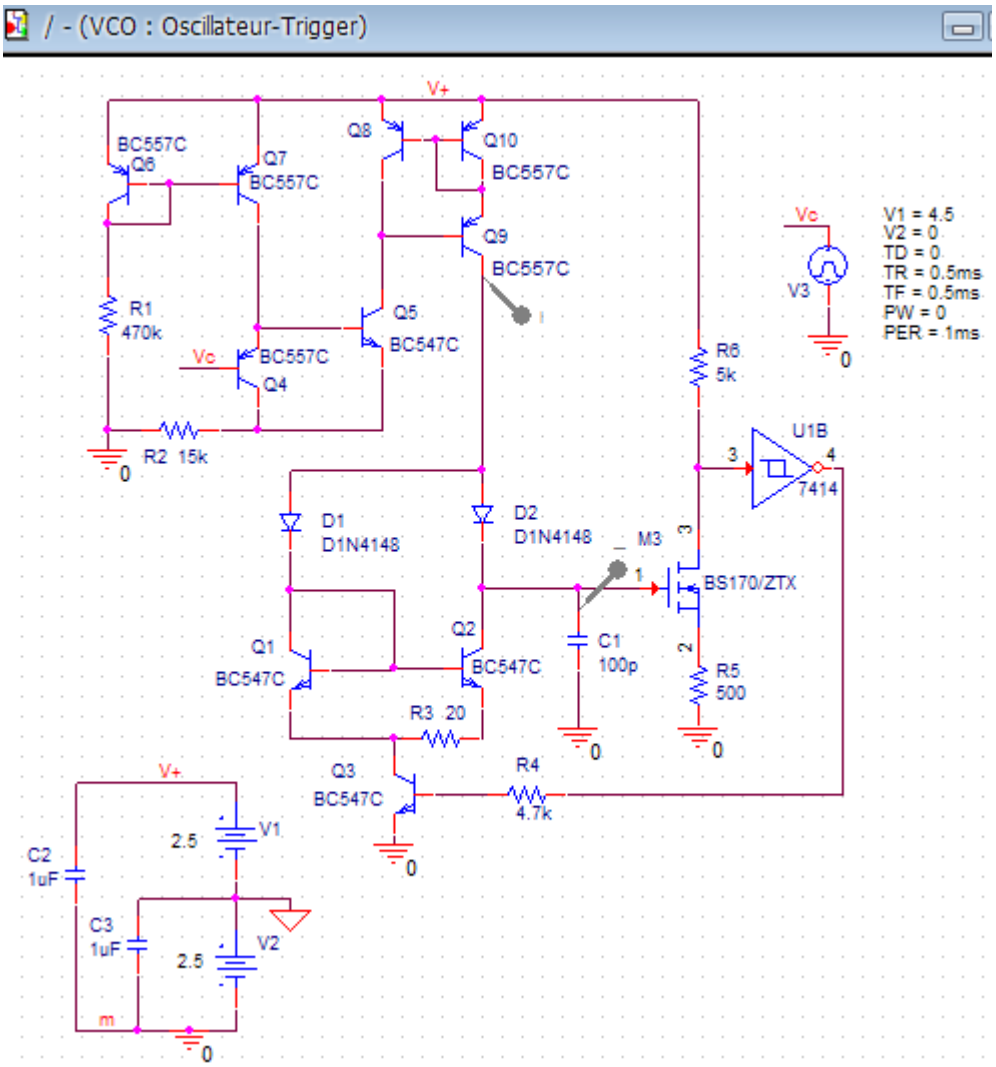


Figure II.24 : VCO finalisé.

Sur le plan de la vérification d'une éventuelle saturation du dernier étage de la source contrôlée (précisément le transistor Q_9), la Figure II.25 montre qu'à partir de l'instant 0.9ms, l'état du minimum de V_{CE} est supérieure à 0.7V. La clause $V_{CE} \geq 0.4$V, est ainsi largement satisfaite. L'instant 0.9ms marque au fait, le début de saturation du transistor Q_7 et blocage du transistor Q_4, Figure II.24, qui font que la progression linéaire de la tension $V(R_2)$ image du courant contrôlé de la source, ne se poursuit plus, Figure II.25.

D'autre part, les mesures portées au bas de la Figure II.25 sont prises aux mêmes instants que celles de la Figure II.18, et reviennent à la fréquence délivrée par le VCO une fois piloté par notre nouvelle source contrôlée. Ces mesures témoignent d'une

monotonie (sans maxima) de la fréquence du VCO sous une tension de commande linéaire, $V(R_2)$ de 0.26V à 3.5V pour $t = 0.52$ms à 0.9ms.

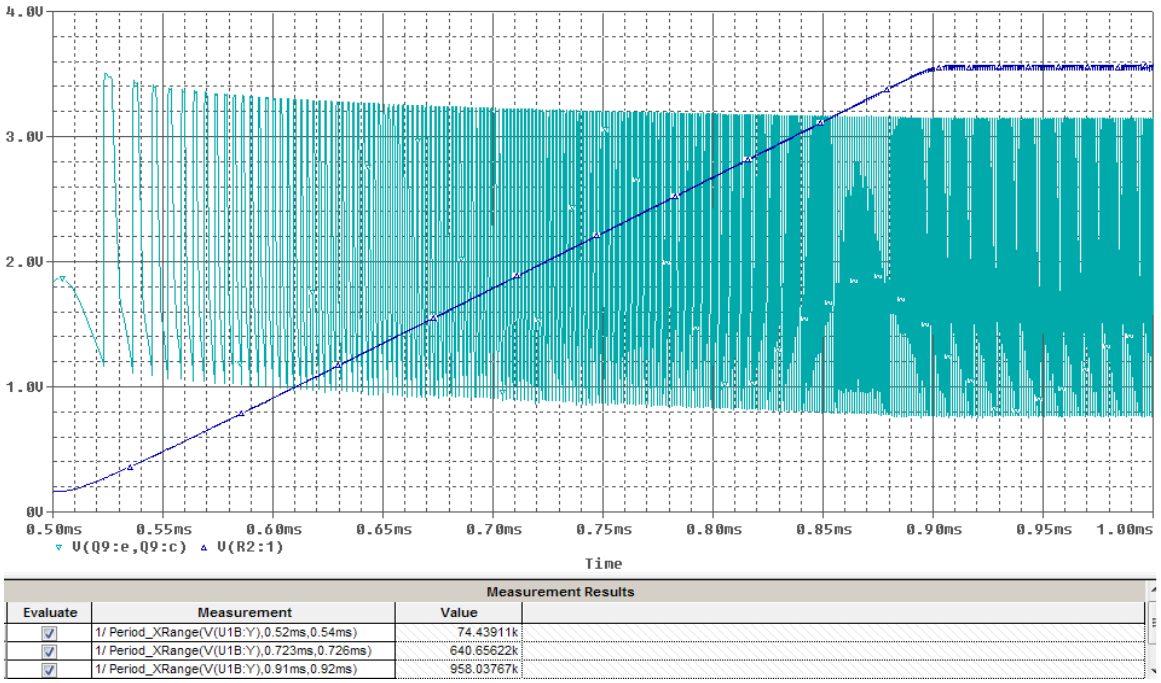


Figure II.25 : Tension V(R_2) image du courant contrôlé et tension V_{CE} du transistor Q_9.

La Figure II.26 réunit les deux courants issus de la source contrôlée par tension au niveau de son transistor Q_9, $I_C(Q9)$, et celui circulant à travers le condensateur C_1, $I(C_1)$, Figure II.24.

L'augmentation des intensités des deux courants (bandes foncées) se fait en fonction de la rampe ascendante de la tension de commande $V(R_2)$ de la Figure II.25. La stagnation de $V(R_2)$ à partir de 0.9ms, amène celle des deux courants.

Cependant, la remarque importante s'attache à la disparition de la zone de saturation dans le tracé de $I_C(Q9)$, en comparaison avec celui de la Figure II.19. Les impulsions parasites passant par la diode bloquée D_2, y demeurent. Les courants de charge-décharge de C_1 sont bien balancés positivement et négativement autour de zéro.

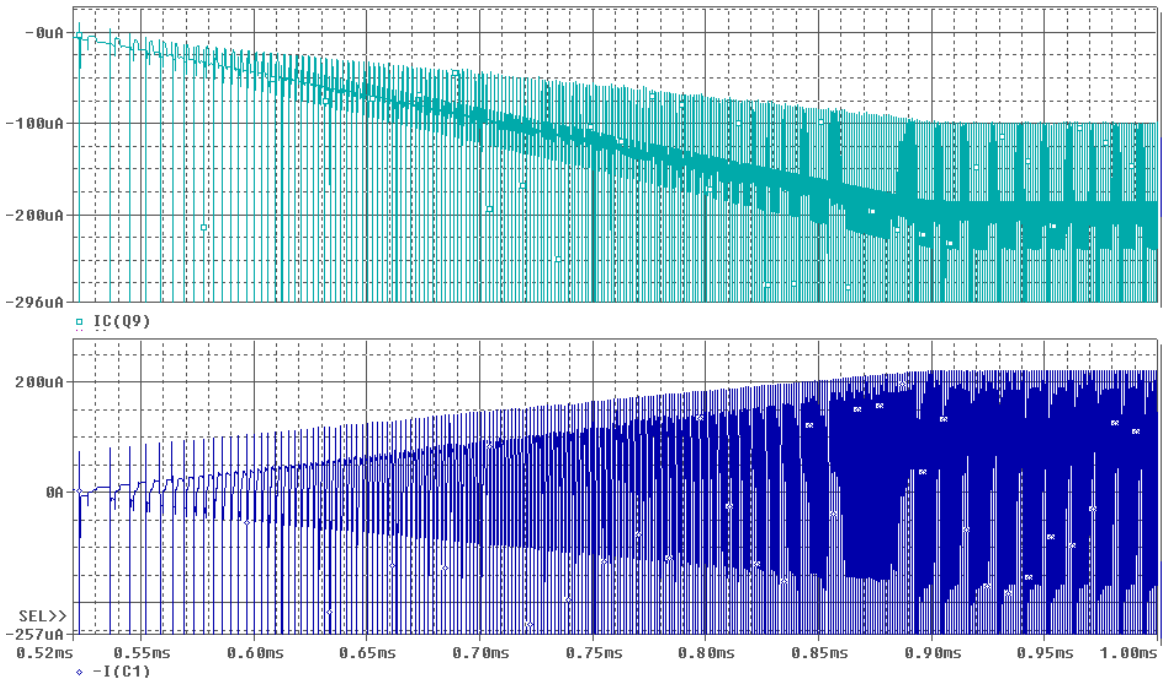


Figure II.26 : Courant collecteur du transistor Q_9 (en haut) et courant dans la capacité C_1 (en bas).

La caractéristique du VCO en Figure II.27 est presque linéaire sur la plage de commande, $V_f = 0.1\text{V}$ à 3.6V, pour laquelle la fréquence varie de 50kHz à 967kHz (plus d'une décade).

La pente moyenne de la caractéristique se mesure par la tangente au point milieu de la courbe ($V_f = 1.8\text{V}$, $f = 585.6$ kHz).

Soit, une sensibilité du VCO : $K_0 = 325$ kHz/V $= 2.10^6$ rd/Vs.

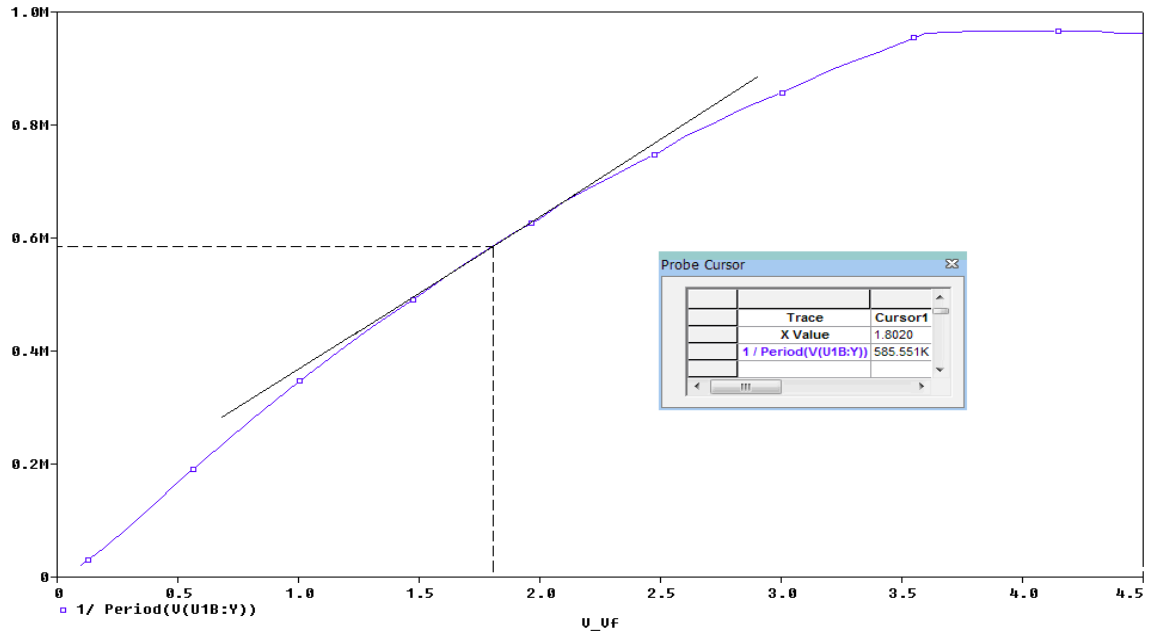


Figure II.27 : Caractéristique tension-fréquence du VCO amélioré.

II.1.3 Le filtre de boucle

En général, la sortie du comparateur de phase contient des termes indésirables souvent situés dans la partie HF du spectre relativement au terme utile de la détection. Ces composantes inutiles sont filtrées par le filtre de boucle, qui est donc nécessairement de type passe-bas. Dans la plupart des conceptions de PLL un filtre passe-bas du premier ordre est utilisé, dont la Figure II.28 donne la version fréquemment rencontrée [1] et [39]. Ce filtre est à avance-retard de phase possédant un pôle et un zéro. Un réseau à avance de phase dont l'action provient du zéro (numérateur de la fonction de transfert du filtre), est donc combiné avec un autre à retard de phase dont l'action provient du pôle (dénominateur de la fonction de transfert du filtre).

Le filtre passif, Figure II.28, possède la fonction de transfert :

$$F(P) = \frac{1 + \tau_2 P}{1 + (\tau_1 + \tau_2)P} \tag{II.17}$$

Où $\tau_1 = R_1 C$ et $\tau_2 = R_2 C$. La réponse en amplitude du filtre est montrée en Figure II.28.

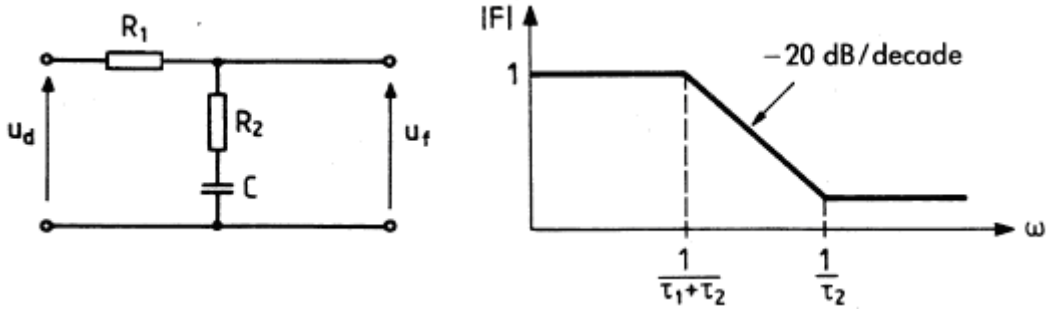


Figure II.28 : Filtre passe-bas de boucle et son diagramme d'amplitude.

Cependant, un comportement spécial du filtre de boucle se manifeste en présence du détecteur de phase de type DFP. Ce dernier n'assure aucune circulation de courant dans les résistances R_1 et R_2 lorsqu'il est dans l'état à haute impédance. Sous cette condition, la tension aux bornes du condensateur reste inchangée. En négligeant les courants de fuite, le signal de sortie U_f du filtre présente un niveau constant lorsque le DFP est dans son état 0. Ainsi, le filtre agit comme un intégrateur idéal, et sa fonction de transfert devient :

$$F(P) = \frac{1 + \tau_2 P}{(\tau_1 + \tau_2)P} \tag{II.18}$$

La combinaison du DFP avec le filtre passif, est souvent référenciée comme pompe de charge. Ce qui signifie qu'à l'état haut du signal de sortie du DFP, la charge circule vers le condensateur du filtre autrement dit, la charge est pompée à l'intérieur du condensateur. Lorsque le signal de sortie du DFP est à son état bas, la charge circule depuis le condensateur du filtre, c-à-d que la charge est pompée à l'extérieur du condensateur vers la masse. Le pompage s'arrête cependant durant l'état 0 du DFP.

Dès lors, nous retenons pour notre conception de la DPLL, la pompe de charge constituée du détecteur DFP associé au filtre passif de la Figure II.28.

Des filtres passe-bas d'ordres supérieurs peuvent être utilisés dans certaines applications des PLLs, mais comme chaque pôle additionnel du filtre amène un déphasage supplémentaire, il est difficile de maintenir la stabilité de ces systèmes d'ordre élevé. Cependant, le dimensionnement du filtre retenu réclame certaines connaissances relatives au comportement de la DPLL en boucle fermée, surtout vis-à-vis du bruit. Nous commencerons ci-après, par l'étude comportementale d'une PLL sans bruit, puis avec bruit.

II.2 Fonctionnement d'une PLL

Une fois les blocs constitutionnels de la PLL en Figure II.1 sont mis en boucle à contre réaction, et en assumant que la fréquence angulaire du signal d'entrée $U_1(t)$ est égale à la fréquence centrale ω_0 du VCO, l'erreur de phase est alors nulle et par conséquent, les signaux de sortie du détecteur de phase et du filtre de boucle sont aussi nuls, Figure II.29. C'est la condition qui permet au VCO d'opérer à sa fréquence centrale. Si initialement, l'erreur de phase θ_e n'est pas nulle, le détecteur de phase développe un signal de sortie U_d non nul. Après un certain retard, le filtre de boucle produit lui aussi, un signal fini U_f. Ceci cause un changement de fréquence du VCO de manière à ce que l'erreur de phase finale disparaisse.

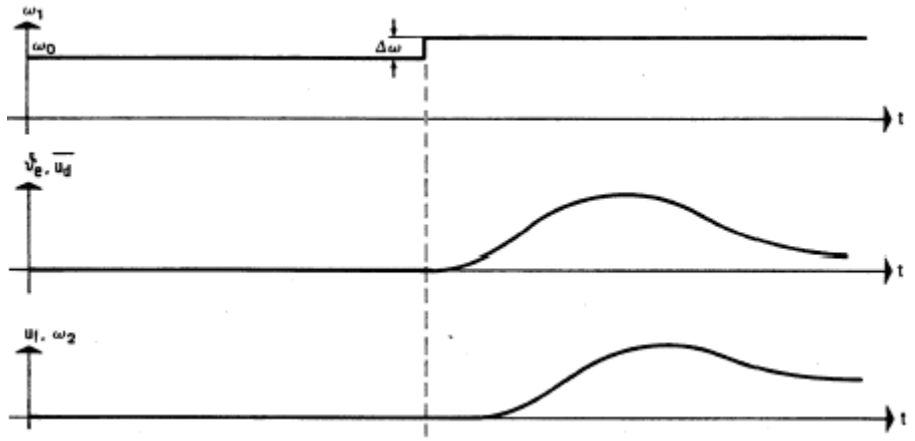


Figure II.29 : Erreur de phase ou tension moyenne du DFP (au milieu) et tension de filtre ou fréquence du VCO (en bas) en réponse à un échelon de fréquence (en haut).

Si la fréquence du signal d'entrée change subitement au temps t_0 d'une augmentation $\Delta\omega$, Figure II.29, la phase associée commence alors en avance par rapport à la phase du signal de sortie. L'erreur de phase apparaît et augmente avec le temps. Le détecteur de phase développe un signal $U_d(t)$ qui augmente aussi avec le temps. Après un retard causé par le filtre de boucle, $U_f(t)$ croît aussi. Ceci oblige le VCO à augmenter sa fréquence. L'erreur de phase devient maintenant plus faible, et après quelques instants, le VCO oscillera à une fréquence exactement égale à celle d'entrée $(\omega_0 + \Delta\omega)$. Dépendant du type de filtre de boucle utilisé, l'erreur de phase finale est réduite à zéro

ou à une valeur finie. Le signal $U_f(t)$ est forcé à une valeur finale de $U_f = \Delta\omega / K_0$. Si la fréquence centrale du signal d'entrée est modulée en fréquence par un signal BF arbitraire alors, le signal de sortie du filtre de boucle est le signal démodulé. Une PLL peut par conséquent, être utilisée comme un détecteur FM. En pratique, une PLL peut aussi assurer le rôle d'un détecteur AM ou PM.

L'une des propriétés importantes de la PLL est sa capacité à supprimer le bruit superposé à son signal d'entrée [1]. Considérons le cas du signal d'entrée de la PLL, noyé dans le bruit. Le détecteur de phase essaie de mesurer l'erreur de phase entre les signaux d'entrée et de sortie. Le bruit à l'entrée fait avancer ou reculer les passages par zéro du signal d'entrée d'une manière stochastique. Ceci cause au signal de sortie du détecteur de phase une fluctuation autour d'une valeur moyenne. Si la fréquence de coupure du filtre de boucle est assez basse, la plupart du bruit se trouve éliminé dans le signal de contrôle du VCO. Ce dernier opère donc d'une manière à ce que la phase de son signal soit égale à la phase moyenne du signal d'entrée. Par conséquent, on peut conclure qu'une PLL est capable de détecter un signal noyé dans le bruit.

II.2.1 Performance dynamique d'une DPLL en dehors de bruit [1]

Lorsque la DPLL est à l'état de verrouillage, sa performance peut être analysée par un modèle linéaire. La Figure II.30 montre le modèle mathématique de la DPLL verrouillée, dont on peut calculer la fonction de transfert de phase $H(P)$ qui relie la phase θ_1 du signal d'entrée à la phase θ_2 du signal de sortie :

$$H(P) = \frac{\theta_2(P)}{\theta_1(P)} = \frac{NK_0K_dF(P)}{NP + NK_0K_dF(P)} = \frac{\omega_2(P)}{\omega_1(P)} \qquad \text{(II.19)}$$

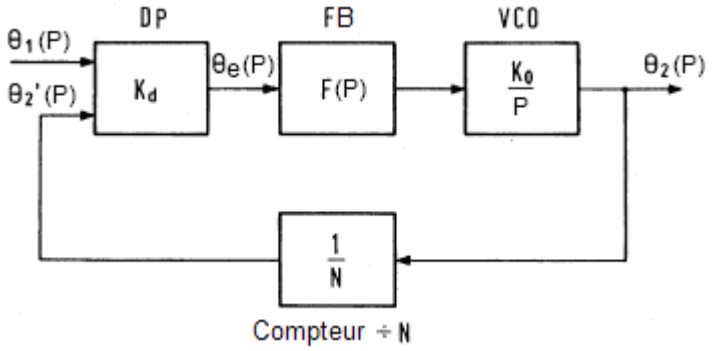


Figure II.30 : Schéma-bloc fonctionnel de la DPLL.

La fonction de transfert d'erreur $H_e(P)$ liant l'erreur de phase θ_e à la phase d'entrée θ_1, est définie par :

$$H_e(P) = \frac{\theta_e(P)}{\theta_1(P)} = 1 - \frac{\hat{\theta_2}(P)}{\theta_1(P)} = 1 - \frac{H(P)}{N} = \frac{NP}{NP + K_0K_dF(P)} \qquad \text{(II.20)}$$

En considérant la fonction de transfert $F(P)$ du filtre de boucle passif (II.17) qui se transforme en (II.18) en présence d'un DFP, la transmittance $H(P)$ de la DPLL présente un dénominateur du deuxième ordre qu'il est plus commode de le mettre sous forme normalisée, $P^2 + 2\varepsilon\omega_n P + \omega_n^2$ dont ω_n représente la fréquence naturelle de la DPLL, et ε, son facteur d'amortissement. On a donc :

$$H(P) = \frac{\frac{K_0 K_d}{(\tau_1 + \tau_2)}(1 + \tau_2 P)}{P^2 + \frac{K_0 K_d \tau_2}{N(\tau_1 + \tau_2)} P + \frac{K_0 K_d}{N(\tau_1 + \tau_2)}} \qquad \text{(II.21)}$$

$$\text{où} \qquad \begin{cases} \omega_n = \sqrt{\dfrac{K_0 K_d}{N(\tau_1 + \tau_2)}} \\ \varepsilon = \dfrac{\omega_n \tau_2}{2} \end{cases}$$

Dans l'équation (II.21), le produit $K_0 K_d$ de dimension S^{-1} représente le gain de boucle. L'équation représente le système DPLL du second ordre, qui acte comme un filtre passe-bas pour les signaux d'entrée $\theta_1(t)$ et dont la bande passante plate s'étend approximativement de zéro à la fréquence naturelle ω_n. Une DPLL est donc capable de traquer les modulations de phase et de fréquence du signal de référence tant que celles-ci ne sortent pas de la bande passante de la DPLL.

Le facteur d'amortissement ε a une importante influence sur la performance dynamique de la DPLL. Pour $\varepsilon = 1$, le système est juste amorti. Si ε est plus faible que l'unité, la réponse transitoire de la DPLL devient oscillatoire. Plus ε diminue, plus augmente le dépassement de la réponse. Dans la plupart des systèmes pratiques, la fonction de transfert est optimale pour $\varepsilon = 1/\sqrt{2}$ et la bande passante à – 3dB devient alors $\omega_{3dB} \approx 2.06\,\omega_n$ [1], qui est le double de la fréquence naturelle.

II.2.2 Plages de fonctionnement d'une DPLL [1]

La performance dynamique de la DPLL est gouvernée par un ensemble de paramètres-clé.

II.2.2.1 La plage de maintien

C'est la plage $\Delta\omega_H$ dans laquelle une PLL est capable de maintenir un verrouillage statiquement. La PLL se déverrouille toujours lorsque la fréquence du signal d'entrée excède la rangée de maintien. Pour atteindre la limite de stabilité statique, il est nécessaire d'effectuer un balayage de la fréquence de référence d'une manière lente.

Si la fréquence de référence est augmentée et que le gain statique du filtre de boucle est fini, l'erreur de phase augmente en proportion. Lorsque la valeur maximale est atteinte pour laquelle le détecteur de phase opère linéairement, la plage de maintien est atteinte. Ainsi, pour un détecteur de phase de type DFP qui présente une erreur de phase maximale de 2π, l'étendue de la plage de maintien peut être obtenue en calculant l'écart fréquentiel à l'entrée de référence qui cause une erreur de phase de 2π.

Soit :

$$\omega_1 = \omega_0 + \Delta\omega_H \tag{II.22}$$

Le signal de phase en réponse à cet accroissement, devient :

$$\theta_1(t) = \Delta\omega_H . t \tag{II.23}$$

Sa transformée de Laplace est :

$$\theta_1(P) = \Delta\omega_H / P^2 \tag{II.24}$$

L'erreur de phase est tirée de l'expression (II.20) :

$$\theta_e(P) = \theta_1(P) H_e(P) = \frac{\Delta\omega_H}{P^2} \frac{NP}{NP + K_0 K_d F(p)} \tag{II.25}$$

Utilisant le théorème de la valeur finale, l'erreur de phase finale dans le domaine temporel est :

$$\lim_{t\to\infty} \theta_e(t) = \lim_{P\to\infty} P\theta_e(P) = \frac{N\Delta\omega_H}{K_0 K_d F(0)} = 2\pi \tag{II.26}$$

La quantité $K_0 K_d F(0)/N$ n'est que le gain statique de la boucle ouverte de la DPLL, où $F(0)$ est le gain statique du filtre de boucle, qui vaut 1 pour le filtre passif de la Figure II.28 mais vaut l'infini lorsque ce filtre devient intégrateur une fois associé au détecteur de phase DFP à trois états.

La plage de maintien de la DPLL utilisant un DFP devient ainsi infinie.

II.2.2.2 La plage de verrouillage (ou plage de capture rapide [39])

C'est la plage $\Delta\omega_L$ dans laquelle une PLL se verrouille en moins d'un cycle d'écart entre la fréquence de référence et la fréquence obtenue par division de celle du VCO. On assume que la DPLL est non verrouillée initialement et que le VCO oscille sur sa fréquence centrale $N\omega_0$.

La fréquence de référence est décalée de $\Delta\omega$ de sa fréquence centrale ω_0, telle que $\omega_1 = \omega_0 + \Delta\omega$.

Les signaux d'entrées du détecteur de phase peuvent ainsi être représentés par des fonctions de Walsh :

$$u_1(t) = E_1 W[(\omega_0 + \Delta\omega)t] \quad \text{et} \quad u_2'(t) = E_2 W(\omega_0 t) \tag{II.27}$$

Où E_1 et E_2 sont les amplitudes des signaux carrés d'entrées.
L'erreur de phase est la différence de phase de ces deux signaux, soit la rampe :

$$\theta_e = \Delta\omega.\,t \tag{II.28}$$

Pour le détecteur de fréquence-phase (DFP), sa sortie moyenne $\bar{u}_d(\theta_e)$est une fonction en dent de scie de l'erreur de phase (II.4), $\bar{u}_d(t)$ devient donc une fonction en dent de scie du temps, Figure II.31.

A la sortie du filtre de boucle apparaît le signal :

$$u_f(t) = |F(\Delta\omega)|\bar{u}_d(t) = K_d|F(\Delta\omega)|\Delta\omega t \tag{II.29}$$

La forme du signal u_f est ainsi en dent de scie, et module la fréquence ω_2 du VCO, Figure II.31, et sa dérivée ω'_2 à la sortie du diviseur de fréquence.

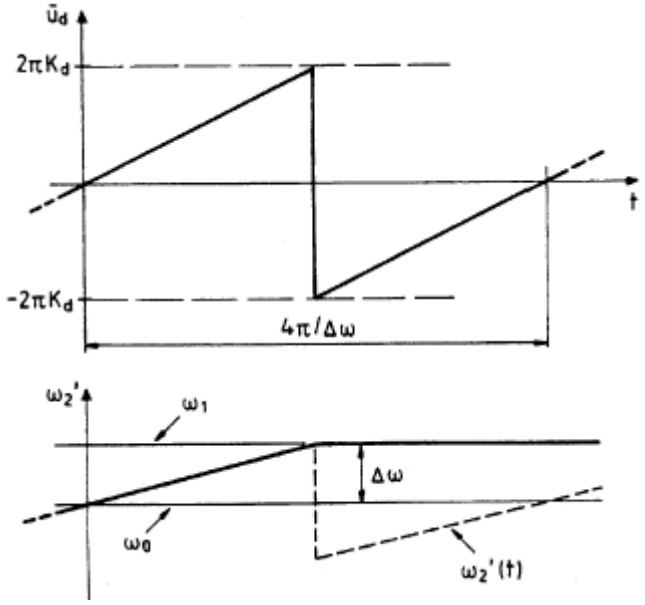


Figure II.31 : Tension moyenne du DFP (en haut) et fréquence de VCO (en bas).

La condition de verrouillage rapide est vérifiée lorsque le pic de la courbe ω'_2 dépasse la fréquence de référence ω_1 ou l'atteint tout juste :

$$2\pi K_0 K_d|F(\Delta\omega)| \geq \Delta\omega \tag{II.30}$$

L'écart fréquentiel $\Delta\omega$ devient égal à la rangée de verrouillage $\Delta\omega_L$, lorsque la condition (II.30) est vérifiée avec égalité de ses deux membres. L'équation non linéaire qui en

découle pour $\Delta\omega_L$, peut facilement être résolue si on considère une approximation sur le gain du filtre $|F(\Delta\omega_L)|$. A partir de considérations pratiques, la plage de verrouillage est toujours plus grande que les fréquences de coupure du filtre de boucle [1], $\Delta\omega_L >> 1/\tau_2 > 1/\tau_1$. Le gain HF du filtre devient alors pour le filtre passif :

$$F(\Delta\omega_L) = \tau_2/(\tau_1 + \tau_2) \tag{II.31}$$

D'autre part, de l'expression (II.20) on a la fonction de transfert :

$$\theta_2'(P)/\theta_1(P) = H(P)/N \tag{II.32}$$

La fréquence naturelle associée à cette fonction de transfert peut être déduite de l'expression (II.21), en lui omettant uniquement le facteur *'N'*. Cependant, le facteur d'amortissement de la même expression (II.21), reste conservé. En assumant encore une boucle à fort gain statique, le facteur d'amortissement devient égal à $\varepsilon \approx \omega_n\tau_2/2$.

La plage de verrouillage d'une DPLL à base du détecteur de fréquence-phase (DFP), devient donc:

$$\Delta\omega_L = 2\pi K_0 K_d F(\Delta\omega_L) = 4\pi\varepsilon\omega_n \tag{II.33}$$

Lorsque la DPLL se verrouille rapidement, les signaux u_d et u_f accusent une oscillation amortie (pour $\varepsilon < 1$) dont la fréquence angulaire est approximativement ω_n. Comme au maximum, le processus de verrouillage rapide se termine au bout d'un seul cycle de l'oscillation amortie, il est raisonnable de prendre pour temps de verrouillage T_L, une seule période de l'oscillation amortie. Soit :

$$T_L \approx 2\pi/\omega_n \tag{II.34}$$

Qui est valable pour tout type de filtre de boucle.

II.2.2.3 La plage de capture lente [39] (pull in)

C'est la plage $\Delta\omega_C$ dans laquelle une PLL peut toujours se verrouiller mais sous éventuellement un lent processus, il s'agit d'un phénomène non linéaire [1]. Ceci, se passe lorsque la condition (II.30) n'est pas vérifiée à cause d'un écart fréquentiel $\Delta\omega$ entre référence et VCO devenant supérieur à la rangée de verrouillage $\Delta\omega_L$. Autrement dit, lorsque la DPLL n'est pas verrouillée initialement et que son VCO opère à sa fréquence centrale $N\omega_0$ en présence d'une fréquence du signal de référence $\omega_1 = \omega_0 + \Delta\omega$, dont le large écart $\Delta\omega$ ne permet pas un processus de capture rapide. Dans le cas du détecteur de phase DFP, la moyenne $\bar{u}_d(t)$du signal de sa sortie est une fonction en dent de scie comme en Figure II.31 mais cette fois-ci avec allure asymétrique, Figure II.32.

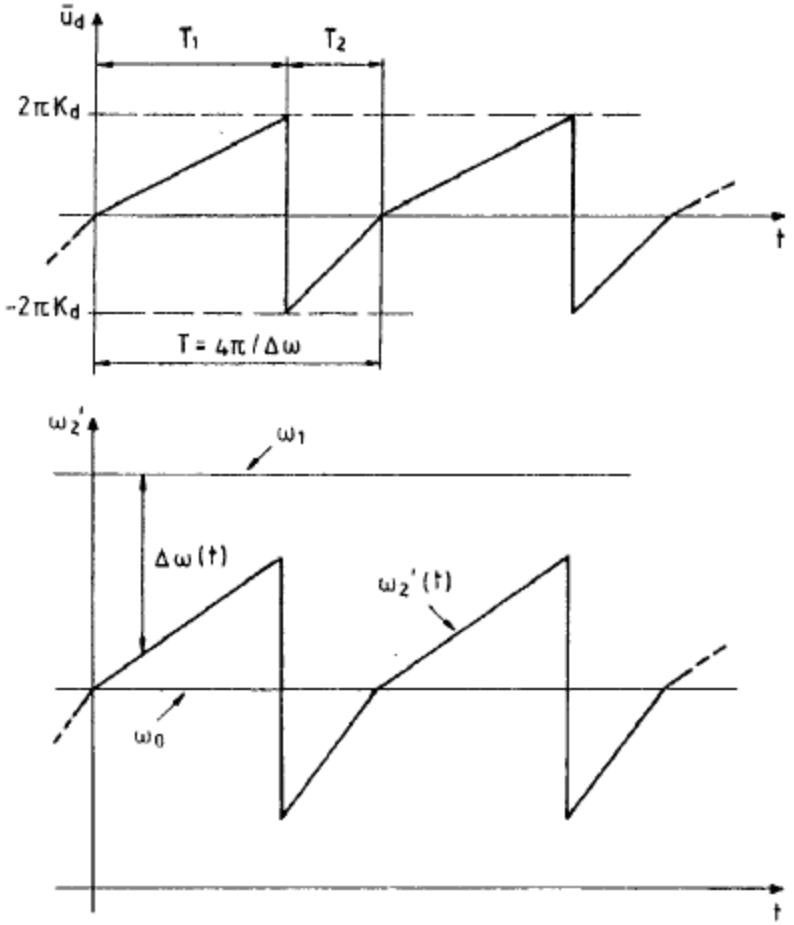


Figure II.32 : Tension moyenne du DFP (en haut) et fréquence du VCO (en bas).

Le signal de sortie $u_f(t)$ du filtre de boucle suit la forme du signal $\bar{u}_d(t)$ et module instantanément la fréquence $\acute{\omega}_2$ du VCO, Figure II.32. Comme le pic de la fréquence $\acute{\omega}_2$ n'atteint pas la fréquence d'entrée ω_1, on peut imaginer que la DPLL ne pourra jamais se verrouiller. Ceci sera vrai si la forme d'onde triangulaire est symétrique (c-à-d, $T_1 = T_2$), alors la moyenne de la fréquence $\acute{\omega}_2$ restera constante et égale à ω_0. Heureusement et comme montrée à la Figure II.32, l'écart de fréquence $\Delta\omega$ n'est pas constant mais donné par la différence entre la fréquence de référence ω_1 et la fréquence instantanée $\acute{\omega}_2$ dérivée du VCO. Par conséquent, $\Delta\omega(t)$ devient faible durant la demie onde positive du signal $\bar{u}_d(t)$ et devient forte durant la demie onde négative. Ainsi, la forme d'onde $\bar{u}_d(t)$ devient asymétrique de façon à ce que sa valeur moyenne n'est pas nulle mais légèrement positive. Ceci cause une augmentation de la fréquence moyenne du VCO. Maintenant si le gain statique $K_d K_0 F(0)/N$ en boucle ouverte de la DPLL est très faible, la fréquence du VCO est augmentée d'un très faible pas et reste pratiquement autour d'une valeur finale [1]. Cependant, si le gain de boucle est élevé, le processus de capture devient régénératif : la fréquence moyenne $\acute{\omega}_2$ est augmentée suffisamment pour que la forme d'onde $\bar{u}_d(t)$ devient davantage asymétrique (c-à-d, le rapport T_1 / T_2 devient significativement plus grand). Ceci force la fréquence moyenne $\acute{\omega}_2$ à augmenter encore, etc, jusqu'à ce que la fréquence dérivée du VCO devient proche de celle de référence laissant place au processus de verrouillage, voir plus loin les exemples de la Figure II.34.

Le phénomène décrit ci-dessus ne s'amorce que si l'écart de fréquence $\Delta\omega$ initial ne dépasse pas une valeur critique $\Delta\omega_C$ qui représente la rangée de capture lente. Au delà, l'effet de capture n'est plus régénératif [1]. Le calcul de la rangée $\Delta\omega_C$ reste approximatif et plusieurs auteurs ont proposé des expressions pour une PLL contenant un filtre passif.

Dans le cas du DFP utilisé comme détecteur de phase, la plage de capture lente devient infinie car le filtre de boucle est piloté par une source à trois états, dont l'état haute impédance permet de garder inchangée la charge du filtre de boucle. Sous cette condition, même un filtre passif travaillera comme un intégrateur, voir relation (II.18). Cependant, le temps de capture lente demeure fini et on présentera ici, son calcul approximatif [1].

Considérons d'abord une DPLL à filtre de boucle passif dans son état déverrouillé en présence d'une fréquence de référence ω_1 plus élevée de $\Delta\omega$ que celle dérivée en sortie $\acute{\omega}_2$, Figure II.33. Comme expliqué sur la Figure II.4, la sortie u_d du DFP bascule entre les états 0 et +1, Figure II.33.

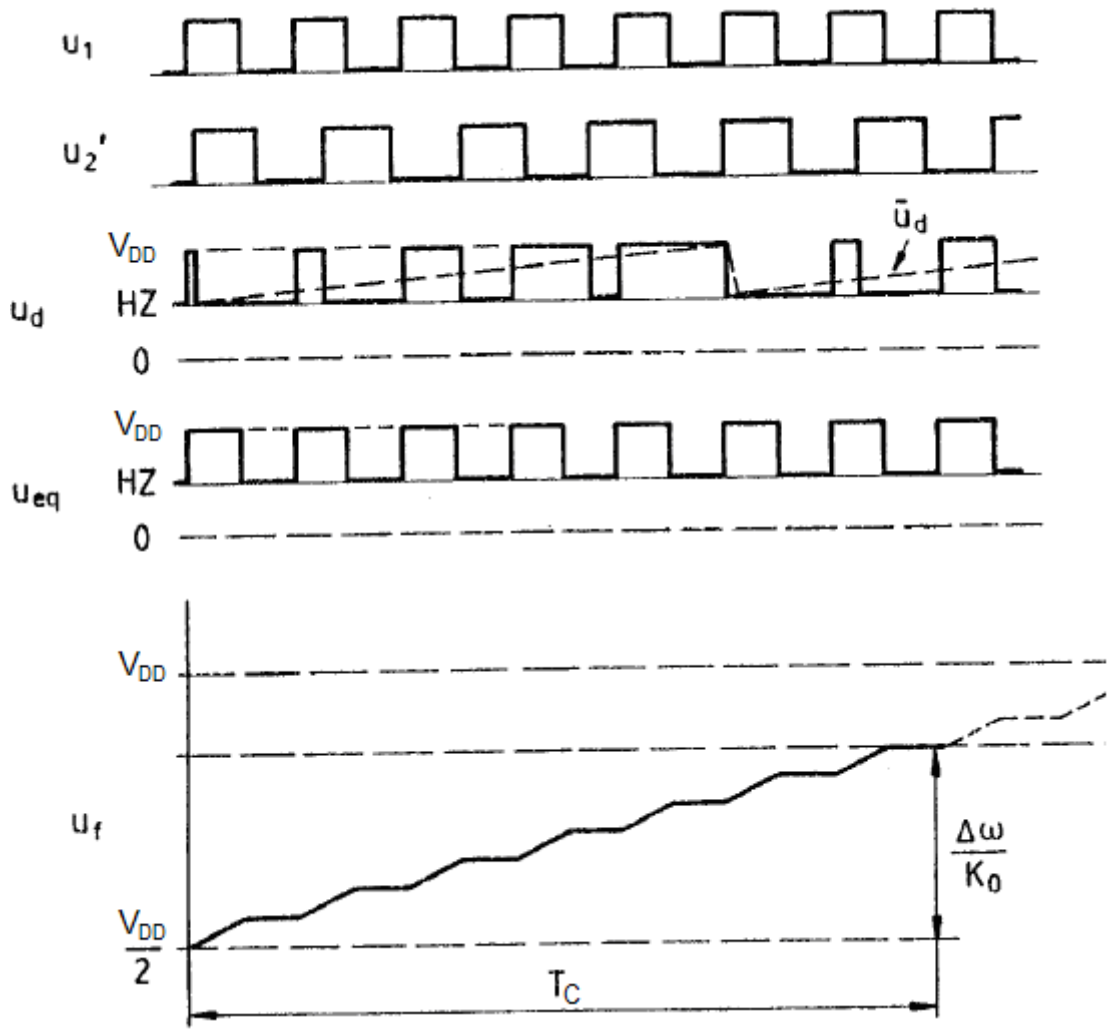


Figure II.33 : Tensions d'entrées et de sortie du DFP avec tension équivalente et tension de filtre.

L'alimentation du DFP est supposée être unipolaire entre 0 et $+V_{DD}$. La ligne centrale du signal u_d représente l'état haute impédance (HZ) du DFP. Le signal moyen de u_d est tracé en discontinu et a la forme d'une dent de scie. Le signal $\bar{u}_d$ n'est autre que le rapport cyclique de u_d, évoluant périodiquement entre 0 et 1. Le rapport cyclique moyen est 50% et peut être considéré comme équivalent au signal $\bar{u}_d$.

Le filtre passif se charge sous la constante de temps $\tau_1 + \tau_2 = (R_1 + R_2)C$, Figure II.28, qui est beaucoup plus élevée que la période du signal u_1. Un signal équivalent u_{eq} ayant un facteur de forme de 50% aura le même effet sur le filtre de boucle, Figure II.33. Si le signal équivalent u_{eq} a un facteur de forme de 100%, la capacité du filtre se chargera simplement vers le tension d'alimentation V_{DD} sous la constante de temps $\tau_1 + \tau_2$. A cause du facteur de forme de 50%, la capacité a besoin d'un double temps pour se charger. C'est équivalent à dire que le filtre de boucle présente une constante de temps $2(\tau_1 + \tau_2)$. D'autre part, le VCO de la DPLL est assumé travailler à sa fréquence centrale $N\omega_0$ au début du processus de la capture lente, la valeur initiale du signal u_f à la sortie du filtre de boucle (aux bornes du condensateur C) est donc $V_{DD}/2$. Par conséquent, le condensateur se charge de $V_{DD}/2$ à V_{DD} durant le processus de la capture lente. Cependant, lorsque la tension de charge u_f arrive à la valeur $\Delta\omega/K_0$, Figure II.33, le VCO génère la bonne fréquence et le processus de capture lente prend fin.

Le temps de capture lente T_C est le temps requis par le condensateur pour augmenter sa tension de la valeur $\Delta\omega/K_0$ et permettre ainsi de couvrir l'écart fréquentiel initial,$\Delta\omega = \omega_1 - \omega_0$.

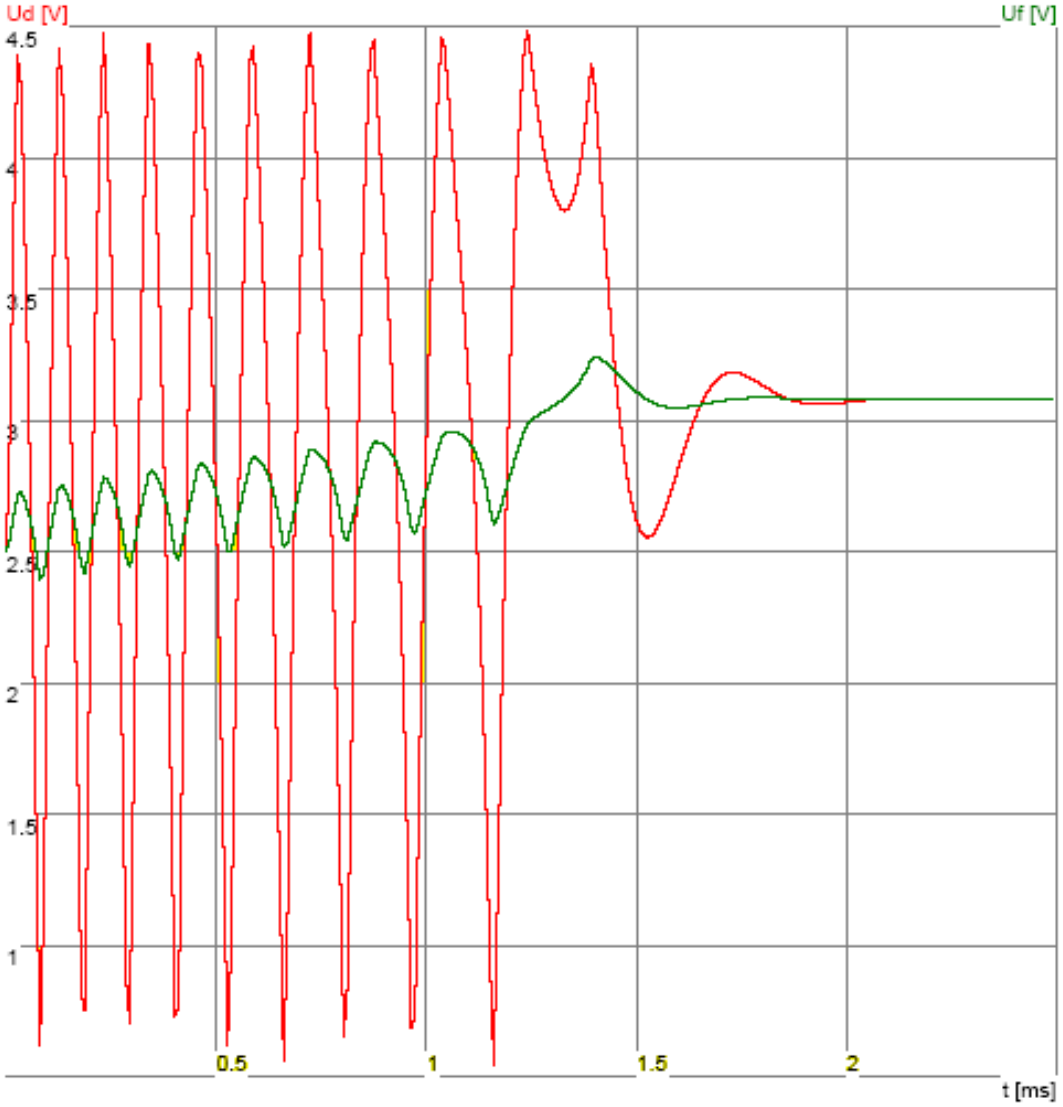
$$\frac{\Delta\omega}{K_0} = \frac{V_{DD}}{2}\left[\left(1 - exp\left(-\frac{T_C}{2(\tau_1 + \tau_2)}\right)\right)\right] \Rightarrow T_C = 2(\tau_1 + \tau_2)\ln\frac{K_0(V_{DD}/2)}{K_0(V_{DD}/2) - \Delta\omega} \qquad \text{(II.35)}$$

La Figure II.34 illustre le processus de la capture lente à l'aide du simulateur téléchargé, phaselockloop. Dans cette simulation, la DPLL est configurée avec le filtre passif de la Figure II.28 en liaison avec l'un des trois détecteurs de phase : porte XOR, bascule JK, et détecteur DFP, pris à tour de rôle. Cela permet de mettre en évidence, la supériorité de la pompe de charge sur les autres variantes. Pour cela, les constantes de temps du filtre sont choisies telles que $\tau_1 = 500\mu s$ et $\tau_2 = 50\mu s$. La fréquence centrale du VCO ainsi que sa sensibilité sont choisies telles que f_o=100kHz et $K_0 = 130$ $(V.ms)^{-1}$, respectivement.

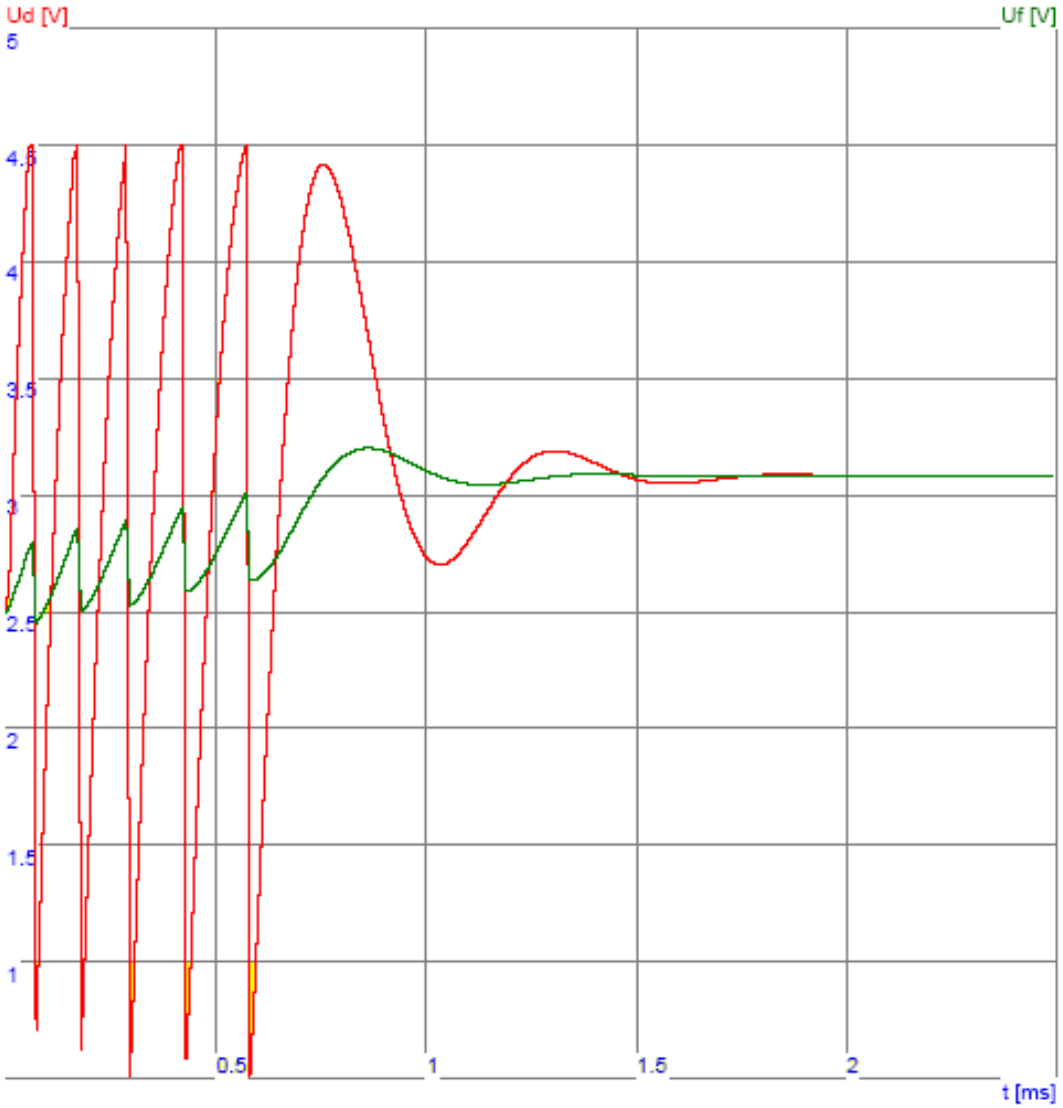
La DPLL munie d'une porte XOR avec une sensibilité portée à $K_d = 1.27$V/rad. La Figure II.34(a) montre l'évolution temporelle des tensions de sortie du détecteur de phase et du filtre de boucle, comme réponses à un fort échelon fréquentiel d'entrée de 12kHz (à la limite de la plage de capture lente). Le temps de capture simulé est d'environ 1.3ms (laissant un cycle U_d pour la capture rapide).

Pour le cas d'une DPLL munie d'une bascule JK à sensibilité $K_d = 0.637$V/rad, un fort échelon fréquentiel d'entrée de 12kHz est appliqué dont la réponse est fournie à la Figure II.34(b). Le temps de capture simulé est d'environ 0.7ms.

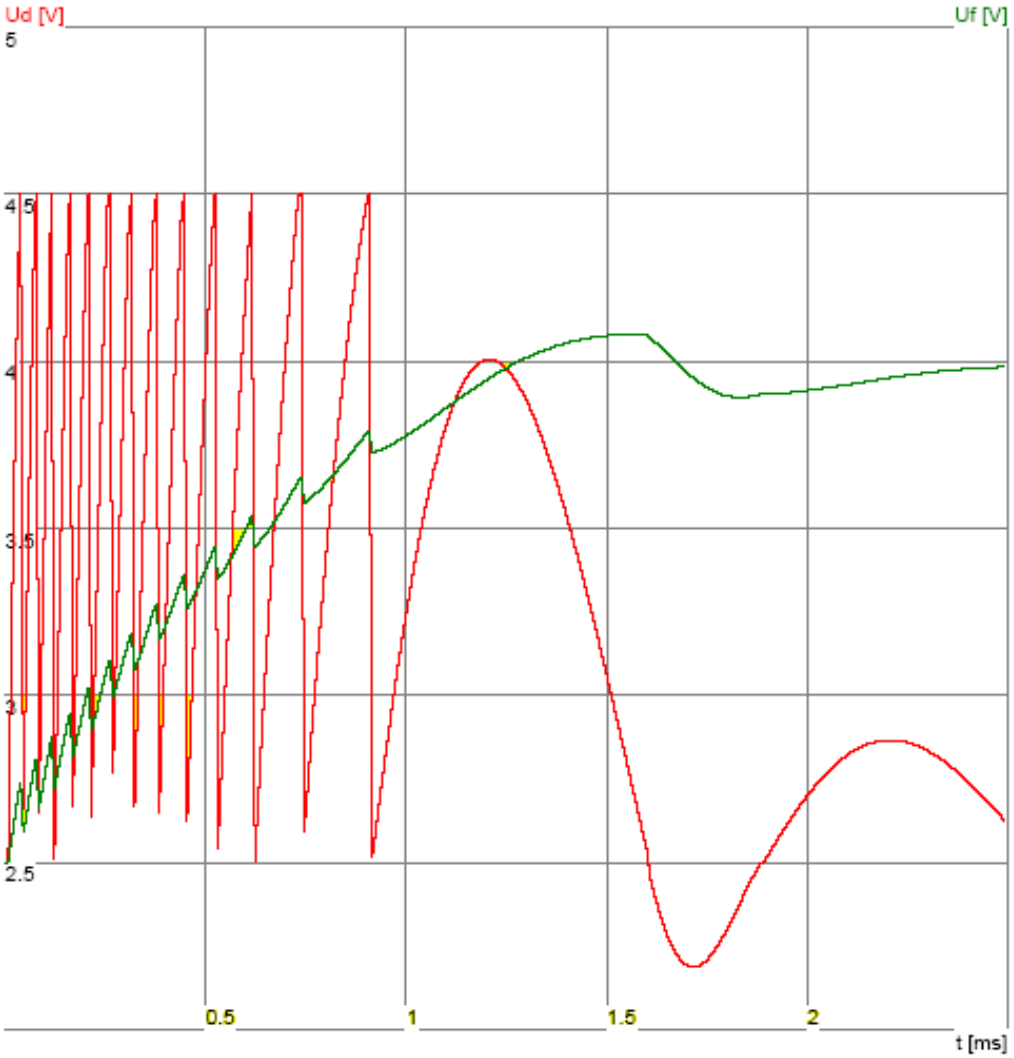
Pour le cas d'une DPLL munie d'un DFP à sensibilité $K_d = 0.318\text{V/rad}$, les paramètres correspondants du tableau II.1, deviennent fixés aux valeurs $\omega_n = 8674\,\text{sec}^{-1}$ et $\varepsilon = 0.217$, avec une plage de capture lente prédite à l'infinie. Un fort échelon fréquentiel d'entrée de 30kHz est appliqué dont la réponse est fournie à la Figure II.34(c). Le temps de capture simulé est d'environ 1.2ms comparé devant le temps prédit au tableau II.1, de 0.95ms.



(a) DPLL Simulation 31-8-2014 10:11
PD = EXOR
LF = passive Filter
Center frequency = 100000 Hz
Response on frequency step 12000 Hz



(b) DPLL Simulation 31-8-2014 10:14
PD = JK-Flipflop
LF = passive Filter
Center frequency = 100000 Hz
Response on frequency step 12000 Hz



(c) DPLL Simulation 31-8-2014 10:17
PD = Phase-Frequency Detector
LF = passive Filter
Center frequency = 100000 Hz
Response on frequency step 30000 Hz

Figure II.34 : Processus de capture lente avec détecteur de phase à porte XOR (a), à bascule JK (b), et à DFP présentant un temps de capture similaire mais pour un pas d'entrée plus grand (c).

II.2.2.4 La plage de poursuite (tracking) (pull out)

Cette rangée $\Delta\omega_P$ est par définition, la taille du pas fréquentiel qui cause un déverrouillage de la PLL une fois appliqué à l'entrée de référence. Donc, c'est la limite dynamique d'une opération stable d'une PLL. Dans la plupart des cas pratiques, on montre que la plage de poursuite est comprise entre les deux plages de capture, rapide et lente [39] :

$$\Delta\omega_L < \Delta\omega_P < \Delta\omega_C \qquad \text{(II.36)}$$

Plusieurs expressions empiriques sont proposées [1] pour une DPLL, selon le type du détecteur de phase utilisé. Ainsi, pour une DPLL à base d'un DFP :

$$\Delta\omega_P = 11.55\ \omega_n(\varepsilon + 0.5) \qquad \text{(II.37)}$$

Les caractéristiques les plus importantes de la DPLL sont récapitulées dans le tableau II.1.

Catégorie de paramètres	Symbole et valeur	Paramètre	Définition
Générale	ω_0	Fréquence centrale du VCO	Fréquence du VCO à $u_f = 0$
	$\tau_1,\ \tau_2$	C^{stes} de temps du filtre de boucle	–
	$\omega_n = \sqrt{\dfrac{K_0 K_d}{N(\tau_1 + \tau_2)}}$	Fréquence naturelle de la PLL	Fréquence de la réponse oscillatoire amortie de la boucle
	$\varepsilon = \dfrac{\omega_n \tau_2}{2}$	Facteur d'amortissement	–
Acquisition	$\Delta\omega_H = \infty$	Plage de maintien	Rangée de fonctionnement stable statiquement (verrouillage maintenu pour une lente variation de la fréquence d'entrée)
	$\Delta\omega_L = 4\pi\varepsilon\omega_n$	Plage de capture rapide	Rangée de verrouillage se terminant en moins d'un cycle de la réponse oscillatoire amortie de la boucle
	$T_L \approx\ 2\pi\ /\ \omega_n$	Temps de capture rapide	Durée requise pour la capture rapide
	$\Delta\omega_C = \infty$	Plage de capture lente	Rangée de verrouillage se terminant en plusieurs cycles de la réponse oscillatoire amortie de la boucle
	$T_C = 2(\tau_1 + \tau_2)\ln\dfrac{K_0(V_{DD}/2)}{K_0(V_{DD}/2) - \Delta\omega}$	Temps de capture lente	Durée requise pour la capture lente
Poursuite	$\Delta\omega_P = 11.55\ \omega_n(\varepsilon + 0.5)$	Plage de poursuite	Rangée de fonctionnement stable dynamiquement (verrouillage pour une rapide variation de la fréquence d'entrée)

Tableau II.1 : Récapitulation des paramètres de la DPLL munie d'une pompe de charge (DFP + filtre passif).

II.2.3 Performance dynamique d'une DPLL en présence de bruit [1]

Dans cette étude, nous nous limitons à certains paramètres qui ont une influence sur la conception des PLLs.

Tout d'abord commençons par voir l'effet du bruit sur la phase du signal d'entrée u_1 de la DPLL, Figure II.35. Le premier tracé montre une forme d'onde d'un signal d'information binaire u_1 démuni de bruit. Souvent, un tel signal est transmis à travers une liaison de donnée dont la bande est limitée. En dehors de bruit additionnel, les transitions de u_1 accusent maintenant des temps de montée et de descente finis (signal u_{1b}). Cependant si un bruit est superposé, le signal bruité u_{1n} apparaît à la fin de la ligne de transmission. En vue d'une remise en forme du signal, un trigger de Schmitt est utilisé dont les seuils haut et bas sont respectivement, u_{haut} et u_{bas}. Le signal restitué par remise en forme est u_{1r} au dernier tracé de la Figure II.35. L'effet évident d'une bande étroite de canal et un bruit additif, est la gigue de phase accompagnant les transitions du signal mis en forme. Plus grande est l'intensité du bruit, plus grande est la gigue de phase (régions hachurées de u_{1r}). Cependant vu que l'amplitude du signal mis en forme est constante, nous ne pouvons décider quelle portion de u_{1r} est due à la contribution du signal ou du bruit.

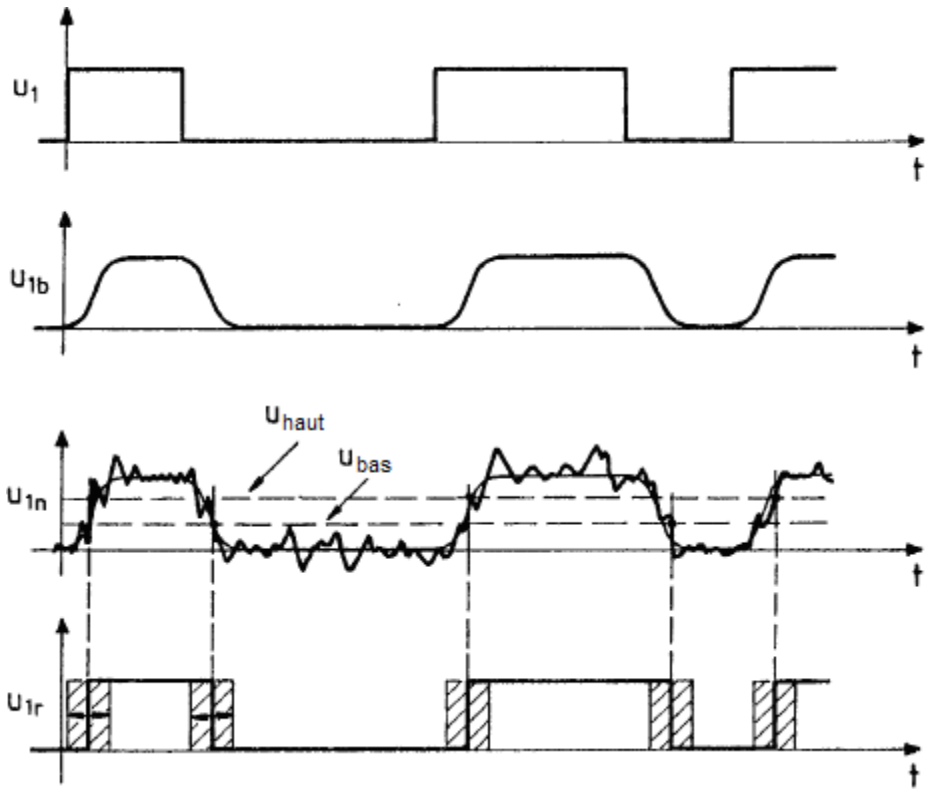


Figure II.35 : Signal U_1 d'information, signal U_{1b} à bande réduite par le canal, signal U_{1n} superposé de bruit, et signal U_{1r} remis en forme.

D'autre part, une DPLL offre par sa nature d'asservissement, une fonction de transfert de phase $H(j\omega)$, expression (II.19), à filtrage passe-bas qui aide à supprimer les bruits en superposition à son signal d'entrée. Si la fréquence du bruit de phase est au-delà de la bande équivalente de bruit B_n de la DPLL (définie plus loin), cette composante de

bruit sera éliminée. Seule les composantes spectrales situées à l'intérieure de B_n, contribuent à la gigue de phase du signal de sortie de la DPLL. Souvent, la bande de bruit est limitée à B_i par un pré-filtre passe-bande ou passe-bas, généralement disposé à l'entrée de la DPLL, Figure II.36. Le filtrage peut être dû simplement à la bande de canal de transmission, comme en Figure II.35.

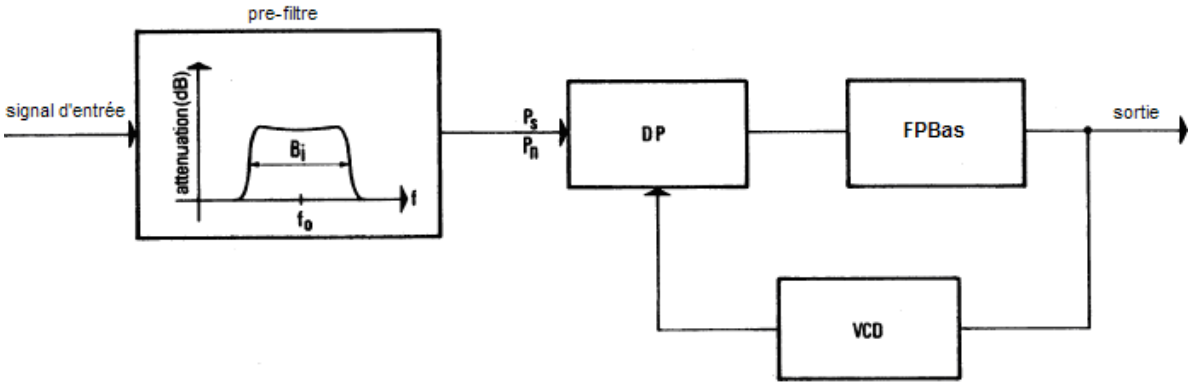


Figure II.36 : DPLL munie d'un pré-filtre d'entrée.

Soit le signal d'entrée u_{1n} de la Figure II.35, qui est filtré mais infecté encore de bruit $n(t)$ (en Volts).

Sa représentation par la fonction de Walsh est :

$$u_{1n}(t) = E\,W(\omega_1 t + \theta_1) + n(t) \tag{II.38}$$

Où E et θ_1 sont respectivement, l'amplitude et la phase du signal carré (unipolaire) d'entrée.

Son rapport signal-sur-bruit s'écrit :

$$SNR_1 = (E^2/2) / \sigma_n^2 \tag{II.39}$$

Avec la variance σ_n^2 ou puissance du bruit $n(t)$ qui est égale à la densité spectrale de puissance (PSD) du bruit blanc (N_0 pour un spectre unilatéral, W/Hz) filtrée à la bande B_i :

$$\sigma_n^2 = N_0 B_i \tag{II.40}$$

Après mise en forme de u_{1n}, Figure II.35, le signal d'entrée u_{1r} de la DPLL est à amplitude E' constante et s'écrit :

$$u_1(t) = E'\,W(\omega_1 t + \theta_{n1}(t)) \tag{II.41}$$

Où $\theta_{n1}(t) = \theta_1 + n'(t)$ est la phase imprégnée de bruit $n'(t)$ (sans dimension).
Pour θ_1 considérée nulle initialement, la moyenne du carré de la valeur RMS de $\theta_{n1}(t)$ n'est autre que la variance de bruit $n'(t)$ [3] :

$$\overline{\theta_{n1}^2} = \sigma_{\acute{n}}^2 \tag{II.42}$$

La relation entre les deux bruits, n (superposé à l'amplitude E) et n' (superposé à la phase θ_1), est bien établie dans le cas d'une PLL munie d'un multiplieur comme détecteur de phase [3] :

$$n' = n / E \tag{II.43}$$

ici E, est relative à l'amplitude du signal sinusoïdal d'entrée.

Dans le cas d'une DPLL, on postule [1] la réutilisation de la relation (II.43) qui en termes de variances de bruit, s'écrit :

$$\sigma_{\acute{n}}^2 = \sigma_n^2 / E^2 \tag{II.44}$$

En utilisant (II.39) et (II.42), (II.44) devient :

$$\overline{\theta_{n1}^2} = \frac{1}{2\,SNR_1} \tag{II.45}$$

Dans les applications pratiques, la gigue de phase est mesurée à l'aide d'un diagramme en œil, obtenu sur oscilloscope par superposition du signal $u_1(t)$ bruité sur quelques périodes.

En reprenant le calcul dans le domaine spectral, la PSD du bruit de phase d'entrée limitée à la bande de base $B_i/2$ [1] par le pré-filtre de la Figure II.36, devient comme indiquée sur la Figure II.37, une constante qui vaut :

$$\Phi_1 = \frac{\overline{\theta_{n1}^2}}{B_i/2} \qquad rad^2\,Hz^{-1} \tag{II.46}$$

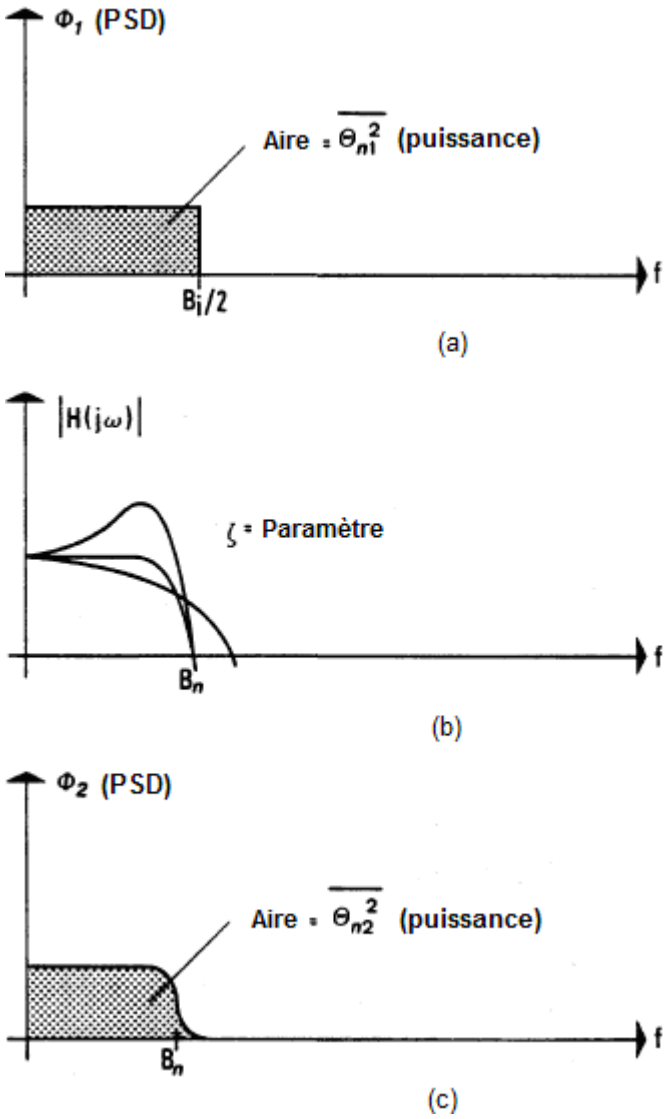


Figure II.37 : DPLL avec sa PSD d'entrée (a), sa réponse impulsionnelle (b), et sa PSD de sortie (c).

Comme la PSD du bruit de phase à l'entrée de la DPLL est maintenant connu, on peut calculer également celle de la sortie en tenant en compte du filtrage passe-bas $|H(j\omega)|$ propre à la DPLL comme expliqué plus haut :

$$\Phi_2 = |H(j\omega)|^2 \Phi_1 \tag{II.47}$$

La Figure II.37(c) n'est que la multiplication des deux courbes des figures I.37(a) et I.37(b), selon l'équation (II.47).

La puissance du bruit $\overline{\theta_{n2}^2(t)}$ à la sortie de la DPLL est donnée par l'aire de sa PSD, soit :

$$\overline{\theta_{n2}^2} = \int_0^\infty \Phi_2(j2\pi f) df \tag{II.48}$$

En réunissant les expressions (II.47) et (II.48), il vient :

$$\overline{\theta_{n2}^2} = \int_0^\infty \Phi_1 |H(j2\pi f)|^2 df \qquad \text{(II.49)}$$

Par analogie à l'expression (II.40), cela permet de définir la bande équivalente de bruit B_n :

$$\overline{\theta_{n2}^2} = \Phi_1 B_n \implies B_n = \int_0^\infty |H(j2\pi f)|^2 df = \frac{\omega_n}{2}\left(\varepsilon + \frac{1}{4\varepsilon}\right) \qquad \text{(II.50)}$$

Cette bande est proportionnelle à la fréquence naturelle ω_n de la DPLL et présente un minimum $B_{n-min} = \omega_n/2$, pour un facteur d'amortissement $\varepsilon = 0.5$. Cependant pour $\varepsilon = 0.7$ choisi en pratique, la performance au bruit ne change pas notablement : $B_n = 0.53\,\omega_n$ [1].

Par analogie à l'équation (II.45), la variance de bruit de sortie peut s'écrire en fonction du rapport signal-sur-bruit SNR_2 à la sortie de la DPLL :

$$\overline{\theta_{n2}^2} = \frac{1}{2SNR_2} \qquad \text{(II.51)}$$

En combinant les équations (II.45), (II.46), (II.50), avec (II.51), on obtient :

$$SNR_2 = SNR_1 \frac{B_i}{2B_n} \qquad \text{(II.52)}$$

Qui indique que la DPLL améliore le SNR_1 du signal d'entrée d'un facteur $B_i/2B_n$. Plus étroite est la bande B_n, plus grande sera l'amélioration.

On a par exemple [1], une opération stable de la DPLL pour $SNR_2 = 4$ (6 dB), où selon (II.51), la valeur RMS du bruit de phase à la sortie de la DPLL, est :
$\sqrt{\overline{\theta_{n2}^2}} = {}^1/_{\sqrt{8}} = 0.353\,rd = 20^0$.

La gigue de phase étant de $20°$, la valeur $180°$ qui représente la limite de stabilité dynamique, est donc rarement dépassée.

Il faut remarquer que les DPLLs se comportent différemment vis-à-vis d'un bruit d'évanouissement [1]. Lorsqu'un paquet de bits est transmis à travers une liaison de transmission et qu'un nombre de bits successifs est perdu par évanouissement, le signal remis en forme u_{1r} de la Figure II.35 reste au niveau logique haut ou bas pour un certain temps. Le signal moyen à la sortie d'une porte XOR reste nul pendant ce temps et la fréquence du VCO reste inchangée ou dérive lentement. Cependant, le signal de sortie d'une bascule JK ou DFP, reste à un haut niveau, ce qui oblige la fréquence du VCO à changer très rapidement, ce qui est indésirable. En conclusion, une porte XOR en présence de bruit d'évanouissement, fournit une meilleure performance comparativement aux autres. Pour un bruit naturel, un DFP en reste le meilleur.

II.3 Conception de la DPLL

Tous les blocs constituant notre DPLL, Figure II.1, ont été présentés et discutés un par un, aboutissant à leur dimensionnement. Ainsi au paragraphe II.1.1, a été choisi le détecteur de phase de type DFP selon la Figure II.3, présentant une caractéristique la plus large. Sa sensibilité était calculée avec la valeur $K_d = 0.4$V/rd.

Au paragraphe II.1.2, un VCO était conçu et amélioré suivant la Figure II.24. Sa sensibilité est mesurée avec la valeur $K_0 = 2\,10^6$ rd/Vs. Au dernier paragraphe II.1.3, un filtre de boucle passif du premier ordre a été choisi selon le type de la Figure II.28, vue sa simplicité et la performance atteinte une fois couplé au DFP. La pompe de charge résultante promet des plages de maintien et de capture lente théoriquement infinies, et pratiquement limitées à la plage couverte par le VCO. D'autres types de filtre de boucle n'apportent pas d'additionnels bénéfices [1].

Ces étapes d'étude nous ont permis d'établir le schéma bloc fonctionnel de notre DPLL en Figure II.30, dont, la fonction de transfert en boucle fermée a été établie et exprimée par l'équation (II.21). Seul reste le calcul du filtre de boucle pour fixer les propriétés attendues de la DPLL.

Il faut, au cours de la conception d'une DPLL, décider si celle-ci doit offrir une suppression de bruit ou pas. Dans notre cas, le bruit éventuel doit être supprimé, les paramètres telles que la bande équivalente de bruit B_n et autres paramètres relatifs doivent être considérés.

Généralement, le rapport signal-sur-bruit d'entrée SNR_1 doit être estimé à partir d'une mesure W de la gigue de phase d'entrée θ_{n1}. Dans notre application, on estime avoir une gigue de phase d'entrée maximale de $45°$, soit $\theta_{n1} = 0.785$ rd. La variance de bruit $n'(t)$ selon l'expression (II.42) s'élève donc à $\sigma_n^2 = \overline{\theta_{n1}^2} = 0.617$. On en déduit le SNR_1 de l'expression (II.45) :

$$SNR_1 = \frac{1}{2\overline{\theta_{n1}^2}} = 0.81$$

La bande équivalente de bruit $B_n$doit être spécifiée selon la formule (II.52) pour un rapport signal-sur-bruit de sortie $SNR_2 \geq 4$ (6 dB). Dans cette formule, la bande passante B_i du signal source ou du pré-filtre éventuel, doit être connue. Dans notre cas, un signal carré d'entrée à trois fréquences successives sont choisies le long de la caractéristique utile du VCO, Figure II.27. Sa bande minimale se borne successivement aux fréquences d'entrée choisies, 100kHz, 200kHz et 625kHz, qui correspondent aux fondamentaux dont les harmoniques impaires sont à négliger. La restitution éventuelle de la forme carrée se fait par une procédure classique de mise en forme.

Pour la fréquence d'entrée la plus faible, $B_i = 100$ kHz, le calcul du filtre de boucle se poursuit comme suit :

Pour un SNR_2 minimal, on trouve :

$$B_n = \frac{B_i}{2}\frac{SNR_1}{SNR_2} = 10\ kHz$$

Le calcul de la fréquence naturelle ω_n de la DPLL se fait à partir de l'expression (II.50) dont le facteur d'amortissement est choisi optimal $\varepsilon = 0.7$, et lui correspond une bande $B_n = 0.53\,\omega_n$. La valeur de la fréquence naturelle devient $\omega_n = 118.5\,10^3$ rd/s, de laquelle se fixent les propriétés dynamiques du système DPLL dont dépend étroitement son utilisation pratique, comme la plage de capture rapide $= 166kHz$ ($\Delta\omega_L = 10^6$ rd/s), la rangée de poursuite $= 261.4kHz$ ($\Delta\omega_P = 1.6\,10^6$ rd/s), ou bien le temps de verrouillage $T_L = 53\mu s$ pour une capture rapide, etc.

Il devient aisé maintenant de calculer les composants du filtre de boucle sélectionné, Figure II.28.

Ainsi, étant donnés les paramètres ω_n, ε, K_0, et K_d, on déduit les constantes de temps du filtre à partir de l'équation (II.21) :

$$\tau_1 + \tau_2 = \frac{K_0 K_d}{N\omega_n^2} = 57\ \mu s \quad \text{et} \quad \tau_2 = \frac{2\varepsilon}{\omega_n} = 11.8\ \mu s \quad \text{soit} \quad \tau_1 = 45.2\ \mu s\,.$$

Ainsi pour $C = 1$nF, on a $R_1 = \tau_1/C \approx 45$kΩ, et $R_2 = \tau_2/C \approx 12$kΩ.
Afin de ne pas amortir la sortie du filtre, un circuit d'adaptation au premier étage du VCO est prévu. La première fréquence de coupure étant de $f_c = \frac{1}{2\pi(\tau_1+\tau_2)} = 2.8kHz$, le signal du filtre est donc BF et un simple ampli opérationnel de type 741 monté en suiveur suffit comme étage adaptateur, Figure II.38.

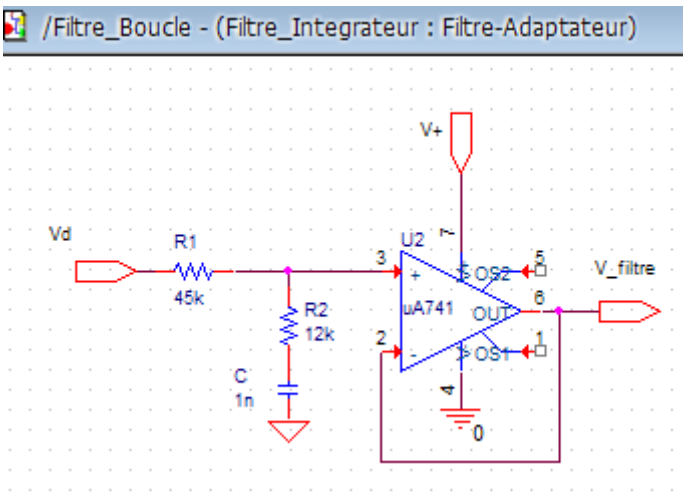


Figure II.38 : Filtre de boucle.

Les trois blocs fonctionnels de la DPLL, des figures II.3, II.24, et II.38, sont assemblés en boucle fermée, comme montré en Figure II.39.

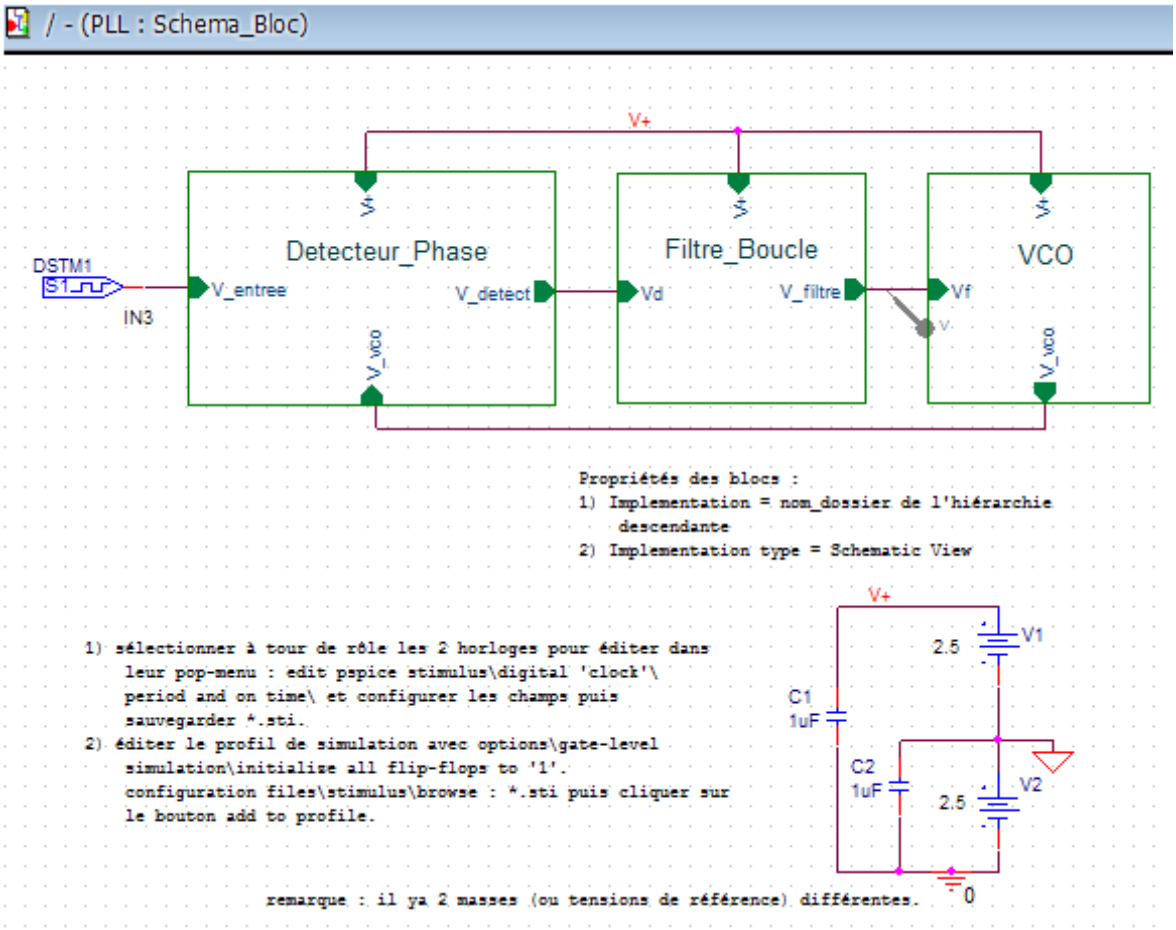
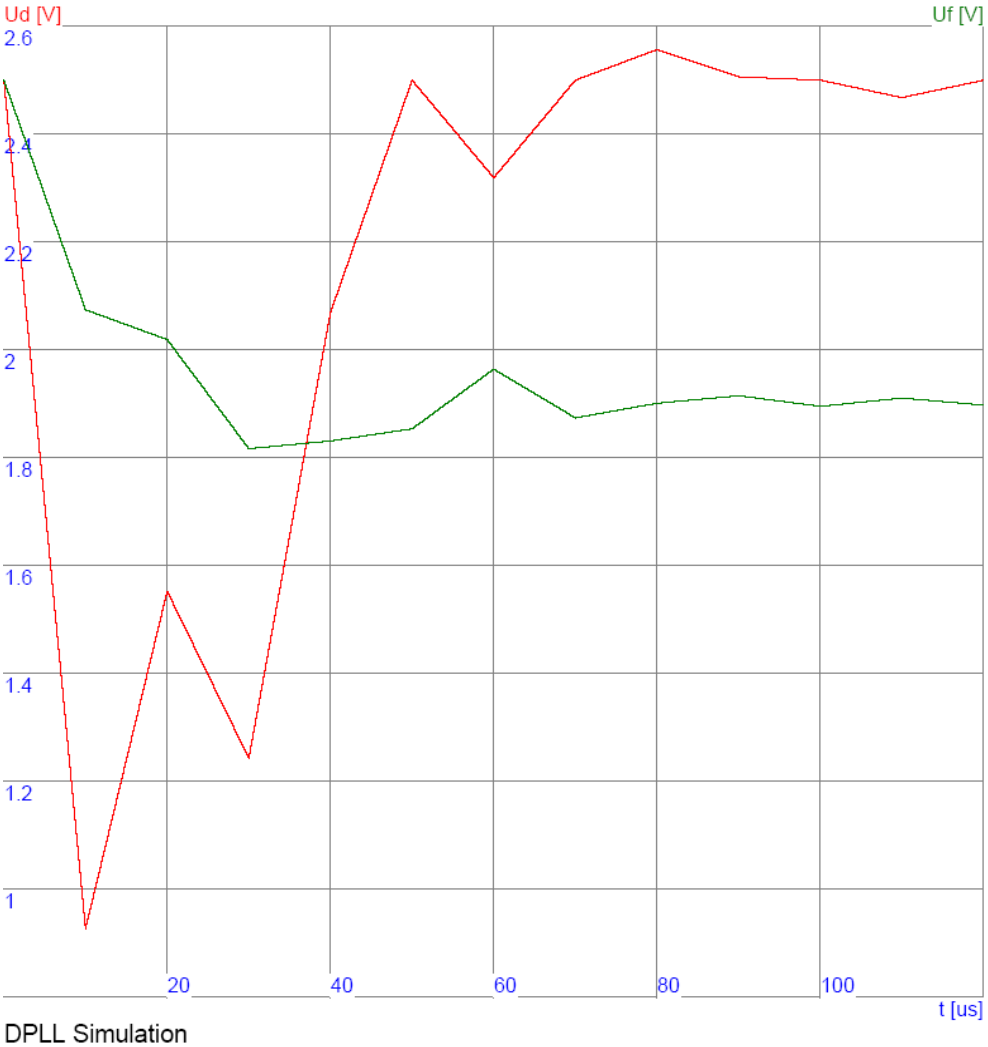


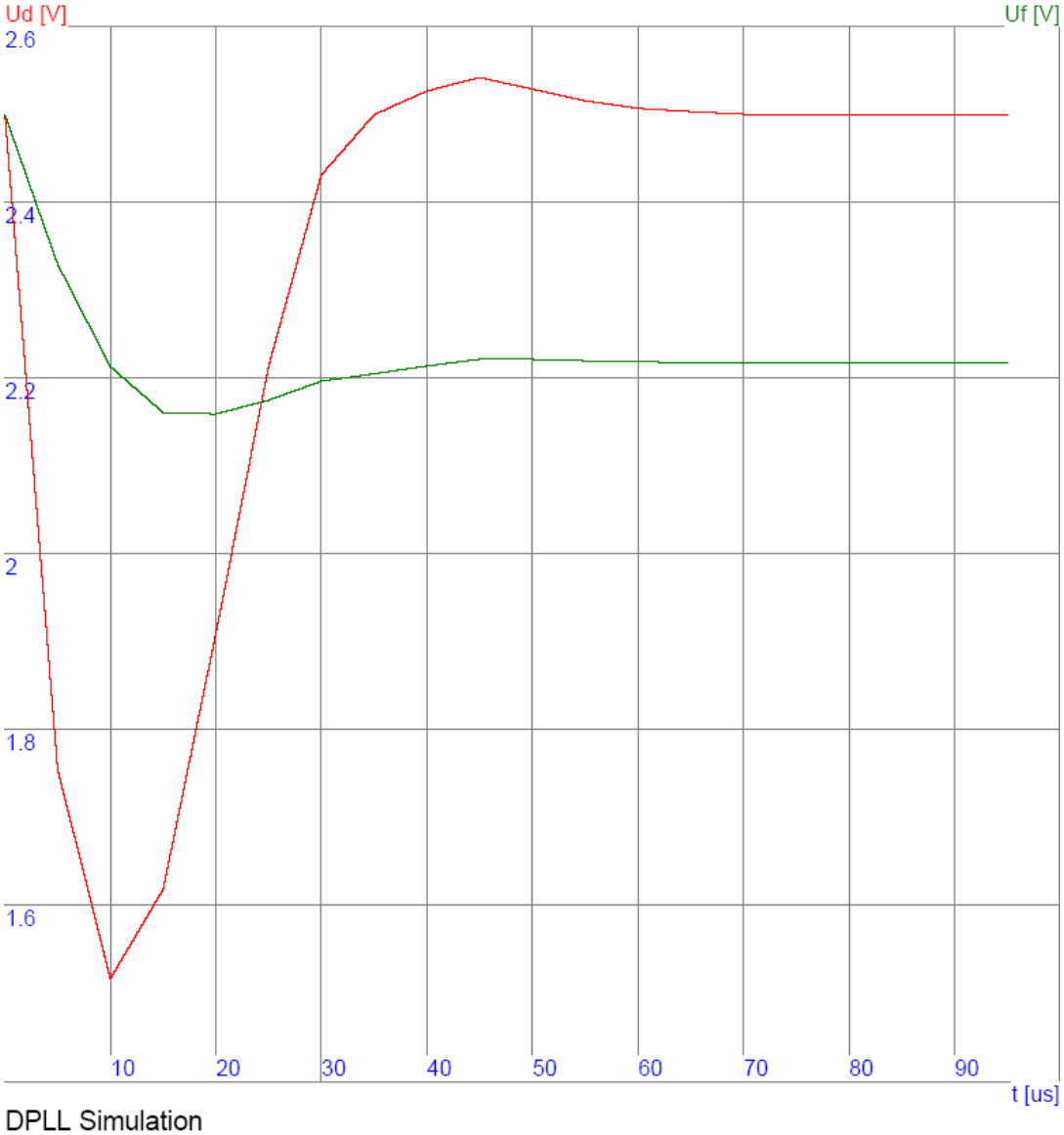
Figure II.39 : Schéma bloc de la DPLL simulée.

Les données de notre DPLL conçue étant fournies d'abord au simulateur de PLL téléchargé, ce dernier estime sa fréquence naturelle être à 31 kHz et son facteur d'amortissement à 0.63.

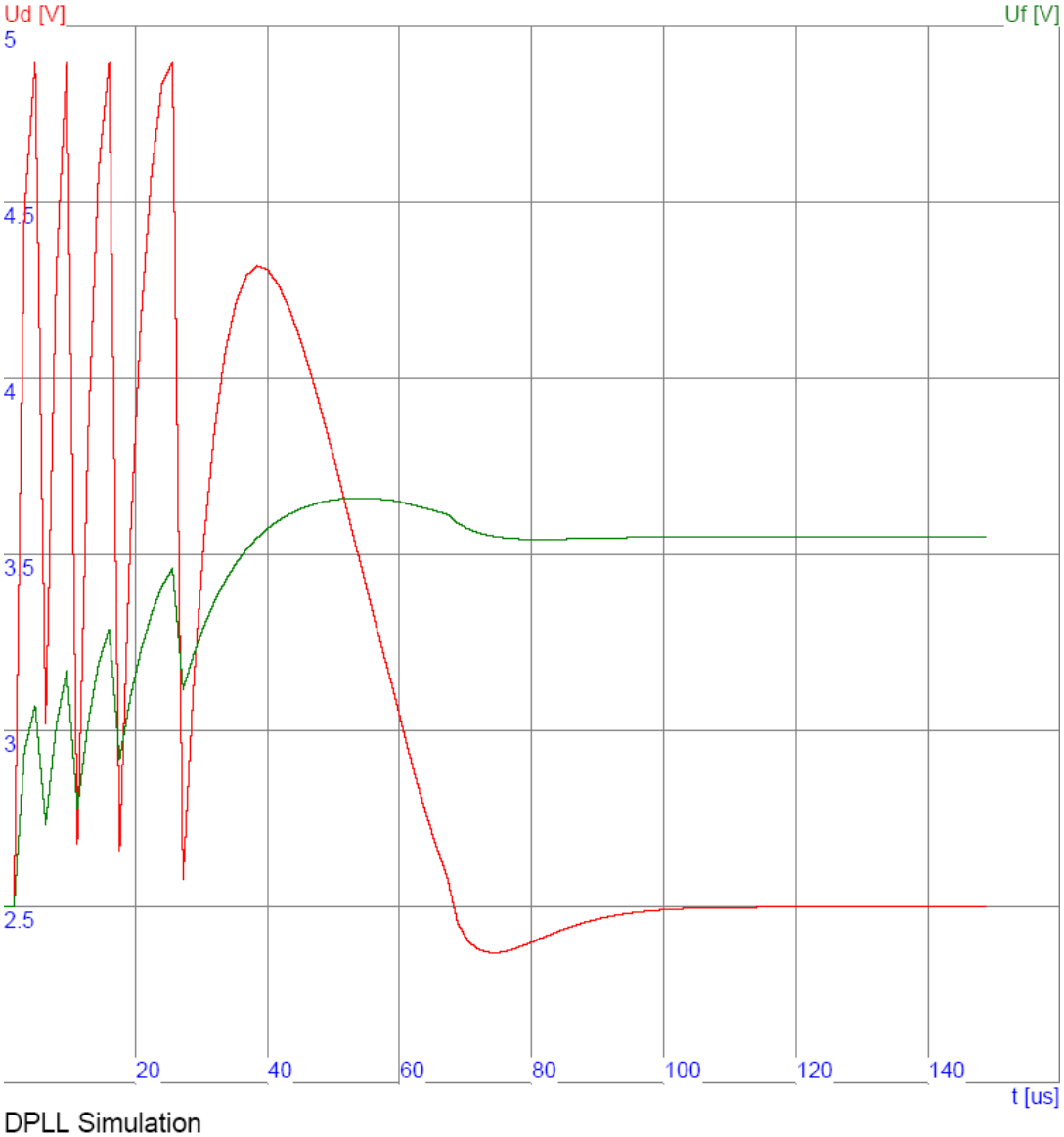
La fréquence centrale est estimée être à 290kHz. La figure II.40 rassemble trois simulations séparées dont la première est effectuée sous une fréquence d'entrée de 100kHz (a), la deuxième à la fréquence 200kHz (b), et la dernière à la fréquence 625kHz (c). Les trois réponses du filtre de boucle convergent vers des tensions moyennes fixes de 1.9V, 2.23V, et 3.55V respectivement.



(a) DPLL Simulation
12-12-2015 11:39
PD = Phase-Frequency Detector
LF = passive lag
OSC = VCO
Center frequency = 290000 Hz
Response on frequency step -190000 Hz



(b) DPLL Simulation
12-12-2015 11:40
PD = Phase-Frequency Detector
LF = passive lag
OSC = VCO
Center frequency = 290000 Hz
Response on frequency step -90000 Hz



(c) DPLL Simulation
12-12-2015 11:42
PD = Phase-Frequency Detector
LF = passive lag
OSC = VCO
Center frequency = 290000 Hz
Response on frequency step 335000 Hz

Figure II.40 Tracés de tension du détecteur de phase (rouge) et du filtre (vert), pour les fréquences d'entrée 100kHz (a), 200kHz (b), et 625kHz (c).

Par ailleurs, la simulation sur Pspice d'un tel assemblage de circuits analogiques et logiques de la Figure II.39, nécessite certaines étapes de configuration du profil de simulation.

Il faut ainsi, remplir certaines conditions au niveau de la fenêtre de configuration du profil de simulation :

- Sous l'onglet "Options", catégorie "Gate-level Simulation" ; choisir le mode timing typique avec "Initialize all flip-flops to" 0 ou 1.
- Mettre l'option "Default I/O level for A/D interfaces" sur niveau 2.

La figure II.41 rassemble les trois simulations séparées. La première avec une entrée à la fréquence 625kHz (tracés du haut), la deuxième à la fréquence 200kHz (tracés du milieu), et la dernière à la fréquence 100kHz (tracés du bas). Les trois réponses du filtre de boucle convergent vers des tensions moyennes fixes, mais ne sont pas fines à cause des impulsions parasites arrivant aux mêmes fréquences d'entrée. Ces impulsions sont triangulaires dans le cas de la réponse propre à la fréquence d'entrée 625kHz (courbe bleue). La différence constatée entre ces réponses et celles issues du simulateur de PLL téléchargé, réside dans l'option de filtrage des tracés que possède ce dernier pour ne retenir que leurs tendances moyennes. Les valeurs initiales et finales ne sont pas les mêmes, du fait que la tension de repos du filtre au départ n'est pas considérée à une même valeur suite à la caractéristique non linéaire du VCO, figure II.27, alors que celle du simulateur téléchargé est prise parfaitement linéaire.

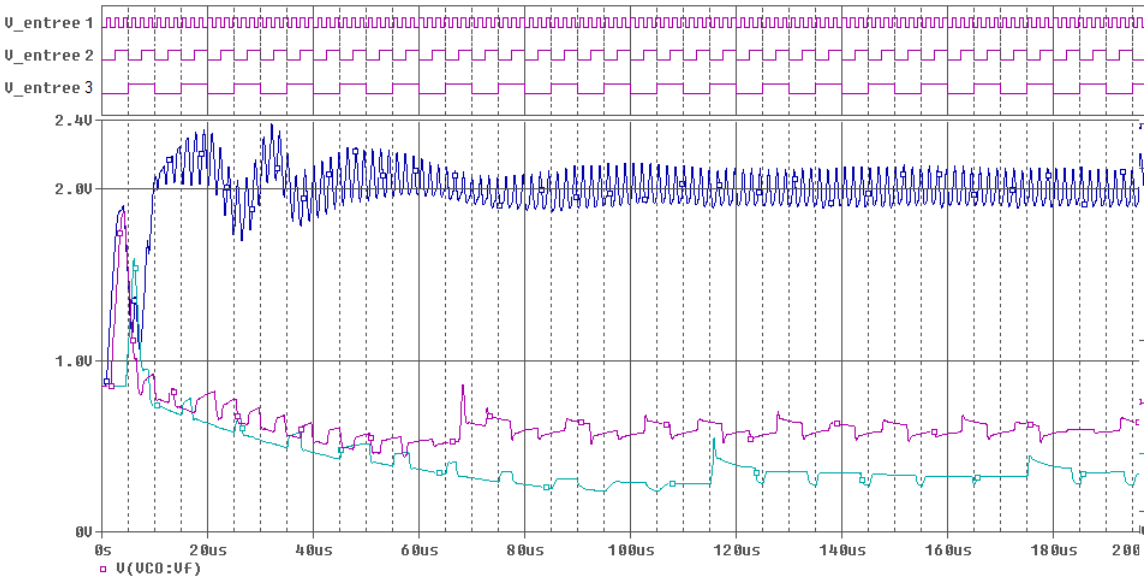


Figure II.41 : Réponses du filtre de boucle pour trois différentes fréquences d'entrée.

Les résultats obtenus par Pspice prouvent que le filtre de boucle ne joue pas bien son rôle, même si ses deux fréquences de coupure sont assez basses. Pour améliorer le comportement du filtre vis-à-vis des signaux à filtrer, nous avons pris en considération les faits suivants :

- Garder la valeur calculée de la somme des deux résistances, $R_1 + R_2 = 57$ kΩ.
- Atténuer les impulsions parasites arrivant à la même fréquence d'entrée, par le biais du condensateur C du filtre qui doit dans le cas de la fréquence d'entrée la plus élevée, $625kHz$, assurer une faible impédance de l'ordre de 50Ω. Soit, $C = 5$nF qu'on prend 4.7nF.
- L'atténuation apportée par la valeur de C à la fréquence de $625kHz$ est d'autant meilleure que $\tau_2 << \tau_1$. Ainsi pour $R_2 = 100\Omega$, on a $R_1 = 57$kΩ.

Avec ces nouvelles valeurs attribuées aux composants du filtre de boucle, la DPLL conçue, offre de meilleures réponses de son filtre aux fréquences d'entrée, 100kHz, 200kHz, et 625kHz, dont on remarque la disparition des impulsions parasites, figures II.42 à II.44.

Dans la figure II.42, deux mesures de la tension de filtre en réponse à la fréquence d'entrée de 100kHz sont effectuées au début et à la fin de simulation : $V_f = 846.537\text{mV}$ à $t = 0$, et $V_f = 330.833\text{mV}$ à $t = 2\text{ms}$.

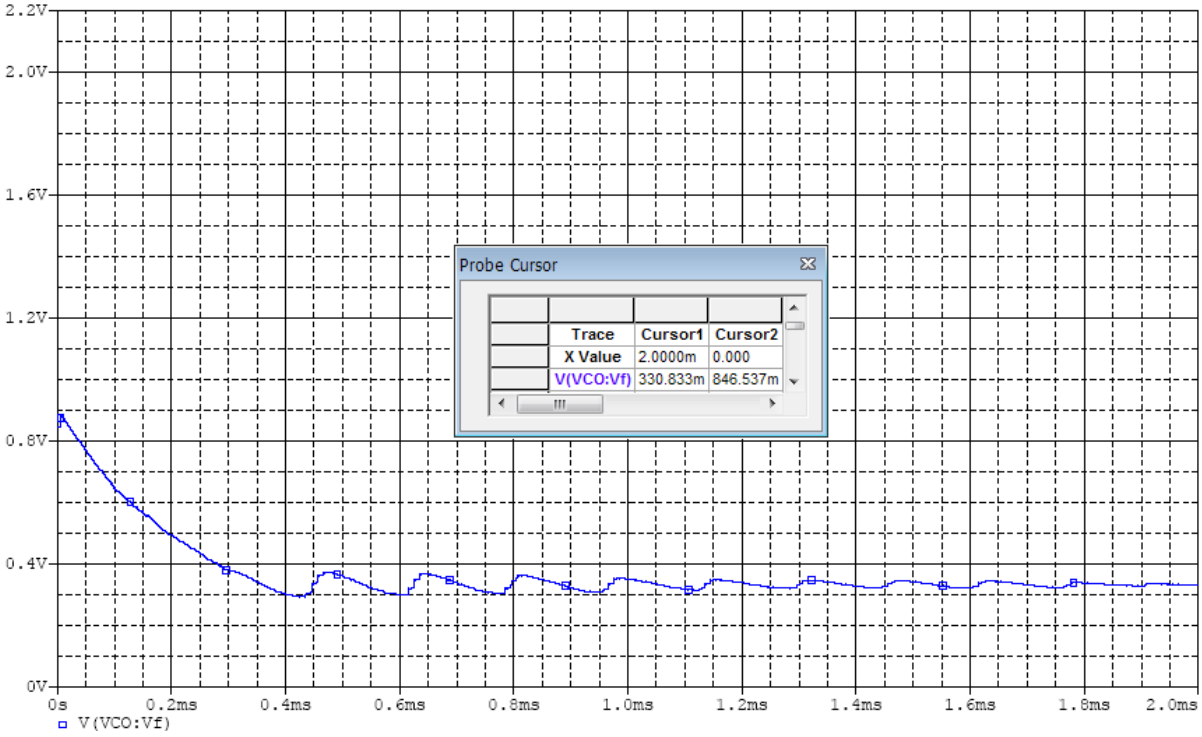


Figure II.42 : Réponse du filtre à la fréquence d'entrée de 100kHz.

Dans la figure II.43, les deux mesures de la tension de filtre en réponse à la fréquence d'entrée de 200kHz et prises au début et à la fin de simulation sont : $V_f = 846.537\text{mV}$ à $t = 0$, et $V_f = 599.292\text{mV}$ à $t = 2\text{ms}$.

De même en figure II.44, les mesures de la tension de filtre en réponse à la fréquence d'entrée de 625kHz et correspondant au début et à la fin de simulation sont : $V_f = 846.537\text{mV}$ à $t = 0$, et $V_f = 1.9589\text{V}$ à $t = 2\text{ms}$.

Les fréquences extrêmes d'entrée, 625kHz et 100kHz, présentent respectivement des sauts de fréquence relativement à f_0, de $\Delta f = 335kHz$ et $\Delta f = 190kHz$, qui sont au-delà de la plage de capture rapide, $166kHz$. Les temps de capture lente théoriques correspondants sont respectivement, $293\mu s$ et $146\mu s$.

La fréquence d'entrée à $200kHz$ présente quant à elle, un saut fréquentiel de $\Delta f = 90kHz$ se trouvant à l'intérieur de la plage de capture rapide, et offre ainsi, un temps de réponse le plus faible.

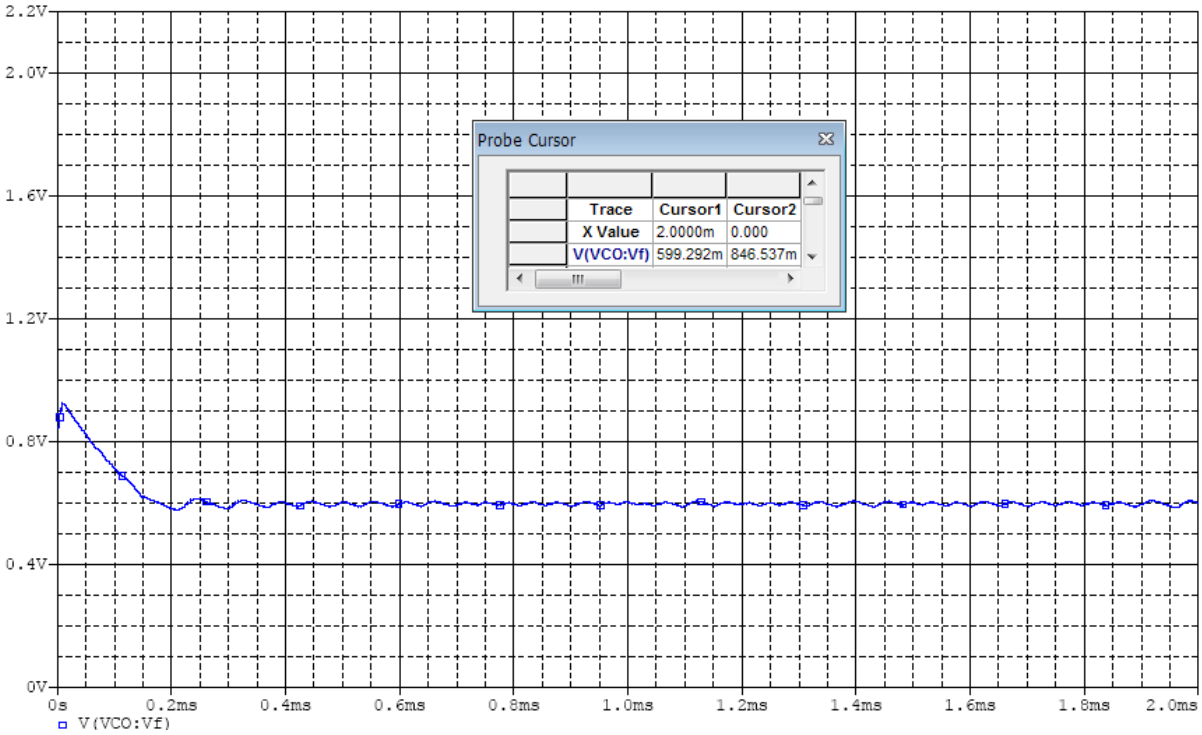


Figure II.43 : Réponse du filtre à la fréquence d'entrée de 200kHz.

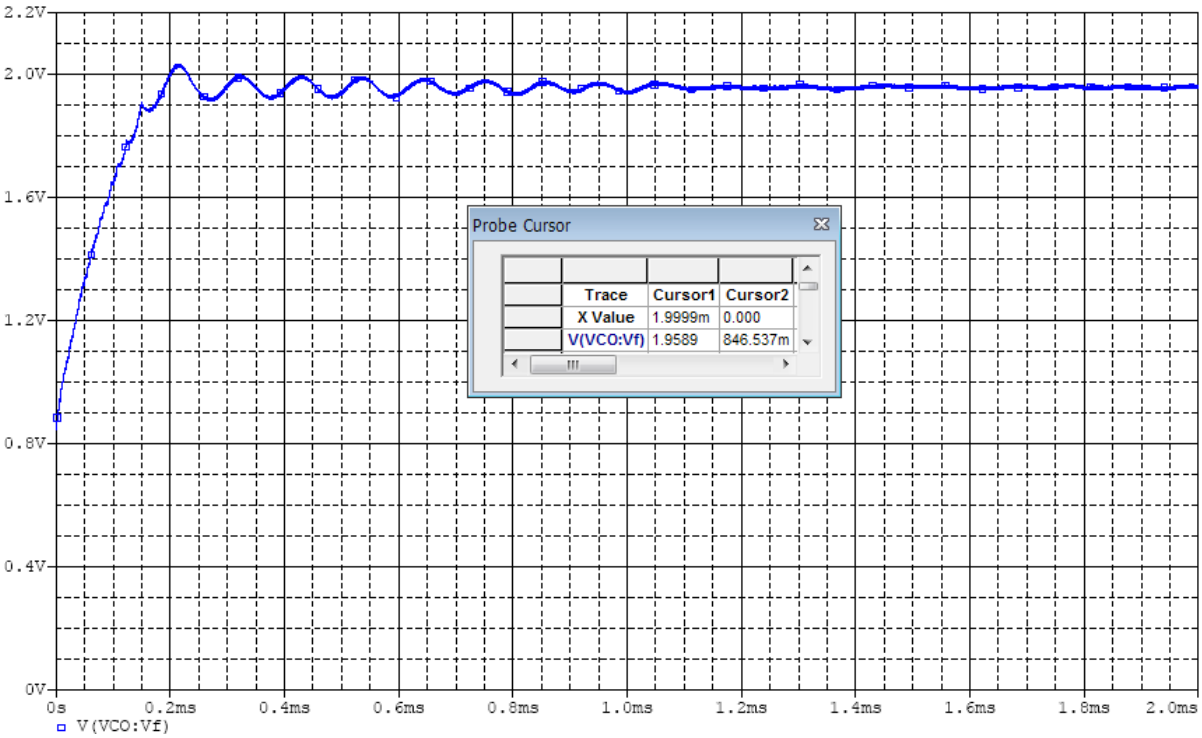


Figure II.44 : Réponse du filtre à la fréquence d'entrée de 625kHz.

Toutes les réponses du filtre présentent un point commun à leur départ, qui revient à la tension de filtre au repos, $V_f = 846.537mV$. Cette tension de repos en sortie du filtre de boucle fixe une fréquence propre ou de repos du VCO, qui est $f_0 = 289.075\text{kHz}$ mesurée sur la caractéristique tension-fréquence de la Figure II.27.

Une quatrième entrée à la fréquence propre 289.075kHz a été rajoutée, dont la réponse de filtre se trouve en Figure II.45, dans laquelle sont aussi mentionnées, les mesures au début et fin de simulation : $V_f = 846.537\text{mV}$ à $t = 0$, et $V_f = 846.714\text{mV}$ à environ 2ms. La tension du filtre rejoint sa valeur de repos pratiquement en 0.4ms mais demeure en oscillation autour, à de très faibles amplitudes.

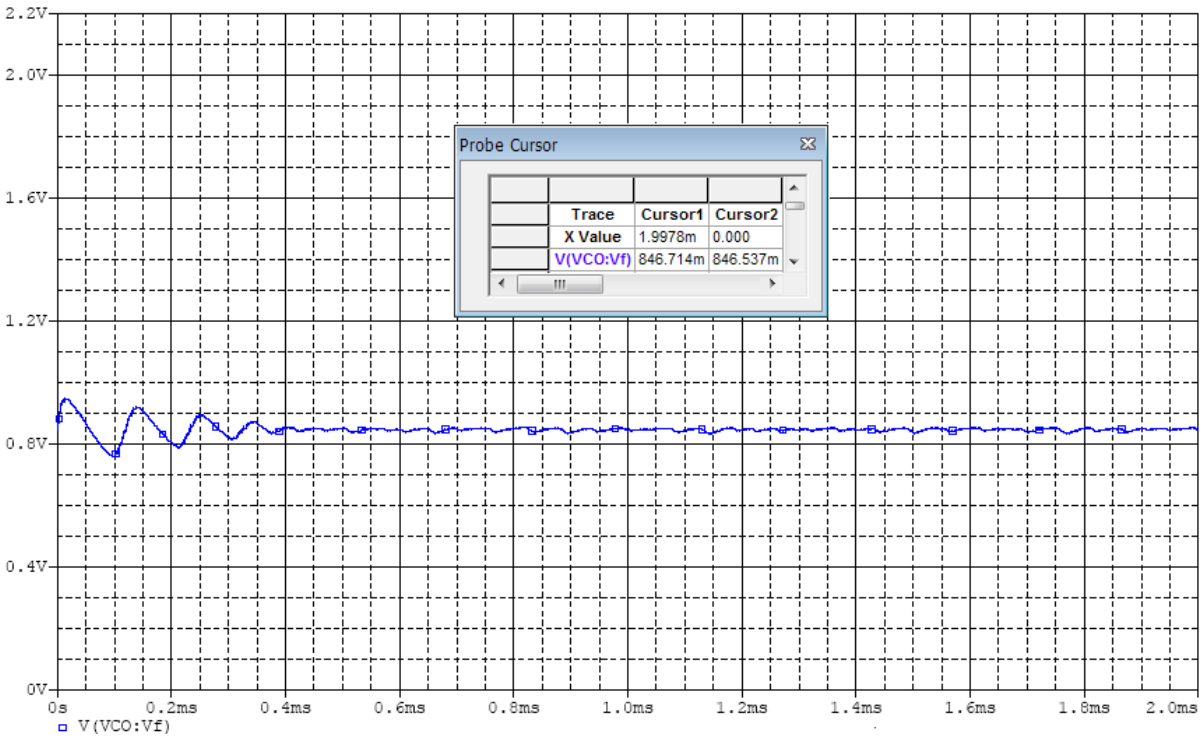
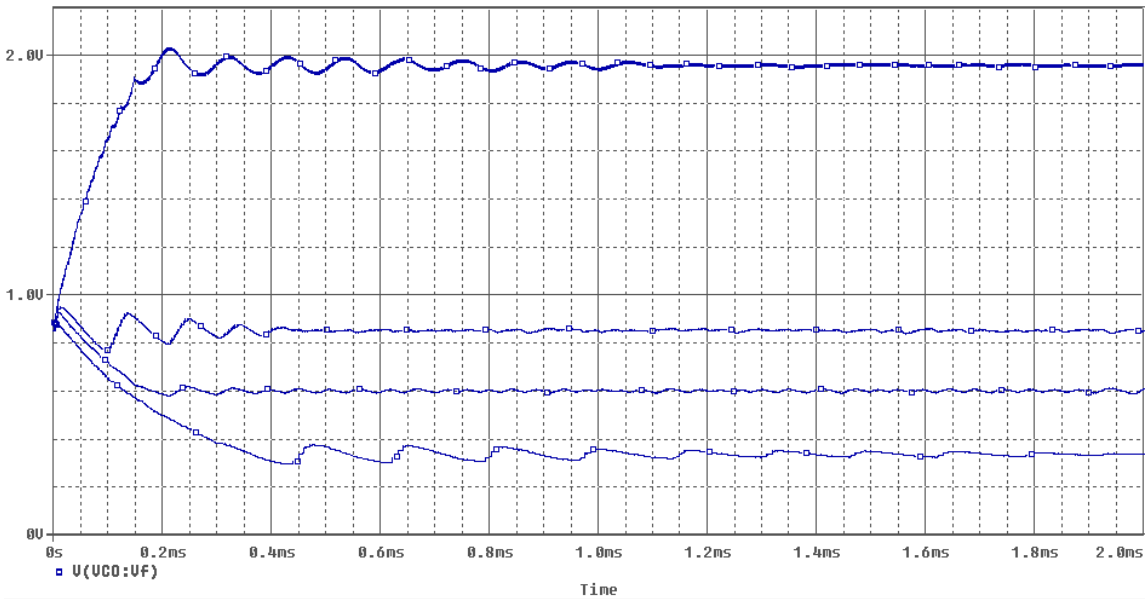


Figure II.45 : Réponse du filtre à la fréquence d'entrée de 289kHz.

Les réponses du filtre aux entrées 625kHz, 289kHz, 200kHz, et 100kHz, sont respectivement disposées du haut vers le bas dans la figure II.46.

Au bas de cette même figure, sont réunies les fréquences atteintes du VCO après 2ms de simulation, en réponse aux quatre signaux indiqués du filtre de boucle. De faibles écarts avec les fréquences d'entrée sont constatés. Cela est dû à ce qu'une fois la fréquence du VCO arrive à être égale à celle d'entrée, la pompe de charge reste dans son état à haute impédance sous lequel, le condensateur C_1 du VCO ne peut garder indéfiniment sa charge à cause de fuites. La tension baissant jusqu'à ce qu'elle soit perceptible par le Trigger, fait qu'une nouvelle charge suffisante de C_1 est lancée par ce dernier. Le cycle recommence de nouveau, faisant ainsi varier la fréquence du VCO dans un faible intervalle autour de la fréquence-consigne d'entrée.



Measurement Results

Evaluate	Measurement	Value
☑	1/ Period_XRange(V(Detecteur_Phase.V_vco),1.994ms,2ms)	625.78223k
☑	1/ Period_XRange(V(Detecteur_Phase.V_vco),1.98ms,2ms)	290.03066k
☑	1/ Period_XRange(V(Detecteur_Phase.V_vco),1.99ms,2ms)	201.40176k
☑	1/ Period_XRange(V(Detecteur_Phase.V_vco),1.95ms,2ms)	100.10812k

Figure II.46 : Réponses réunies du filtre amélioré pour quatre différentes fréquences d'entrée.

La fréquence propre du VCO peut être ajustée à l'aide d'une tension continue réglable, superposée à la tension du filtre de boucle à l'aide d'un circuit additionneur non inverseur, en place et lieu du suiveur de la figure II.38.

CHAPITRE III : CONCEPTION D'UNE SPLL

Au temps des microcontrôleurs et des DSPs, il devient légitime d'implémenter un système PLL par software. Dans ce contexte, les fonctions de la PLL seront réalisées par un programme. Le concepteur de la SPLL remplace les composants électroniques par des microsecondes de temps de calcul. Comme dans une PLL hardware où le nombre de composants augmente avec le niveau de performance demandé, le nombre d'instructions augmente avec la complexité des algorithmes d'une PLL software requise. Notre objectif dans ce chapitre est d'implémenter la DPLL conçue précédemment à l'aide d'une carte DSP TMS320C6713. Après un bref aperçu sur cette carte, nous présentons la méthode adoptée pour l'implémentation de notre DPLL, suivie des organigrammes qui ont servi à l'élaboration de notre programme.

III.1 Compromis soft-Hard

Il est évident, qu'une SPLL peut devenir concurrentielle à une solution hardware uniquement si les algorithmes requis s'exécutent assez rapidement sur la plate forme où réside son programme. La haute vitesse d'exécution et le bas prix des PLLs disponibles en circuits intégrés, rendent plus difficile à une SPLL de concurrencer avec son homologue hardware. Néanmoins, les SPLLs peuvent offrir des avantages particuliers spécialement lorsque la puissance de calcul est disponible [1].

Selon le théorème d'échantillonnage, l'algorithme d'une SPLL doit s'exécuter deux fois ou plus dans chaque cycle du signal de référence. Par exemple, si la fréquence de référence est 100kHz, l'algorithme doit s'exécuter 200000 fois par seconde dans le meilleur des cas, ce qui ne laisse guère plus de $5\mu s$ pour chaque itération.

Les DSPs en plus de leur vitesse d'exécution relativement à leur fréquence d'horloge, offrent une architecture Harvard avec pipeline. L'architecture Harvard signifie que le DSP dispose de mémoires séparées pour le programme et les données, et de là, peut rechercher une instruction et des données durant un même cycle machine. Le terme

pipeline implique que les unités arithmétiques et logiques de la machine sont totalement découplées, de manière à ce qu'une recherche d'une instruction, une recherche de quelques données, une ou plusieurs additions en virgule flottante, une ou plusieurs multiplications en virgule flottante, un ou plusieurs décodages d'instruction, une ou plusieurs opérations registre-à-registre, et peut être d'avantage encore, se font en un seul cycle machine. Ceci, améliore grandement la capacité de calcul mais au détriment d'un prix de revient plus cher que celui d'un simple microcontrôleur.

Dans la prochaine section, nous discuterons les étapes requises pour vérifier la faisabilité et l'économie d'une réalisation SPLL.

III.2 Faisabilité d'une conception de SPLL

Un maximum de degré de liberté est offert dans une conception d'une SPLL à cause de sa maniabilité à volonté pour qu'elle puisse répondre comme une LPLL analogique linéaire ou une DPLL numérique, voire exécuter une nouvelle fonction qu'aucune de ces variantes hardwares n'est capable de faire. Une implémentation software de PLL peut être vérifiée si elle est économiquement justifiée, suite aux étapes suivantes de sa conception [1] :

Etape 1 : Définition de l'algorithme SPLL par des équations algébriques et/ou logiques, ou à l'aide d'organigrammes.

Etape 2 : Définition du langage qui doit encoder l'algorithme défini. L'effort de programmation est minimisé lorsqu'un langage de haut niveau comme C ou C^{++}, FORTH, ADA, est utilisé. Pas tous les compilateurs ne sont capables de générer un code assembleur optimisé en temps d'exécution. Dans le cas de DSP utilisant les techniques de Pipeline, le compilateur doit générer un code assembleur parallèle. C'est-à-dire, un programme assembleur dont un nombre d'instructions différentes est exécuté en même temps. Dans le cas d'absence de compilateurs performants, le concepteur est forcé à écrire le programme directement en langage assembleur. Cependant avec un DSP à calcul parallèle, la tâche ne devient pas simple. Certaines cartes DSP offrent dans leurs bibliothèques, des fonctions écrites en assembleur qui peuvent être utilisées pour assurer des tâches élémentaires comme un filtrage numérique ou autre. Quelque soit le langage utilisé, le code en assembleur doit être disponible afin d'estimer le temps d'exécution de l'algorithme SPLL.

Etape 3 : Estimation de l'exécution en temps réel, suite au temps d'exécution estimé du programme. Si par exemple, le temps d'exécution total de l'algorithme SPLL est de $50\mu s$, et que deux itérations sont requises pendant un cycle du signal de référence, au moins $100\mu s$ de temps de calcul est nécessaire durant une période de référence. Si l'on rajoute le temps d'effectuer des opérations d'entrée-sortie, l'exécution en temps réel peut s'abaisser au dessous de 10KHz dans cet exemple.

Etape 4 : Test du temps réel afin de s'assurer que le système opère comme planifié.

III.3 Support hard et soft du DSP

Le DSP choisi pour notre travail, est à virgule flottante sur 32 bits. L'application est développée d'abord sur un simulateur de Texas Instrument utilisant le DSP TMS320C6713 à architectures VelociTI™ et VLIW avancé (Very Long Instruction Word) de la famille C6000 [42]. Un pipeline à 4 phases permet une exécution en parallèle de huit instructions par cycle. Ceci équivaut à 2 milliards d'opérations en virgule flottante par seconde à la fréquence horloge de 250 MHz [42].

III.3.1 Interruptions

Il y a huit registres de contrôle pour gérer les interruptions. Le DSP supporte 16 interruptions à priorité. L'interruption à plus haute priorité est INT_00 qui est dédiée à l'interruption RESET, alors que l'interruption la plus basse en priorité est INT_15. Les quatre premières interruptions (INT_00-INT_03) sont non masquables. Les autres interruptions (INT_04-INT_15) sont masquables et correspondent par défaut aux sources désignées dans le tableau III.1. L'ensemble des périphériques du C6713 procure jusqu à 32 sources d'interruptions, alors que le CPU n'en peut gérer que 12. Le sélecteur d'interruptions permet alors de choisir et d'établir l'ordre de priorité des interruptions.

Numéro d'INT.	Interruption	Source d'interruption
INT_00	RESET	–
INT_01	NMI	–
INT_02	–	Réservée
INT_03	–	Réservée
INT_04	EXT_INT4	Interruption externe au pin 4
INT_05	EXT_INT5	Interruption externe au pin 5
INT_06	EXT_INT6	Interruption externe au pin 6
INT_07	EXT_INT7	Interruption externe au pin 7
INT_08	EDMA_INT	Interruption EDMA (canal 0 à 15)
INT_09	XINT0	Interruption McBSP0 (transmetteur)
INT_10	SD_INT	Interruption EMIF-SDRAM-Timer
INT_11	RINT0	Interruption McBSP0 (récepteur)
INT_12	XINT1	Interruption McBSP1 (transmetteur)
INT_13	DSP_INT	Interruption HPI (vers DSP)
INT_14	TINT0	Interruption Timer0
INT_15	TINT1	Interruption Timer1
–	RINT1	Interruption McBSP1 (récepteur)
–	–	Réservée

Tableau III.1 Interruptions du DSP (sources par défaut).

III.3.2 Logiciels d'accompagnement

Des outils softwares sont intégrés dans l'environnement de développement CCS [43]. Ces outils constituent le système opératoire DSP/BIOS et ses librairies [44, 45]. Le DSP/BIOS est un ensemble d'API (Application Programming Interface) et d'outils pour le développement d'applications en temps réel sur DSP. Les API constituent des

fonctions exécutables en temps réel et pouvant être utilisées efficacement dans des fichiers sources en langage C/C++.

La génération automatique de code peut être effectuée par un outil DSP/BIOS spécial de configuration, présentant une interface graphique. La librairie CSL (Chip Support Library) [46] est une catégorie de cette interface graphique. Elle permet de configurer aisément les périphériques internes en programmant leurs registres, voire leurs champs en bits, dans des pages de propriétés. Le code de configuration résultant automatiquement, est inséré dans un fichier source en C prêt à être employé dans l'application.

Le périphérique interne au DSP utilisé dans notre application est le Timer. C'est un compteur programmable à 32 bits. L'horloge du Timer peut être dérivée à partir d'une source externe ou interne (FCPU / 4). Les registres à 32 bits d'un Timer sont au nombre de trois [47].

Ainsi, pour le Timer par exemple, l'interface CSL inclut deux sections :

- Un gestionnaire de configuration : permet d'instancier des objets de configuration (en nombre illimité) à partir d'une page de propriétés.

- Un gestionnaire de ressource : permet de sélectionner un périphérique et lui associer un objet de configuration. Pour un objet de configuration donné, l'option « Show Dependency » du menu principal montre quel périphérique l'utilise.

Le DSP/BIOS gère d'autre part, quatre niveaux distincts d'amorçage d'exécution (à priorités décroissantes) : les routines de service d'interruptions hardwares, les interruptions softwares, les tâches, et les fonctions IDLE de fond (boucles sans fin avec économie d'énergie).

Parmi les autres catégories constituant l'outil graphique de configuration DSP/BIOS, et qui sont utilisées dans notre application SPLL, on trouve la catégorie «Scheduling» dans laquelle on a :

Dans le Module «HWI-SR Manager», Voie «HWI_RESERVED1», est programmée la fonction _RTDX_Poll pour la communication RTDX (real time data exchange). Les canaux RTDX sont unidirectionnels dans leurs transferts de données. Le transfert de données est asynchrone. Le DSP appelle les fonctions de la librairie RTDX pour envoyer ou recevoir des données du PC via l'interface JTAG, et ce, de façon transparente à l'application du DSP. Dans la plupart des applications, le RTDX utilise des interruptions pour accomplir le transfert des données. Ce transfert s'opère en temps réel pendant que l'application s'exécute. Sur PC, la librairie RTDX opère en conjonction avec l'environnement CCS de la carte DSP. Le choix des logiciels d'affichage sur PC inclut : Visual Basic, Visual C++, Excel, LabView, Matlab. Dans notre cas, Matlab 7.1 s'en charge de tracer les signaux calculés par notre SPLL.

Le CCS contrôle le flux de données entre le hôte (sur PC) et la carte DSP ou son simulateur. Pour entretenir un flux de données du DSP vers le hôte, il faut déclarer un canal RTDX en sortie et utiliser des routines spécifiques appartenant à 'User Interface'

pour y écrire les données à envoyer. Ces données sont immédiatement enregistrées dans un buffer cible défini dans la librairie RTDX (côté DSP) appelé « target buffer ». Les données du buffer sont alors envoyées vers le hôte via l'interface JTAG. La librairie RTDX (côté hôte) reçoit les données à partir de l'interface, et les enregistre soit dans un buffer en mémoire, appelé « host buffer », soit dans le fichier Log_RTDX (dépendant du mode d'enregistrement spécifié sur le hôte). Le transfert de données à partir de l'application DSP vers la librairie RTDX du côté hôte est effectué en temps réel. Les données enregistrées peuvent être retrouvées par n'importe quelle application hôte permettant à l'utilisateur de visualiser ses données d'une manière plus représentative (graphique par exemple).

Similairement, le DSP peut recevoir des données de la part du hôte, mais cette possibilité n'est pas exploitée dans notre cas.

Un contrôle graphique de la configuration RTDX, existe. Elle permet de valider ou inhiber la liaison RTDX, et d'accéder à la page des propriétés de configuration RTDX.

D'autres modules existent dans la catégorie Scheduling , tels les modules d'interruptions qui sont de deux types :

Interruptions Hardwares :

Les événements hardware se déroulant dans l'environnement du DSP, peuvent déclencher une routine de service d'interruption (ISR). Les interruptions hardware sont utilisées pour des événements à temps critique, et le ISR doit être déclenché directement et traité rapidement. Par conséquent, il doit être court.

Le module HWI [44] est un gestionnaire d'interruption hardware renfermant des objets HWI prédéfinis, qui sont au nombre des 16 numéros d'interruptions de la table des vecteurs d'interruptions hardwares du tableau III.1. Il est possible de modifier les valeurs par défaut de la table des vecteurs d'interruption dans le but de réarranger les priorités des interruptions, de sélectionner une source d'interruption non utilisée, de valider/inhiber l'exécution d'interruptions, ou de choisir la polarité des interruptions externes sur les pins 4 à 7.

Avec l'outil de configuration, on peut assigner une routine à n'importe quelle interruption hardware. Cependant, certaines routines sont assignées automatiquement à certaines interruptions comme le cas de l'interruption du Timer0, configurée pour exécuter une fonction d'incrémentation temporelle du module CLK. Les routines d'interruptions peuvent être écrites en assembleur ou en C. Dans ce dernier cas, l'outil « HWI dispatcher » est fourni pour utiliser la même pile du système en incluant des macros d'encapsulation du code [44].

Interruptions Softwares :

Le module SWI [44] est un gestionnaire d'interruption software. Une interruption software est un objet qui encapsule une fonction à exécuter avec une priorité.

Le module SWI sauvegarde automatiquement les registres du processeur, avant de servir une interruption soft à haute priorité interrompant une autre à basse priorité.

Les registres sont restaurés en fin d'exécution afin d'achever la routine précédente à basse priorité, à condition que d'autres interruptions soft à plus hautes priorités n'ont pas été postées dès lors.

Vu le temps de réponse des SWIs, elles ne doivent être utilisées que pour des événements à longs traitements. Une routine ISR software est appelée lorsque un SWI est postée. L'appelant peut utiliser quelques options additionnelles dans la boîte-aux-lettres (mailbox) du ISR. Ces options sont vérifiées avant le postage du SWI. Ceci rend un ISR software plus flexible dans sa réponse à une interruption SWI. Le gestionnaire des SWI valide ces propriétés afin qu'elles puissent être opérationnelles.

III.4 Fonctionnement d'une SPLL comme DPLL

Lorsque le signal d'entrée u_1 d'une PLL est un signal binaire, il est plus adéquat d'implémenter une SPLL pour opérer comme une DPLL. Dans notre cas, l'algorithme de la SPLL utilisera les fonctions d'un détecteur fréquence-phase (DFP) de la Figure II.3, associé au filtre passif de la Figure II.28, soit en tout, une pompe de charge. Les fonctions requises sont représentées par le schéma bloc de la Figure III.1. Cela consiste en trois blocs fonctionnels : un algorithme DFP, un filtre numérique, et un oscillateur contrôlé numériquement ou DCO (digital controlled oscillator), [1].

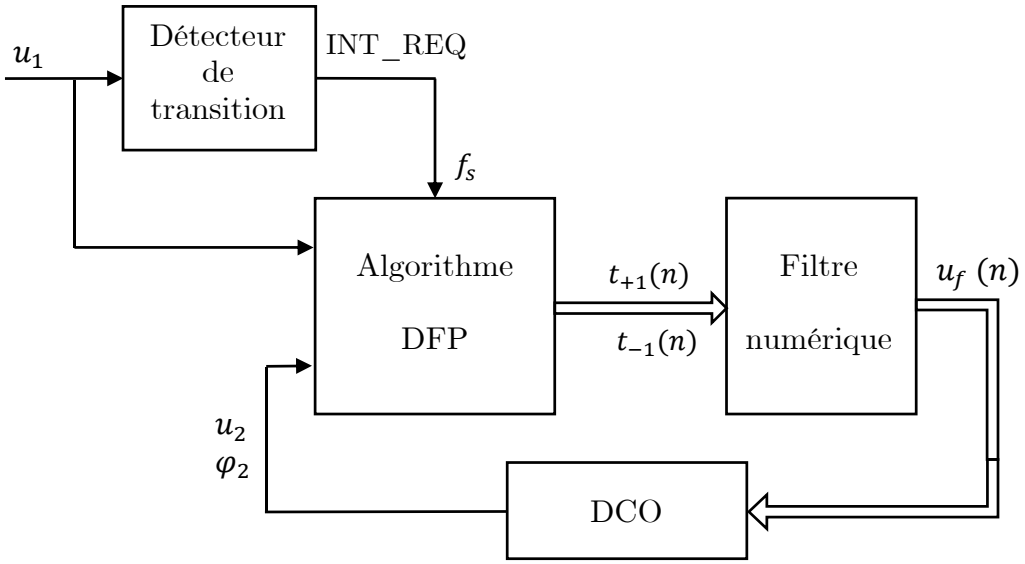


Figure III.1 : Schéma-bloc des opérations d'une SPLL opérant en tant qu'une DPLL.

Le filtre numérique doit opérer de la même manière que le filtre passif de la DPLL, Figure II.28. Le seul signal qui existe physiquement, est le signal d'entrée $u_1(t)$ à forme carrée dont la fréquence peut varier dans la plage fréquentielle du DCO. Le signal fictif $u_2(t)$ de sortie de la SPLL, doit aussi être de forme carrée.

III.4.1 Algorithme SPLL

A partir de la Figure II.4, on sait que l'état logique de sortie du DFP dépend des transitions positives de ses deux signaux d'entrée, u_1 et u_2. Lorsque la PLL est dans son état permanent, les signaux $u_1(t)$ et $u_2(t)$ sont presque en phase. La sortie du DFP est alors à l'état 0 la majorité du temps. Si la fréquence du DCO tend à dériver, le DFP génère des impulsions de correction de sorte à ce que sa sortie devient $+1$ ou -1 sur une courte durée qui est infime relativement à une période du signal de référence. Si on veut détecter les fronts de $u_1(t)$ et $u_2(t)$ par échantillonnage de ces signaux, une bonne précision réclame une fréquence d'échantillonnage au moins 1000 fois celle du signal de référence, ce qui est difficilement réalisable [1].

Comme on a besoin de connaître les instants de commutation des fronts positifs de $u_1(t)$ et $u_2(t)$, nous utilisons les transitions (positives et négatives) de $u_1(t)$ afin de générer des requêtes d'interruption (signal INT_REQ des figures II.1 et II.2). Dès que l'interruption est reconnue, son instant d'apparition est sauvegardé. Le DSP utilisé doit contenir pour cela, un timer. Les instants de détection des interruptions sont appelés $t(0)$, $t(1)$, .., $t(n)$, .., Figure III.2. Avant de discuter l'algorithme SPLL, il faut définir un certain nombre de signaux, Figure III.2.

$u_1(n)$ est la version échantillonnée du signal continu de référence $u_1(t)$ immédiatement après occurrence de la requête d'interruption. Soit au temps $t(n-1)$, $u_1(n) = 1$, et au temps $t(n)$, $u_1(n) = 0$.

$\varphi_2(t)$ est la phase continue fictive du signal DCO. $\varphi_2(n)$ est sa version échantillonnée. De même, les échantillons sont pris aux instants d'arrivée d'interruption.

$u_2(t)$ est le signal DCO continu fictif, qui sera calculé à partir de la phase $\varphi_2(t)$. Lorsque φ_2 dans sa croissance passe par 0 (ou 2π), u_2 génère son front positif. Tandis que lorsque dans sa croissance, φ_2 passe par π, u_2 génère son front négatif.

$U_d(t)$ est la sortie continue fictive du signal ou état du DFP, qui peut avoir les valeurs -1, 0, ou 1. Comme expliqué en Figure II.4, le front montant de u_1 force le DFP à aller vers le prochain état plus élevé, sauf s'il se trouve déjà à l'état $+1$, tandis que le front montant de u_2 force le DFP à aller vers le prochain état moins élevé, sauf s'il se trouve déjà à l'état -1.

$U_d(n)$ est la version échantillonnée de $U_d(t)$ et est définie comme état du DFP juste avant l'occurrence de l'interruption. Par exemple, $U_d(n-1)$ a la valeur 0, car $U_d(t)$ était à l'état 0 avant que l'interruption ne soit déclenchée à $t = t(n-1)$.

$T(n)$ est défini comme intervalle de temps, $t(n) - t(n-1)$, c'est-à-dire entre celui de l'interruption la plus récente et celui de l'interruption précédente. Lorsque $U_d(t)$ est à l'état $+1$ dans une fraction de l'intervalle $T(n)$, la durée correspondante est stockée dans la variable $t_{+1}(n)$, Figure III.2. Lorsque $U_d(t)$ est à l'état -1 dans une fraction

de l'intervalle $T(n)$, la durée correspondante est stockée cette fois-ci dans la variable $t_{-1}(n)$, Figure III.2.

$u_c(n)$ est utilisée pour décrire le signal aux bornes de la capacité fictive du filtre de boucle de la Figure II.28.

$u_f(n)$ est utilisée pour décrire le signal échantillonné en sortie du filtre numérique de la Figure III.1 (qui doit répondre comme le filtre analogique de la Figure II.28). $u_f(n)$ est presque identique à $u_c(n)$ mais peut différer lorsque le "courant" parcourt la résistance fictive R_2.

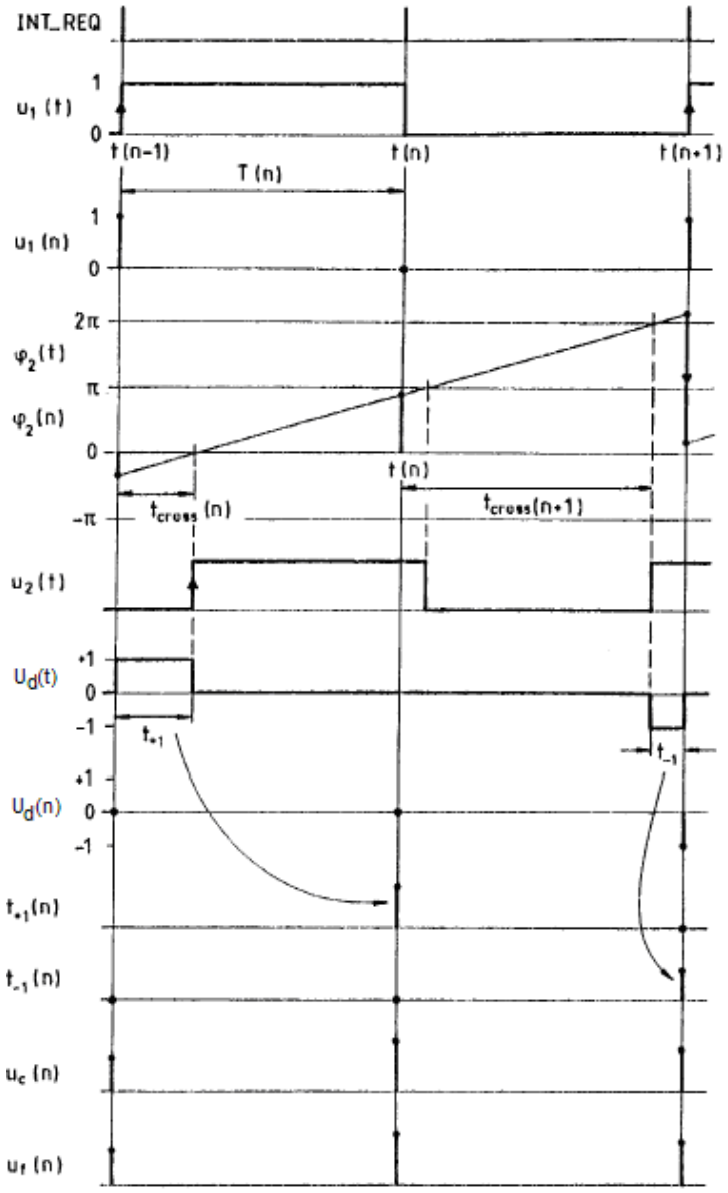


Figure III.2 : Tracé des signaux calculés par l'algorithme SPLL.

L'organigramme de la Figure III.3 montre quoi faire à chaque service d'interruption. Les signaux apparaissant dans cet organigramme sont montrés en Figure III.2. La portion la plus haute de l'algorithme SPLL est triviale et liste l'initialisation de

certaines variables, classées selon l'ordre de leur utilisation. Le programme rentre dans une boucle sans fin où il attend au début, la prochaine interruption. Lorsque l'interruption est détectée, le temps écoulé depuis la dernière interruption est mesuré, $T(n) = t(n) - t(n-1)$. La valeur courante du signal de référence $u_1(t)$ est ensuite échantillonnée, $u_1(n) = u_1(t)$. Ceci est nécessaire parce qu'on a besoin de connaître si l'on est en présence du demi cycle positif ou négatif du signal $u_1(t)$. Le DCO est supposé générer une forme d'onde carrée $u_2(t)$, qui n'est connue qu'aux instants d'échantillonnage, $u_2(n)$. Ce signal doit être calculé indirectement à partir de la phase $\varphi_2(t)$ du DCO, dont la fréquence angulaire instantanée est donnée par :

$$\omega_2(t) = K_0\, u_f(t) \tag{III.1}$$

Où initialement, $\omega_2(0) = K_0\, u_f(0)$.

De la caractéristique de la Figure II.27, on mesure $K_o = 2146.10^3$ rd/Vs et $u_f(0) = 846.54$mV correspondant à $\omega_2(0) = 1817.10^3$ rd/s où $f_2(0) \approx 290$kHz comme fréquence de repos du VCO.

En utilisant la fonction de Walsh, le signal de sortie continu $u_2(t)$ devient donc :

$$u_2(t) = W[\varphi_2(t)] \tag{III.2}$$

$$\text{Où :}\quad \varphi_2(t) = \int \omega_2(t)\, dt \tag{III.3}$$

On assume que le temps courant t est $t(n)$ qui correspond à la seconde requête d'interruption comme montrée au milieu de la Figure III.2, mais nous ne connaissons pas la valeur de la phase $\varphi_2(t)$ en ce temps. La raison à cela est simple, au temps $t = (n-1)T$, la valeur du signal de sortie du filtre numérique $u_f(n-1)$ peut être calculée et par conséquent, nous connaissons aussi la fréquence angulaire instantanée $\omega_2(n-1)$ du DCO.

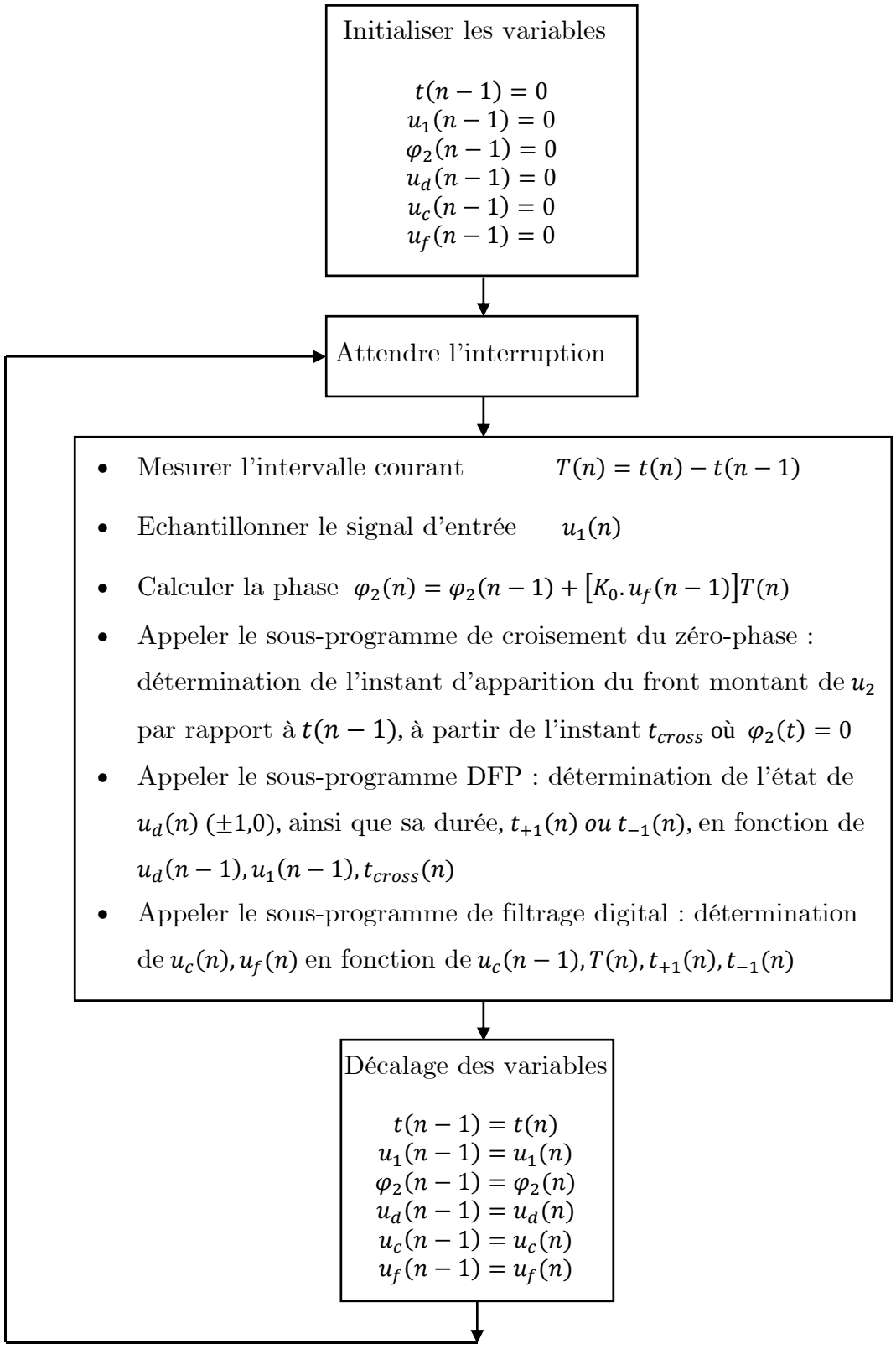


Figure III.3 : Organigramme des opérations effectuées par l'algorithme SPLL.

En admettant que $u_f(n-1)$ reste constante durant l'intervalle $(n-1)T \leq t < nT$, la phase du DCO doit changer d'un incrément égal à :

$$\Delta\varphi_2 = [K_0\, u_f(n-1)]\, T \qquad \text{(III.4)}$$

Comme nous ne connaissons pas encore au temps $t = t(n-1)$ la durée du demi cycle suivant de $u_1(t)$, nous reportons le calcul de $\varphi_2(n)$ au temps $t = t(n)$. Ainsi, au moment $t = t(n)$ on a :

$$\varphi_2(n) = \varphi_2(n-1) + \big[K_0 u_f(n-1)\big]T(n) \qquad \text{(III.5)}$$

Noter que la phase $\varphi_2(n-1)$ au temps $t = t(n-1)$ est connue, car le signal de phase est calculé récursivement avec la valeur initiale $\varphi_2(0) = 0$ à $t = 0$.

On note aussi qu'en Figure III.2, l'allure de $\varphi_2(t)$ est représentée théoriquement par des segments de droite, mais l'allure générale de $\varphi_2(t)$ suit l'intégrale (donc filtrage passe-bas) du signal filtré qui lui, présente la forme des figures II.7 et II.8, ou des figures II.33 et II.34 (voir aussi le début transitoire de la réponse du filtre pour les cas extrêmes à $625kHz$ et $100kHz$ de la Figure II.41).

Dans la suite de l'organigramme de la Figure III.3, trois sous-programmes sont appelés successivement avant de mettre à jour les variables d'initialisation et revenir à l'état d'attente d'une nouvelle interruption.

III.4.2 Sous-programmes fonctionnels de la SPLL

Afin de ne pas charger l'organigramme de l'algorithme SPLL, Figure III.3, les organigrammes des sous-programmes de croisement du zéro-phase en Figure III.4, du DFP en Figure III.6, et du filtre numérique en Figure III.9, sont illustrés séparément.

L'organigramme de la Figure III.4 se charge de déterminer si le signal fictif $u_2(t)$ a montré un front positif dans l'intervalle $T(n)$. C'est le cas lorsque le signal continu de phase $\varphi_2(t)$ croise la valeur 0 (ou $2\pi, 4\pi$, etc..) durant l'intervalle $T(n)$, Figure III.2. Pour interdire tout dépassement arithmétique dans les calculs, nous réduisons périodiquement la phase totale de 2π lorsqu'elle dépasse 2π. Lorsque la phase passe une telle limite, l'instant correspondant est calculé dans la variable $t_{cross}(n)$. Si aucun croisement n'est détecté, $t_{cross}(n)$ est mise à 0.

On note cependant, que le test $[\varphi_2(n) \geq 0\ et\ \varphi_2(n-1) < 0]$ dans la Figure III.4 est équivalent au test plus pratique en programmation, $[\varphi_2(n)\,.\,\varphi_2(n-1) \leq 0]$.

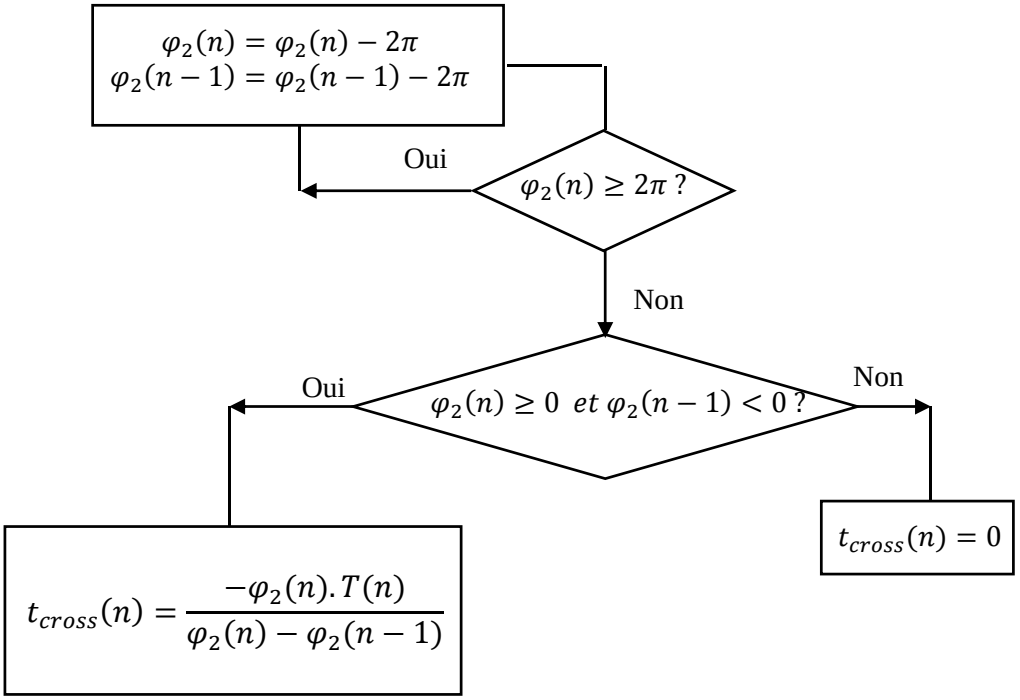


Figure III.4 : Organigramme de croisement du zéro-phase pour $\varphi_2(t)$.

Le calcul de $t_{cross}(n)$ indiqué dans l'organigramme de la Figure III.4, repose sur la constatation géométrique de la Figure III.5 reprise en partie de la Figure III.2. Par symétrie des deux triangles de la Figure III.5, on tire :

$$\text{cotg}(\delta) = \frac{t_{cross}}{-\varphi_2(n-1)} = \frac{T(n)}{\varphi_2(n) - \varphi_2(n-1)} \qquad \text{(III.6)}$$

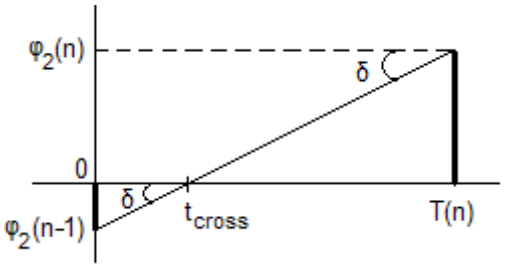


Figure III.5 : Calcul géométrique de t_{cross}.

L'organigramme de la Figure III.6 devient maintenant prêt à calculer l'état $U_d(t)$ du DFP durant l'intervalle $T(n)$ et sauvegarder sa durée dans $t_{\pm 1}(n)$ au temps $t(n)$, Figure III.2. En plus de t_{cross} donnant information sur le front montant de $u_2(t)$, le signal $U_d(t)$ dépend aussi du front montant de $u_1(t)$ qui est détecté lorsque $u_1(n-1) = 1$, figures III.2 et III.6. De même, l'état de $U_d(n-1)$ juste avant le temps $t = t(n-1)$ doit être connu. Ainsi si comme en Figure III.2, $U_d(n-1) = 0$ et $u_1(t)$ marque une transition positive à $t = t(n-1)$, $U_d(t)$ va à l'état $+1$. Lorsque $u_2(t)$ marque par la

suite une transition positive, $U_d(t)$ revient à l'état 0. Lorsque $u_2(t)$ marque encore une transition positive, $U_d(t)$ va à l'état -1. $U_d(t)$ revient à l'état 0 suite à une transition positive de $u_1(t)$ à $t = t(n+1)$.

Tout se décide donc selon le diagramme d'état du DFP de la Figure II.4, et l'algorithme de la Figure III.6 en montre neuf différents cas possibles. Cet algorithme détermine la durée de chaque état ± 1 dans $t_{+1}(n)$ ou $t_{-1}(n)$, et calcule aussi l'état de U_d à la fin de l'intervalle $T(n)$. Cet état sera utilisé comme condition initiale $U_d(n-1)$ au prochain service d'interruption.

Quand il se trouve que $t_{+1}(n)$ est plus grand que 0, cela veut dire que la tension d'alimentation doit être appliquée durant l'intervalle $t_{+1}(n)$ au filtre RC de la Figure II.28. Lorsque $t_{-1}(n)$ n'est pas nul, la capacité C fictive doit être déchargée à la masse durant l'intervalle $t_{-1}(n)$.

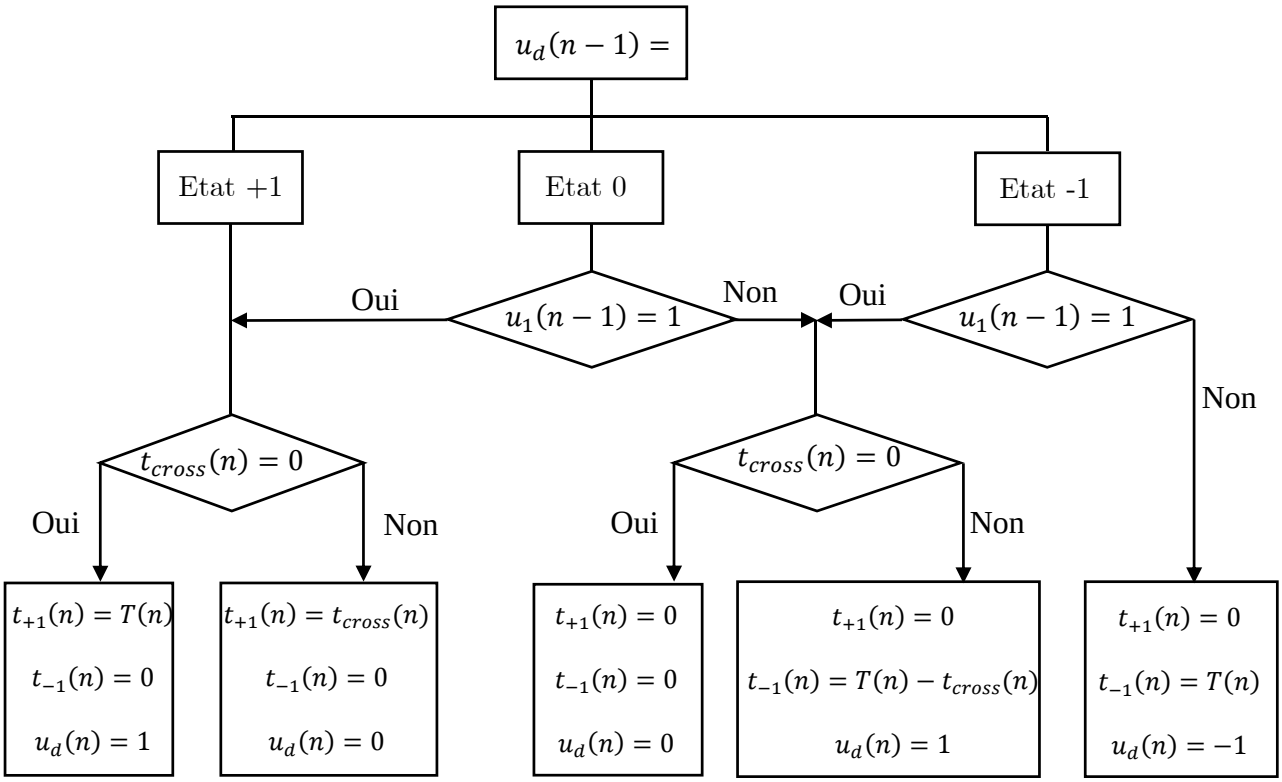


Figure III.6 : Organigramme du DFP.

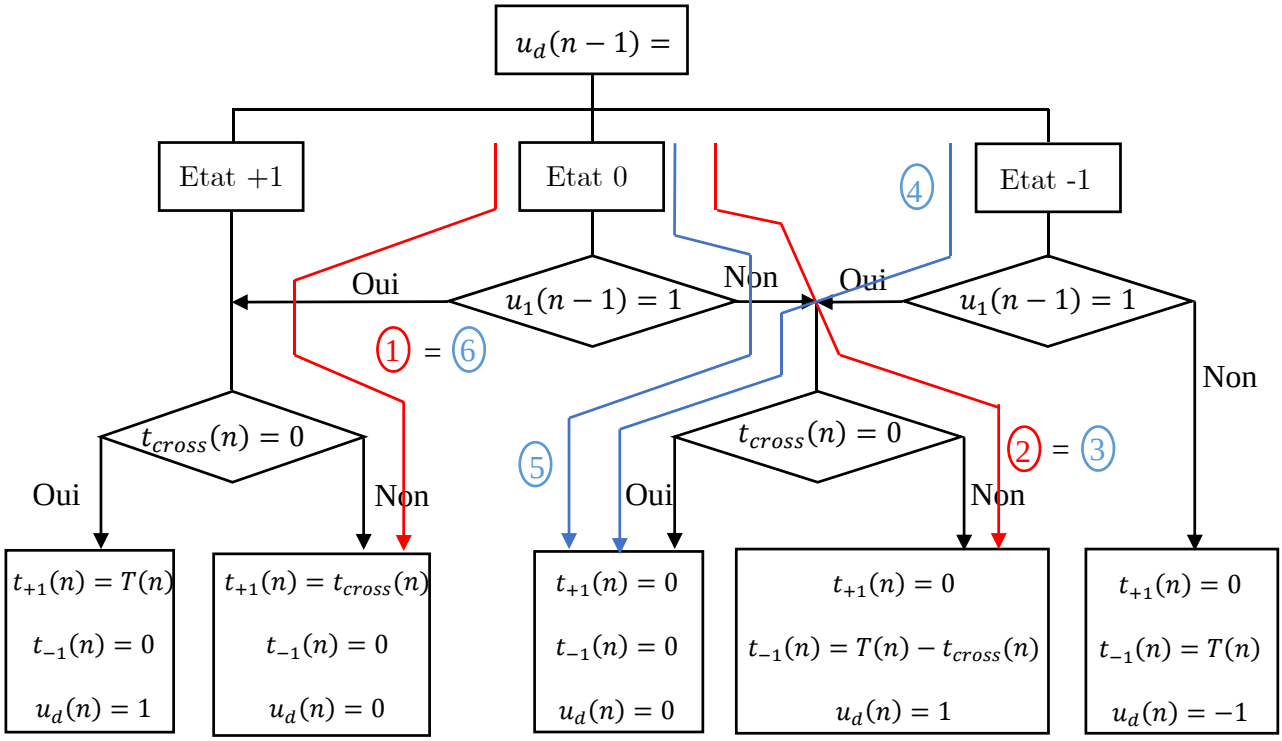
Un premier exemple inspiré des tracés de la Figure III.2 où la fréquence d'entrée $u_1(t)$ est inférieure à celle du DCO, $u_2(t)$, permet une vérification partielle de l'organigramme de la Figure III.6. On en distingue les deux cas sur la Figure III.2 :

1) à $t(n)$:
   $u_d(n-1) = 0$ (considérée juste avant $t(n-1)$)
   $u_1(n-1) = 1$ (considérée juste après $t(n-1)$, donc il s'agit d'un front montant de u_1)

$t_{cross}(n) \neq 0$ (temps d'arrivée du front montant de u_2 à partir de $t(n-1)$)
ainsi,
$t_{+1}(n) = t_{cross}(n)$ (durée de $u_d(n-1) = 1$ juste après $t(n-1)$ causé par le front montant de u_1 puis $u_d(n-1) = 0$ jusqu'à $t(n)$)
$t_{-1}(n) = 0$
$u_d(n) = 0$ (considérée juste avant $t(n)$)

2) à $t(n+1)$:
$u_d(n) = 0$
$u_1(n) = 0$ (pas de front montant de u_1)
$t_{cross}(n+1) \neq 0$ (temps d'arrivée du front montant de u_2 à partir de $t(n)$)
ainsi,
$t_{+1}(n+1) = 0$
$t_{-1}(n+1) = T(n+1) - t_{cross}(n+1)$ (durée de $u_d(n) = -1$ causée par le front montant de u_2 après $t(n)$ et mesurée à partir de $t(n+1)$)
$u_d(n+1) = -1$ (considérée juste avant $t(n+1)$)

Ces évaluations de $t_{+1}(n)$, $t_{-1}(n)$, et $u_d(n)$, sont confirmées par les tracés 1 et 2 (en rouge) de la Figure III.7.



En rouge, pour les 2 cas de la figure III.2, où fréquence de $u_1(t) <$ fréquence de $u_2(t)$.
En bleu, pour les 4 cas de la figure III.8, où fréquence de $u_1(t) >$ fréquence de $u_2(t)$.

Figure III.7: Vérification des cas de figures III.2 et III.8.

Un deuxième exemple pour la vérification de la Figure III.6, est le complémentaire des tracés de la Figure III.2 dont la Figure III.8 en donne une illustration. Dans cet exemple, la fréquence d'entrée $u_1(t)$ est supérieure à celle du DCO, $u_2(t)$.

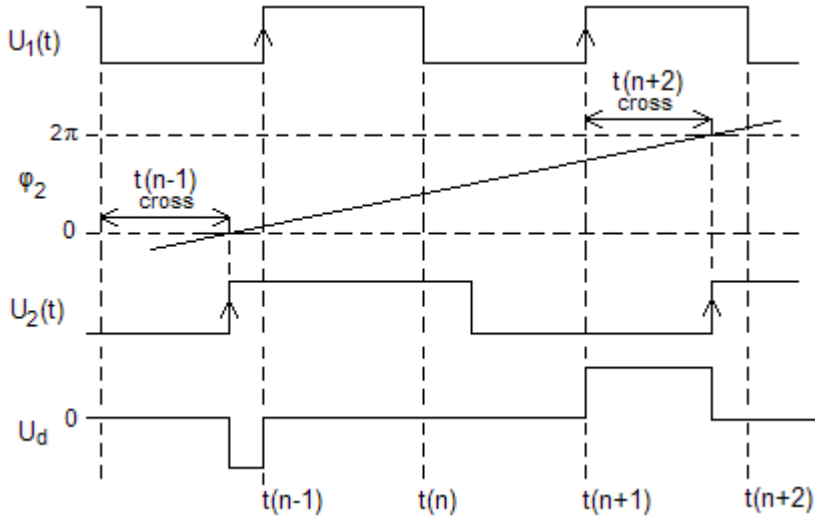


Figure III.8 : Cas de tracés complémentaires de la Figure III.2.

Outre les deux cas de l'exemple précédent, on en distingue les quatre cas suivants sur la Figure III.8:

3) à $t(n-1)$:

$u_d(n-2) = 0$

$u_1(n-2) = 0$ (pas de front montant de u_1)

$t_{cross}(n-1) \neq 0$ (temps d'arrivée du front montant de u_2 à partir de $t(n-2)$)

ainsi,

$t_{+1}(n-1) = 0$

$t_{-1}(n-1) = T(n-1) - t_{cross}(n-1)$

$u_d(n-1) = -1$

4) à $t(n)$:

$u_d(n-1) = -1$

$u_1(n-1) = 1$ (front montant de u_1)

$t_{cross}(n) = 0$ (pas de front montant de u_2)

ainsi,

$t_{+1}(n) = 0$

$t_{-1}(n) = 0$

$u_d(n) = 0$

5) à $t(n+1)$:

$u_d(n) = 0$

$u_1(n) = 0$ (pas de front montant de u_1)

$t_{cross}(n+1) = 0$ (pas de front montant de u_2)

ainsi,

$t_{+1}(n+1) = 0$

$t_{-1}(n+1) = 0$
$u_d(n+1) = 0$

6) à $t(n+2)$:
$u_d(n+1) = 0$
$u_1(n+1) = 1$ (front montant de u_1)
$t_{cross}(n+2) \neq 0$ (temps d'arrivée du front montant de u_2 à partir de $t(n+1)$)
ainsi,
$t_{+1}(n+2) = t_{cross}(n+2)$
$t_{-1}(n+2) = 0$
$u_d(n+2) = 0$

Ces calculs de $t_{+1}(n)$, $t_{-1}(n)$, et $u_d(n)$, sont confirmés par les tracés 3 à 6 (en bleu) de la Figure III.7.

L'algorithme utilisé pour calculer la sortie $u_f(n)$ du filtre diffère catégoriquement des algorithmes conventionnels des filtres numériques. Dans un algorithme classique de filtrage, l'échantillon $u_f(n)$ du signal de sortie est calculé à partir d'un nombre d'échantillons décalés de ce signal de sortie et d'un nombre d'échantillons du signal d'entrée. Ce principe ne s'applique pas à notre cas, car le signal d'entrée de ce circuit est appliqué uniquement durant une fraction de l'intervalle d'échantillonnage et non pendant toute la durée d'échantillonnage. L'entrée est flottante dans le reste du temps. Par conséquent, le signal de sortie du filtre numérique doit être copié du signal de sortie du filtre analogique de la Figure II.28.

L'algorithme du filtrage numérique en Figure III.9 explique comment la tension $u_c(n)$ aux bornes de la capacité C doit être calculée à partir de la valeur précédente $u_c(n-1)$. Si aucun courant ne circule dans la capacité C, Figure II.28, le signal de sortie $u_f(n)$ doit être identique à la tension $u_c(n)$ aux bornes de C. Cependant, dans les intervalles où le courant circule, $u_f(n)$ peut être plus grande ou plus petite que $u_c(n)$, dépendant de la polarité du courant. Comme $u_f(t)$ est non constante dans l'intervalle $T(n)$, nous définissons $u_f(n)$ comme la moyenne de $u_f(t)$ dans l'intervalle $t(n-1) \leq t < t(n)$. On assume aussi que la durée d'un cycle $T(n)$ (moitié de la durée de cycle du signal de référence) est beaucoup plus faible que la constante de temps τ_1 du filtre de la Figure II.28. Sous cette condition, le courant circulant dans C demeure constant durant les intervalles de charge ou décharge. Cette hypothèse permet des expressions simples de $u_c(n)$ et $u_f(n)$.

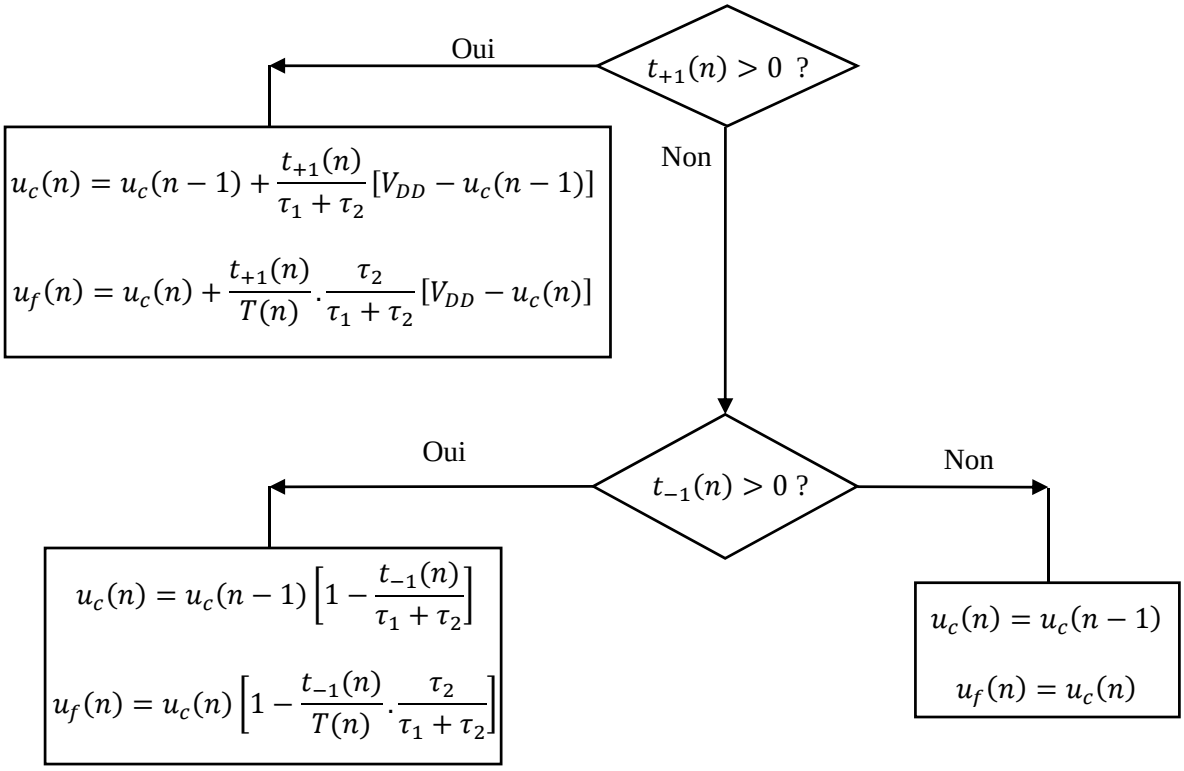


Figure III.9 : Organigramme du filtre digital.

Les expressions listées sur la Figure III.9, proviennent des constatations suivantes effectuées sur la Figure II.28.

Lorsque $u_d(n) = +1$:

$u_c(n) = u_c(n-1) + \Delta V_c$ avec $\Delta V_c = \frac{1}{C}\int_0^{t_{+1}(n)} I dt$

où $I = [V_{DD} - u_c(n-1)]/[R_1 + R_2] = c^{ste}$

$u_f(n) = u_c(n) + R_2 I_{moy}$ avec $I_{moy} = \frac{1}{T(n)}\int_0^{t_{+1}(n)} \frac{V_{DD} - u_c(n)}{R_1+R_2} dt$

lorsque $u_d(n) = -1$:

$u_c(n) = u_c(n-1)e^{\frac{-t_{-1}(n)}{\tau_1+\tau_2}}$ pour $t_{-1}(n) << \tau_1 + \tau_2$ on a $e^{-x} \approx 1 - x$ pour x faible.

$u_f(n) = u_{c(n)} - R_2 I_{moy}$ avec $I_{moy} = \frac{1}{T(n)}\int_0^{t_{-1}(n)} \frac{u_c(n)}{R_1+R_2} dt = \frac{u_c(n-1)(\tau_1+\tau_2)}{T(n)(R_1+R_2)}\left[1 - e^{\frac{-t_{-1}(n)}{\tau_1+\tau_2}}\right]$

soit $I_{moy} = \frac{u_c(n-1)t_{-1}(n)}{T(n)(R_1+R_2)}$ où l'on remplace $u_c(n-1)$ par $u_c(n)$ du fait que $u_f(n)$ doit être fonction de $u_c(n)$ à $t = t(n)$.

Tous les calculs d'un seul service d'interruption viennent d'être présentés. Comme la plupart des échantillons calculés à $t = t(n)$ seront utilisés comme valeurs de départ dans le prochain service d'interruption, ils doivent être décalés dans le temps. Ceci est indiqué au bas de l'organigramme de l'algorithme SPLL de la Figure III.3.

III.5 Implémentation de l'algorithme SPLL sur DSP

Le logiciel de l'application est constitué de quatre fichiers sources, deux écrits en langage C : PLL.c, RTDX.c, un fichier écrit en Matlab : Trace.m, et un fichier texte renfermant des données d'interruptions. Les contenus de ces fichiers sont discutés brièvement ci-après :

PLL.c :

Ce fichier renferme le programme principal, deux routines d'interruptions hardwares, *hwiENTREE_ OFF()* et *hwiENTREE_ ON(),* et une routine d'interruption software, *swiPLL()*.

Le programme principal fait démarrer le timer, autorise les deux interruptions externes, et initialise la liaison RTDX pour rentrer dans une boucle d'attente (idle, à faible énergie consommée).

La routine *hwiENTREE_ OFF()* est appelée par l'interruption externe INT4 qui détecte l'arrivée de chaque niveau bas du signal d'entrée de la PLL. Tandis que La routine *hwiENTREE_ ON()* est appelée par l'interruption externe INT5 qui détecte l'arrivée de chaque niveau haut du signal d'entrée de la PLL. Ces deux routines commencent par lire le contenu du timer, mettre à zéro le drapeau d'interruption, sauvegarder la valeur du niveau d'entrée détecté, pour donner la main à la routine software afin qu'elle fasse le traitement adéquat de la SPLL en exécutant son algorithme. En fin de chaque étape de calcul, elle renvoi les résultats à Matlab via la liaison RTDX pour les tracer.

RTDX.c :

Le fichier contient deux fonctions. L'une d'initialisation de la liaison RTDX, et l'autre pour l'envoi des paquets de données à Matlab pour les tracer.

Trace.m :

Après avoir reconnu la liaison RTDX, Matlab ouvre son canal pour recevoir les paquets de données à tracer.

Le fichier texte renfermant les données sur les interruptions externes, est décrit un peu plus loin.

Une fois les fichiers sources de notre application sur la SPLL sont préparés, la construction du projet gérant l'application sous l'environnement CCS (Code Composer Studio), suit les étapes de développement suivantes :

Tout d'abord, via l'icône « setup du CCS », nous choisissons la plate forme à base du DSP C6713 (DSK ou simulateur). Une fois ce choix additionné à la colonne de configuration de système, le composant est de nouveau sélectionné afin d'actionner le bouton « Modify Properties ». Dans la fenêtre de dialogue qui s'ouvre, décocher les options « detect CPU resource conflicts » et « detect reserved memory access » afin d'améliorer la rapidité du simulateur.

Une fois CCS ouvert et au menu *Project* → *New*, notre projet est nommé« SPLL.pjt » en choisissant le type exécutable (et non librairie) avec la cible *TMS320C67XX*.

Le fichier de configuration renferme diverses configurations d'objets à saisir graphiquement. Pour cela :
Au menu *File* → *New* → *DSP/BIOS Configuration*, on sélectionne la plate-forme C6713.cdb.

L'outil DSP/BIOS de configuration graphique s'affiche et certaines catégories sont à programmer :

- Application de la catégorie «System»
  Dans le Module «Global Settings», on choisit DSP_speed = 250MHz.

- Application de la catégorie Scheduling
  Dans le module « HWI-SR manager » : on laisse par défaut, les polarités des interruptions externes aux pins 4 et 5, sur l'option « low to high ».
  Dans la Voie « HWI_INT4 » : est programmée la routine *_hwiENTREE_OFF*, avec « use Dispatcher » mise sur « true ».
  Dans la Voie « HWI_INT5 » : est programmée la routine *_hwiENTREE_ON*, avec « use Dispatcher » mise sur « true ».
  Dans le module «SWI- Manager» : Objet «swiProcess» : est programmée la routine *_swiPLL* avec priorité=1.

- Application de la catégorie Input/Output
  Dans les propriétés du module «RTDX-Settings»: on choisit le mode simulateur du RTDX, et on initialise la taille du « target buffer » à 2064. On programme aussi le canal de sortie au nom *versMatlab*, assurant l'écoulement des données calculées par DSP vers Matlab pour les tracés.

- Application de la catégorie CSL
  Dans le module «TIMER»: Voie «Configuration Manager»: est programmée la configuration *timerCg1*, qui est destinée au timer1 disponible. Ce temporisateur assure la mesure du temps écoulé entre des interruptions externes consécutives, INT4 et INT5. Dans les propriétés de *timerCg1*, on choisit l'onglet « Advanced » pour configurer les registres internes du timer :
  CTL = 0x000003C1 (timer cadencé par l'horloge interne $F_{CPU} = 250Mhz$ au rythme de $F_{CPU}/4$), et PRD = 0xFFFFFFFF ($T_{max} = 4.6 \text{ milliards} \times 4/F_{CPU}$ correspondant à 69 sec ou 0.015 Hz, donc non atteinte par les fréquences d'entrée de notre DPLL ou SPLL).
  Dans la Voie «Resource Manager» : dans les propriétés du timer1, on choisit :
  Handle = *hTimer1*, et *Pre-Initialize with* = *timerCfg1*.

En quittant l'outil de configuration graphique, nous sauvegardons nos configurations sous le nom de fichier, SPLL.cdb. Ceci génère automatiquement un fichier de liaison ou linkage, *SPLLcfg*.cmd, et deux fichiers C : un fichier de source, *SPLLcfg_ c*.c, et un fichier d'entête (ou header), *SPLLcfg*.h. Le fichier source *SPLLcfg_ c*.c contient, la déclaration des objets et structures de configuration établie graphiquement. Le fichier d'entête *SPLL*cfg.h est à insérer manuellement dans les fichiers source en C de notre application par : #include "SPLLcfg.h". Un autre fichier d'entête target.h propre à la liaison RTDX, doit lui aussi être déclaré au début de programme.

Sous l'option *Add Files to Project* du menu déroulant de *Project*, on ajoute successivement au projet les fichiers sources C de notre application, le fichier de configuration SPLL.cdb.

La construction du projet, passe par le menu *Project* → *Rebuild All* qui compile, assemble, et lie tous les fichiers du projet en un seul fichier exécutable SPLL.out.

Simulation d'évènements externes :

Le simulateur de DSP [48] fourni des moyens pour simuler les interactions du DSP avec les entités externes. On en distingue deux catégories :

1- les signaux de contrôle, comme les interruptions, les signaux d'horloge, et les signaux de synchronisation.
2- les valeurs de données, comme la lecture ou écriture dans un registre de périphérique ou un port d'entrée/sortie.

Au menu « Tools » du CCS, se trouvent les deux outils « Pin connect » et « Port connect » pour la simulation de ces deux types d'interaction, respectivement.

L'outil « Pin connect » permet de simuler les interruptions externes appliquées aux broches (ou pins) correspondantes du DSP. Un fichier avec un format spécifié peut être connecté à ces pins. Les pins simulés sont de deux types : impulsion et forme d'onde. Les pins à impulsion sont sensibles aux fronts montants des signaux, tandis que les pins à forme d'onde en sont sensibles aux niveaux (donc, fronts montants et descendants).

Dans notre application, nous avons à mesurer les intervalles de temps $T(n)$, figures III.2 et III.3, compris entre deux transitions successives du signal d'entrée (donc, mesure du temps que met chaque niveau du signal). Nous utilisons pour cela, les pins INT4 et INT5 qui sont de type à impulsion, donc sensibles qu'aux fronts montants. En considérant un signal carré périodique à l'entrée de la PLL, il suffit alors d'appliquer ce signal au pin INT5 et sa réplique décalée d'une demi-période au pin INT4 pour que cela répond à notre besoin. Ainsi, les fronts descendants du signal d'entrée sont transformés en fronts montants au pin INT4. Selon la figure III.10, le pin INT5 est dédié à l'identification des niveaux hauts du signal à l'aide de la routine d'interruption *hwiENTREE_ ON*, tandis que le pin INT4 est tenu à identifier les niveaux bas du signal à l'aide de la routine d'interruption *hwiENTREE_ OFF*.

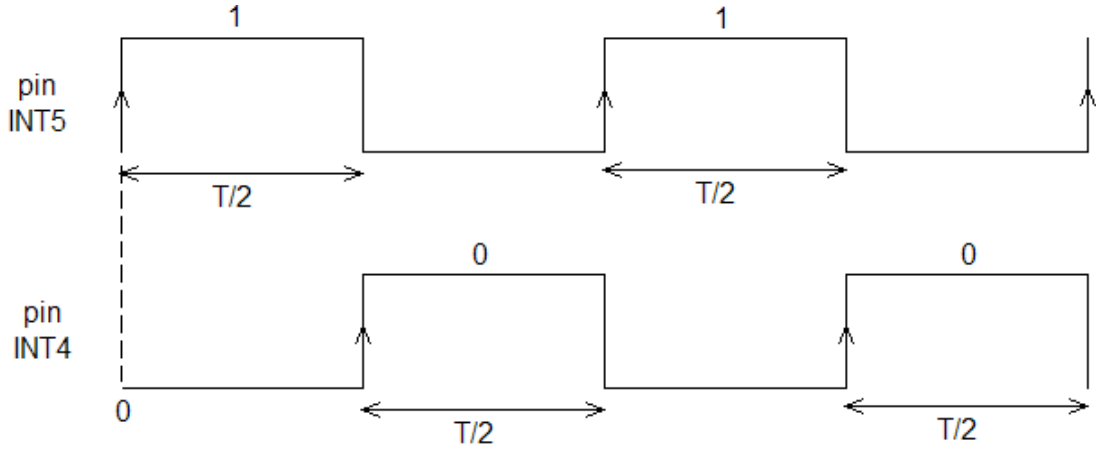


Figure III.10 : Signal d'entrée et sa forme décalée, appliqués aux pins INT5 et INT4.

Le « Pin connect » permet de spécifier les instants d'occurrence des interruptions externes dans un simple fichier texte qu'on doit rattacher au pin spécifié à travers une fenêtre de configuration qui s'ouvre dans ce but. Le fichier texte renferme les intervalles d'interruption en terme de cycles d'horloge CPU [48], F_{CPU}=250MHz.

Deux fichiers de données de type texte, F_E_4.txt et F_E_5.txt, liés chacun à un pin d'interruption externe, INT4 et INT5 respectivement, sont donc attachés à la fréquence F_E du signal d'entrée de la manière répétitive suivante :

F_E_5.txt : $\left(+\frac{F_{CPU}}{F_E}\right)$ rpt EOS

F_E_4.txt : $\frac{F_{CPU}}{2F_E}\left(+\frac{F_{CPU}}{F_E}\right)$ rpt EOS

« rpt EOS » indique une répétition jusqu'à la fin de la simulation (end of simulation). Le signe plus ajoute la valeur en cycles d'horloge à la valeur totale des cycles précédents. Huit fichiers sont donc préparés pour les quatre fréquences d'entrée du cas de la DPLL en figure II.42, à savoir, 100kHz, 200kHz, 290kHz, et 625kHz.

III.6 Test d'acquisition en temps réel du signal d'entrée

Après développement du projet SPLL sous CCS, l'application dénommée SPLL_TempsReel a été exécutée en premier sur le simulateur du C6713 afin de tester l'acquisition en temps réel du signal d'entrée sous les quatre différentes fréquences. La routine software accomplissant le traitement de l'algorithme SPLL a été ignorée, en faveur des seules routines hardwares mesurant les niveaux du signal et leurs durées $T(n)$.

Pour le signal à 100kHz de demie période $T = 5\mu s$, la Figure III.11 tracée par Matlab via RTDX, montre une bonne acquisition de l'information portant sur la demie période du signal d'entrée.

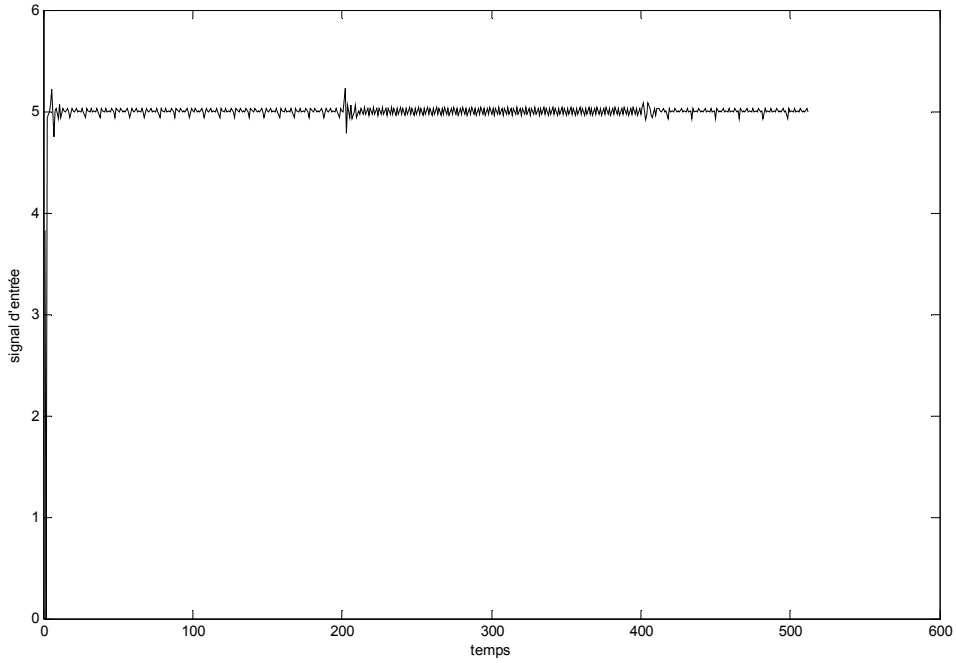


Figure III.11 : Mesure des durées de séparation entre niveaux du signal à 100kHz.

Cette acquisition en temps réel n'a pu se poursuivre pour les autres fréquences d'entrée plus élevées, dû vraisemblablement à certaines interruptions ratées pendant la phase d'acquisition. Pensant que le simulateur était à l'origine de ce problème, nous avons entamé les changements nécessaires à cette application en vue de l'adapter à une carte DSP réelle, le kit DSK C6713. De vraies interruptions émanent d'un générateur d'impulsions, ont été appliquées aux pins INT4 et INT5 du port d'entrée/sortie de la carte. L'alternance des bits d'entrée a bien été détectée à la fréquence 100kHz, à l'inverse des autres fréquences. Il faudra optimiser le programme afin de pouvoir satisfaire son exécution en temps réel pour les fréquences d'entrée supérieures à 100kHz.

III-7 Test d'acquisition en temps différé du signal d'entrée :

Les résultats obtenus étant pareils à ceux du simulateur C6713, nous nous sommes résignés à poursuivre notre étude sur le simulateur sans pour autant, attacher une grande importance à l'aspect d'exécution en temps réel. Par conséquent, des essais en temps différé ont été menés dans une nouvelle application « SPLL_TempsDiff », où les demie périodes d'entrée $T(n)$ étaient initialisées d'avance sans passer à les mesurer par le timer associé à leurs interruptions chronométrées. Tous les autres traitements de l'algorithme SPLL restent fonctionnels y compris les tracés confiés à Matlab via la liaison RTDX.

Les caractéristiques techniques des blocs constituants la DPLL simulée par Pspice, doivent être attribuées à la SPLL implémentée sur DSP afin qu'elle puisse se comporter comme la DPLL. Ainsi, le DCO de la SPLL devra présenter une caractéristique non linéaire similaire à celle du VCO de la Figure II.27. Du moins, la valeur de la sensibilité K_o doit être tirée de la caractéristique non linéaire selon la fréquence considérée. Ainsi,

faute de mesurer directement les tensions de filtre associées aux fréquences-test d'entrée suite à la résolution du graphe de la Figure II.27, on mesure donc sur la caractéristique de cette figure, les deux tensions de filtre x_1 et x_2 correspondant aux deux fréquences adjacentes y_1 et y_2 de chaque fréquence d'entrée. Desquelles, on tire par interpolation linéaire, la tension de filtre U_f nécessaire à l'obtention de la fréquence d'entrée désirée. La sensibilité K_o en est déduite par calcul, Tableau III.2 :

Fréquence d'entrée	Y_1 (kHz)	X_1 (mV)	Y_2 (kHz)	X_2 (mV)	U_f (mV)	K_0 (kHz/V)	K_0 (rd/Vs)
100 kHz	99.406	331.683	101.287	336.634	333.247	300.078	1885.10^3
200 kHz	199.708	599.010	201.525	603.960	599.806	333.441	2095.10^3
289.075 kHz	289.074	846.535	290.811	851.485	846.537	341.479	2146.10^3
625 kHz	623.986	1.9703 V	625.128	1.9753 V	1.9745 V	316.536	1989.10^3

Tableau III.2 : Mesure de points adjacents relatifs aux fréquences d'entrée et calcul de U_f et K_o

En considérant des demi-périodes d'entrée $T = 5\mu s$, $2.5\mu s$, $1.729655\mu s$, et $0.8\mu s$, correspondant respectivement, à 100kHz, 200kHz, 289.075kHz, et 625 kHz, les tensions à la sortie du filtre de boucle sont réunies sur la Figure III.12. Toutes les réponses présentent des oscillations plus ou moins prononcées autour de valeurs moyennes fixes. La durée de simulation est de 2ms, comme celle menée avec Pspice sur la DPLL en Figure II.46. Les deux figures marquent un même point commun de départ des signaux filtrés, qui est 846.537mV.

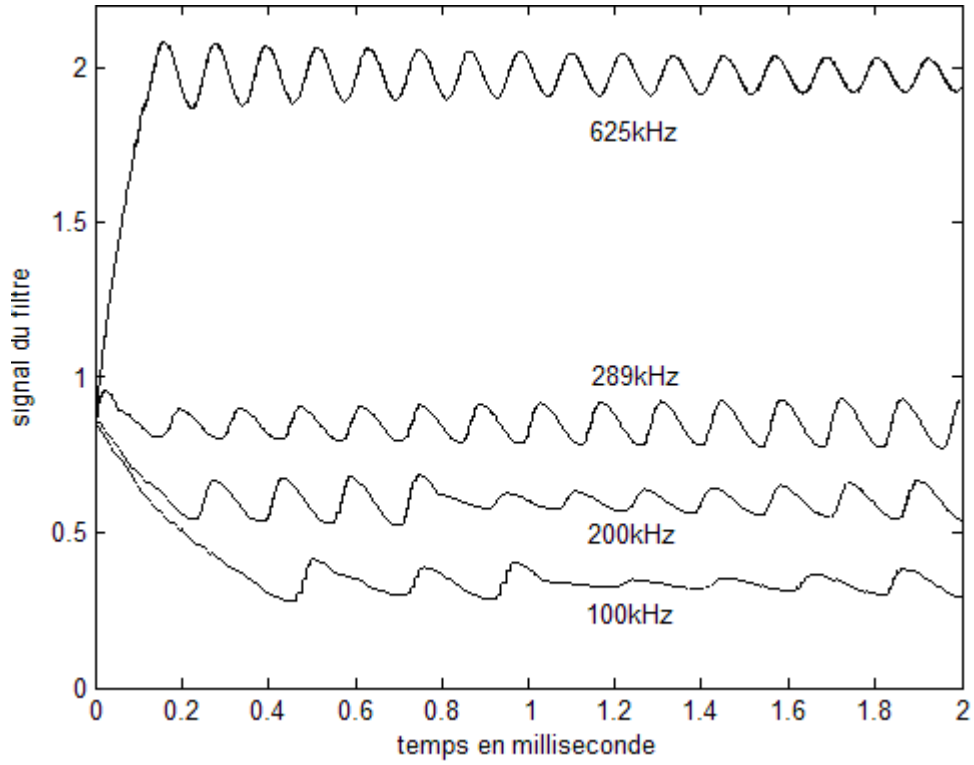


Figure III.12 : Tensions de filtre à c^{stes} de temps modifiées en réponse aux quatre fréquences de test

Dans le cas de la DPLL, les anciennes constantes de temps du filtre ont été modifiées dans l'intention d'améliorer les réponses de la Figure II.41, aboutissant ainsi aux réponses de la Figure II.46. Cependant, les constantes de temps modifiées et présentant les valeurs de $\tau_1 = 268\mu s$ et $\tau_2 = 0.47\mu s$, assurent une fréquence naturelle de la SPLL égale à 8.7 kHz ($\omega_n = 54.6\ 10^3$ rd/s) sous un facteur d'amortissement $\varepsilon = 0.013$. Ce faible facteur est tenu responsable des réponses oscillatoires liées aux régimes transitoires de la Figure III.12.

Dans le but d'améliorer les performances dynamiques de la SPLL, nous réutilisons de nouveau les anciennes constantes de temps du filtre de boucle établies initialement dans Pspice, à savoir :
$\tau_1 = 45.2\mu s$ et $\tau_2 = 11.8\mu s$. Ce qui instaure une nouvelle dynamique pour la SPLL, à travers un facteur d'amortissement optimal de $\varepsilon = 0.7$ et une fréquence naturelle de 18.9 kHz ($\omega_n = 118.5\ 103$ rd/s) assurant ainsi, une bande passante plus large de la SPLL avec un temps de réponse minime.

Pour une fréquence d'entrée de 100kHz, la nouvelle réponse filtrée de la SPLL est exempte d'oscillations pendant le régime permanent. Au bout de 2ms, la tension du filtre est stabilisée sur la valeur 0.3333V accusant par là, un écart relatif de 0.76% avec la réponse filtrée de la DPLL qui arrive à 330.833mV, Figure II.42. Le temps de réponse est lui aussi amélioré. Il passe d'environ 0.5ms, Figure III.12, à environ 0.2ms, Figure III.13.

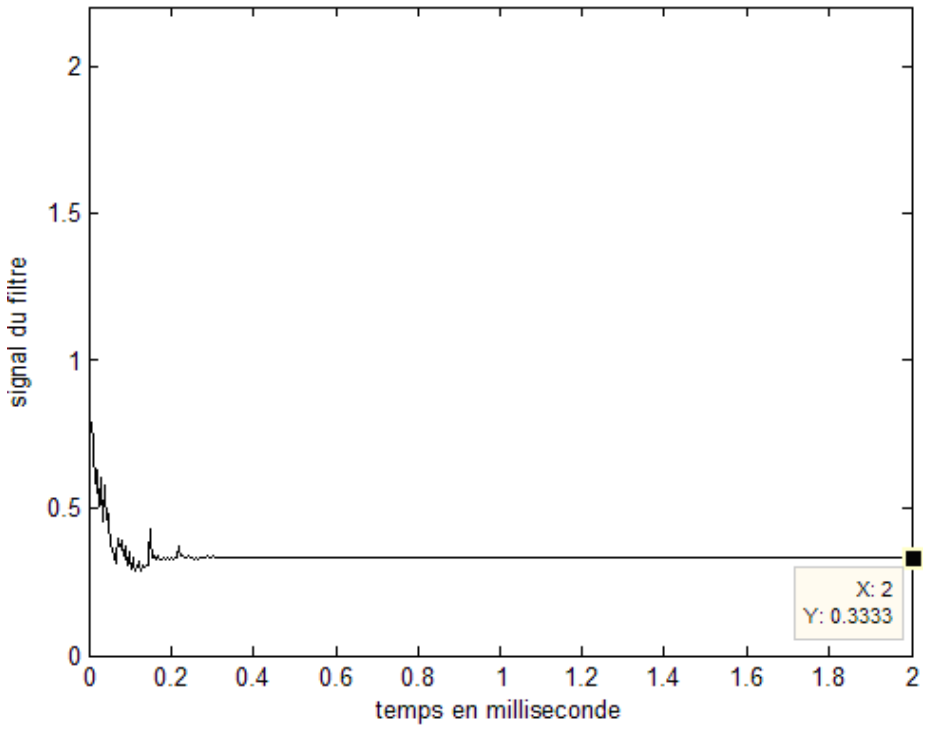


Figure III.13 : Tension de filtre en réponse à 100kHz d'entrée.

Pour une fréquence d'entrée de 200kHz, la nouvelle réponse filtrée de la SPLL est aussi exempte d'oscillations en régime permanent et présente au bout de 2ms, un écart relatif de 0.085% avec la réponse filtrée de la DPLL (599.292mV en Figure II.43). Le temps de réponse est de même amélioré. Il est de 0.1ms en Figure III.14 et de 0.2ms en Figure III.12.

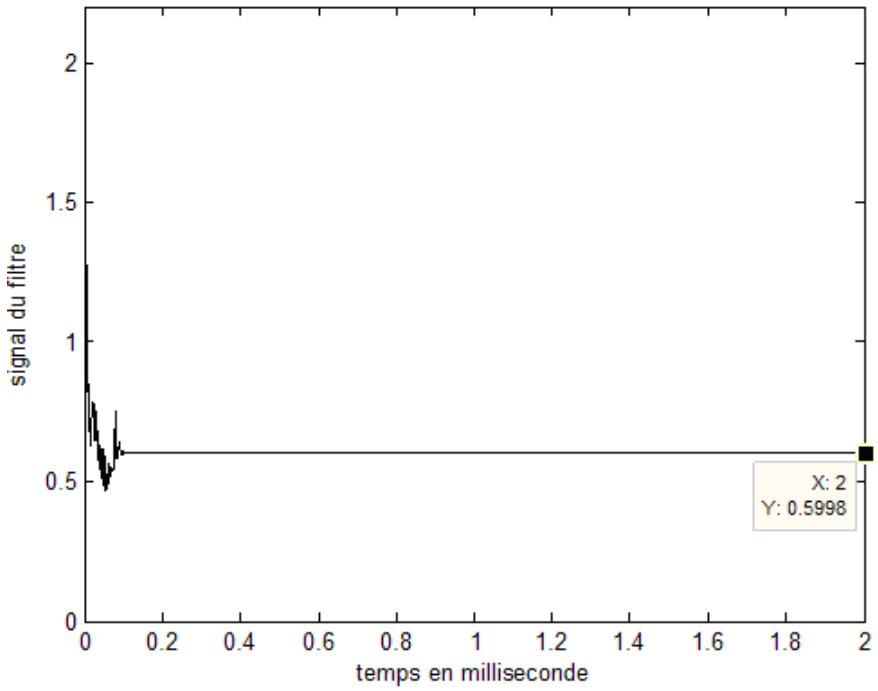


Figure III.14 : Tension de filtre en réponse à 200kHz d'entrée.

Pour une fréquence d'entrée de 289.075kHz, la nouvelle réponse filtrée de la SPLL démunie d'oscillations apparentes, présente au bout de 2ms, un écart relatif de 0.06% avec la réponse filtrée de la DPLL (846.714mV en Figure II.45). Quant au temps de réponse, il est inférieur à 0.1ms, Figure III.15, en comparaison au caractère toujours oscillatoire en Figure III.12.

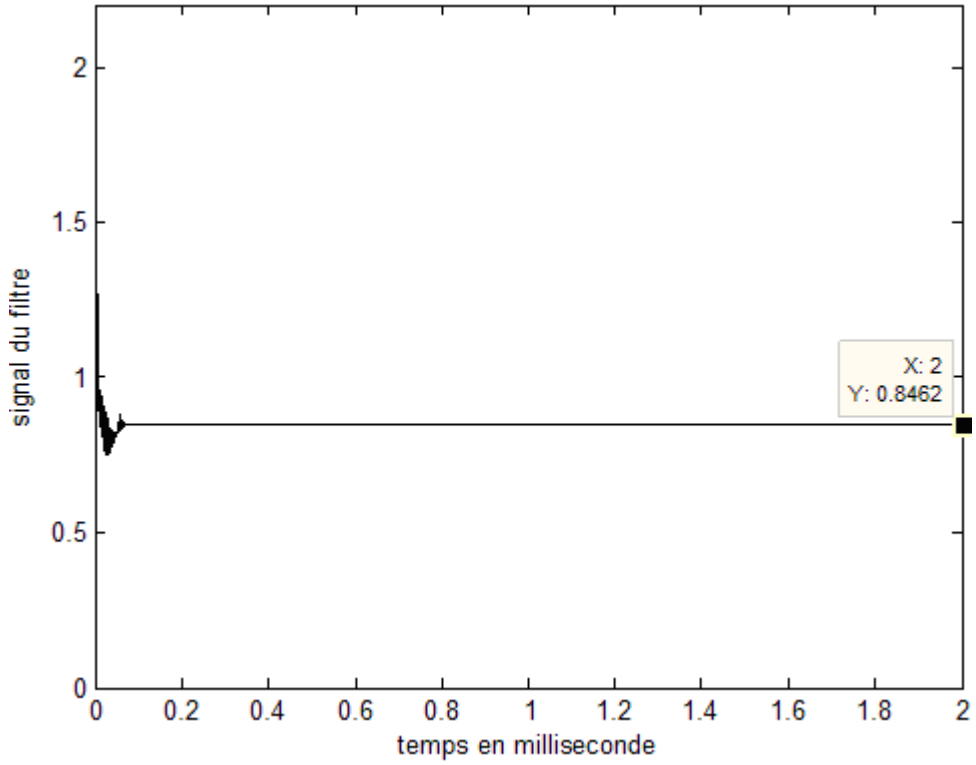


Figure III.15 : Tension de filtre en réponse à 289kHz d'entrée.

Est associée à la fréquence d'entrée 625kHz, une nouvelle réponse filtrée de la SPLL démunie d'oscillations apparentes et présentant au bout de 2ms, un écart relatif de 0.77% avec la réponse filtrée de la DPLL (1.9589V en Figure II.44). Le temps de réponse est inférieur à 0.1ms, Figure III.16, comparé à celui juste supérieur à 0.1ms de la Figure III.12.

Les écarts statiques relevés dans les quatre figures précédentes marquent de très faibles valeurs (inférieurs à 0.1%) relativement aux fréquences avoisinant celle de repos, à savoir, 289kHz.

Les deux réponses extrêmes liées aux fréquences d'entrée, 100kHz et 625kHz, sont un peu moins précis en terme d'écart statique mais restent faibles en ne dépassant pas le 1%.

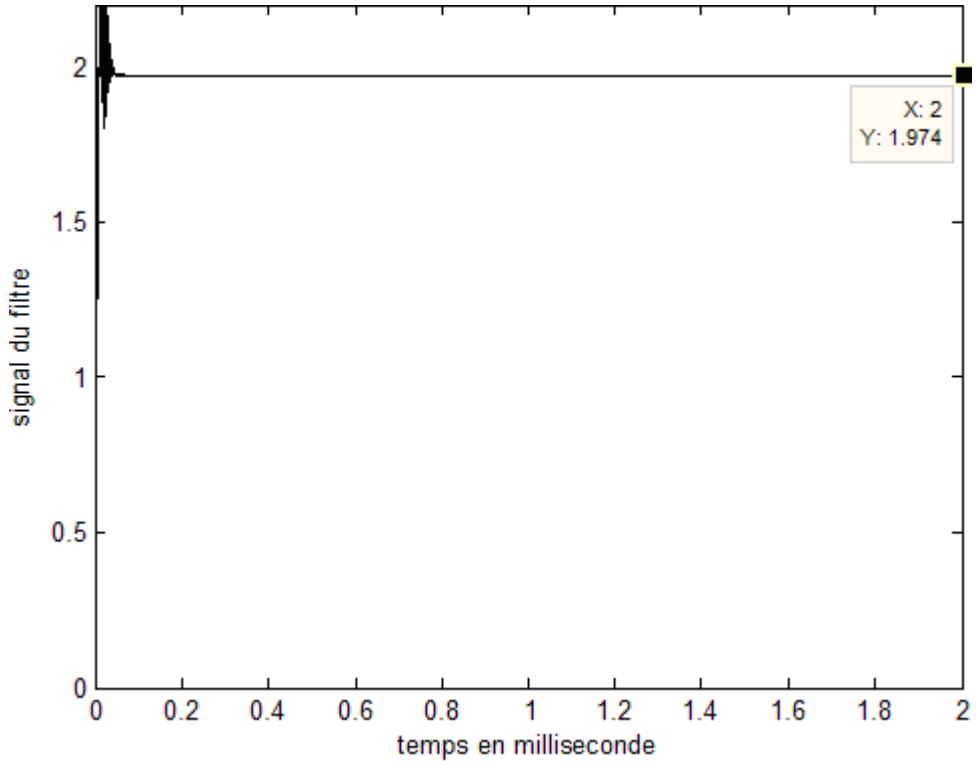


Figure III.16 : Tension de filtre en réponse à 625kHz d'entrée.

Les Figures de III.13 à III.16 sont réunies en Figure III.17 dans laquelle sont rassemblées aussi, les réponses filtrées de la Figure III.12. Les premières réponses servent bien de valeurs moyennes en régime permanent autour desquelles, oscillent les secondes.

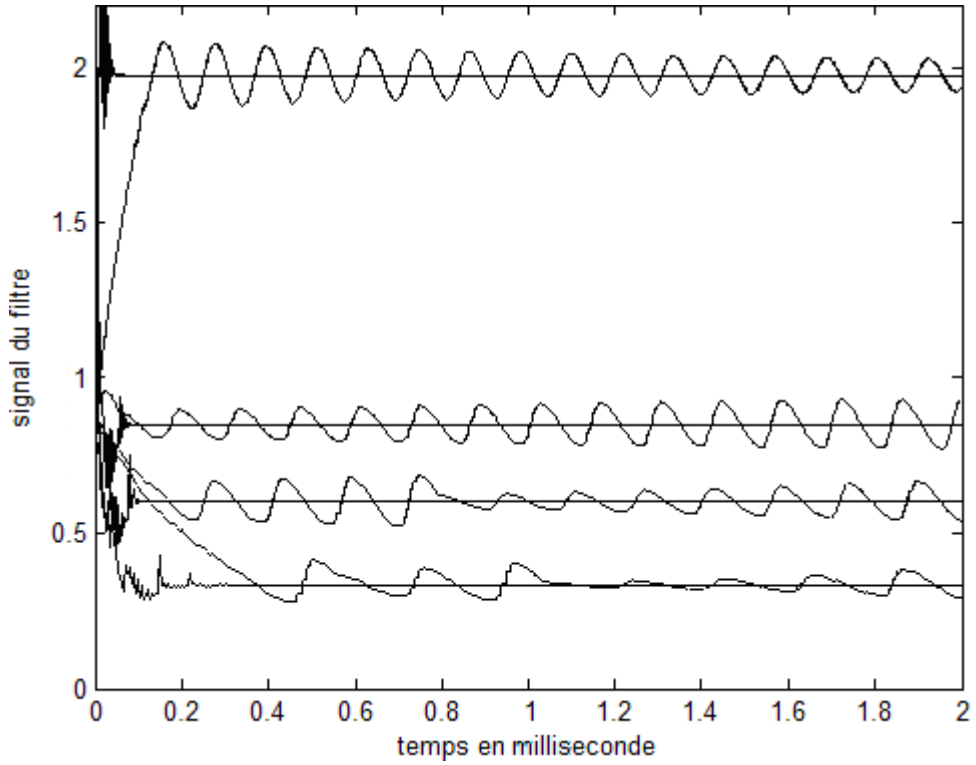


Figure III.17 : Tensions de filtre à c^{stes} de temps optimales comme valeurs moyennes de celles à c^{stes} de temps modifiées.

Les réponses du filtre à c^{stes} de temps optimales sont presque toutes améliorées en précision statique et dynamique relativement à celles à c^{stes} de temps modifiées.

Les valeurs de simulation Pspice se reportant à la Figure II.46, ont été exportées vers des fichiers de données pour visualisation à l'aide de Matlab. Ainsi et à titre comparatif, la Figure III.18 regroupe les quatre réponses filtrées optimales précédentes de la SPLL, en superposition avec celles de la DPLL munie du filtre à c^{stes} de temps modifiées. On peut affirmer de même qu'en Figure III.17, que ces dernières oscillent en régime permanent autour des valeurs moyennes que constituent les premières. Bien que les tendances des réponses de la DPLL et la SPLL, soient les mêmes dans les deux Figures III.17 et III.18, seulement, les oscillations de la DPLL sont bien amorties que celles présentées par la SPLL.

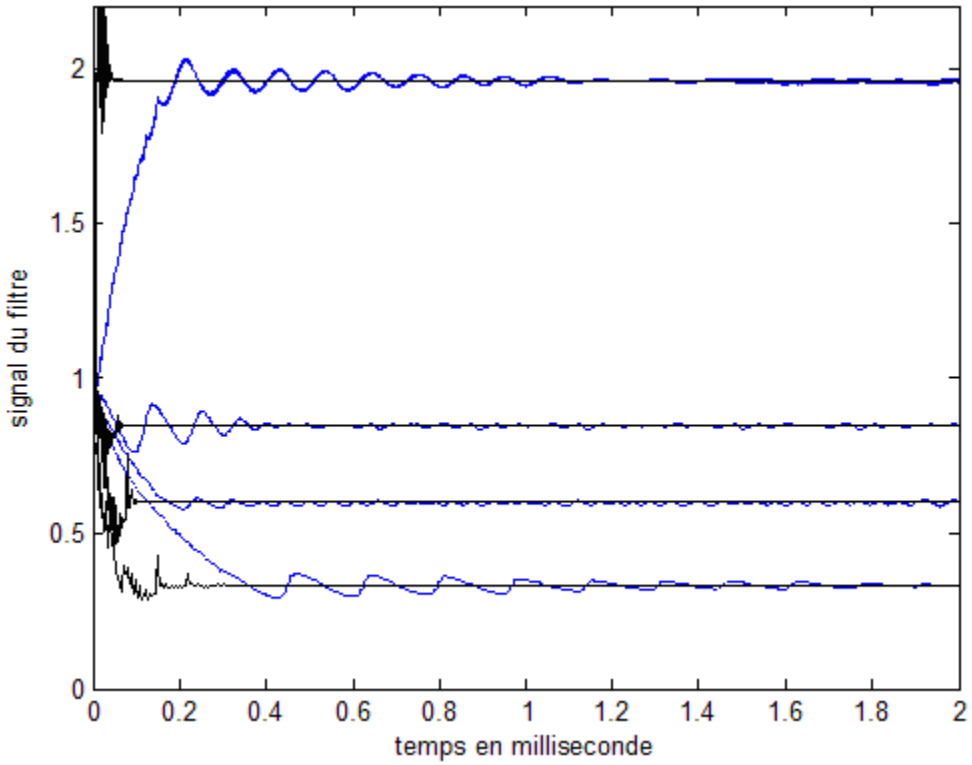


Figure III.18 : Tensions de filtre superposées de la DPLL et la SPLLen réponse aux quatre fréquences de test.

CONCLUSION GENERALE

Depuis leur apparition en début du siècle précédent, les boucles à asservissement de phase ont connu beaucoup de développements aussi bien sur le plan théorique que pratique. On compte aujourd'hui plusieurs types de PLLs, les plus connues d'entre elles sont les PLLs analogiques, les PLLs semi-numériques, les PLLs entièrement numériques et les PLLs software.

L'objectif de notre travail était d'étudier, de concevoir et de réaliser deux types de PLLs, une PLL semi-numérique et une PLL software.

Ayant voulu accomplir un travail aussi complet que possible, nous avons commencé par donner une idée générale sur les boucles à verrouillage de phase. Nous avons fait brièvement le tour de leur principe de fonctionnement, leur classification leur historique et leurs applications.

Une conception assistée par Pspice a été menée afin d'aboutir à une DPLL optimisée dans son comportement statique et dynamique. Pour cela, un détecteur de type fréquence-phase est choisi pour sa grande sensibilité en phase d'entrée, arrivant jusqu'à $j2\pi$. Outre des plages de fonctionnement étendues sinon infinies, procurées par la pompe de charge que forme ce détecteur DFP une fois associé à un simple filtre passif du 1er ordre, une précision statique optimale est atteinte suite au comportement intégrateur inné à cette pompe. La partie relevant de l'oscillateur contrôlé par tension était la plus ardue. A plusieurs reprises, il fallait le retoucher pour l'améliorer vis-à-vis de l'élargissement de la fenêtre d'hystérésis de son trigger de Schmitt, de la stabilisation de l'impédance de sortie de sa source de courant contrôlée par tension, et de la linéarisation de sa caractéristique tension-fréquence. La transmittance du deuxième ordre obtenue de la DPLL en boucle fermée s'est vue attribuée un facteur d'amortissement optimal de $1/\sqrt{2}$, réduisant ainsi le temps de réponse de la DPLL à son minimum. Dans le dimensionnement du filtre de boucle, une gigue de phase d'entrée maximale de 45 $^\circ$ et correspondant à une bande équivalente de bruit de 10 kHz a été considérée. Le fonctionnement de la DPLL testée initialement à l'aide d'un simulateur de PLL phaselockloop, a été très concluant. Le simulateur Pspice reste le plus précis du fait qu'il tient compte du comportement individuel des composants électroniques réalisant la DPLL. Ainsi, il prévoit des réponses stables et convergentes à la sortie du filtre de boucle mais restant parasitées par le fonctionnement non linéaire de la pompe de charge. Le filtre est redimensionné par conséquent, donnant par là des signaux filtrés satisfaisants.

Dans notre partie d'étude réservée à la conception de la SPLL, nous avons voulu lui conférer l'aspect de notre DPLL. Dans ce contexte, le comportement non linéaire du DFP ainsi que le comportement linéaire du filtre de boucle ont été traduits par des algorithmes qui ont été démontrés, et dont celui du DFP a été vérifié sur des exemples comprenant des fréquences d'entrée supérieure et inférieure à celle du DCO. Cette dernière fonction à l'instar du VCO de la DPLL, est par contre considérée parfaite et présente une caractéristique tension-fréquence tout à fait linéaire. Nous lui avons donc attribué autant de valeurs de sensibilité K_o qu'il y a de fréquences d'entrée-test, et cela, par interpolation linéaire menée sur la caractéristique non linéaire de la tension-fréquence du VCO de la DPLL en Figure II.27.

L'implémentation de l'algorithme SPLL sur un DSP à virgule flottante de 32 bits, a été effectuée par le biais d'un simulateur du C6713 de Texas instruments. L'essentiel du programme est partagé entre l'acquisition du signal d'entrée à l'aide de deux routines d'interruptions hardwares externes, le traitement SPLL assuré par une routine d'interruption software, et la récupération des résultats graphiques sur Matlab basée sur une liaison RTDX. Dans cette première version du programme, l'exécution en temps réel de la SPLL s'est estompée pour une fréquence d'entrée de 200kHz et plus.

Lors d'exécutions en temps différé, des réponses du filtre de boucle à caractères oscillatoires plus ou moins amorties sont relevées pour des fréquences d'entrée fixées à 100kHz, 200kHz, 289kHz, et 625kHz. Ces réponses filtrées se rapprochent dans leurs tendances moyennes de celles du filtre de la DPLL simulée sous Pspice. Voulant améliorer le fonctionnement de la SPLL, les anciennes constantes de temps du filtre de boucle dimensionnées lors de la conception de la DPLL, ont été adoptées de nouveau. Le facteur d'amortissement optimal résultant, amortit les oscillations de la SPLL pendant son régime transitoire tout en réduisant son temps de réponse. Ainsi et au bout de 2ms, les nouvelles réponses filtrées de la SPLL correspondant aux mêmes fréquences croissantes d'entrée précédentes, accusent respectivement des écarts relatifs de $0.76\%, 0.085\%, 0.06\%$, et 0.77%, par rapport aux réponses filtrées de la DPLL. Les régimes permanents obtenus de la SPLL sont tous exempts d'oscillations résiduelles et sont atteints sous un temps de réponse n'excédant pas 0.2ms. Ces réponses optimales constituent des valeurs moyennes pour les réponses oscillatoires amorties de la DPLL et de la SPLL, une fois que ces deux PLLs sont munies du filtre à constantes de temps modifiées.

Le filtre de boucle est un élément essentiel dans toute PLL. En plus des signaux indésirables accompagnant le signal du détecteur de phase qu'il doit filtrer, il est appelé à améliorer le comportement statique et dynamique de la boucle asservie de la PLL. Les constantes de temps du filtre ont été initialement optimisées, assurant par là, les meilleures réponses de la SPLL aux différentes fréquences d'entrée. La SPLL n'est qu'un programme exécutant parfaitement le comportement théorique prédit par les modèles d'équations pré-établis. La DPLL par contre, n'a pas pu suivre à la lettre son modèle mathématique optimisé et a exigé un raffinement empirique des constantes de temps de son filtre de boucle afin d'améliorer ses réponses aux différentes fréquences d'entrée. La circuiterie formant la DPLL cache ses multiples surprises en termes de capacités parasites des composants utilisés, les temps de transition non nuls des signaux traités suite aux bandes limitées des circuits, les caractéristiques non linéaires de la majorité des dispositifs, les désadaptations éventuelles entre étages, la dissymétrie même légère des courants de charge et décharge du condensateur de la pompe de charge

et ce en fonction de la fréquence d'entrée, affectant ainsi la symétrie des signaux mis en jeu, etc..

Pspice est d'autant précis dans la prise en compte de ces différents aspects que les modèles descriptifs émanent des constructeurs de composants utilisés, sont plus sophistiqués. Ces remarques de fond expliquent pourquoi que les constantes de temps optimales du filtre de boucle n'ont pas pu drainer d'aussi bonnes réponses dynamiques en DPLL simulée par Pspice, comparativement à sa version imitée en SPLL par implémentation sur DSP et qui est parvenue à la surpasser.

Dans une SPLL, les signaux existants physiquement peuvent se réduire à l'unique signal de référence $u_1(t)$. Mais un système SPLL réel et fonctionnel doit avoir un ou plusieurs signaux de sortie. Lorsque une SPLL est utilisée en tant que décodeur FSK par exemple, le signal démodulé est représenté par $u_f(n)$ en sortie du filtre, qui est directement calculé à chaque service d'interruption. Il est plus simple d'appliquer u_f à un port de sortie du DSP chaque fois qu'il est calculé. La situation devient plus gênante lorsque le système SPLL est appelé à délivrer continûment le signal $u_2(t)$ en sortie DCO, car en plus de la détermination des instants de fronts montants de u_2, il faut aussi régénérer ses fronts descendants lorsque φ_2 dans sa croissance, croise π cette fois-ci, Figure III.2.

Comme constaté à partir des formes d'onde de la Figure III.2, l'algorithme SPLL détermine au temps $t = t(n)$, l'instant de la dernière transition émise par le signal u_2. C'est-à-dire, qu'il connaît uniquement ce qui est arrivé dans le passé. Pour délivrer un signal u_2 en temps réel délivrant ses transitions même entre les instants d'échantillonnage, l'algorithme SPLL peut extrapoler au temps $t(n)$, le temps nécessaire devant s'écouler jusqu à la prochaine transition de u_2. Le temps de retard correspondant peut être chargé dans un timer, qui cause une autre requête d'interruption une fois ce temps achevé. La routine d'interruption correspondante doit actualiser le bit d'un port de sortie avec l'état courant du signal u_2. Ce simple exemple met en évidence la lourdeur éventuelle lorsqu'un simple circuit hardware d'une PLL est à remplacer par software chez une SPLL.

BIBLIOGRAPHIE

[1] R.E. BEST, "Phase-Locked Loops : Design, Simulation and Applications", 4th Edition, McGraw-Hill, 1999.

[2] Wikipedia, the free encyclopedia, "Phase-locked loop", lien sur le net : http://en.wikipedia.org/wiki/Phase-locked_loop

[3] F. M. Gardner, "Phaselock Techniques", 2nd edition, New York, John Wiley and sons, 1979.

[4] J. Crawford, "Frequency Synthesizer Design Handbook", Artech House, 1994.

[5] D. Abramovitch, "Phase-Locked Loops: A Control Centric Tutorial", Proceedings of the 2002 American Control Conference, AACC, IEEE, 8-10 May, 2002.

[6] S. Yichuang, "Wireless Communication Circuits and Systems", The IEE London, UK, 2004.

[7] B. Razavi, "RF Microelectronics", Prentice-Hall, 1998.

[8] H. Meyer, M. Moeneclay et S. Fechtel, "Digital Communications Receivers: Channel Estimation and Signal Processing", John Wiley and Sons, 1998.

[9] R. E Best, "Phase Locked Loops : Theory, Design and Applications", McGraw-Hill, 1984.

[10] V. Prasad et C. Sharma, "A review of phase locked loop". International J. of emerging tech. and adv. Eng. Vol 2, issue 6, 2012.

[11] K. A. Lacanette, "Basic Introduction to Filters - Active, Passive, and Switched-Capacitor", National Semiconductor Application Note 779, 1991.

[12] S. Goldman, "Phase-locked loop engineering handbook for integrated circuits" ARTECH HOUSE, INC., 2007.

[13] J. H. Vincent, "On Some Experiments in Which Two Neighboring Maintained Oscillatory Circuits Affect a Resonating Circuit", Proceedings Royal Society, Vol. 32, part 2, 84-91, 1919-1920.

[14] E. V. Appleton, "Automatic synchronization of triode oscillators, part iii", Dans Proc. Cambridge Phil. Soc., tome 21, 231–248, 1922.

[15] B. Van der Pol, "Forced oscillators in a circuit with non-linear resistance. (reception with reactive triode)", Phil. Mag., tome 3, 64-80, 1927.

[16] H. De Bellescize, "La réception synchrone", Onde Electr., tome 11, 230-240, June 1932.

[17] K. R. Wendt et G.L. Fredendall, "Automatic Frequency and Phase Control of Synchronization in Television Receivers", Proc. IRE, 31, 1943.

[18] R. D. Huntoon, "Synchronization of Oscillators", Proc. IRE, 1947.

[19] S. C. Gupta, "Phase-locked loop", Proceedings of the IEEE, Vol. 63, 291-306, February 1975.

[20] W. C. Lindsey et C.M. Chie "A survey of digital phase-locked loops", Proc IEEE 69, 410–431, 1981.

[21] D. H. Wolaver, Phase-locked loop circuit design, Prentice Hall Edition, 1991.

[22] A. J. Viterbi, "Principles of coherent communications", Mc Graw–Hill, 1966.

[23] J. Klapper et J. T. Frankle, "Phase-locked and frequency-feedback systems", New York Academic Press, 1972.

[24] A. Blanchard, "Phase–locked loops", Wiley, 1976.

[25] F. M. Gardner et W. C. Lindsey, "Special issue on synchronisation, part i. " IEEE trans. on Comm., tome COM–28, pp 1105-1388, 1980.

[26] W. C. Lindsey et C. M. Chie, "Special issue on pll, part i." IEEE trans. on Comm., tome COM–28, pp 2221–2411, 1982.

[27] P. R. Westlake, "Digital phase control techniques", IRE Trans Commun Syst 8, 237–246, 1960.

[28] C. J. Byrne, "Properties and design of the phase-controlled oscillator with a sawtooth comparator", Bett. Syst. Tech. J. 41, 559–602, 1962.

[29] S. C. Gupta, "On optimum digital phase locked loops". IEEE Trans. Commun. Technol. COM-16(2), 340-344, 1968.

[30] T. Saeki et al., "A 2.5-ns clock access, 250-MHz, 256-Mb SDRAM with synchronous mirror delay", IEEE J. Solid-State Circuits 31, 1656-1665, 1996.

[31] A. Hatakeyama et al., "A 256-Mb SDRAM using a register-controlled digital DLL", IEEE J. Solid-State Circuits 32, 1728-1732, 1997.

[32] C.H. Kim et al., "A 64-Mbit, 640-Mbyte/s bidirectional data strobed, double-data-rate SDRAM with a 40-mW DLL for a 256-Mbyte memory system", IEEE J Solid-State Circuits 33, 1703-1708, 1998.

[33] G. Geannopoulos et X. Dai, "An adaptive digital deskewing circuit for clock distribution networks", In: ISSCC digest of technical papers, 400-401, 1998.

[34] S. Eto et al., "A 1Gb SDRAM with ground level precharged bit line and nonboosted 2.1-V word line", IEEE J. Solid-State Circuits 33, 1697-1701, 1998.

[35] B. Garlepp et al., "A portable digital DLL for high-speed CMOS interface circuits", IEEE J. Solid-State Circuits 34(5), 632–642, 1999.

[36] F. Lin, J. Miller, A. Schoenfeld, M. Ma et R. J. Baker "A register-controlled symmetrical DLL for double-data-rate DRAM". IEEE J. Solid-State Circuits 34(4):565–568, 1999.

[37] R. B. Staszewski, D. Leipold, K. Muhammad et P. T. Balsara "Digitally controlled oscillator (DCO)-based architecture for RF frequency synthesis in a deep-submicrometer CMOS process", IEEE Trans. Circuits Syst II 50(11), 815-828, 2003.

[38] N. D. Dalt, E. Thaller, P. Gregorius et L. Gazsi, "A compact tripleband low-Jitter digital LC PLL with programmable coil in 130-nm CMOS", IEEE J. Solid-State Circuits, 40(7),1482-1490, 2005.

[39] J. Encinas, "Systèmes à verrouillage de phase (PLL) : réalisations et applications", éditions Masson, 1990.

[40] N. Batel, "Conception microélectronique : Sources contrôlées", cours dispensé en magister instrumentation et microélectronique, 2012.

[41] N. Batel, "Conception microélectronique : Miroirs de courant", cours dispensé en magister instrumentation et microélectronique, 2012.

[42] Texas Instruments, "TMS320C6000 CPU and instruction set reference guide", SPRU189F, 2000.

[43] Texas Instruments, "Code Composer user's guide", SPRU296, 1999.

[44] Texas Instruments, "TMS320C6000 DSP/BIOS Application Programming Interface reference guide", SPRU403C, 2001.

[45] Texas Instruments, "TMS320 DSP/BIOS user's guide", SPRU423A, 2001.

[46] Texas Instruments, "TMS320C6000 Chip Support Library API user's guide" SPRU401C, 2001.

[47] Texas Instruments, "TMS320C6000 peripherals reference guide", SPRU190D, 2001.

[48] Texas Instruments, "TMS320C6000 Instruction Set Simulator", Technical Reference, SPRU609F, 2005.