Optimizing the performance of dynamic sensor networks by controlling the synchronization in ultra wide band systems

Rshdee Alhakim

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Signal, Image, Parole, Télécom**

Arrêté ministériel : 7 août 2006

Présentée par

Rshdee ALHAKIM

Thèse dirigée par **Emmanuel SIMEU**
codirigée par **Kosai RAOOF**

préparée au sein du **Laboratoire TIMA**
dans **l'École Doctorale Electronique, Electrotechnique, Automatique & Traitement du Signal (E.E.A.T.S)**

Optimisation des Performances de Réseaux de Capteurs Dynamiques par le Contrôle de Synchronisation dans les Systèmes Ultra Large Bande

Thèse soutenue publiquement le **29 Janvier 2013**,
devant le jury composé de :

M. Salvador MIR
Directeur de recherche CNRS à l'université de Grenoble, Président
M. Abbas DANDACHE
Professeur à l'université de Lorraine, Rapporteur
M. Daniel ROVIRAS
Professeur au CNAM-Paris, Rapporteur
M. Eric SIMON
Maître de conférences à l'université de Lille, Examinateur
M. Kosai RAOOF
Professeur à l'université du Maine, Directeur de thèse
M. Emmanuel SIMEU
Maître de conférences à l'université de Grenoble, Directeur de thèse

Acknowledgments

I owe my gratitude to all the people who have made this thesis possible with their help, support and contributions.

First and foremost, I would like to thank my supervisors, Prof. Kosai RAOOF and Dr. Emmanuel SIMEU, who have given me an invaluable opportunity to do research and work on challenging and extremely interesting subjects over the past three years. They have been great mentors throughout my Ph.D. by helping me establish a direction of research and providing valuable guidance and advice. I will never forget the fun moments and the dialogues we had on various subjects.

Many thanks go to Prof. Salvador MIR, the jury president, Prof. Abbas DANDACHE, Prof. Daniel ROVIRAS, Dr. Eric SIMON, for coming from the different corners of France to serve on my thesis committee, and for sparing their invaluable time reviewing the manuscript.

All my colleagues at TIMA have enriched my graduate life in many ways. I would like to thank my colleagues Nourredine, Rafik, Louay, Brice, Laurent, Martin, Asma, Diane, as well as, Haralampos and the rest of the staff. Special thanks go to Youssef SERRESTOU for his helpful advice and mathematical expertise. I would like to express my sincere thank to Damascus University for supporting this work.

Last but not least, I owe my deepest thanks to my wonderful family; I would like to thank my precious wife Nada for her love, support, and unlimited kindness. She has always been by my side, especially during the hardest moments of the Ph.D. I would like to thank my parents, my sisters and my brothers-in-law Ali and Ashraf, who are always there for me even though we are a thousand miles apart. I express my gratitude to my parents for having guided me through life, and supported and encouraged me to move to France to pursue my Master’s and Ph.D. studies.

Contents

Résumé Étendu en Français

La technologie Ultra Large Bande (UWB) est en pleine croissance depuis 2002 en particulier dans le domaine des communications sans fil de courte portée. Parmi les applications les plus prometteuses, il y a les réseaux de capteurs communicants sans fil (WSNs). Il s'agit d'un réseau ad-hoc avec un grand nombre de nœuds communicants entre eux et distribués sur une zone donnée. Leur but est par exemple de surveiller un évènement ou de récolter et de transmettre des données environnementales d'une manière autonome (température, humidité, pression, accélération, sons, image, vidéo, etc.). Dans les WSNs, les critères les plus importants à prendre en compte sont : la conception d'un nœud à faible coût, un facteur de forme réduit, une faible consommation énergétique et des communications robustes.

Ces travaux de thèse portent principalement sur les transmissions Radio Impulsionnelle Ultra Large Bande (IR-UWB) qui présentent plusieurs avantages de par la très large bande utilisée (entre 3.1GHZ et 10.6GHZ) : des débits élevés et une très bonne résolution temporelle. Ainsi, la très courte durée des impulsions émises assure une transmission robuste dans un canal multi-trajets et dense. De plus, la faible densité spectrale de puissance du signal permet au système UWB de coexister avec les applications existantes. De ce fait, la technologie UWB a été considérée comme une technologie prometteuse pour les applications WSN. Cependant, plusieurs défis technologiques subsistent pour l'implémentation des systèmes UWB, à savoir : gestion des distorsions différentes de la forme d'onde du signal reçu liées à chaque trajet, conception d'antennes très large bande de petites dimensions et peu coûteuses, synchronisation d'un signal impulsionnel, utilisation de modulations d'ondes d'ordre élevé pour améliorer le débit, etc. Les travaux présentés dans cette thèse relèvent de ces problématiques et concernent plus particulièrement l'étude et l'amélioration de la synchronisation temporelle dans les systèmes UWB.

La synchronisation temporelle dans les systèmes de communication sans fil dépend généralement du corrélateur glissant entre le signal reçu et un signal de référence local, généré dans le récepteur. Cependant, dans les systèmes IR-UWB, cette approche n'est pas seulement sous-optimale en présence de canal à multi-trajets résolvables, mais subit aussi une grande complexité de calcul et un long temps de synchronisation. Nous sommes donc très motivés à

rechercher une approche rapide et à faible complexité pour réaliser une synchronisation temporelle satisfaisante pour les systèmes IR-UWB. Dans cette thèse, nous proposons une approche efficace de synchronisation IR-UWB, appelé « *Dirty Template* : DT ». Cette technique est basée sur la corrélation du signal reçu avec un gabarit de référence extrait du signal reçu. Ce gabarit est appelé « sale » (*dirty*), car il est déformé par le canal inconnu et par le bruit ambiant. L'approche DT permet au niveau du récepteur d'augmenter l'énergie émis captée même lorsque le canal multi-trajets et les codes de sauts temporels (TH) sont tous deux inconnus. En outre, elle n'a besoin d'aucune estimation du canal de propagation et d'aucune génération de signal de référence « propre » au niveau de récepteur. Par conséquent, l'approche DT contribue à améliorer la performance de synchronisation IR-UWB et de réduire la complexité de la structure du récepteur.

Le système de synchronisation au niveau du récepteur est généralement constitué de trois phases (voir Figure 1): la détection du signal, l'acquisition temporelle et la poursuite. La **détection du signal** est la première unité, utilisée pour décider si le signal reçu est une impulsion désirée UWB ou seulement du bruit. L'**acquisition temporelle** est employée pour trouver un point de départ pour chaque symbole reçu et à réduire l'erreur de décalage temporel à une fraction de la durée d'impulsion UWB. La troisième étape est la **poursuite**. Son objectif est de verrouiller et maintenir une synchronisation satisfaisante tout au long de la période de transmission, même si des variations de décalage temporel de la forme d'onde reçue se produisent à la suite de mouvement relatif entre émetteur et récepteur (l'effet Doppler).

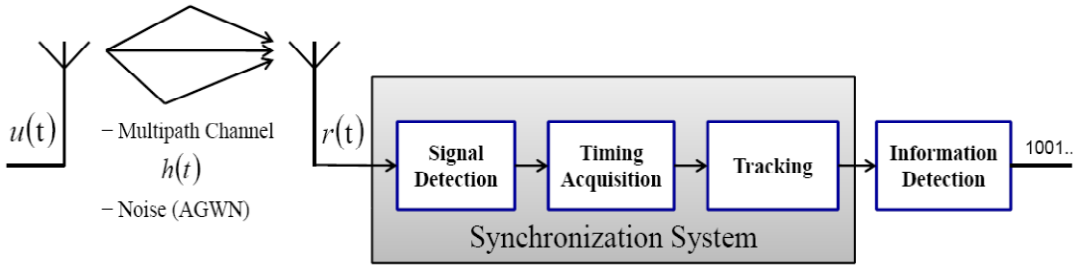


Figure 1. Système de synchronisation

Ce résumé est organisé comme suit : La Section 1 présente le modèle de signal UWB et la technique de synchronisation « *Dirty Template* : DT ». Dans la Section 2, nous allons concevoir un schéma de détection de signal basé sur l'approche DT. Le théorème de Neyman-Pearson sera appliqué pour définir un seuil optimal. Ce seuil sera comparé avec les échantillons reçus observées afin de prendre la décision si le signal UWB est présent ou non. La Section 3 extraira un estimateur d'acquisition temporelle basé sur les algorithmes DT. La performance de cet estimateur sera améliorée en modifiant la structure du corrélateur croisée. La Section 4 se focalisera sur la troisième étape du système de synchronisation: Poursuite. La boucle à verrouillage de délai DLL est une technique de poursuite fondamentale utilisée pour le maintien

d'une synchronisation satisfaisante. Pour concevoir la structure de DLL, nous proposerons deux approches: la première est la méthode classique basée sur un filtre de Wiener de second ordre, la seconde est basée sur une nouvelle stratégie de contrôle dans le domaine de la communication appelée Commande par Modèle Interne « *Internal Model Control* : IMC ». Le résumé se terminera par un plan du manuscrit et une liste de publications associées.

1. **Modèle de signal UWB et technique de synchronisation « *Dirty Template* »**

L'élément de base d'un signal UWB est un symbole $p_T(t)$ de durée T_s, qui correspond à la transmission d'un symbole d'information. Chaque temps symbole T_s est constitué de N_f trames de durée T_f. A chaque trame est subdivisée en N_c intervalles de temps (*chip*) de durée T_c. Le symbole $p_T(t)$ s'écrit : $p_T(t) = \sum_{j=0}^{N_f-1} p(t - jT_f - c_j T_c)$; où $p(t)$ est un impulsion UWB de durée très courte $T_p(\ll T_f)$. Dans une trame donnée un seul chip est porteur d'une impulsion UWB $p(t)$; ce chip est déterminé par un code pseudo-aléatoire de saut temporel (*Time-Hopping Code*) $\{c_j\} \in [0, N_c - 1], j \in [0, N_f - 1]$, comme le montre la Figure 2.

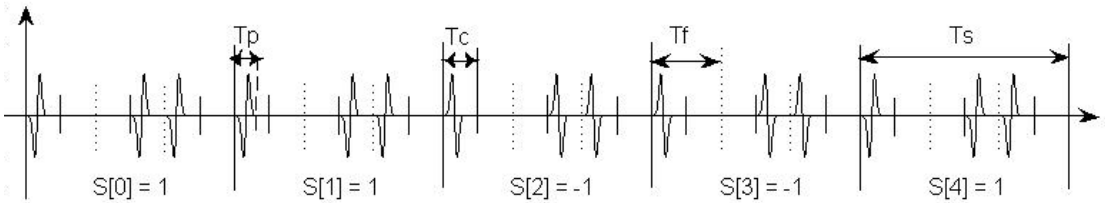


Figure 2. Présentation du signal transmis UWB ($N_f = 3, N_c = 2$, $c_j = [0,1,0]$)

Nous considérons que le symbole UWB a unité d'énergie ($\int {p_T}^2(t)dt = 1$). En se concentrant sur la modulation d'amplitude de l'impulsion (PAM), où les symboles porteurs d'information $s[n] \in \{\pm 1\}$ sont modélisés sous forme binaire indépendantes et identiquement distribués (i.d.) avec énergie $\mathcal{E}_s$ dispersée sur N_f trames. Le signal transmis peut alors s'écrire comme :

$$u(t) = \sqrt{\mathcal{E}_s} \sum_{n=0}^{\infty} s[n] p_T(t - nT_s) \quad (1)$$

On suppose que le signal $u(t)$ est transmis dans un canal à trajets multiples dont la réponse impulsionnelle s'écrit sous la forme $h(t) := \sum_{l=0}^{L-1} \alpha_l \delta(t - \tau_0 - \tau_{l,0})$; où le décalage temporel τ_0 se réfère à l'instant d'arrivée pour le premier trajet, α_l représente l'atténuation du $l^{ème}$ trajet et $\tau_{l,0}$ son retard relatif d'arrivée par rapport au premier trajet. L'impulsion reçue à l'intérieur de chaque trame est donnée par $p_r(t) := \sum_{l=0}^{L-1} \alpha_l p(t - \tau_{l,0})$. Le signal reçu dans la sortie de l'antenne de réception s'écrit :

$$r(t) = \sqrt{\mathcal{E}_s}\sum_{n=0}^{\infty} s[n]p_R(t - nT_s - \tau_0) + w(t) \quad (2)$$

où $w(t)$ est un Bruit Blanc Additif Gaussien (BBAG) de variance $N_0/2$, et $p_R(t)$ est la forme d'onde reçue de chaque symbole :

$$p_R(t) := \sum_{j=0}^{N_f-1} p_r\left(t - jT_f - c_jT_c\right) = \sum_{l=0}^{L-1} \alpha_l p_T\left(t - \tau_{l,0}\right) \quad (3)$$

Nous supposons dans cette étude que le signal reçu arrive avec un décalage τ_0 non connu à priori. Dans le cas où on désire synchroniser et détecter le signal reçu $r(t)$, le décalage τ_0 peut être développé comme suit $\tau_0 = n_sT_s + \tau$ tel que $n_s = \lfloor \tau_0/T_s \rfloor \geq 0$ représente le décalage temporel du niveau de symbole, la notation $\lfloor \cdot \rfloor$ représente la partie entière par défaut, et $\tau \in [0, T_s)$ est le décalage fin. Pour effectuer la détection des symboles émis, on doit estimer n_s. Puis pour effectuer la synchronisation, on doit estimer τ. En substituant τ_0 dans (2), le signal reçu peut être exprimé par :

$$r(t) = \sqrt{\mathcal{E}_s}\sum_{n=0}^{\infty} s[n]p_R(t - nT_s - n_sT_s - \tau) + w(t) \quad (4)$$

En présence du décalage temporel $\tau_0 \neq 0$, tout segment reçu de $r(t)$ de durée T_s peut être représenté par des parties de deux symboles consécutifs (voir Figure 3), comme le montre l'équation suivante; où $x(t)$ est un signal reçu de durée T_s, observé à la sortie de l'antenne :

$$x(t - nT_s) = w(t - nT_s) + \begin{cases} \sqrt{\mathcal{E}_s}s[n - n_s - 1]p_R(t - T_s - \tau) : t \in [0, \tau) \\ \sqrt{\mathcal{E}_s}s[n - n_s]p_R(t - \tau) \quad\quad : t \in [\tau, T_s) \end{cases} \quad (5)$$

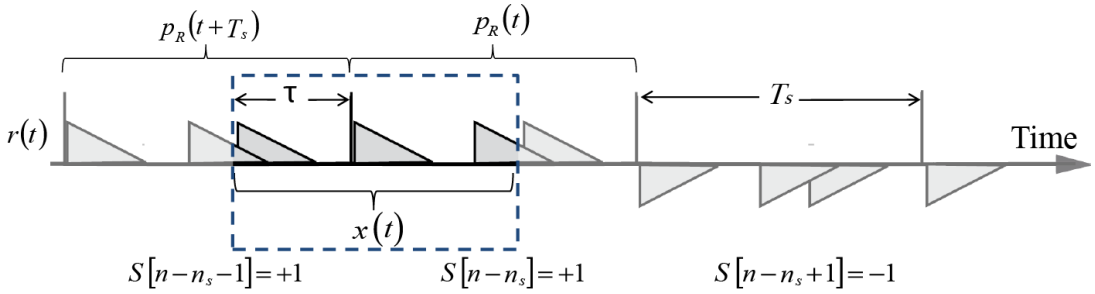


Figure 3. Signal reçu observé $x(t)$ de durée T_s

Il existe dans la littérature plusieurs techniques de synchronisation permettant la récupération du τ_0. Nous proposons dans ce travail d'utiliser la technique du « *Dirty Template* » introduite en 2005 par Giannakis. Cette technique est basée sur la corrélation croisée entre des paires de segments successifs reçus de durée T_s. En d'autres termes, un segment de chaque paire de symboles successifs reçue sert de signal de référence pour l'autre segment, comme indiqué ci-dessous :

$$R_{x,x}[n] = \int_0^{T_s} x(t - nT_s)x(t - (n+1)T_s)\, dt \quad (6)$$

Après une série d'opérations mathématiques, $R_{x,x}[n]$ peut s'écrire :

$$R_{x,x}[n] = \breve{\omega}[n] + A \int_{T_s-\tau}^{T_s} p_R^2(t)\, dt + B \int_0^{T_s-\tau} p_R^2(t)\, dt \quad (7)$$

où $A = \mathcal{E}_s s[n - n_s - 1].s[n - n_s]$, $B = \mathcal{E}_s s[n - n_s].s[n - n_s + 1]$, et $\breve{\omega}[n]$ est le bruit d'échantillonnage en sortie du corrélateur. Nous avons trouvé que $\breve{\omega}[n]$ peut être représenté comme un bruit gaussien de moyenne nulle $\breve{\omega}[n] \sim \mathcal{N}(0, \sigma_w^2)$. Les valeurs possibles de A et B sont exposées dans le Tableau 1.

Tableau 1 Les valeurs possibles de A et B dans (7)

$\boldsymbol{s[n-1]}$	+1	+1	-1	-1	+1	-1	+1	-1
$\boldsymbol{s[n]}$	+1	-1	-1	+1	+1	-1	-1	+1
$\boldsymbol{s[n+1]}$	-1	-1	+1	+1	+1	-1	+1	-1
$\boldsymbol{A}$	$+\mathcal{E}_s$	$-\mathcal{E}_s$	$+\mathcal{E}_s$	$-\mathcal{E}_s$	$+\mathcal{E}_s$	$+\mathcal{E}_s$	$-\mathcal{E}_s$	$-\mathcal{E}_s$
$\boldsymbol{B}$	$-\mathcal{E}_s$	$+\mathcal{E}_s$	$-\mathcal{E}_s$	$+\mathcal{E}_s$	$+\mathcal{E}_s$	$+\mathcal{E}_s$	$-\mathcal{E}_s$	$-\mathcal{E}_s$

La technique « Dirty Template » peut être appliquée à un récepteur UWB, même en présence de codes de sauts temporels TH ou d'interférences entre trames consécutives (*Inter-Frame Interference :* IFI). En outre, cette méthode exploite la diversité de trajets multiples fournis par les canaux UWB, et elle n'a besoin d'aucune estimation du canal de propagation et d'aucune génération de signal de référence propre au niveau de récepteur. Par conséquent, cette technique réduit la complexité du récepteur avec une grande quantité de l'énergie émise captée au niveau du récepteur.

2. La phase de détection du signal UWB

Après avoir examiné la technique *Dirty Template*, nous pouvons maintenant entrer dans les détails du processus de synchronisation. La première étape est d'étudier la détection du signal. Comme nous l'avons dit plus tôt, le signal transmit UWB $u(t)$ se propage dans un canal de transmission imparfait, et subit des modifications et le bruit ambiant lors de sa propagation. La tâche du détecteur est donc de retrouver le symbole émis à partir d'observations "noyées" dans un bruit, ou de prendre une décision si le signal reçu est une impulsion désirée UWB ou que du bruit, en faisant le minimum d'erreurs de décision. Alors, le récepteur a à choisir entre deux hypothèses notées $\mathcal{H}_0$ (le signal UWB est présent) et $\mathcal{H}_1$ (le signal UWB est absent) respectivement.

Afin d'effectuer la détection du signal $r(t)$ et l'estimation de la valeur de n_s, nous utilisons la technique *Dirty Template*. Supposons que nous envoyons une séquence d'apprentissage de longueur M_1 symboles et de valeur égale 1 $\{s[n]=1\}_{n=0}^{M_1-1}$. Dans ce cas, en consultant le Tableau 1 et en prenant les valeurs correspondantes de A et B, puis en les substituant dans (7), la valeur de $R_{x,x}[n]$ observée à la sortie du corrélateur est :

$$R_{x,x}[n] = \underbrace{\mathcal{E}_s \int_0^{T_s} P_R^2(t)\,dt}_{\mathcal{E}_r} + \widetilde{\omega_d}[n] = \mathcal{E}_r + \widetilde{\omega_d}[n] \tag{8}$$

où l'énergie $\mathcal{E}_r$ est une valeur constante au cours de la présence de la séquence d'apprentissage. Alors, le problème de détection est de faire la distinction entre ces deux hypothèses:

$$\begin{cases} \mathcal{H}_0 \colon R_{x,x}[n] = \widetilde{\omega_{d0}}[n] & n = 0,1,\dots,M_1-1 \\ \mathcal{H}_1 \colon R_{x,x}[n] = \mathcal{E}_r + \widetilde{\omega_{d1}}[n] & n = n_s, n_s+1,\dots,n_s+M_1-1 \end{cases} \tag{9}$$

où $\widetilde{\omega_{d0}}[n]$ et $\widetilde{\omega_{d1}}[n]$ représentent le bruit de l'échantillonnage sous $\mathcal{H}_0$ et $\mathcal{H}_1$ respectivement, et sont considérés comme bruit gaussien de moyenne nulle mais de variances différentes $\widetilde{\omega_{d0}}[n] \sim \mathcal{N}(0,{\sigma_0}^2)$, $\widetilde{\omega_{d1}}[n] \sim \mathcal{N}(0,{\sigma_1}^2)$ respectivement. Nous remarquons que dans les deux hypothèses l'observation $R_{x,x}[n]$ suit une distribution normale et sa fonction de densité de probabilité est représentée par les équations suivantes:

$$\begin{cases} p\big(R_{x,x}[n];\mathcal{H}_0\big) = \frac{1}{(2\pi{\sigma_0}^2)^{M_1/2}}\, exp\left[-\frac{1}{2{\sigma_0}^2}\sum_{m=n}^{n+M_1-1}\big(R_{x,x}[m]\big)^2\right] \\ p\big(R_{x,x}[n];\mathcal{H}_1\big) = \frac{1}{(2\pi{\sigma_1}^2)^{M_1/2}}\, exp\left[-\frac{1}{2{\sigma_1}^2}\sum_{m=n}^{n+M_1-1}\big(R_{x,x}[m]-\mathcal{E}_r\big)^2\right] \end{cases} \tag{10}$$

Pour concevoir la structure du processus de détection du signal, nous employons le théorème de Neyman Pearson (NP). Ce théorème affirme que si le rapport de vraisemblance $J(n)$ mentionné dans (11) est supérieur à une certaine valeur (η) on dit que le signal est présent, sinon on parle de signal absent, où η est un seuil fixé par la probabilité de fausse alarme désirée :

$$J(n) = \frac{p\big(R_{x,x}[n];\mathcal{H}_1\big)}{p\big(R_{x,x}[n];\mathcal{H}_0\big)} > \eta \tag{11}$$

En développant (11), le test de Neyman Pearson (T) peut s'écrire comme indiqué ci-dessous.

$$T(n) = \sum_{m=n}^{n+M_1-1}\left\{R_{x,x}^2[n] + \frac{2\sigma_0^2}{N_0}R_{x,x}[n]\right\} > \underbrace{\left(ln(\eta) - M_1 ln\left(\frac{\sigma_0}{\sigma_1}\right) + \frac{M_1{\mathcal{E}_r}^2}{2\sigma_1^2}\right)\frac{2\sigma_0^2\sigma_1^2}{\mathcal{E}_r N_0}}_{\xi} \tag{12}$$

La Figure 4 montre le schéma de la détection du signal UWB en utilisant la technologie *Dirty Template*. Alors la processus de détection compare entre la valeur générée de test T et le seuil ξ, si $T > \xi$ on dit que le signal est présent, sinon il est absent.

Les performances d'un test d'hypothèse ou d'un détecteur sont caractérisées par les probabilités de détection P_D et de fausse alarme P_{FA}. En effet, un test performant est un test pour lequel la probabilité de détection P_D est suffisamment importante tout en garantissant le plus petit niveau de probabilité de fausse alarme P_{FA}. Les probabilités P_D et P_{FA} sont définies, comme suit:

$$\begin{cases} P_{FA} = Pr\{T > \xi; \mathcal{H}_0\} \\ P_D = Pr\{T > \xi; \mathcal{H}_1\} \end{cases} \tag{13}$$

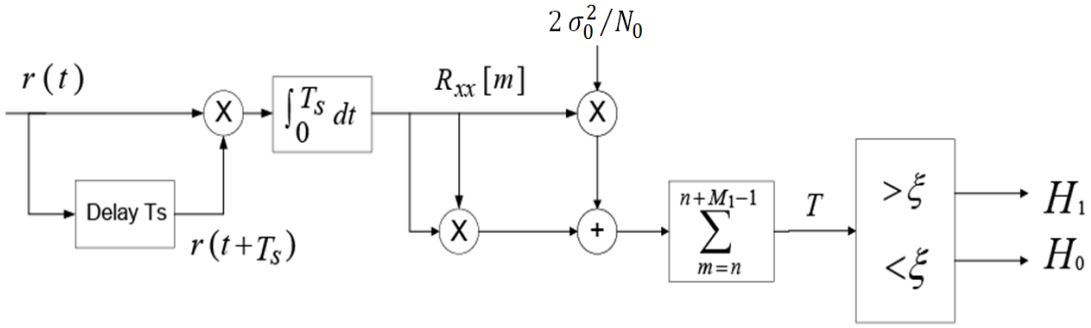


Figure 4. Schéma du processus de détection du signal pour le système *Dirty Template*

La Figure 5 montre un moyen de représentation de la performance de détection appelée « caractéristique opérationnelle de réception » (*Receiver Operating Characteristic* : ROC). Cette figure exprime P_D en fonction de P_{FA} pour différentes valeurs du seuil ξ, du SNR et du nombre de symboles observés M_1. Nous remarquons que l'augmentation du seuil ξ entraîne une diminution de P_{FA}. Toutefois, cela a malheureusement pour effet secondaire de diminuer P_D. Le cas idéal est représenté sur la Figure 5 par la courbe verte, où P_D est égal à 1 quelle que soit la valeur de P_{FA}. Nous remarquons que lorsque le SNR ou M_1 augmente, la courbe ROC s'approche le cas idéal (ligne verte), et la performance de détection est alors améliorée.

Après la détection du signal, le décalage temporel du niveau de symbole n_s peut être estimé en utilisant un algorithme de recherche linéaire pour trouver la valeur optimale de n qui maximise la valeur de test $T(n)$, comme indiqué ci-dessous :

$$\hat{n}_s = arg \max_n T(n); \qquad T(n) = \sum_{n=m}^{m+M_1-1} \left(R_{x,x}^2[n] + \frac{2\sigma_0^2}{N_0} R_{x,x}[n] \right) \tag{14}$$

Par conséquent, le problème de la détection dans les systèmes UWB peut être conçu en deux étapes : La première étape est d'observer la valeur de test $T(n)$ à la sortie du détecteur représenté sur la Figure 4. Si $T(n)$ dépasse le seuil ξ, la détection du signal est déclarée et nous passons à la deuxième étape qui à son tour applique l'algorithme de recherche linéaire proposé en ligne (14) pour estimer le décalage temporel du niveau de symbole n_s.

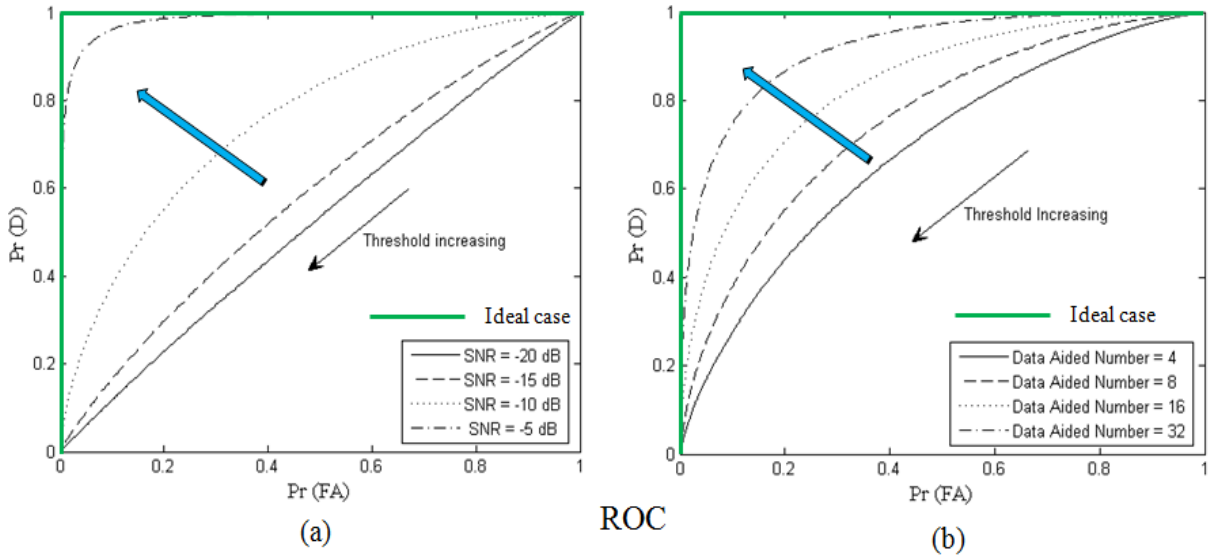


Figure 5. Caractéristique opérationnelle de réception : (a) $M_1 = 8$, (b) SNR= -10 dB

3. La phase d'acquisition temporelle

Après avoir détecté le signal UWB, nous passons à la deuxième étape du système de synchronisation : Acquisition temporelle. Dans cette section, nous allons essayer de concevoir un estimateur de synchronisation basé sur l'approche « *Dirty Template* ». Cet estimateur nous permettra de trouver la valeur du décalage temporel $\tau \in [0, T_s)$ entre le segment reçu et la fenêtre d'observation (voir Figure 3). L'estimateur d'acquisition DT est développé en utilisant la méthode du maximum de vraisemblance ou de la moyenne quadratique. Nous avons trouvé analytiquement que l'acquisition peut être accomplie par :

$$\hat{\tau} = \underset{m}{arg\,max}\ \frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n, \tau - \tau_e] \quad : \tau_e = m\Delta_t \in [0, T_s) \tag{15}$$

où R_{xx} est la sortie de la corrélation entre le signal reçu et le signal de référence bruité, exprimée dans (6), M_2 est le nombre d'échantillons de sortie utilisés pour la réalisation de l'opération d'acquisition DT, $\tau_e = m\Delta_t$: Δ_t représente le pas d'incrémentation et m représente le nombre d'incrémentations. La Figure 6 montre le schéma de l'acquisition DT. ICI, l'acquisition est effectuée en faisant varier la valeur de τ_e entre 0 et T_s tandis que l'unité de décision recherche une énergie de pic maximale (une crête) à la sortie, et décide de la valeur optimale correspondante de τ_e.

Cependant, lorsque nous appliquons cet estimateur, plusieurs points de maximum autour du pic à la sortie de la corrélation sont observés. Ceux-ci peuvent dégrader l'estimation de l'erreur de décalage temporel (TOE) (voir Figure 7). Pour pallier cette difficulté, un filtre à fenêtre

(window filter) $W(t)$ adapté à la structure de la corrélation est ajouté, permettant d'obtenir un seul pic maximal dans l'intervalle d'estimation et améliorant l'estimation de l'erreur de décalage temporel. Ce filtre à fenêtre contribue également à réduire les effets indésirables du bruit sur l'estimation de l'acquisition. Dans ce cas, l'estimation dans (15) peut être obtenue de la manière suivante :

$$\hat{\tau} = \underset{m}{arg\,max}\ \frac{1}{M_2}\sum_{n=0}^{M_2-1}\left(\int_0^{T_s} x(t+nT_s-\tau_e).x(t+(n+1)T_s-\tau_e).W(t)dt\right)^2 \quad (16)$$

où la fenêtre W (t) contient des informations de code de sauts temporels TH, comme indiqué ci-dessous :

$$W(t) = \sum_{j=0}^{N_f-1} c_j p\left(t-jT_f\right);\ p(t) = \begin{cases} 1 & si\ \ 0 \le t \le T_w \\ 0 & sinon \end{cases} \quad (17)$$

où la largeur de la fenêtre doit être : $T_w \in \{T_c, 2T_c\}$. La Figure 7 montre l'énergie captée à la sortie du corrélateur glissant pour différentes valeurs de $\tau_e = m\Delta_t \in [0, T_s)$. La figure comporte trois graphiques, l'un (en pointillés) représente l'estimateur DT sans fenêtre, les autres (en pointu et solide) représentent l'estimateur DT avec une fenêtre de largeur égale à $6T_c$ et à $2T_c$ respectivement. Cette figure montre que la sortie du corrélateur sans fenêtre comporte plusieurs points de maximum autour du pic. En outre, la diminution de la taille de la fenêtre $W(t)$ conduit à diminuer le nombre de points de maximum, jusqu'à ce que nous atteignons un point désiré ($\tau_e \approx \hat{\tau}$). Les résultats des simulations sous Matlab ont montré la fiabilité de l'estimateur proposé et confirment que la performance de l'estimateur du décalage temporel s'améliore quand M_2 ou SNR augmente.

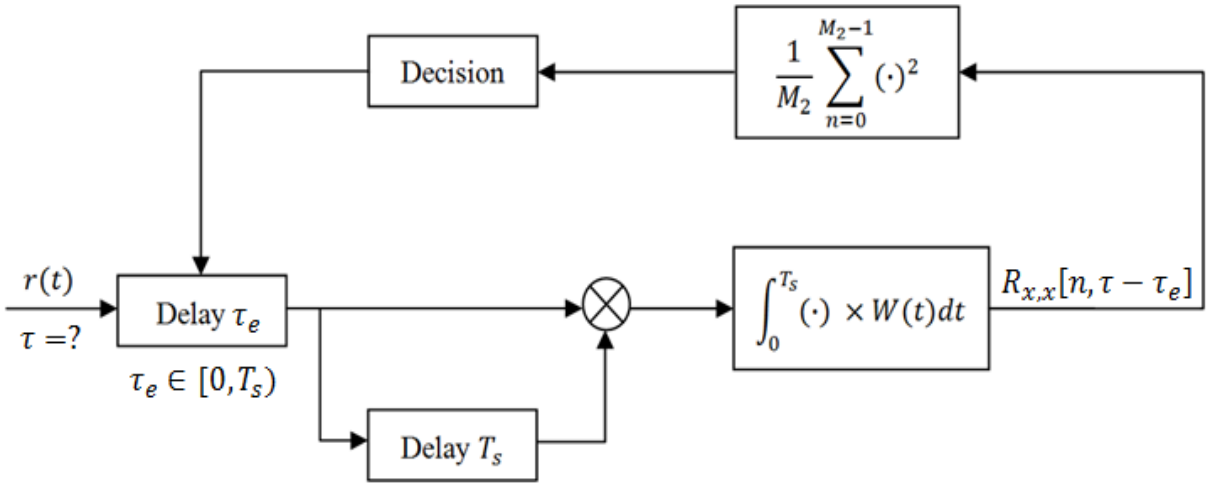


Figure 6. Schéma de l'acquisition DT

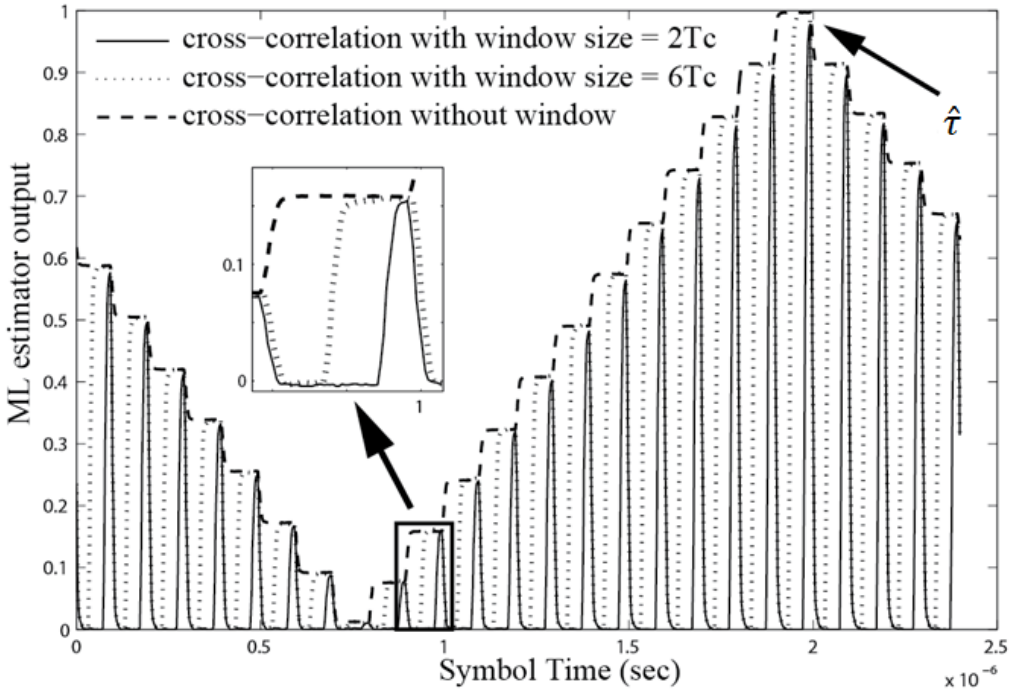


Figure 7. Sortie de l'estimateur DT

4. La phase de poursuite

Une fois la tâche d'acquisition temporelle a été correctement remplie, et nous pouvons trouver un point de départ approximatif de chaque symbole reçu. Nous allons maintenant nous concentrer sur la troisième phase du système de synchronisation qui est la poursuite. L'objectif de cette étape est de maintenir et verrouiller une synchronisation satisfaisante entre le récepteur et l'émetteur, même en présence de l'effet Doppler. Pour réaliser la poursuite, nous utilisons la technique fondamentale appelée boucle à verrouillage de délai (*Delay Locked Loop* : DLL). La Figure 8 montre la méthode DLL.

L'effet Doppler produira le décalage temporel ($\tau \neq 0$) entre le signal de référence et le signal reçu. La poursuite est effectuée en estimant le décalage temporel τ entre $r(t)$ et $y(t)$. Ce décalage est ensuite compensé en décalant la position de signal de référence vers $y(t-\hat{\tau})$, où $\hat{\tau}$ est l'estimation de τ. En d'autres termes, la poursuite DLL est utilisée pour minimiser l'erreur de décalage $\epsilon = \hat{\tau} - \tau$.

La poursuite est effectuée à l'aide de deux branches de corrélateur. Dans la première branche, le signal reçu est en corrélation croisée avec la version avancée de la référence par Δ; et R_e représente la sortie de corrélateur avancé. Et dans la seconde branche, le signal reçu est corrélé avec la version retardée du signal de référence par Δ; et R_l représente la sortie de corrélateur retardé. Les sorties de corrélation sont données par :

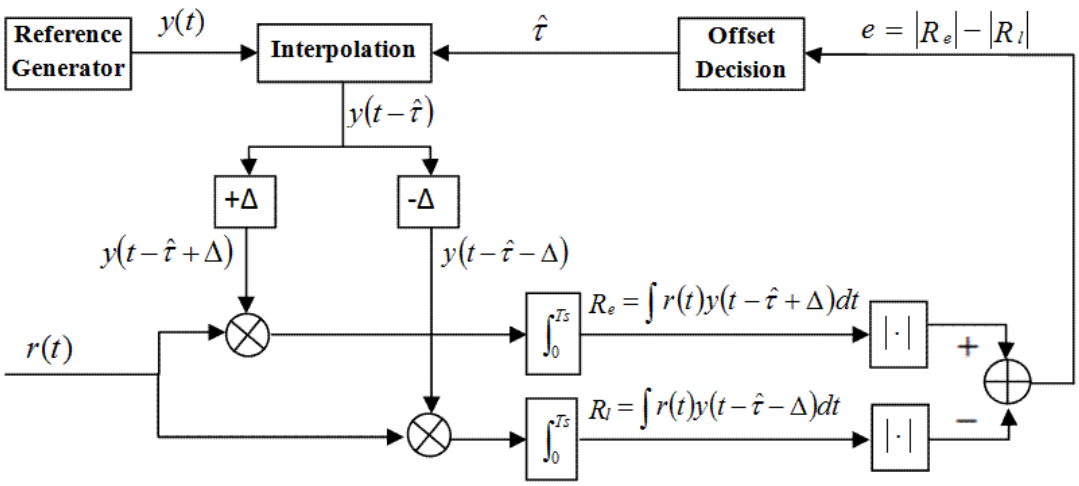


Figure 8. Diagramme de boucle à verrouillage de délai

$$\begin{cases} R_e = \int_0^{T_s} r(t)y(t-\hat{\tau}+\Delta)\,dt \\ R_l = \int_0^{T_s} r(t)y(t-\hat{\tau}-\Delta)\,dt \end{cases} \tag{18}$$

La valeur de l'erreur (e) est la différence entre ces deux sorties de corrélateur, et il est utilisé comme indicateur de l'erreur de synchronisation. e doit dépendre du décalage de synchronisation (ϵ) entre le signal reçu $r(t)$ et le signal de référence $y(t)$. La valeur de l'erreur (e) est proche de zéro lorsque le décalage est égal à zéro, et il augmente progressivement avec l'augmentation de décalage. Pour éviter l'influence du signe inconnu de données reçues sur la décision de la poursuite, on ajoute un bloc de valeur absolue à l'intérieur du diagramme de DLL. Après une série de manipulations mathématiques, on déduit l'erreur e :

$$e[\epsilon, n] = \mathbb{S}[\epsilon] + \eta[n] \tag{19}$$

où $\eta[n]$ est le bruit additif équivalent du système DLL, et peut être considéré comme un bruit gaussien de moyenne nulle $\eta[n] \sim \mathcal{N}(0, \sigma_\eta^2)$. La fonction $\mathbb{S}[\cdot]$ est la caractéristique de discriminateur de boucle (courbe en « S »), considéré comme le terme util pour le DLL. La courbe en « S » est représentée sur la Figure 9 pour valeurs différentes de ϵ. Pour être en mesure d'effectuer le processus de poursuite, le décalage temporel ϵ doit être compris dans l'intervalle $[-\Delta, +\Delta]$. La poursuite ne fonctionne que dans cet intervalle, sinon il y a risque de perdre la synchronisation. Pour cette raison, nos futurs efforts se concentreront sur la partie solide de la courbe en « S » (voir Figure 9).

Pour plus de simplification, le système DLL pourrait être représenté par le modèle équivalent montré dans la Figure 10. Nous nous concentrons maintenant sur la conception d'un bloc de décision (*Offset Decision*) qui est utilisé pour ajuster la position de référence du signal $y(t)$ afin de verrouiller la synchronisation entre $y(t)$ et $r(t)$, et de maintenir e proche de zéro.

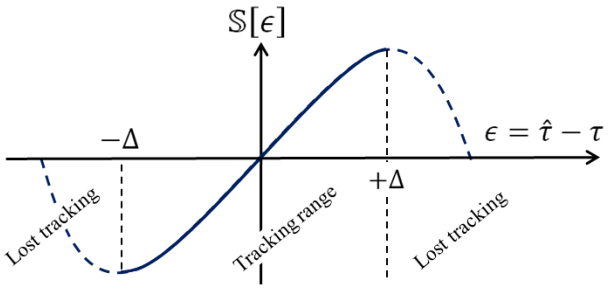


Figure 9. La courbe en « S »

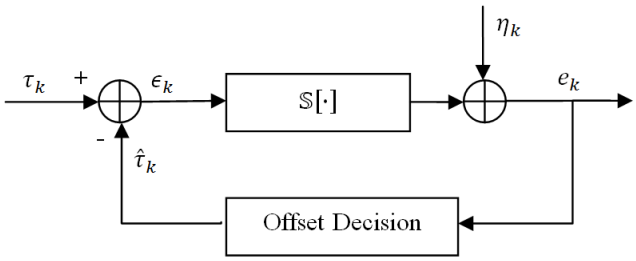


Figure 10. Modèle équivalent de DLL

Pour concevoir la boucle fermée appropriée, nous prenons en compte deux considérations :

- Nous avons besoin de minimiser les effets du bruit.
- Et nous devons être en mesure de suivre le signal d'entrée dans la présence d'un effet Doppler linéaire, représenté par cette équation :

$$\tau(t) = \tau_0 + mt \tag{20}$$

où τ_0 représente le temps de retard constant et m est le facteur de Doppler qui est produit par une vitesse constante par rapport à l'émetteur et le récepteur.

Pour atteindre cet objectif, nous proposons deux approches: la première est la méthode classique basée sur un filtre de Wiener de second ordre, la seconde est basée sur une nouvelle stratégie de contrôle dans le domaine de la communication et appelée « Commande par Modèle Interne » (*Internal Model Control* : IMC).

4.1 Filtre de Wiener de second ordre

En utilisant cette méthode, La fonction de transfert en boucle fermée de la DLL peut être exprimée par l'équation suivante, où ω_n et ξ sont respectivement la fréquence naturelle et le coefficient d'amortissement, et $A = \frac{d\mathbb{S}(\epsilon)}{d\epsilon}\Big|_{\epsilon=0}$ est la pente de la courbe en « S » :

$$F(s) = \frac{1+2\xi(s/\omega_n)}{A(s/\omega_n)^2} \tag{21}$$

La bande passante équivalente de bruit de la boucle s'écrit :

$$B_L = \frac{\xi\omega_n}{2}\left(1+\frac{1}{4\xi^2}\right) \ll 1/T_s \tag{22}$$

Pour sélectionner la valeur optimale de $\{\omega_n, \xi\}$ on applique la théorie de filtre de Wiener. Cette méthode d'optimisation permet de minimiser le critère de conception $\sigma_\epsilon^2 + \lambda^2\epsilon_T^2$, où σ_ϵ^2 est l'énergie d'erreur (variance) dû au bruit, ϵ_T^2 l'énergie d'erreur due au régime transitoire, et λ le multiplicateur de Lagrange (considéré comme poids relatif entre l'énergie du bruit et l'énergie de l'erreur transitoire). Sur la base de la théorie de filtre de Wiener, les paramètres optimaux de boucle du second ordre sont donnés comme suit :

$$\omega_n^2 = m\,\lambda\,\sigma_\eta^{-1}, \quad \xi = \sqrt{2}/2, \qquad B_L = 0.53\omega_n \ll 1/T_s \tag{23}$$

La Figure 11 présente un comportement de DLL en présence de variations de temps d'entrée $\tau(t)$ pour valeurs différents de ω_n. On voit que DLL suit le signal d'entrée $\tau(t)$ rapidement lorsque la fréquence naturelle ω_n augmente. Les comparaisons de BER pour différentes valeurs de ω_n sont représentées dans la Figure 12. On peut observer que pour des valeurs élevées du SNR, une augmentation de ω_n (ou B_L) conduit à accélérer l'opération de poursuite et à améliorer la performance du BER. D'autre part, pour les valeurs de SNR faible, la bande passante B_L doit être diminuée pour atténuer les effets du bruit; mais cela entraîne aussi la dégradation des performances du régime transitoire. Par conséquent, ω_n croissante contribue à réduire l'effet d'erreur transitoire, mais dans le prix de la capacité de gestion du bruit.

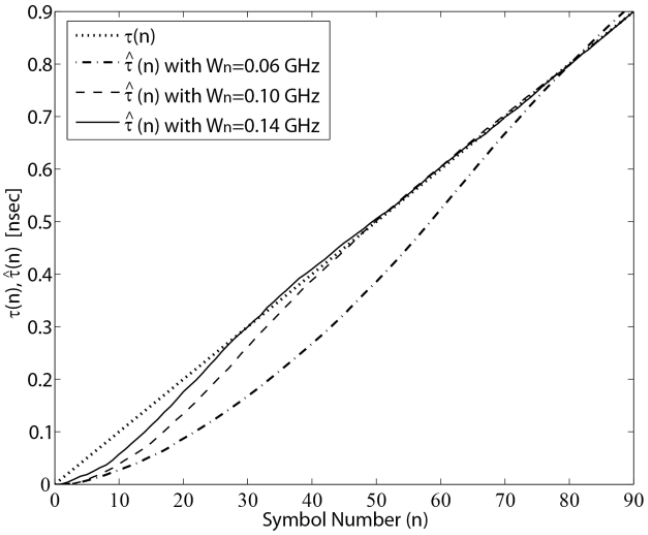


Figure 11. Réponse transitoire de DLL pour l'entrée de la rampe ($\tau(n) = 0.01nT_p$, SNR=3 dB)

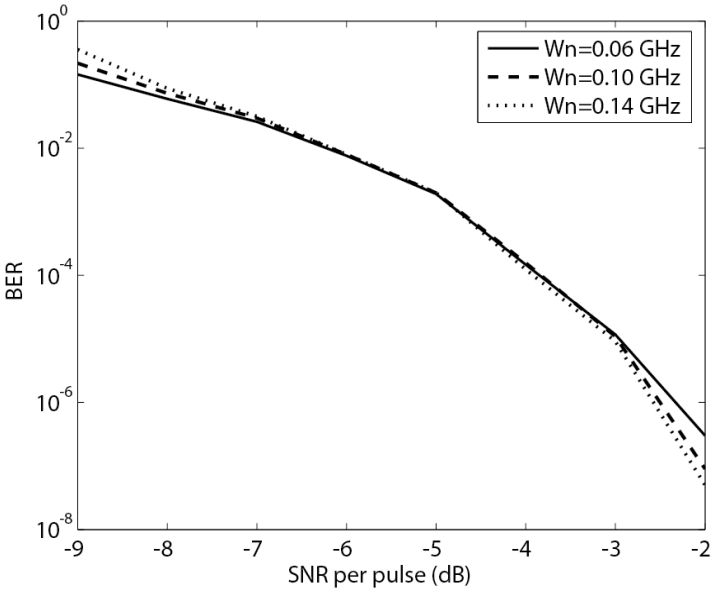


Figure 12. Performance de BER ($\tau(n) = 0.01nT_p$)

4.2 **Commande par Modèle Interne**

Dans cette section, nous traitons toujours de la boucle DLL de poursuite mais en utilisant maintenant une architecture basée sur IMC. La Figure 13 représente le schéma de l'IMC appliqué à la boucle DLL. Sur cette figure, on retrouve que la boucle IMC est composée de deux blocs (modèles) : le modèle inverse (modèle de contrôle) connecté en série avec le système DLL en boucle ouverte, et un modèle direct (modèle du système) connecté en parallèle avec la DLL en boucle ouverte. Afin de définir le modèle approprié pour ces deux blocs, le modèle du system et le modèle de contrôle doivent respectivement correspondre au modèle de courbe en « S » et à son inverse. Nous constatons que l'estimation de la courbe en « S » et son inverse peut être réalisée en utilisant un polynôme d'ordre 3 avec puissances impaires, comme suit :

$$\begin{cases} \text{modèle du système :} & \hat{e} = a_1\hat{\tau} + a_3\hat{\tau}^3 \\ \text{modèle du contrôle :} & \hat{\tau} = b_1\hat{d} + b_3\hat{d}^3 \end{cases} \tag{24}$$

où $\{a_1, a_3, b_1, a_3\}$ sont les coefficients qui définissent les deux modèles. En pratique, les coefficients de ces modèles sont obtenus par identification paramétrique. Cette identification est basée sur une acquisition des données d'entrée/sortie $\{\hat{\tau}_i, e_i\}$ pour $\mathrm{i} = 1, \ldots, \mathrm{N}$ et d'une recherche des coefficients du modèle en utilisant l'algorithme des Moindres Carrés.

Pour évaluer la performance du système de poursuite proposé, Nous exécutons les simulations en Matlab dans deux cas: dans le premier cas, l'effet Doppler est absent et le décalage temporel est constant $\tau = 0.5$. Nous pouvons voir sur la Figure 14 que $\hat{\tau}$ converge vers τ et reste très

proche de lui. Dans le second cas, nous avons un décalage Doppler linéaire. La Figure 15 montre que quand τ change, $\hat{\tau}$ le suit efficacement, mais à un certain point il y a perte de poursuite. Cette perte de poursuite est due au fait que les modèles du système de contrôle deviennent invalides lorsque $\hat{\tau}$ dépasse $\Delta = 1$ (voir la Figure 15).

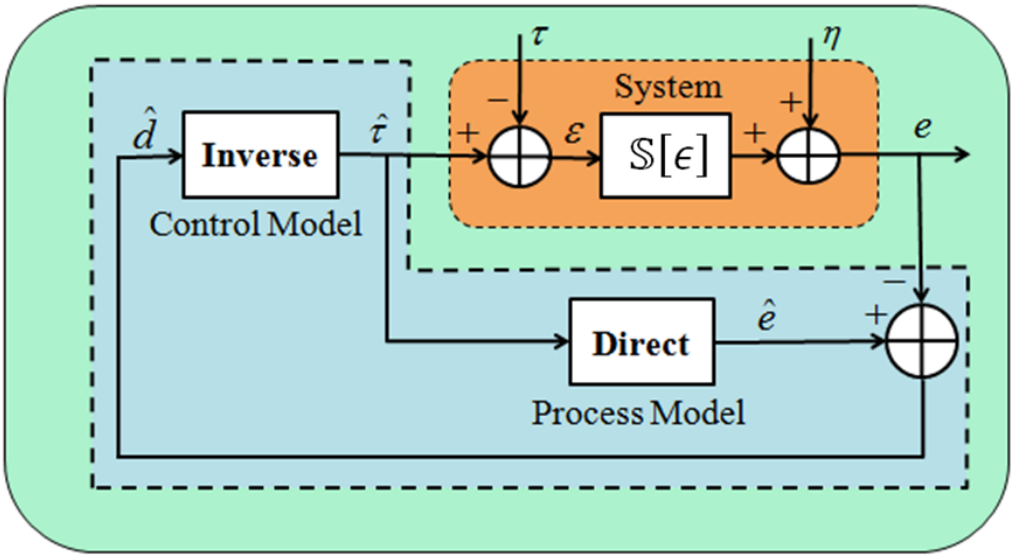


Figure 13. Conception du système DLL avec la technique IMC

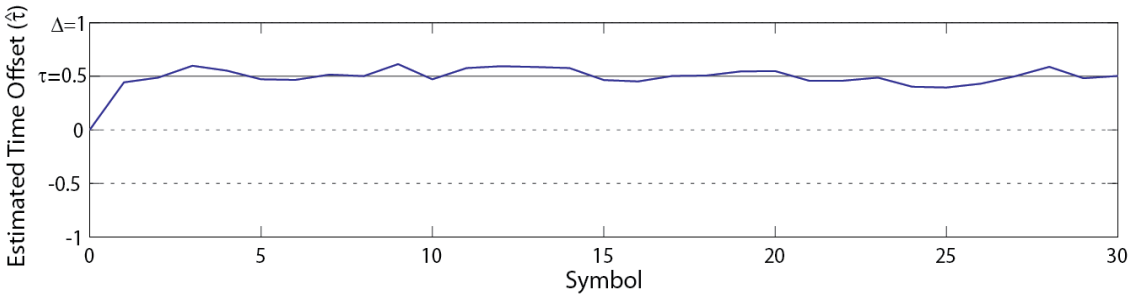


Figure 14. Evolution de la boucle IMC en l'absence de décalage Doppler (SNR=3dB et $\tau = 0.5$)

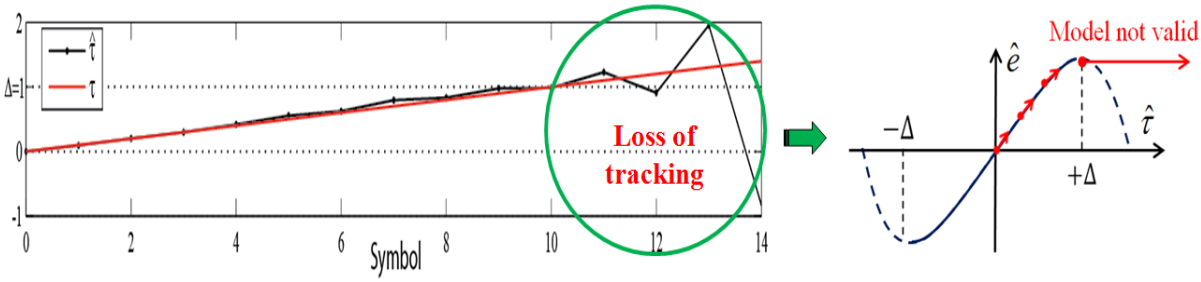


Figure 15. Evolution de la boucle IMC en la presence de décalage Doppler (SNR=20dB et $\tau(n) = 0.01nT_p$)

Pour résoudre la perte de poursuite due au décalage Doppler, nous modifions la structure de l'IMC par l'ajout de deux filtres : Filtre à moyenne mobile (*Moving Avewrage Filter*) et Filtre Adaptatif (*Adaptive Filter*), comme indiqué dans la Figure 16. Le but du Filtre à moyenne mobile est d'observer la valeur de τ (ou $\hat{\tau}$) : si elle dépasse un certain seuil, le filtre décale (met à jour) les modèles du système et du contrôle pour éviter la perte de poursuite. Ces changements sont représentés par les équations suivantes :

$$\begin{cases} \text{modèle du système :} & \hat{e} = a_1\hat{\tau} + a_3\left(\hat{\tau}_i - c_j\right)^3 \\ \text{modèle du contrôle :} & \hat{\tau} = b_1\hat{d} + b_3\hat{d}^3 + c_j \end{cases} \tag{25}$$

où c_j représente la valeur de décalage.

Le filtre à moyenne mobile, indiqué dans la Figure 16, a l'objective d'éliminer le biais de l'estimateur (le biais à la sortie du système : e). L'idée de ce filtre est d'observer le biais (la moyenne) de la sortie du système et de le compenser en le réinjectant dans l'entrée.

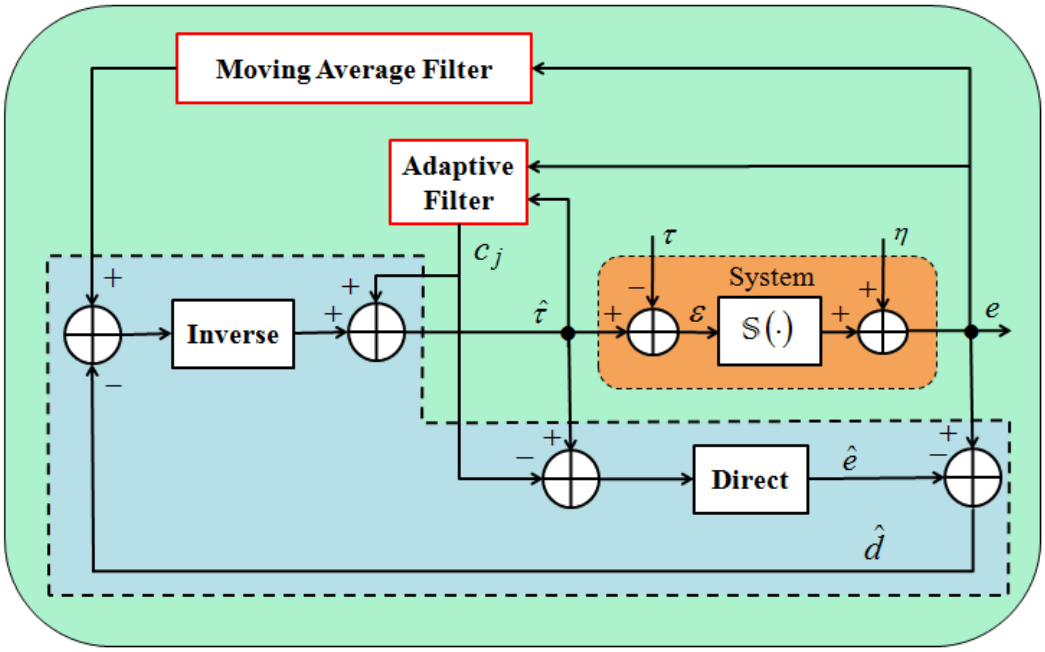


Figure 16. Conception du système DLL avec la boucle IMC modifiée

La Figure 17 présente le comportement des deux systèmes DLL de poursuite (DLL avec boucle fermée classique et DLL avec boucle fermée de l'IMC) en présence de décalage Doppler linéare. Il est clair que le DLL avec la boucle de l'IMC suit le signal d'entrée $\tau(n)$ environ dix fois plus vite que la DLL classique. Les comparaisons de BER pour les deux systèmes sont représentés sur la Figure 18. Nous observons que le système DLL classique a un meilleur BER. Ces figures montrent que si la vitesse relative entre l'émetteur et le récepteur est constante, la

poursuite du système DLL conventionnelle est meilleure. Toutefois, lorsque la vitesse relative accélère ou décélère, le système DLL IMC a une réponse transitoire rapide, capable de suivre efficacement la variation de l'entrée, tandis que le système DLL conventionnel a de forte chance de décrocher la poursuite. La recherche d'une structure hybride entre les deux méthodes mérite d'être poursuivie et généralisée.

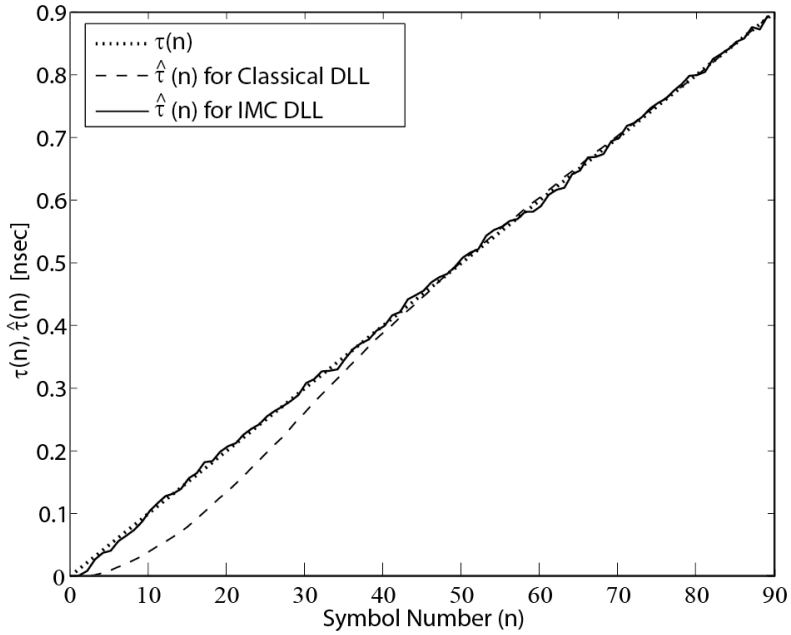


Figure 17. Réponse transitoire de DLL classique et de DLL avec IMC : $\tau(n) = 0.01nT_p$ et SNR=3 dB

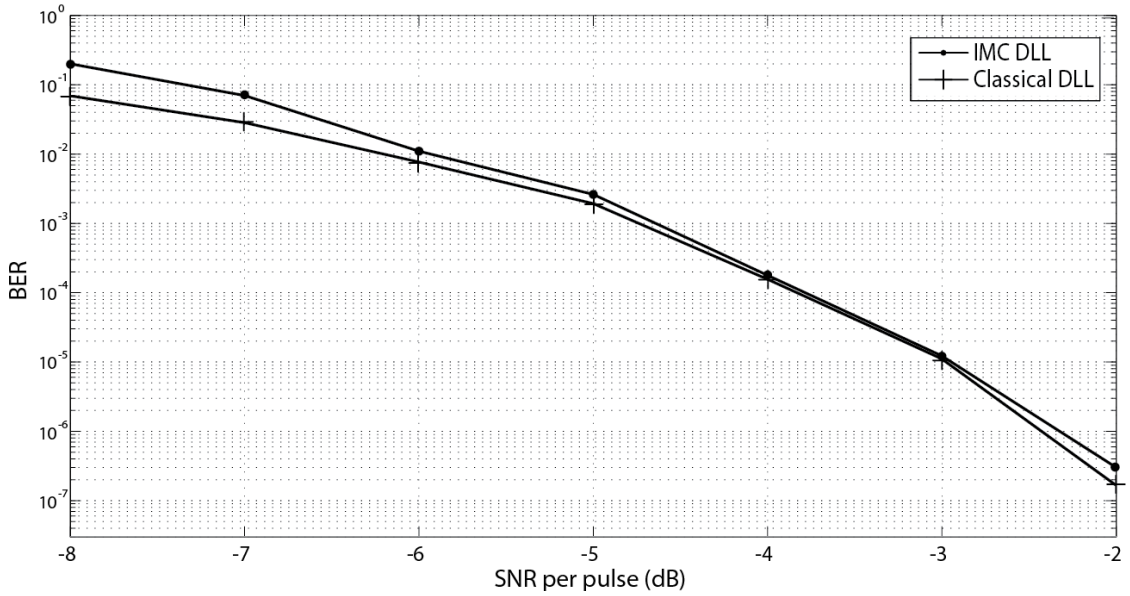


Figure 18. BER de DLL classique et de DLL avec IMC : $\tau(n) = 0.01nT_p$

Plan du manuscrit

La thèse est consacrée à l'étude des techniques de synchronisation temporelle (détection, acquisition et poursuite) dans des systèmes de transmission Radio Impulsionnelle Ultra Large Bande (IR-UWB). En effet, la richesse des multi-trajets résolvables du canal de transmission complexifie la synchronisation des récepteurs UWB. La technique utilisée durant tout le travail de thèse est l'emploi dans le récepteur d'une référence locale construite à partir du signal reçu. Cette technique est baptisée le « *Dirty Template* ». Les chapitres 2 à 5 sont dédiés respectivement à l'étude de la détection, de l'acquisition et de la poursuite avec boucle de type DLL. Les chapitres 6 et 7 traitent de la poursuite avec boucle DLL en utilisant une structure de boucle basée sur la technique « *Internal Model Control* ». Le manuscrit comporte sept chapitres ainsi que dernière partie dédiées à la conclusion et aux perspectives :

Le Chapitre 1 concerne l'introduction générale. Il présente le cadre du travail de thèse, à savoir, les systèmes IR-UWB ainsi que les problèmes inhérents à la synchronisation, en particulier pour les applications des réseaux de capteurs dynamiques. Il présente également un plan succinct du manuscrit.

Le Chapitre 2 présente un état de l'art de la technologie UWB et de la synchronisation temporelle. Ce chapitre est divisé en deux parties principales. La première présente la technologie UWB et son application sur les réseaux de capteurs dynamiques sans fils. Cette partie expose les avantages, les caractéristiques et les régulations de la technologie UWB. La deuxième partie introduit le problème de la synchronisation temporelle dans les systèmes UWB et propose une approche de résolution utilisant la temporisation sur référence bruitée, appelée « *Dirty Template* : DT ». Ceci permet d'obtenir une synchronisation temporelle rapide, précise et d'une complexité faible. L'approche proposée est basée sur la corrélation entre le signal reçu et un signal de référence (signal reçu avec un retard égal à la durée d'un seul bit d'information). Le système de synchronisation temporelle considéré est composé de trois blocs principaux: *détection du signal, acquisition temporelle et poursuite*. Chacun des trois blocs est détaillé dans un chapitre dédié.

Le Chapitre 3 met l'accent sur la première étape du système de synchronisation temporelle : *la détection du signal*. La première partie de ce chapitre introduit les concepts de la théorie de la détection. La détection du signal est d'abord réalisée en comparant les échantillons reçus avec un seuil (si l'échantillon dépasse ce seuil, le signal est présent, sinon il est absent). L'étude analytique montre que le choix de la valeur seuil affecte les performances de détection. L'application du théorème de Neyman-Pearson est suggérée afin de définir la valeur seuil appropriée. Dans la deuxième partie, les critères de détection sont appliqués au système de DT-UWB afin de mettre en place une structure pour une meilleure détection et de définir des valeurs seuils optimales. Les résultats sont analysés sous trois formes: un graphique des probabilités de fausse alarme et de détection par rapport au seuil, des histogrammes statistiques de test sous

chacune des hypothèses et un graphique de la courbe ROC « *Receiver Operating Characteristic* ». Les résultats des simulations montrent que les performances de détection de l'approche DT sont améliorées en augmentant le rapport signal-sur-bruit ou la taille de l'échantillon des données d'apprentissage « *Data-aided symbols*». Mais ces améliorations réduisent le rendement énergétique et compliquent la conception du récepteur.

Le Chapitre 4 expose la deuxième étape de la synchronisation : *l'acquisition temporelle*. Sur la base de la méthode de Cramér-Rao, la borne de Cramér-Rao (limite inférieure) pour le système DT-UWB est tout d'abord calculée. Cette borne est utilisée comme limite de performance fondamentale pour tout estimateur temporel. Ensuite, l'estimateur d'acquisition DT est développé avec et sans données d'apprentissage en utilisant la méthode du maximum de vraisemblance ou de la moyenne quadratique. La synchronisation DT proposée repose sur la recherche d'une crête dans la sortie de la corrélation glissante entre le signal reçu et le signal de référence bruité. Cependant, plusieurs points de maximum autour du pic à la sortie de la corrélation sont observés. Ceux-ci peuvent dégrader l'estimation de l'erreur de décalage temporel (TOE). Pour pallier cette difficulté, un filtre à fenêtre (*window filter*) adapté à la structure de la corrélation est ajouté, permettant d'obtenir un seul pic maximal dans l'intervalle d'estimation et améliorant l'estimation de l'erreur de décalage temporel. Ce filtre à fenêtre contribue également à réduire les effets indésirables du bruit sur l'estimation de synchronisation. L'analyse théorique et les résultats des simulations sous Matlab ont montré la fiabilité de l'estimateur proposé et confirment que l'estimateur du décalage temporel est plus précis et plus rapide à calculer avec des données d'apprentissage qu'en l'absence de ces dernières, mais au détriment de l'efficacité de la bande passante.

Le Chapitre 5 traite la dernière étape de la synchronisation : la *poursuite*. L'objectif de ce chapitre (ainsi que les chapitres 6 et 7) est de concevoir une unité de poursuite appropriée pour le système DT-UWB, qui contribue à maintenir une synchronisation satisfaisante tout au long de la période de transmission entre le récepteur et l'émetteur, et à atténuer les variations du décalage temporel dues au mouvement relatif entre émetteur et récepteur. La boucle à verrouillage de retard DLL est la technique fondamentale utilisée pour la poursuite dans les systèmes UWB. Le système DLL se compose de deux blocs principaux: un système en boucle ouverte (*early-late loop*) et un contrôle en boucle fermée. La première partie du chapitre 5 étudie les caractéristiques de la boucle ouverte de la DLL pour le système DT-UWB. Deux approches sont proposées pour la conception du contrôleur en boucle fermée. La première approche est basée sur des algorithmes classiques tels que la méthode itérative et les filtres de Wiener, tandis que la seconde approche est basée sur la technique de la commande par modèle interne (*Internal Model Control* : IMC), qui est une technique novatrice et prometteuse pour des applications de radio et de télécommunication. La deuxième partie du chapitre 5 utilise des algorithmes classiques pour trouver la structure de contrôle en boucle fermée. La première phase débute par la conception d'un contrôleur en boucle fermée de premier ordre basé sur un simple procédé itératif. Toutefois,

des études théoriques ont montré qu'en cas de présence de l'effet Doppler, la DLL du premier ordre n'est plus en mesure de suivre les variations temporelles du signal reçu. Pour régler ce problème, une DLL en boucle fermée du deuxième ordre est proposée. Les paramètres de la DLL proposée ont été sélectionnés en appliquant la théorie des filtres de Wiener afin d'améliorer le comportement transitoire ainsi que la capacité de traitement du bruit.

Les chapitres 6 et 7 emploient la technique IMC pour déterminer la structure de contrôle en boucle fermée. **Le Chapitre 6** décrit la conception du système DLL avec la technique IMC et analyse le comportement du système en l'absence d'effet Doppler. La technique IMC présentée est bien adaptée en termes de linéarité et non linéarité à l'application au système DLL. La technique IMC est composée de deux blocs (modèles): un modèle inverse (modèle de contrôle) connecté en série avec le système DLL en boucle ouverte, et un modèle direct (modèle du système) connecté en parallèle avec la DLL en boucle ouverte. Cette structure a pour objectif de surmonter les problématiques de perturbations et de déviations des paramètres du modèle. La méthode des moindres carrés (LS) est ensuite utilisée pour déterminer la valeur optimale des coefficients du modèle de contrôle et du modèle de système.

Le Chapitre 7 présente et analyse le comportement du système DLL avec la technique IMC en tenant compte de l'effet Doppler. Les résultats de simulations montrent que la performance transitoire pour la structure standard de l'IMC est décevante et que les performances de poursuite sont affectées par l'effet Doppler. Dans ce chapitre, une structure IMC-DLL est développée pour garantir une synchronisation satisfaisante, même en présence de l'effet Doppler. Ce chapitre est composé de trois parties principales: Dans la première, une approche multi-modèle (IMMC) est proposée comme solution potentielle au problème de l'effet Doppler. Selon une division du modèle de base en plusieurs sous-régions, le système de DLL au sein de chaque sous-région est représenté par un modèle local approprié. Une commutation entre ces modèles locaux se produit à chaque fois que le point de fonctionnement dépasse un certain seuil. La deuxième partie développe un filtre à moyenne glissante dans le système IMMC pour limiter la dégradation du système, due aux erreurs statiques (*residual steady-state error*) induites par l'effet Doppler. La troisième partie montre, par simulation sur Matlab, que la performance transitoire du système IMMC est améliorée en utilisant la technique de commande adaptative entre les modèles locaux, car la valeur du seuil diminue. Enfin, une comparaison des performances de la poursuite est présentée entre le système DLL utilisant la méthode classique et la DLL combinée avec la technique IMC. Cette comparaison montre que si la vitesse relative entre l'émetteur et le récepteur est constante, la poursuite du système DLL conventionnelle est meilleure. Toutefois, lorsque la vitesse relative accélère ou décélère, le système DLL IMC a une réponse transitoire rapide, capable de suivre efficacement la variation de l'entrée, tandis que le système DLL conventionnel a de grandes difficultés à suivre ces variations. La recherche d'une structure hybride entre les deux méthodes mérite d'être poursuivie et généralisée.

Enfin, **la Conclusion générale et Perspective** récapitule les travaux réalisés et les principaux résultats obtenus pour améliorer la technique de synchronisation temporelle dans les systèmes UWB et d'optimiser les performances des réseaux de capteurs sans fils. Cette méthodologie originale et d'un intérêt incontestable mérite d'être poursuivie et généralisée en prenant en compte d'autres facteurs tels que la complexité fonctionnelle et architecturale ainsi que les différents types d'implantation matérielle.

Liste des publications

Papiers de Journal

[J1] R. Alhakim, K. Raoof, E. Simeu and Y. Serrestou, "Cramer-Rao lower bounds and maximum likelihood timing synchronization for dirty template UWB communications," Signal Image and Video Processing Journal, Ed. Springer, pp. 1-17, October 13, 2011

[J2] R. Alhakim, K. Raoof and E. Simeu, "Design of tracking loop with dirty templates for UWB communication systems," Signal Image and Video Processing Journal, Ed. Springer, pp.1-17, June 27, 2012

[J3] R. Alhakim, K. Raoof and E. Simeu, "Detection of UWB signal using dirty template approach," Signal Image and Video Processing Journal, Ed. Springer, 2013 (accepted)

Chapitre de Livre

[B1] R. Alhakim, K. Raoof and E. Simeu, "Chapter 2: Timing synchronisation for IR-UWB communication systems," In: Ultra wideband - current status and future trends, edited by M. A. Matin, Published by InTech, ISBN: 9789535107811, pp. 15-40, October 03, 2012

Conférences Internationales

[C1] R. Alhakim, K. Raoof and E. Simeu, "A novel fine synchronization method for dirty template UWB timing acquisition," Wireless Communications Networking and Mobile Computing (WiCOM), 6th International Conference, Chengdu, China, pp. 1-4, September 23-25, 2010

[C2] R. Alhakim, K. Raoof, E. Simeu and Y. Serrestou, "Data-aided timing estimation in UWB communication systems using dirty templates," IEEE International Conference on Ultra Wideband (ICUWB), Bologna, Italy, pp. 435-439, September 14-16, 2011

[C3] R. Alhakim, E. Simeu and K. Raoof, "Internal model control for a self-tuning delay-locked loop in UWB communication systems," 17th IEEE International On-Line Testing Symposium (IOLTS), Athens, Greece, pp. 121-126, July 13-15, 2011

[C4] R. Alhakim, K. Raoof and E. Simeu, “Design of tracking loop for UWB systems,” International Conference on Information Processing and Wireless Systems (IP-WIS), Sousse, Tunisia, March 16-18, 2012

Conférences Nationales

[C5] R. Alhakim, E. Simeu and K. Raoof, “Internal model control for a self-tuning delay-locked loop in UWB communication systems,” Journées scientifiques du projet SEmba 2011, L'Épervière, Valence, France, October 20-21, 2011

[C6] R. Alhakim, E. Simeu and K. Raoof, “A novel design for delay-locked loop using internal model control approach,” Journées scientifiques du projet SEmba 2013, Domaine des Hautannes, Lyon, France, April 4-5, 2013

List of Figures

List of Tables

Acronyms

ADC	Analog-to-Digital Converter
AGWN	Additive White Gaussian Noise
ASIC	Application-Specific Integrated Circuit
BER	Bit Error Rate
BPSK	Bi-Phase Shift Keying (Antipodal Modulation)
CM	Channel Model
CRLBs	Cramer-Rao Lower Bounds
CS	Cyclo-Stationarity
DA	Data Aided
DLL	Delay Locked Loop
DS	Direct Sequence
DSP	Digital Signal Processor
DT	Dirty Template
ECC	European Commission Committee
EIMMC	Enhanced Internal Multi-Model Control
ELG	Early-Late Gate
ENR	Energy to Noise Ratio
EWMA	Exponentially Weighted Moving Average
FA	False Alarm
FCC	Federal Communications Commission
FI	Fisher Information

FPGA	Field Programmable Gate Array
GLRT	Generalized Likelihood Ratio Test
IFI	Inter-Frame Interference
IID	Independent and identically distributed
IMC	Internal Model Control
IMMC	Internal Multi-Model Control
IR-UWB	Impulse-Radio Ultra Wide-Band
ISI	Inter-Symbol Interference
K-S	Kolmogorov-Smirnov
LMS	Least Mean Squares
LOS	Line Of Sight
LRT	Likelihood Ratio Test
LS	Least-Squares
LWMA	Linearly Weighted Moving Average
MA	Moving Average
MIMO	Multi-Input Multi-Output
ML	Maximum Likelihood
MLE	Maximum Likelihood Estimator
MSE	Mean Square Error
MUI	Multiple-User Interference
MVU	Minimum Variance Unbiased
NDA	Non-Data Aided
NP	Neyman-Pearson

OOK	On-Off Keying
PAM	Pulse Amplitude Modulation
PDF	Probability Density Function
PPM	Pulse Position Modulation
PSD	Power Spectral Density
PSM	Pulse Shape Modulation
ROC	Receiver Operating Characteristic
SISO	Single-Input Single-Output
SNR	Signal to Noise Ratio
S-V	Saleh-Valenzuela
TDT	Timing with Dirty Template
TH	Time-Hopping
TOE	Timing Offset Error
TR	Transmitted Reference
UWB	Ultra Wide-Band
WSN	Wireless Sensor Network
WUSB	Wireless Universal Serial Bus

Chapter 1

Introduction

Since the Federal Communications Commission (FCC) in the USA released the First Report and Order in 2002 covering commercial use of **Ultra Wide-Band** (UWB) [1], the interest for ultra wideband technology is growing fast especially in the short-range indoor wireless communication [2]. Among various potential communication applications, one of the most promising is wireless sensor network (WSN) [3-5]. In dynamic WSN systems, many wireless terminals with sensors are placed or moved within a specific geographical region, and they are employed to collect environmental information such as temperature, movement, humidity, and so on. They transmit then the information to a base station which gathers in its turn all the sensed data and uses them for several applications. Dynamic WSN systems usually require: low cost node circuit with small form factor, low energy consumption, robust communications and high-precision ranging capabilities [6-8].

The basic concept of **Impulse-Radio** UWB (IR-UWB) technology is to transmit and receive baseband impulse waveform streams of very low power density and ultra-short duration pulses (typically at nanosecond scale). These properties of UWB give rise to fine time-domain resolution, rich multipath diversity, low power and low cost on-chip implementation facility, high secure and safety, enhanced penetration capability, high user capacity, and potential spectrum compatibility with existing narrowband systems [8-10]. Due to all these features, UWB technology has been considered as a feasible technology for WSN applications, including locating and imaging of objects [11], perimeter intrusion detection [12], video surveillance [13], in-vehicle sensing [14], etc.

While UWB has many reasons to make it a useful and exciting technology for wireless sensor networks and many other applications, it also has some challenges which must be overcome for it to become a popular approach, such as interference from other UWB users, accurate modelling of the UWB channel in various environments, wideband RF component (antennas, low noise amplifiers) designs, accurate synchronization, high sampling rate for digital implementations, and so on [15-19].

In this thesis, we will focus only on one of the most critical issues in ultra wideband systems: **Timing Synchronization**. Since UWB system has a dense multipath channel, and its pulses are so narrow (typically at nanosecond scale) with low power density [17]. It follows that the ability to maintain the satisfactory timing synchronization imposes major challenges in achieving UWB potential. Numerical tests show that a timing offset error higher than a tenth of the impulse width leads to a total loss of information [18, 19].

Timing synchronization in wireless communication systems typically depends on the sliding correlator between the received signal and a transmit-waveform template (Clean Template) [17]. In Impulse-Radio Ultra-Wideband devices however, this approach is not only sub-optimum in the presence of rich resolvable multipath channel, but also incurs high computational complexity and long synchronization time [17, 20]. We are thus motivated to look for a rapid and low-complexity approach for realizing satisfactory timing synchronization for IR-UWB systems. Several timing algorithms in the literature have been proposed for IR-UWB [21-26]. For example: bit reversal search approach [21], coded beacon sequence in conjunction with a bank of correlators [22], the inherent cyclo-stationarity (CS) approach [23], transmitted reference (TR) approach [25], etc. Each of these approaches requires one or more of the following assumptions, as: 1) the absence of multipath; 2) the absence of time-hopping (TH) codes; 3) the multipath channel is known; 4) high computational complexity and long synchronization time; and 5) degradation of bandwidth and power efficiencies. In this thesis, we propose an efficient synchronization approach for IR-UWB, called: **Timing with Dirty Templates** (TDT) [27-34]. This technique is based on correlating the received signal with "dirty" template extracted from the received waveforms. This template is called dirty; because it is distorted by the unknown channel and by the ambient noise. TDT allows the receiver to enhance energy capture even when the multipath channel and the Time-Hopping (TH) spreading codes are both unknown. Consequently, TDT approach contributes to enhance synchronization performance for IR-UWB and to reduce receiver structure complexity [33].

Timing Synchronization stage at the receiver usually consists of three units: signal detection, timing acquisition and tracking. **Signal Detection** is the first unit, used for deciding if the signal or noise is received [33-34]. **Timing Acquisition** unit is, a coarse synchronization, employed to find approximately a starting point of each received symbol and to reduce the timing offset error to within a fraction of UWB pulse duration [27-29]. The third step is a **Tracking.** Its objective is to maintain and lock the satisfactory synchronization throughout the transmission period, even if timing offset variation in the received waveform occurs as a result of oscillator drifts or relative transmitter-receiver motion (Doppler Effect) [30-33].

This dissertation is organized as follows: In **Chapter 2,** we present a state of the art of the UWB technology and its synchronization problem statement. The first part of this chapter

proposes an overview of wireless sensor networks and feasibility of UWB for it as a communication system. We discuss also the UWB characteristics and regulations, and describe its modulations and channel model. The second part introduces the timing synchronization challenge in UWB systems and proposes the timing with dirty template approach as an adequate solution for UWB systems, in order to achieve rapid, accurate and low-complexity timing synchronization.

In **Chapter 3,** we focus on the first stage of timing synchronization system: signal detection. First, we introduce the concepts of detection theory. Basically, the signal detection is achieved by comparing the received captured samples with a threshold, if the sample exceeds this threshold we say that the signal is present; otherwise, we consider that the signal is absent. We find further that selecting the threshold value affects detection performance. For setting the suitable threshold value, we suggest an approach termed the Neyman-Pearson theorem. Next, we apply the detection criteria on dirty template (DT) UWB system to establish a good detector structure and to define optimal threshold values. Finally, the simulation results illustrate the detection performance for different values of signal to noise ratio (SNR), training sequence length and threshold. The proposed detection algorithms for DT-UWB and the results presented in this chapter have led to publications [J3 and B1].

In **Chapter 4**, we move on to the second stage of synchronization: timing acquisition. First of all, it is important to derive the Cramer-Rao lower bound for DT-UWB system, which is used as a fundamental performance limit for any timing estimator. Next, we develop the TDT acquisition estimator with and without training symbols by using maximum likelihood and mean-square algorithms. We assert that TDT synchronization relies on searching a peak in the output of the sliding correlation between the received signal and its dirty template. However, we find multiple maxima points around the peak at the output of the correlator, which may degrade the estimation performance of timing offset error (TOE). To avoid this problem, we add a suitable window filter to the structure of the cross-correlator. This modified approach guarantees to obtain a single maximal peak inside the estimator range and improve the estimation error performance. This window filter also contributes to reduce the unwanted noise effects on the timing estimation. Both the theoretical analysis and Matlab simulation results show the performance of the proposed timing estimator, and confirm that the timing estimator with training symbols has high performance and fast execution compared to that without data-aided symbols, but at the expense of bandwidth efficiency. This work has led to publications [J1, B1, C1 and C2].

In **Chapter 5**, we propose the third unit of the synchronization system: Tracking. According to the literature, we propose the main typical technique used for tracking purpose in IR-UWB and spread-spectrum systems: Delay Locked Loop (DLL). This approach is considered as an approximation to the maximum-likelihood estimate of the timing offset error. So, the goal

which we hope to reach in this and following chapters is to design a suitable tracking unit for UWB systems, which contributes to alleviate the effects of timing offset variations in the received waveform, and to enhance the transmission performance (BER). The first part of this chapter deals deeply with DLL system structure without feed-back loop and combines it with DT-UWB. The objective of the second part of this chapter is to design the closed loop controller using conventional methods: iterative method and Winner filter. So the second part starts with deriving an appropriate first-order closed-loop controller based on a simple iterative method. Then, the essential parameters of the proposed DLL control are set and the tracking performance is analyzed in terms of output noise variance (Jitter) and transient behavior. We see later that in the presence of dynamic Doppler Effect, the first-order DLL is no longer able to track the timing variations in received signal. For efficiently tracking the input UWB waveform in the presence of ambient noise and Doppler Effect, we suggest DLL with second-order closed-loop controller. The parameters of suggested DLL are selected by applying Wiener-filter theories, in order to improve transient behavior as well as noise handling ability. The proposed tracking algorithms for DT-UWB and the simulation results presented in this chapter have led to publications [J2, B1 and C4].

In **Chapter 6**, we decide to find another approach for realizing the closed-loop control structure by employing supervisory control concepts, instead of applying the conventional methods. The problem we encounter with the classical approach is that the DLL system is non-linear and it is so difficult to design an appropriate closed-loop filter without taking into account stability and robustness issues. To avoid the complexity of such a study, we present an original DLL structure used to achieve satisfactory and accurate tracking. The proposed DLL scheme is based on **Internal Model Control** (IMC). This technique is a well-known and widely used in various automatic control areas, offering fast respond to input signals with high performance in robustness and stability of the system. IMC is considered as a novel control approach in the communication systems. In this chapter, we give an overview of Internal Model Control concept, and combine it with DLL tracking system. We see that IMC controller is composed of two blocks (models): an inverse model (control model) connected in series with the DLL system and a forward model (system model) connected in parallel with the DLL system. We prove that this structure has a good performance of overcoming disturbance and deviations of model parameters. Next, we apply Least-Squares (LS) estimation algorithm methods in order to determine the optimal coefficients for the system and control models. This chapter focuses on designing IMC-DLL system and analyzing the tracking behavior in the absence of Doppler problems, which will be studied in the next chapter. The original work of this chapter and its simulation results are led to publication [C3, C5 and C6].

In **Chapter 7,** we study the behavior of IMC tracking system taking into account Doppler Effect. The simulation results show that the transient performance for the proposed IMC structure is disappointing, and the tracking extremely suffers from Doppler Effect. Hence, in

this chapter we develop the IMC-DLL structure to be able to keep the satisfactory synchronization even in the presence of Doppler Effect. To solve Doppler problem, the multi-model concepts are applied; where the tracking region is divided into several sub-regions, the DLL system within each sub-region is represented by a private local-model. Switching between these local-models occurs at each time when the operating point exceeds certain threshold. Further, we notice that the transient performance of the modified IMC system is improved as the threshold value decreases. Therefore, instead of switching between sub-models with dramatic jumps, the model switches could smoothly be executed by using adaptive control technique. At the end of the chapter, we compare the tracking performance between the classical DLL, which has been proposed in Chapter 5, and the DLL combined with IMC technique.

Finally, in the **Conclusion and Perspectives** section, we summarize this thesis and give some concluding remarks as well as suggestions for future research directions.

List of publications

Journal Papers

[J1] R. Alhakim, K. Raoof, E. Simeu and Y. Serrestou, "Cramer-Rao lower bounds and maximum likelihood timing synchronization for dirty template UWB communications," Signal Image and Video Processing Journal, Ed. Springer, pp. 1-17, October 13, 2011

[J2] R. Alhakim, K. Raoof and E. Simeu, "Design of tracking loop with dirty templates for UWB communication systems," Signal Image and Video Processing Journal, Ed. Springer, pp.1-17, June 27, 2012

[J3] R. Alhakim, K. Raoof and E. Simeu, "Detection of UWB signal using dirty template approach," Signal Image and Video Processing Journal, Ed. Springer, 2013 (accepted)

Book Chapter

[B1] R. Alhakim, K. Raoof and E. Simeu, "Chapter 2: Timing synchronisation for IR-UWB communication systems," In: Ultra wideband - current status and future trends, edited by M. A. Matin, Published by InTech, ISBN: 9789535107811, pp. 15-40, October 03, 2012

International Conferences

[C1] R. Alhakim, K. Raoof and E. Simeu, "A novel fine synchronization method for dirty template UWB timing acquisition," Wireless Communications Networking and Mobile Computing (WiCOM), 6th International Conference, Chengdu, China, pp. 1-4, September 23-25, 2010

[C2] R. Alhakim, K. Raoof, E. Simeu and Y. Serrestou, "Data-aided timing estimation in UWB communication systems using dirty templates," IEEE International Conference on Ultra Wideband (ICUWB), Bologna, Italy, pp. 435-439, September 14-16, 2011

[C3] R. Alhakim, E. Simeu and K. Raoof, "Internal model control for a self-tuning delay-locked loop in UWB communication systems," 17th IEEE International On-Line Testing Symposium (IOLTS), Athens, Greece, pp. 121-126, July 13-15, 2011

[C4] R. Alhakim, K. Raoof and E. Simeu, "Design of tracking loop for UWB systems," International Conference on Information Processing and Wireless Systems (IP-WIS), Sousse, Tunisia, March 16-18, 2012

National Conferences

[C5] R. Alhakim, E. Simeu and K. Raoof, "Internal model control for a self-tuning delay-locked loop in UWB communication systems," Journées scientifiques du projet SEmba 2011, L'Épervière, Valence, France, October 20-21, 2011

[C6] R. Alhakim, E. Simeu and K. Raoof, "A novel design for delay-locked loop using internal model control approach," Journées scientifiques du projet SEmba 2013, Domaine des Hautannes, Lyon, France, April 4-5, 2013

Chapter 2

UWB in WSNs and Synchronization Problem Statement

Ultra Wide-Band (UWB) is a fast emerging technology that has attracted considerable interest for the wireless communication applications in short-range such as wireless sensor networks (WSNs), due to its very good time-domain resolution, rich multipath diversity, low power and low cost on-chip implementation facility, high secure and safety, potential spectrum compatibility with existing narrowband systems, etc [9]. However, one of the most critical challenges in enabling the unique benefits of UWB transmissions is timing synchronization [17].

This chapter presents a general introduction of UWB communication technology. It describes the characteristics and benefits that make UWB system as extremely attractive for WSNs applications. Then, we present a state of the art of the UWB technology, including the following points adopted in this thesis: UWB regulatory, signal waveform, data modulation and multiple access techniques, a statistical model of the UWB multipath channel. Finally, after having presented a general overview of UWB systems, this chapter introduces the timing synchronization problem statement and proposes the timing with dirty template approach, considered in our study, as an adequate solution for UWB systems, in order to achieve rapid, accurate and low-complexity timing synchronization.

2.1 Overview of WSNs

Recently, wireless sensor network (WSN) has been considered as one of the most important technologies for the 21st century [35]. It consists of a possibly large number of wireless devices, called nodes, which are spatially distributed in order to collect environmental

measurements. Typical examples include temperature, humidity, presence (absence), sound, intensity, vibration, pressure, motion, pollutants, and so on. Then, the smart nodes transmit these environmental readings over a wireless channel to a base station that makes decisions based on all the sensed data and uses them for several applications [36].

2.1.1 Applications

Several potential applications have been developed based on wireless sensor networks [37, 38]. The following are a few examples:

- Environmental Monitoring: Sensor network can be used for animal tracking, flood detection, forest surveillance, weather forecasting, etc. One important example is to monitor the levels of pollution in the air. Where sensor networks have been deployed in several cities, such as London and Stockholm. The purpose is to monitor the concentration of dangerous gazes for citizens. Another example is to control of dangerous goods transported by trains. For example, SNCF which is the French national railway authority uses WSN systems for detecting critical events, such as gas outflow, gate-opening, temperature & pressure changes, etc. In emergency case, the WSN could remotely alert driver to the good and wagon situation [39].

- Military Applications: WSNs can be used for friendly or enemy force surveillance, equipment and ammunition monitoring, targeting, and chemical, biology and nuclear attack detection.

- Health Monitoring: The patients in the hospital can be equipped with wireless sensor nodes that monitor their vital signs and track their location. The wireless sensor networks allow patients to move freely in the hospital and keep them under constant supervision. In emergency case, the WSN could alert doctors to the patient's location and his condition. Another example is to constantly monitor blood glucose level in diabetic patient.

- Industrial Sensing: For instance, sensor nodes can be embedded into industrial machine to monitor machine health and to ensure its safe operation without any catastrophic failure. Another industrial example is to monitor corrosion levels of pipelines and tanks in the oil and gas industry. WSN provides reliable and economic method to efficiently monitor corrosion and identify the situation before any harm is done.

Hence, the basic goals of a WSN are so variety as follows: determine the value of physical variables, detect the occurrence of events of interest, estimate parameters of the detected events, classify a detected object, and track or localize an object [37]. Besides, the sensor nodes can be static or dynamic (mobile) based on the specific application.

2.1.2 Components

The usual structure of each sensor node includes an embedded processor, internal and external memories, a power source, one or more sensors, and a radio transceiver [38]. Figure 2.1 illustrates the typical architecture of the sensor node.

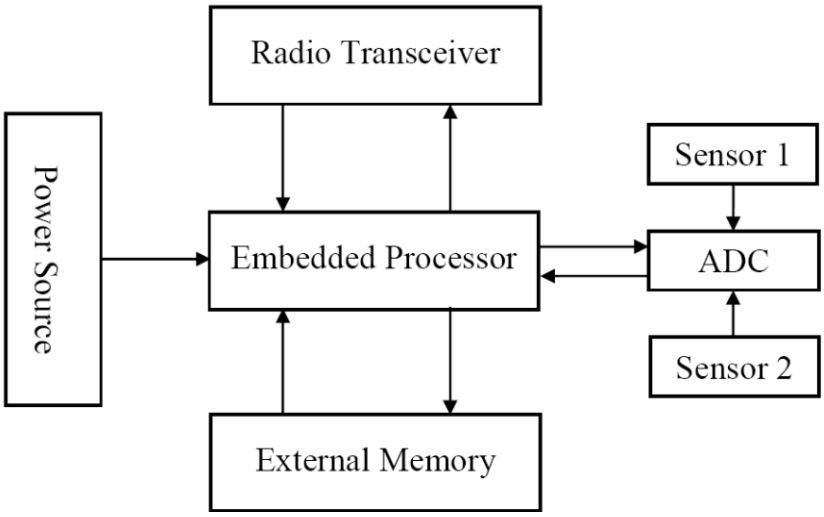


Figure 2.1 Sensor node architecture

- Embedded processor: It performs different tasks, such as processes collected data and controls the functionally of other hardware components. In a sensor node, the processor might be Microcontroller, Digital Signal Processor (DSP), Field Programmable Gate Array (FPGA) or Application-Specific Integrated Circuit (ASIC).

- Memory: The most relevant kinds of memory in sensor node applications are external Flash memory or in-chip memory of a microcontroller. The memory is classified into two categories according to its use: date and program memories. The purpose of the first memory is to store the environmental information, while the program memory contains the identification data of the device and is used also for programming the device.

- Power source: The power in a node sensor is consumed by sensing, data processing and radio communication. Power can be stored in batteries or capacitors. Due to the battery limit of each node, it is always needed to reduce the energy consumption. Current sensors are able to harvest energy from ambient supply, such as solar sources, vibration energy and temperature differences.

- Sensors: They are used, for instance, to produce a measurable response signal corresponding to a change in a physical condition, as temperature, humidity and pressure. The

produced continual analog signal is then digitized by an analog-to-digital converter (ADC) and sent to the embedded processor for processing.

- Transceiver: It provides communication links between the remote nodes and the base station. The node can be connected with the base station either directly by only single-hop, or indirectly via the other nodes of the network (multi-hop). In the case of multi-hop connection, the sensor nodes may also act as routers and provide guarantee to transfer data to the base station. The operational states of a transceiver usually includes transmit, receive, idle and sleep. There are several essential requirements for transceivers that must be fulfilled in the implementation of WSNs, such as low cost, small form factor, low energy consumption, robustness against interference and fading and high reliability of data communication, variable data rate (high, medium, low data rate) according to the application request, and high-precision ranging capabilities [6-8, 40-41]. For data communication and geo-location tasks, the WSNs rely upon one of the following three wireless communication technologies: ZigBee, WiFi and UWB. Table 2.1 provides a comparison among these three technologies. Obviously, ZigBee and UWB are intended for short-distance WSN applications, while WiFi is oriented to the wireless applications with a longer communication distance. WiFi can also provide high data rates, but it requires much higher circuit complexity and much higher energy consumption than that of ZigBee and UWB. UWB is better than the rest for geo-location and ranging applications. This dissertation focuses only on UWB technology as a potential wireless transceiver method for dynamic WSN applications.

Table 2.1 Comparison of wireless transceiver technologies

	ZigBee	**WiFi**	**UWB**
Data rate	Low, 250 kbps	High, 11 to 100 Mbps	Low – Medium – High up to 480 Mbps
Transmission distance	Short, < 30 meters	Long, up to 100 meters	Short, < 30 meters
Location accuracy	Low, several meters	Low, several meters	High, < 50 cm
Power consumption	Low, 20 to 40 mW	High, 0.5 to 1 W	Low, 30 mW
Multipath performance	Poor	Poor	Good
Interference to other systems	High	High	Low
Complexity and cost	Low	High	Low – Medium - High

2.2 UWB principles and characteristics

Ultra wide band (UWB) is a promising technology for wireless communication and localization applications, and it is attracted wide attention from academic sector as well as from commercial sector. According to the FCC, Ultra Wide Band is refers to either a signal occupying at least 500 MHz spectrum or a signal having a fractional bandwidth of at least 0.25 (by comparison, a narrowband signal typically has a fractional bandwidth less than 0.01), where the fractional bandwidth B_f is defined as [1]:

$$B_f = 2\frac{f_H - f_L}{f_H + f_L} \tag{2.1}$$

where f_H and f_L are respectively the highest and lowest frequencies of the signal at the -10 dB point in the spectrum, as shown in Figure 2.2. The nature of the short-duration pulses used in UWB technology offers unique benefits that are attractive for communications and radar applications. The main advantages of UWB systems can be summarized as follows [2]:

- Provide high data rates
- Have high resistance to multipath effects and interference
- Have potential small size, low cost
- Have high-precision ranging and localization at the centimeter level

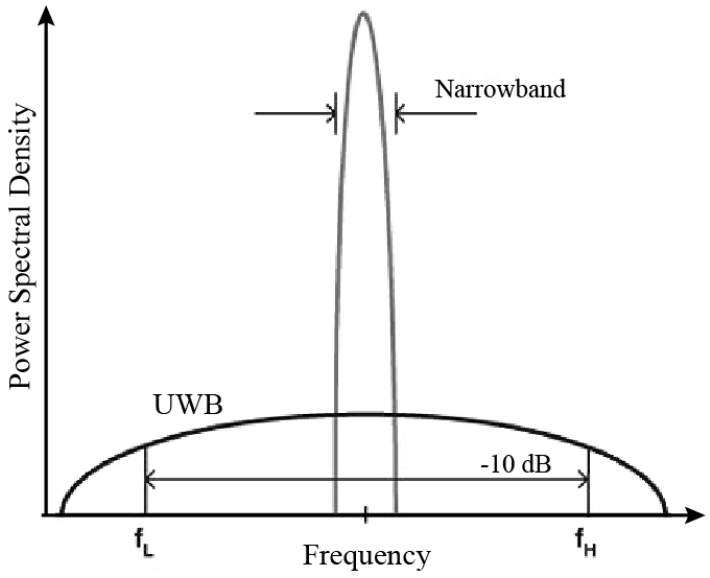


Figure 2.2 Comparison between narrowband and ultra wideband signals

Since the bandwidth of UWB signals is much wider than that of conventional communication systems, a high channel capacity, or high data rate, can be achieved, where

channel capacity is defined as the maximum amount of data that can be transmitted per second over a communications channel, and expressed by Shannon's theorem [42], as follows:

$$C = W \log_2 \left(1 + \frac{S}{N}\right) \tag{2.2}$$

where C is the maximum channel capacity in b/s, W denotes the signal bandwidth in Hz, and S/N is the signal-to-noise ratio (SNR). This equation shows that the channel capacity can be improved by increasing either the signal bandwidth or SNR. However, the capacity grows linearly with the bandwidth, but only logarithmically with the signal power. Therefore, from Shannon's capacity equation, we can see that the UWB systems can provide bit rate much higher than narrow bandwidth systems even in a low SNR environments.

The Shannon's formula also indicates that due to its large bandwidth, UWB signal is capable to convey information with very low energy density. Since the power density of UWB signals is so low and under the noise floor of a typical narrowband receiver. UWB signals are able to share the frequency spectrum and coexist with existing narrowband radio services with minimal or no interference. Besides, this low power density provides high secure and low probability to detection by other communication systems; this feature is a significant benefit for both military and commercial applications.

Another important advantage of using UWB is to provide a high resistance to multipath effects. In wireless telecommunications, the transmitted signal reaches the receiving antenna via different paths of multiple channel as the consequence of reflection, refraction, diffraction, and scattering occurs [43]. In conventional narrowband systems, this multipath phenomenon may cause the significant distortion of the signal received (such as fading effect), leading to degrade the transmission quality. On the other hand, the very short duration of UWB pulses makes them less sensitive to the multipath effect. Because the multipath components of the signal with their extremely time duration have so low opportunity to collide with each other and to cause the degradation of signal received [44]. Thus the fine temporal resolution of UWB signals results in low fading margins, implying robustness against multipath. The fine time resolution ability of UWB systems can be used in several applications: such as radar, accurate positioning or ranging applications and propagation channel estimation.

Another advantage is that UWB transceiver architecture is simpler and cheaper to build than that of typical narrowband transceiver. Unlike conventional radio system, the UWB transmitter generates very-short pulses, which are able to directly propagate through the channel without the need for additional radio-frequency mixer or local oscillators to translate the carrier frequency to the required frequency band, for that UWB transmission is termed as: carrierless system. Besides, the transmission of low-powered pulses eliminates the need for a power amplifier in UWB transmitter [45]. This simplicity makes the implementation of UWB

transceivers on-chip possible, including low production cost, small form factor, and low power consumption.

In addition, since the wide frequency spectrum of UWB contains the low frequency components with long wavelength. These components allow UWB signals to penetrate effectively through different materials. This property makes UWB technology viable for ground-penetrating radars and through-the-wall communications [46].

Due to all these advantages, UWB is well suitable to a lot of applications in wireless communications, networking, radar imaging, and localization systems. UWB wireless communication applications can be divided into: high data-rate and low data-rate applications.

- The high data-rate systems have been defined by the IEEE as applications needed to transmit at data-rates greater than 110 Mbps and at a distance of 10 m. An example of this category is high-speed wireless universal serial bus (WUSB), which can be use for providing wireless link between personal computers and several peripherals (such as printers, scanners, TV and other consumer electronic), enabling the devices to share high-quality real-time video, audio and file transmission [47]. In late 2001, the IEEE established the 802.15.3a to define a new physical layer concept for short-range high data-rate UWB applications [48], involving imaging and multimedia.

- On the other hand, under the low data-rate operation mode, UWB technology is potentially suitable to sensor, positioning, and identification networks. So one of the most promising is wireless sensor network (WSN) [3-5] whose all specific requirements, mentioned in the section 2.1, can be satisfied by UWB system in low-rate operation mode. Numerous sensor network applications proposed in the literature have employed UWB technology to provide communications and high-precision ranging and location capability; such as locating of objects [11], perimeter intrusion detection [12], video surveillance [13], in-vehicle sensing [14], etc. In March 2004, the IEEE established the 802.15.4a to define a new physical layer for low data-rate (from 50 Kbps to 1 Mbps) wireless communication applications [49], involving low-rate wireless personal area networks, sensors, and small networks.

While UWB has many reasons to make it an exciting and useful technology for future wireless communications and many other applications, it also has some challenges which must be overcome for it to become a popular technology. Some of the challenges are: interference from other UWB users, accurate modelling of the UWB channel in various environments, wideband RF component (antennas, low noise amplifiers) designs, accurate synchronization, high sampling rate for digital implementations, and so on [15-19]. The most critical issue in ultra wideband systems, considered in the thesis work, is accurate timing synchronization. That is because, UWB system has a dense multipath channel, and its pulses are so narrow (typically at nanosecond scale) with low power density under the noise floor [17]. The ability of the receiver to maintain the satisfactory synchronization between the received signal and

the generated reference signal imposes major challenges to UWB systems in achieving their potential bit error rate (BER) performance, capacity, and throughput. Numerical tests show that a delay higher than a tenth of the impulse width leads to a total loss of information [18, 19].

2.3 FCC regulations

On February 14, 2002 the Federal Communications Commission (FCC) authorized in the USA the unlicensed use of UWB in the frequency range from 3.1 to 10.6 GHz [1]. However, as depicted in Figure 2.3, the UWB spectrum may overlay several existing spectrums which are allocated for other narrowband applications (UMTS, GSM, WLAN, WiMAX, fixed satellite services…). Thus, we have two problems: First, UWB systems must be able to effectively operate even in the presence of strong interferers with the existing narrowband devices. Second, reciprocally, UWB must not provide noticeable interference to the users of these narrowband services. In order to optimize spectrum use and reduce this interference, the FCC imposed very restrictive spectral mask that must be respected by the UWB devices.

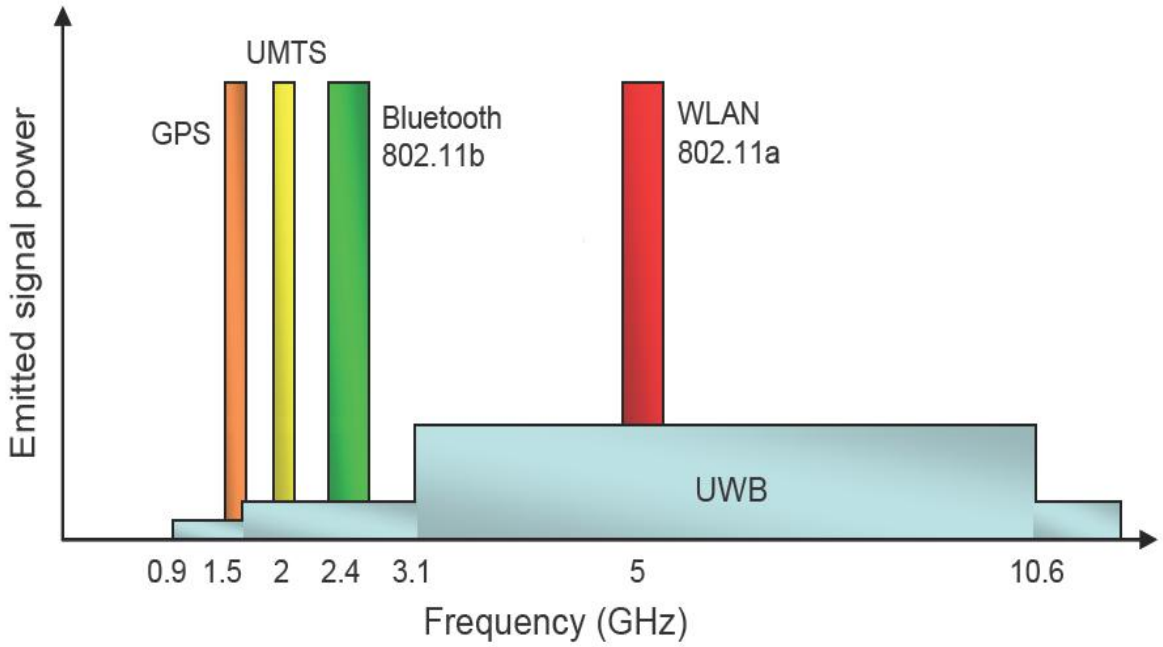


Figure 2.3 Spectrum of UWB and existing narrowband systems

According to the FCC spectral mask, illustrated in Figure 2.4, the power spectral density (PSD) of an UWB signal measured in 1 MHz bandwidth must not exceed -41.3 dBm. We also notice that the PSD limit is even much lower for particularly sensitive bands, such as the GPS

band (0.96–1.61 GHz). Thus, respecting FCC mask contributes on minimize the UWB interference on the existing narrowband devices. In addition, since the UWB signal has so wide frequency range from 3.1 to 10.6 GHz, it follows that the narrowband interferes occur only within the so small frequency range of UWB spectrum and do not significantly impact the UWB operation. The FCC has classified UWB operation into three separate categories: communications, imaging, and vehicular radar, each category has been allocated to a specific spectral mask [1]. For example Figure 2.4 depicts the specific spectral mask for indoor and outdoor communication applications. Similar to the FCC spectral mask in the USA, In Europe, the European Commission Committee (ECC) made a final decision in February 2007, authorizing unlicensed UWB transmission with PSD of -41.3 dBm/MHz between 6 and 8.5 GHz, Both the FCC and ECC spectral masks are depicted in Figure 2.4.

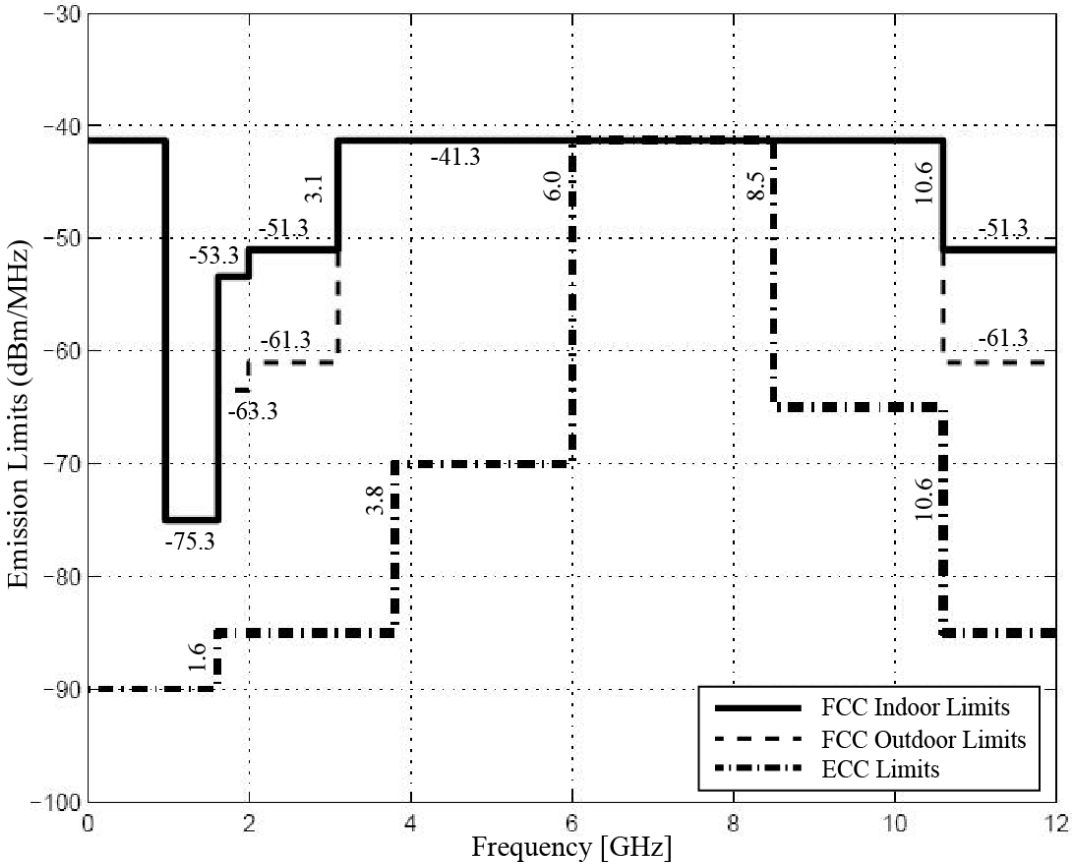


Figure 2.4 FCC spectrum mask for indoor and outdoor communication applications

2.4 UWB impulse shape

There are two common forms of UWB systems. The first approach, called *Impulse Radio*, is based on generating very short duration pulses (on the order of hundreds of picoseconds).

The energy of each pulse spreads over a very wide spectrum, respecting the FCC spectral mask requirements. The impulse radio approach does not require the use of additional carrier modulation and considers a single frequency band for UWB signals. On the contrary, the second approach, termed Multi-Band approach, which divides the available UWB frequency spectrum into multiple smaller and non-overlapping sub-bands, typically the bandwidth of each sub-band must be greater than 500MHz to obey the FCC's definition of UWB signals [16].

This thesis focuses only on impulse radio (IR) UWB approach, which is widely considered as a perfect candidate for wireless communication applications which require precise position location, ranging and imaging through materials. However, the most critical challenge in realizing the full potential of impulse radio communications is timing synchronization [16]. Several waveforms have been proposed in the literature for IR-UWB systems, such as Gaussian [50], Rayleigh, Laplacian, cubic [51], and modified Hermitian monocycles [52]. In all these waveforms the goal is to obtain a nearly flat frequency domain spectrum of the transmitted signal over the bandwidth of the pulse and to avoid a DC component. For instance, because of the DC energy in the transmitted pulse will be lost when the signal passes through the antennas which consider as a lower cut-off frequency filter [19, 53]. Many researches typically consider a derivative of the Gaussian pulse as the theoretical pulse shape for UWB communication systems. In this thesis, we focus only on the first and second derivatives of the Gaussian pulse [15]. The first derivative of a Gaussian pulse (*Gaussian monocycle*) has the form:

$$p_1(t) = k_1 \frac{t}{\tau_1^2} e^{-(t/\tau_1)^2} \tag{2.3}$$

where k_1 is a constant and τ_1 is the time-scaling factor. While the second derivative of a Gaussian pulse (*Gaussian doublet*) is defined by:

$$p_2(t) = k_2 \left[1 - 4\pi \left(\frac{t}{\tau_2}\right)^2\right] e^{-2\pi(t/\tau_2)^2} \tag{2.4}$$

where k_2 is a constant and τ_2 is the time-scaling factor. Figures 2.5 and 2.6 display the first and second derivatives. The pulses here are all plotted for unit energy (by adjusting k_1 and k_2) and represented in time and frequency domains. The duration of generated pulse is selected by adjusting the parameters τ_1 (or τ_2). In these figures, τ_1 and τ_2 are assumed to be 1 and 0.4 ns respectively. We can see from the figures that the pulses spread its power over a wide range of frequencies, and it has virtually no DC. Thus, these pulses are suitable for UWB communications.

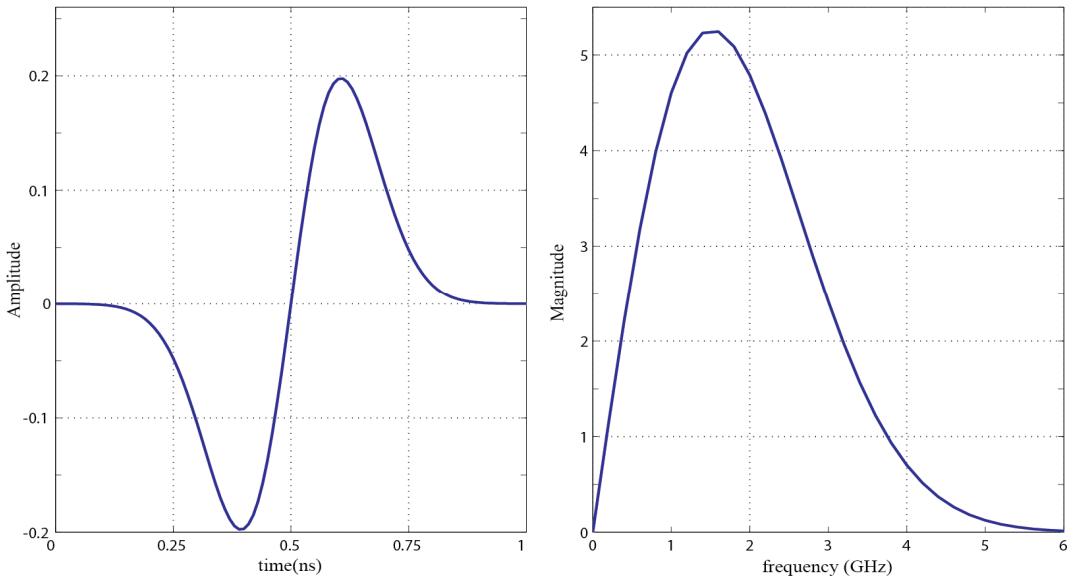


Figure 2.5 A Gaussian monocycle with unit energy in time and frequency domains

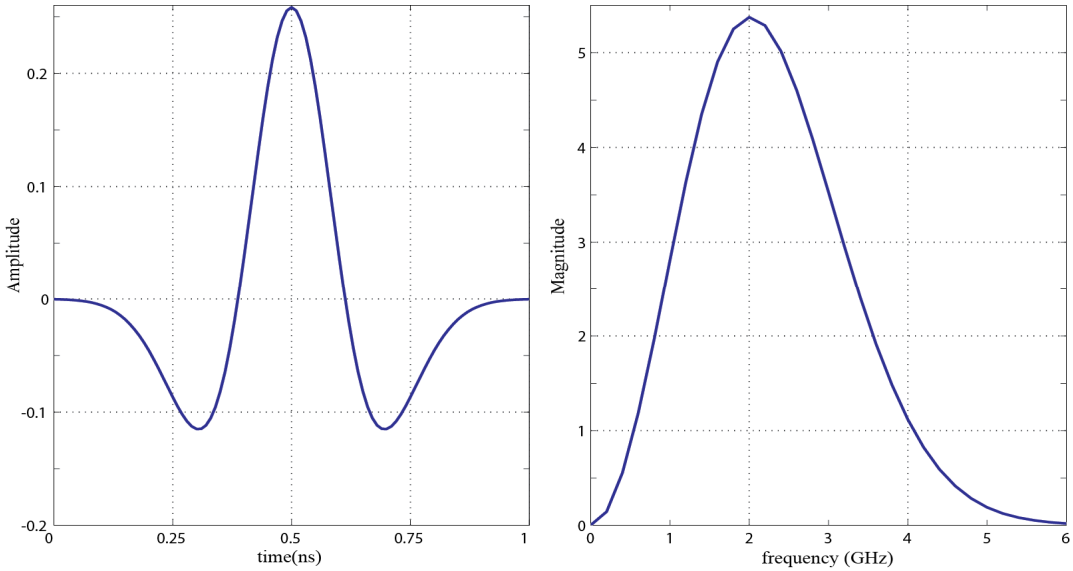


Figure 2.6 A Gaussian doublet with unit energy in time and frequency domains

2.5 Data modulation techniques

In order to transmit information, additional processing is needed to modulate the digital information onto UWB monocycle pulse sequence. A number of modulation techniques may be used with the UWB impulse radio systems, such as Pulse Amplitude Modulation (PAM), Pulse Position Modulation (PPM), On-Off Keying (OOK), Antipodal Modulation (BPSK), or Pulse Shape modulation (PSM). In these techniques, the UWB monocycle which is modulated

by an "1" is differentiated from the pulse modulated by a "0" either by the amplitude of the signal, or by the position (arrival time) of signal or also by its shape [54]. Of these possible modulation techniques, two are predominately considered in research and industry, taking into account the effect of data modulation on power efficiency, interference robustness and FCC mask respect. These techniques are PPM and BPSK [15]. Here, we focus only on explaining the Antipodal Modulation (Bi-Phase Shift Keying), which is strictly a binary scheme and consists of changing the polarity of the transmitted pulses according to the incoming data. For instance, a binary "1" is represented by a positive pulse and a binary "0" is represented by a negative pulse, as can be shown in the Figure 2.7. The BPSK modulation can be expressed as:

$$p_T(t) = d_i \cdot p(t), \quad d_i = \begin{cases} -1, & i = 0 \\ +1, & i = 1 \end{cases} \tag{2.5}$$

where $p(t)$ is the UWB pulse waveform, i is the transmitted bit "0" or "1". Note: based on (2.5) we can say that BPSK modulation could be consider as a particular case of PAM modulation, whereas a binary "0" is encoded into a pulse with small amplitude $d_0 p(t)$ and binary "1" is encoded into a pulse with large amplitude $d_1 p(t)$, considering that $d_0 = -1$ and $d_1 = +1$. Therefore, in this thesis PAM will later be employed as data modulation technique, in effect, the BPSK is which will be applied.

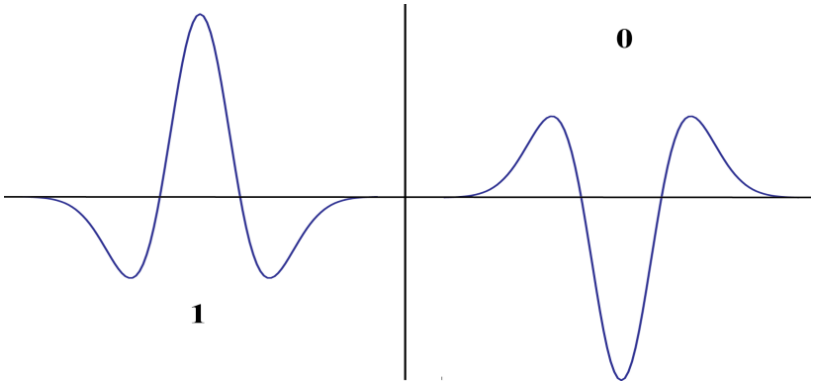


Figure 2.7 Antipodal modulation (BPSK)

2.6 Multiple access schemes

Multiple access technology is additional processing, needed in order to enable different users to transmit their digital information in the same propagation channel with high interference robustness. There are two spread-spectrums multiple access approach that have typically been considered to be used with IR-UWB signals: direct sequence (DS-UWB) and

time hopping (TH-UWB) [19]. The TH-UWB will be considered in this thesis; it is given by the following expression and plotted in Figure 2.8:

$$p_T(t) = \sum_{j=0}^{N_f-1} p(t - jT_f - c_jT_c) \tag{2.6}$$

where $p(t)$ is the UWB pulse waveform, T_f is the frame repetition time, N_f is the number of pulses (or frames) used to represent one data symbol, j is the frame index inside the symbol period, c_j is a pseudo-random code different for each user and T_c is a chip duration. The TH technique can be combined with PAM modulation, as follows:

$$u(t) = d_i \cdot p_T(t) = \sum_{j=0}^{N_f-1} d_i \cdot p(t - jT_f - c_jT_c), \quad d_i = \begin{cases} -1, & i = 0 \\ +1, & i = 1 \end{cases} \tag{2.7}$$

Figure 2.8 illustrates that the total symbol duration T_s is divided into N_f frames of duration T_f, thus $T_s = N_fT_f$. Each frame contains only one pulse $p(t)$ and is divided into N_c chips (slots) of duration T_c, thus $T_f = N_cT_c$. The position of the pulse $p(t)$ inside the frame is determined by the pseudo-random TH code sequence c_j which, in this example, takes the following value: $\{c_j\} \in [0, N_c - 1]$.

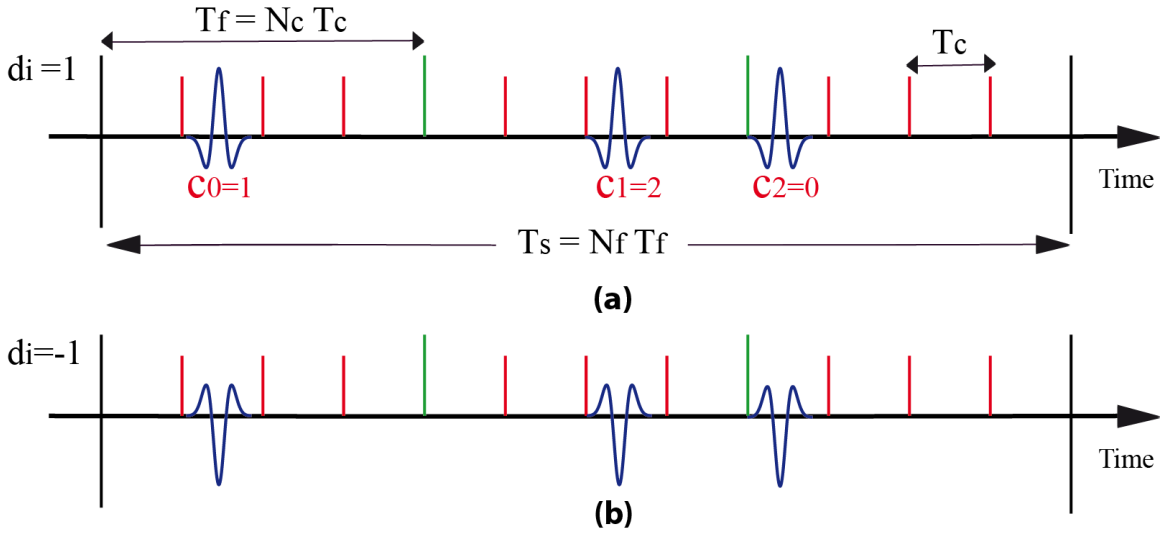


Figure 2.8 TH-UWB signal with PAM modulation ($N_f = 3, N_c = 4$, TH codes=[1, 2,0])
(a) for data bit "1", (b) for data bit "0"

As previously stated, TH technology actually is useful for minimizing collisions between users in multiple access systems, where each user is assigned a unique pseudo-random code. To detect the TH signal at the receiver, it should be correlated with the correct pseudo-random code which is already known at receiver unit. Additionally, this approach provides data

transmission more secure in a hostile environment, because the detection of signal will not be correctly achieved if the receiver does not know the user's TH code.

Besides, it has been conformed in the literature that a regular pulse-to-pulse interval signal train introduces spectral peaks in the pulse's spectrum, these energy spikes can cause interference with other narrowband RF systems (exceeding FCC mask) and limit the amount of useful energy transmitted (less energy-efficient). TH technology makes a random interval (random offset) between the pulses, contributing in reducing the periodic nature of the pulse train and in smoothing the spectrum of the UWB signal (which becomes much more noise-like) [15, 19].

Furthermore, since time hopping method is considered as a form of spread-spectrum; where a large number of pulses are being used to represent digital information bit, the transmission quality and processing gain (the received energy capture) are improved at expense of data transfer rate and energy consumption. Note, the operation which represents one data bit by several number of UWB pulses is called repetition coder. The goal of adding the redundant pulses is to minimize the effect of channel noise and to make transmission more reliable. Thus increasing the pulse number per bit N_f, mentioned in (2.6), leads to improve the processing gain, but it provides a lower bit-rate. Consequently, for applications that require high bit-rate N_f should be relatively small. While for applications, such as WSNs, that require reliable transmission more than the channel bandwidth efficiency and bit-rate, N_f should be big enough.

2.7 UWB indoor channel model

Analysis and design of UWB communication systems require an accurate channel model. In recent years, a lot of research effort has been devoted in order to construct UWB channel model, such as [55-60]. The most common statistical channel model for wideband indoor applications is the Saleh-Valenzuela (S-V) model [61]. The basic assumption behind S-V model is that multi-path components arrive in clusters, and each cluster consists also of ray arrivals. Thus, multipath arrivals are grouped into two different categories: a cluster arrival and a ray arrival within a cluster, as illustrated in Figure 2.9. The channel impulse response of the S-V model is given by [62, 63]:

$$h(t) = \sum_{c=0}^{C-1} \sum_{l=0}^{L-1} \alpha_{c,l} \delta(t - \tau_c - \tau_{c,l}) \tag{2.8}$$

where C is the total number of clusters and L is the total number of paths (rays) within each cluster. $\alpha_{c,l}$ denotes the gain of the l^{th} path component in the c^{th} cluster. The time duration τ_c represents the delay of the c^{th} cluster, and $\tau_{c,l}$ is the delay of the l^{th} path in the c^{th} cluster

relative to the c^{th} cluster arrival time, by definition we have $\tau_{c,0} = 0$. The cluster and path inter-arrival times are described by Poisson processes:

$$p_{\tau_c}(\tau_c|\tau_{c-1}) = \Lambda \cdot \exp[-\Lambda(\tau_c - \tau_{c-1})], \qquad c > 0 \tag{2.9}$$

$$p_{\tau_{c,l}}(\tau_{c,l}|\tau_{c,l-1}) = \lambda \cdot \exp[-\lambda(\tau_{c,l} - \tau_{c,l-1})], \qquad l > 0 \tag{2.10}$$

where Λ is the cluster arrival rate and λ is the ray arrival rate (where $\lambda > \Lambda$). The average power of both the clusters and the rays within the clusters is assumed to decay exponentially such that the average power of a multipath component at a given delay, $\tau_c + \tau_{c,l}$, is given by:

$$\Omega_{c,l} = \mathrm{E}\left[|\alpha_{c,l}|^2\right] = \Omega_{0,0} \cdot \exp\left(-\frac{\tau_c}{\Gamma} - \frac{\tau_{c,l}}{\gamma}\right) \tag{2.11}$$

where $\Omega_{0,0}$ is the mean energy of the first path of the first cluster, Γ & γ are the cluster and ray decay factors respectively. In this thesis, we consider that there is no attenuation in the transmission channel. So the desired value of $\Omega_{0,0}$ is set to normalize the total average power: $\sum_{c=0}^{C-1}\sum_{l=0}^{L-1}\Omega_{c,l} = 1$. To simplify the channel model structure we consider also that the UWB channel consists only of one cluster with small path numbers.

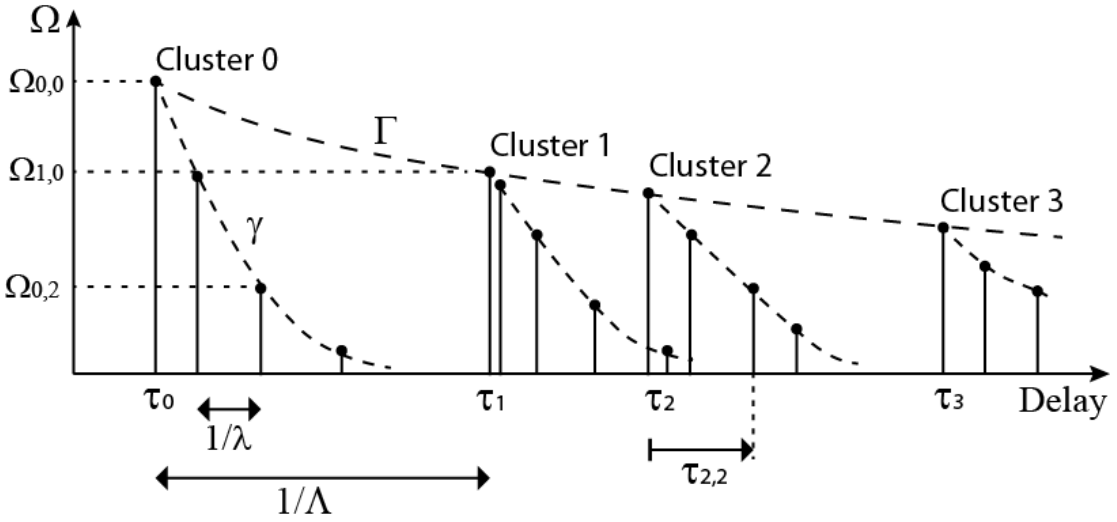


Figure 2.9 Principle of the Saleh-Valenzuela channel model

The four main parameters of UWB channel model (Λ, λ, Γ and γ) can be changed for different environments. The IEEE 802.15 3a has defined four different channel models (CM1, CM2, CM3 and CM4) for the UWB system modelling. CM1 describes a line-of-sight (LOS) scenario with a separation between the transmitter and receiver of less than 4 m. CM2 describes the same range but for a non-LOS situation. CM3 describes a non-LOS scenario for distances

of 4 to 10 m between a transmitter and a receiver. CM4 describes an environment with strong delay dispersion, resulting in a delay spread of 25 ns. Table 2.2 provides the model parameters (Λ, λ, Γ and γ) of CM1 to CM4.

Table 2.2 Multipath channel model parameters

Parameters	CM1	CM2	CM3	CM4
Λ (ns-1)	0.0233	0.4	0.0667	0.0667
λ (ns-1)	2.5	0.5	2.1	2.1
Γ	7.1	5.5	14	24
γ	4.3	6.7	7.9	12

2.8 Time synchronization

As we have previously seen, UWB communications is an extremely promising technology for dynamic wireless sensor networks. However, one of the most critical challenges in enabling the unique benefits and potential of UWB transmissions is timing synchronization.

In general, timing synchronization system consists of three units: signal detection, timing acquisition and tracking. Figure 2.10 depicts the synchronization system block. Signal detection is the first unit, for deciding if the signal received is desired UWB signal or noise only [33, 34]. Timing Acquisition unit is, a coarse synchronization, employed to find approximately a starting point of each received symbol and to reduce the timing error to within a fraction of UWB pulse duration [27-29]. The third step is a tracking to maintain and lock the satisfactory synchronization in the presence of timing offset variations in the received waveform, as a result of oscillator drifts or relative motion between the transmitter and receiver motion (Doppler effects) [30-32]. Doppler effects often arise in dynamic wireless sensor networks, where the nodes are able to move within the application area; one example of dynamic WSNs is to monitor the patients in all hospital units.

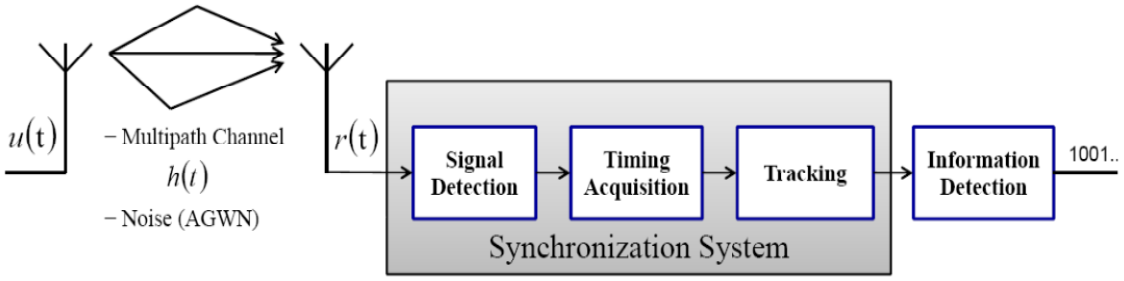


Figure 2.10 Synchronization system block

This section is organized as follows: First, the UWB signal model is introduced. Also, the timing synchronization problem will mathematically be represented by timing offset error, which refers to the difference between the beginning of the received waveform and that of the generated reference template. Then, we will describe the matched filter which is considered as a well-known method used to realize timing synchronization for wireless communication systems, in particular, for narrow-band communication devices. Unfortunately, we will see that this approach for UWB systems requires high computational complexity and long synchronization time. Therefore, we will search a rapid and low-complexity approach for perfectly achieving synchronization for UWB systems. The timing with dirty template (TDT) approach will be proposed as an adequate solution. The proposed timing method will further be combined with the UWB signal model.

2.8.1 UWB signal modelling

Consider an impulse radio IR-UWB system, where every symbol is transmitted over T_s period that consists of N_f pulses over N_f frames (one pulse per frame). Every frame of duration T_f contains N_c chips. The symbol waveform of duration $T_s := T_f N_f$ is:

$$p_T(t) = \sum_{j=0}^{N_f-1} p(t - jT_f - c_j T_c) \tag{2.12}$$

where $p(t)$ is un ultra-short pulse that has duration $T_p (\ll T_f)$, and $T_c := T_f / N_c$ is the chip duration with pseudo-random time-hopping (TH) codes $\{c_j\} \in [0, N_c - 1], j \in [0, N_f - 1]$ (see Figure 2.11).

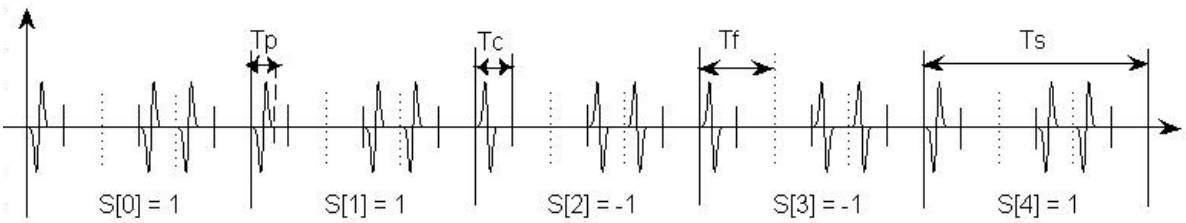


Figure 2.11 TH-UWB signal with PAM modulation ($N_f = 3, N_c = 2$, TH codes=[0, 1,0])

We consider that the symbol waveform has unit energy ($\int p_T^2(t)dt = 1$). The digital information is modulated based on pulse amplitude modulation (PAM), where the information-bearing symbols $s[n] \in \{\pm 1\}$ are modelled as binary independent and identically distributed (i.i.d.) with energy $\mathcal{E}_s$ spread over N_f frames. The transmitted UWB waveform is then given by [64]:

$$u(t) = \sqrt{\mathcal{E}_s} \sum_{n=0}^{\infty} s[n] p_T(t - nT_s) \tag{2.13}$$

The signal $u(t)$ propagates through a multipath channel, whose impulse response given:

$$h(t) := \sum_{l=0}^{L-1} \alpha_l \delta(t - \tau_l) \tag{2.14}$$

where $\delta(t)$ is the Dirac function, L denotes the total number of resolvable channel paths, α_l and τ_l denote l-channel path attenuation and delay respectively, obeying $\tau_l < \tau_{l+1}$.The timing offset τ_0 refers to the first arrival time. To isolate τ_0, we define $\tau_{l,0} \coloneqq \tau_l - \tau_0$ as the relative time delay of each channel tap, where $\tau_{L-1,0}$ is channel delay spread. By definition we have $\tau_{0,0} = 0$. To avoid inter-symbol interference (ISI), T_f is selected to satisfy the following condition [17]:

$$T_f \geq \left(c_{N_f-1} - c_0\right) T_c + \tau_{L-1,0} + T_p \tag{2.15}$$

The received signal is distorted by the unknown multipath channel and by the ambient noise. Moreover, due to the effects of propagation channel (such as Frequency Selective Fading) and the variation of antenna characteristics caused by large bandwidth, the shape of the UWB received signal is different from that of the transmitted signal $p(t)$ [65]. We suppose that the received monocycle pulse has a similar form of the transmitted pulse $p(t)$ for simplification. Accordingly, the received pulse within each frame can be modeled as:

$$p_r(t) := \sum_{l=0}^{L-1} \alpha_l p(t - \tau_{l,0}) \tag{2.16}$$

Thus, the waveform in the output of the receiver antenna is:

$$r(t) = \sqrt{\mathcal{E}_s} \sum_{n=0}^{\infty} s[n] p_R(t - nT_s - \tau_0) + w(t) \tag{2.17}$$

where $p_R(t)$ is the received waveform of each symbol:

$$p_R(t) := \sum_{j=0}^{N_f-1} p_r\left(t - jT_f - c_j T_c\right) = \sum_{l=0}^{L-1} \alpha_l p_T\left(t - \tau_{l,0}\right) \tag{2.18}$$

and $w(t)$ represents the bandpass filtered zero-mean additive white Gaussian noise (AGWN) with power spectral density (PSD) $N_0/2$ and with double sided bandwidth B ($\gg 1/T_s$) (Appendix 2A). The timing offset τ_0 could be represented by:

$$\tau_0 = n_s T_s + \tau \tag{2.19}$$

where $n_s = \lfloor \tau_0 / T_s \rfloor \geq 0$ denotes the symbol-level timing offset, $\lfloor \cdot \rfloor$ represents the floor operation, and $\tau \in [0, T_s)$ the fine-timing offset [17]. By substituting τ_0 in (2.17), the received signal can be expressed by:

$$r(t) = \sqrt{\mathcal{E}_s}\sum_{n=0}^{\infty} s[n]p_R(t - nT_s - n_sT_s - \tau) + w(t) \tag{2.20}$$

Under mistiming ($\tau_0 \neq 0$), any T_s-long received segment of $r(t)$ can be represented by parts of two consecutive symbols (see Figure 2.12: the received UWB pulses are represented by triangles to illustrate the rich multipath effects of the channel), as bellow:

$$x(t + nT_s) = w(t + nT_s) + \begin{cases} \sqrt{\mathcal{E}_s}s[n - n_s - 1]p_R(t + T_s - \tau): t \in [0, \tau) \\ \sqrt{\mathcal{E}_s}s[n - n_s]p_R(t - \tau) \quad : t \in [\tau, T_s) \end{cases} \tag{2.21}$$

where $x(t)$ is the received segment of duration T_s.

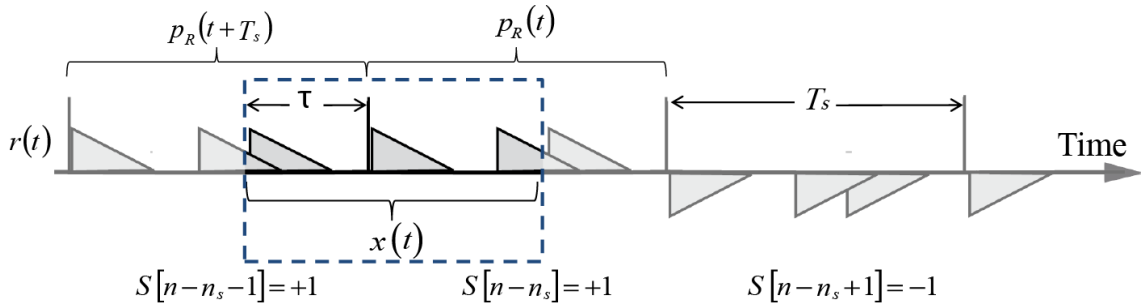


Figure 2.12 T_s-long observed received symbol $x(t)$

To extract correctly the information $s[n]$ from the received signals, the receiver should acquire correct timing information before demodulating. For a carrier-less system, such as IR-UWB system, the timing information is only the timing offset between the received pulse and the reference template signal. However, identifying timing offset error (timing synchronization) in UWB signal becomes so difficult due to the low power density (under the noise floor) of the impulses transmitted and due to their very low duty cycle (much smaller than 1%, where the UWB pulse only exists in a relatively short period during one cycle). Obtaining high precision timing is also difficult, because the transmitted impulse has extremely short width comes along with the ultra-wide bandwidth. Usually, the pulse width is at sub-nanosecond order (typically 0.1~1ns). The transmitted impulses are further deteriorated in the presence of dense multipath channel. Hence, Accurate timing offset estimation (TOE) imposes major challenges to IR-UWB systems in achieving their potential bit error rate (BER) performance, capacity, and throughput [17]. In the following subsections we describe the techniques used to detect the timing information in the received signal $r(t)$.

2.8.2 Correlation receiver (matched filter)

The correlation receiver is a well-known optimal method that is widely used in acquiring the timing information for a narrowband signal in an additive white Gaussian noise (AWGN)

channel. To estimate timing offset at the matched filter, a sliding correlator is applied between the received signal and the local reference signal, called a clean template signal because it is generated at the receiver system so it is clean from the ambient noise and from the channel interference. In the absence of signal distortion throughout the transmission channel, the optimal shape of receiver template is: $f(t) = p_T(t)$ [66]. Figure 2.13 depicts a sliding correlator with a reference signal $f(t)$. Timing synchronization is realized by searching a peak (a maximum point) at the output of the sliding correlation between $r(t)$ and $f(t)$ for different value of τ_n. After acquiring the timing offset, the output signal $y(t)$ is used to demodulate the received signal $r(t)$. Thus, Timing acquisition is given as:

$$\hat{\tau} = \underset{\tau_n}{arg\ max}\ |y(t)|; \quad y(t) = \int_0^{T_s} r(t+\tau_n) \cdot f(t)dt \tag{2.22}$$

The major challenge for a matched filter in UWB system is to construct a proper template $f(t)$ needed to estimate timing offset $\hat{\tau}$ in (2.22). Since the clean template should perfectly match the aggregate receive-waveform $r(t)$ [66], which was produced from convolving the transmitted signal $p_T(t)$ with the unknown multipath channel $h(t)$. Thus, the challenge here is that no clean template is available for perfectly matching $r(t)$, as long as the multipath channel parameters $\{\alpha_l, \tau_{l,0}\}$ are unknown.

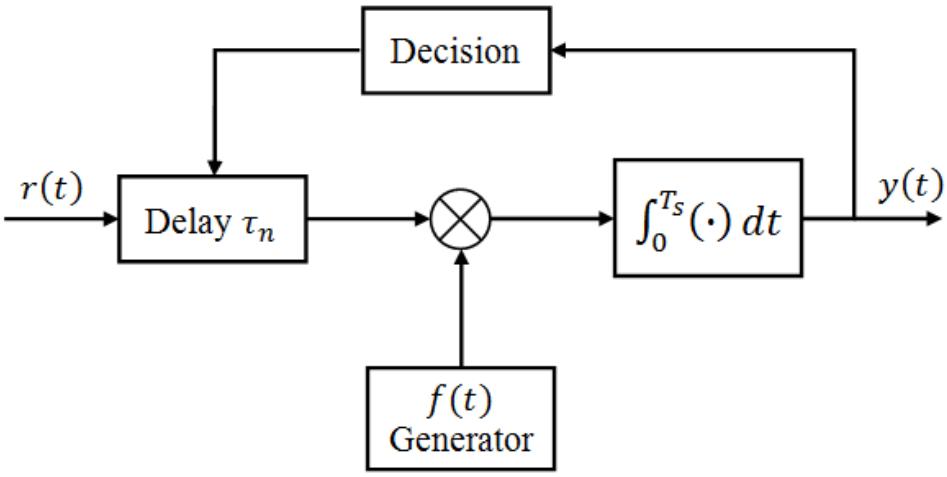


Figure 2.13 Matched filter for UWB with clean template generator

One suggestion could bypass this problem is to estimate channel path attenuations and delays $\{\alpha_l, \tau_{l,0}\}$ by applying an Maximum Likelihood (ML) estimator, the objective function for ML channel estimator could be given as the following expression in the absence of data modulation [67]:

$$\{\hat{\alpha}_l, \hat{\tau}_{l,0}\} = \underset{\{\alpha_l, \tau_{l,0}\}_{l=0}^{L_r-1}}{arg\ max} \int_0^{T_0} \left| r(t) - \sum_{l=0}^{L_r-1} \alpha_l p_T\left(t - kT_s - \tau_{l,0}\right) \right|^2 dt \quad (2.23)$$

where T_0 is length of a synchronization window over which ML estimator searches the optimal value of L_r-channel path parameters $\{\alpha_l, \tau_{l,0}\}: l = 0,1,\cdots, L_r - 1$. After ML channel estimation, the estimates $\{\hat{\alpha}_l, \hat{\tau}_{l,0}\}$ would be used to form an L_r-finger RAKE receiver template, as follows:

$$f_{RAKE}(t) = \sum_{l=0}^{L_r-1} \hat{\alpha}_l p_T\left(t - \hat{\tau}_{l,0}\right) \quad (2.24)$$

RAKE receiver is realized by replacing the proper template $f(t)$ mentioned in Figure 2.13 by $f_{RAKE}(t)$ template. RAKE receiver is usually employed to explore the rich multipath components and increase thus the received energy capture at the correlator output (energy of $y(t)$), leading to improve the information decision quality (such as BER). Unfortunately, two major issues prevent this ML estimator to realize for a practical UWB receiver: First is that requires sampling at or above the Nyquist rate (over 10 GHz), that rates cannot be reached by cheap technology today [67]. Second is that for a dense multipath channel, there is a very large number of resolvable multipath channel parameters that must be estimated accurately, which incurs high computational complexity and long synchronization time [20, 67]. Furthermore, the optimal RAKE receiver should use the same shape of the received pulses as the correlation template. In the absence of antenna and channel distortion, we could say that the RAKE receiver knows in prior the shape of the received signals and uses $p_T(t)$ as the correlation template. In contrast, due to distortion in the propagation channel, the RAKE receiver would have a lack of knowledge of the distorted pulse shape at the receiver. As a result, non-ideal receiver templates would be used. Imperfect correlations can occur by processing the data with an incorrect pulse shape, causing degradation in correlation energy and in system quality. Therefore, this approach is not only sub-optimal in the presence of rich resolvable multipath channel, but also incurs high computational complexity and long synchronization time [17, 20].

We are thus motivated to look for rapid and low-complexity approach for detecting the received signal and recovery the timing information. Several timing algorithms in the literature have been proposed recently for UWB Impulse Radio systems [21-26]. For example, coarse bin reversal searching [21], coded beacon sequence in conjunction with a bank of correlators [22], the inherent cyclo-stationarity (CS) approach [23], transmitted reference (TR) approaches [25]. Each of these approaches requires one or more of the following assumptions: 1) the absence of multipath; 2) the absence of time-hopping (TH) codes; 3) the multipath

channel is known; 4) high computational complexity and long synchronization time; and 5) degradation of bandwidth and power efficiency.

2.8.3 Timing with dirty template method

In this thesis, we have proposed Timing with Dirty Templates (TDT) as an efficient synchronization approach for IR-UWB, introduced in [27-33]. This technique is based on correlating the received signal with "dirty template" extracted from the received waveforms. This template is called dirty; because it is distorted by the unknown channel and by the ambient noise. TDT allows the receiver to enhance energy capture even when the multipath channel and the Time-Hopping (TH) spreading codes are both unknown. Consequently, TDT approach contributes to enhance synchronization performance for IR-UWB and to reduce receiver structure complexity [64, 68].

The original TDT method proposed in [17], is defined as the cross-correlation between the pairs of successive received segments of duration T_s. That means: one segment in each pair of successive received symbols serves as the reference template signal for the other segment, as shown below:

$$R_{x,x}[n] = \int_0^{T_s} x(t + nT_s)x(t + (n+1)T_s)\, dt \tag{2.25}$$

where $x(t + nT_s)$ is T_s-long received segment of $r(t)$. Substituting (2.21) in (2.25) yields:

$$R_{x,x}[n] = \int_0^{T_s} \{w(t + nT_s) + \sqrt{\mathcal{E}_s}\sum_{m=0}^{1} s[n - n_s - m]p_R(t + mT_s - \tau)\} \times \{w(t + (n+1)T_s) + \sqrt{\mathcal{E}_s}\sum_{m=0}^{1} s[n - n_s - m + 1]p_R(t + mT_s - \tau)\}dt$$

Thus:

$$R_{x,x}[n] = \breve{\omega}[n] + A\int_0^{T_s} p_R^2(t + T_s - \tau)\, dt + B\int_0^{T_s} p_R^2(t - \tau)\, dt \tag{2.26}$$

where the sampled noise $\breve{\omega}[n]$ explained in Appendix 2B, is composed of three terms, two of them are the result of correlation between the symbol and the noise, and the third term is between shifted noises (noise-cross-noise). These terms could be treated approximately as mutually uncorrelated zero mean Gaussian noises. So $\breve{\omega}[n]$ could represent as zero-mean Gaussian noise, and its statistic properties are given as (see Appendix 2B):

$$\breve{\omega}[n] \sim \mathcal{N}\left(0, \sigma_w^2 = \frac{{N_0}^2}{2}BT_s + N_0\underbrace{\mathcal{E}_s\mathcal{E}_{max}}_{\mathcal{E}_r}\right) \tag{2.27}$$

where $\mathcal{E}_{max} = \int_0^{T_s} p_R(t)^2\, dt$ and $\mathcal{E}_r = \mathcal{E}_s \mathcal{E}_{max}$ represents T_s-long received segment energy. The dirty template sample $R_{x,x}[n]$ is thus given as:

$$R_{x,x}[n] = \breve{\omega}[n] + A \int_{T_s-\tau}^{T_s} p_R^2(t)\, dt + B \int_0^{T_s-\tau} p_R^2(t)\, dt \tag{2.28}$$

where: $\begin{cases} A = \mathcal{E}_s s[n - n_s - 1].\, s[n - n_s] \\ B = \mathcal{E}_s s[n - n_s].\, s[n - n_s + 1] \end{cases}$

The possible values of *A* and *B* are exhibited in Table 2.3.

Table 2.3 Possible values of A and B in (2.28)

$\boldsymbol{s[n-1]}$	+1	+1	-1	-1	+1	-1	+1	-1
$\boldsymbol{s[n]}$	+1	-1	-1	+1	+1	-1	-1	+1
$\boldsymbol{s[n+1]}$	-1	-1	+1	+1	+1	-1	+1	-1
$\boldsymbol{A}$	$+\mathcal{E}_s$	$-\mathcal{E}_s$	$+\mathcal{E}_s$	$-\mathcal{E}_s$	$+\mathcal{E}_s$	$+\mathcal{E}_s$	$-\mathcal{E}_s$	$-\mathcal{E}_s$
$\boldsymbol{B}$	$-\mathcal{E}_s$	$+\mathcal{E}_s$	$-\mathcal{E}_s$	$+\mathcal{E}_s$	$+\mathcal{E}_s$	$+\mathcal{E}_s$	$-\mathcal{E}_s$	$-\mathcal{E}_s$

The TDT technique can be applied to UWB receiver even in the presence of time hopping TH codes or Inter-Frame Interference (IFI), because the received segment and its dirty template contain the same TH codes and IFI properties regardless of the unknown channel characteristics (e.g. unknown time offset), but with the condition of the absence of ISI. Moreover, this method exploits the rich multipath diversity provided by UWB channels, and does not need to neither estimate the propagation channel nor generate clean correlation template at the receiver. Consequently, it reduces receiver complexity with high received energy capture and high synchronization rapidity.

2.9 System components

The proposed IR-UWB system is shown in Figure 2.14, which consists of the transmitter, multipath channel and receiver. The transmitter structure has previously been explained in this chapter, including Data-aided bits block which is used to add training bits within transmitted data sequence. These redundant bits are known a priori at the receiver and exploited for achieving signal detection and improving the synchronization performance, as we will see later in next chapters. In the propagation channel, the transmitted signal is distorted by the AGWN noise and by multipath channel generated form S-V model. The first arrival time parameter τ_0 is unknown at the receiver which seeks to estimate it. In the thesis the synchronization system, in particular the tracking stage, will be evaluated for both in absence of

Doppler effects (τ_0 is constant) and in presence of Doppler effects (τ_0 varies with the time). It is so important to take Doppler effects into account, because our work is oriented to dynamic wireless sensor network applications. From the thesis's point of view, the receiver consists of two main blocks: timing synchronization and data demodulation blocks. Timing synchronization in its turn consists of three blocks: signal detection, timing acquisition and tracking. Signal detection stage seeks to perfectly detect the presence of the received signal $r(t)$ and to identify the symbol-level offset n_s. Timing acquisition stage focuses on finding the timing offset τ accurately and quickly. While tracking stage contributes to maintaining and locking the synchronization between the transmitter and receiver even in the presence of variations in the first arrival time of the received signal as a result of Doppler effects.

As shown in Figure 2.14, the comparison and evolution of performance of the proposed methods can be done by studying the mean square of timing offset error (MSE) and bit error rate (BER).

2.10 **Conclusion**

In this chapter, we have presented an overview on UWB technology which makes the reader familiar with the UWB. This thesis focuses on impulse-radio UWB which holds enormous potential for wireless sensor network applications, due to its very good time-domain resolution, rich multipath diversity, low power and low cost system, potential spectrum compatibility with existing narrowband systems, etc. However, timing synchronization is one of the most crucial tasks for a UWB receiver in achieving the benefits of UWB transmissions.

Then TDT algorithms have been proposed as a solution for carrying out low-complexity high-performance timing synchronization. TDT technique is based on correlating the received signal with "dirty template" extracted from the received waveforms. This template is called dirty; because it is distorted by the unknown channel and by the ambient noise. TDT allows enhancing received energy capture and reducing receiver structure complexity. The described technique can be applied to UWB systems even in the presence of time hopping TH codes, Inter-Frame Interference (IFI) and rich multipath environment, provided that Inter-Symbol Interference (ISI) is absent.

Finally, we have considered that timing synchronization system consists of three main blocks: signal detection, timing acquisition and tracking. Each block of them will be explained later in a separated chapter of this dissertation.

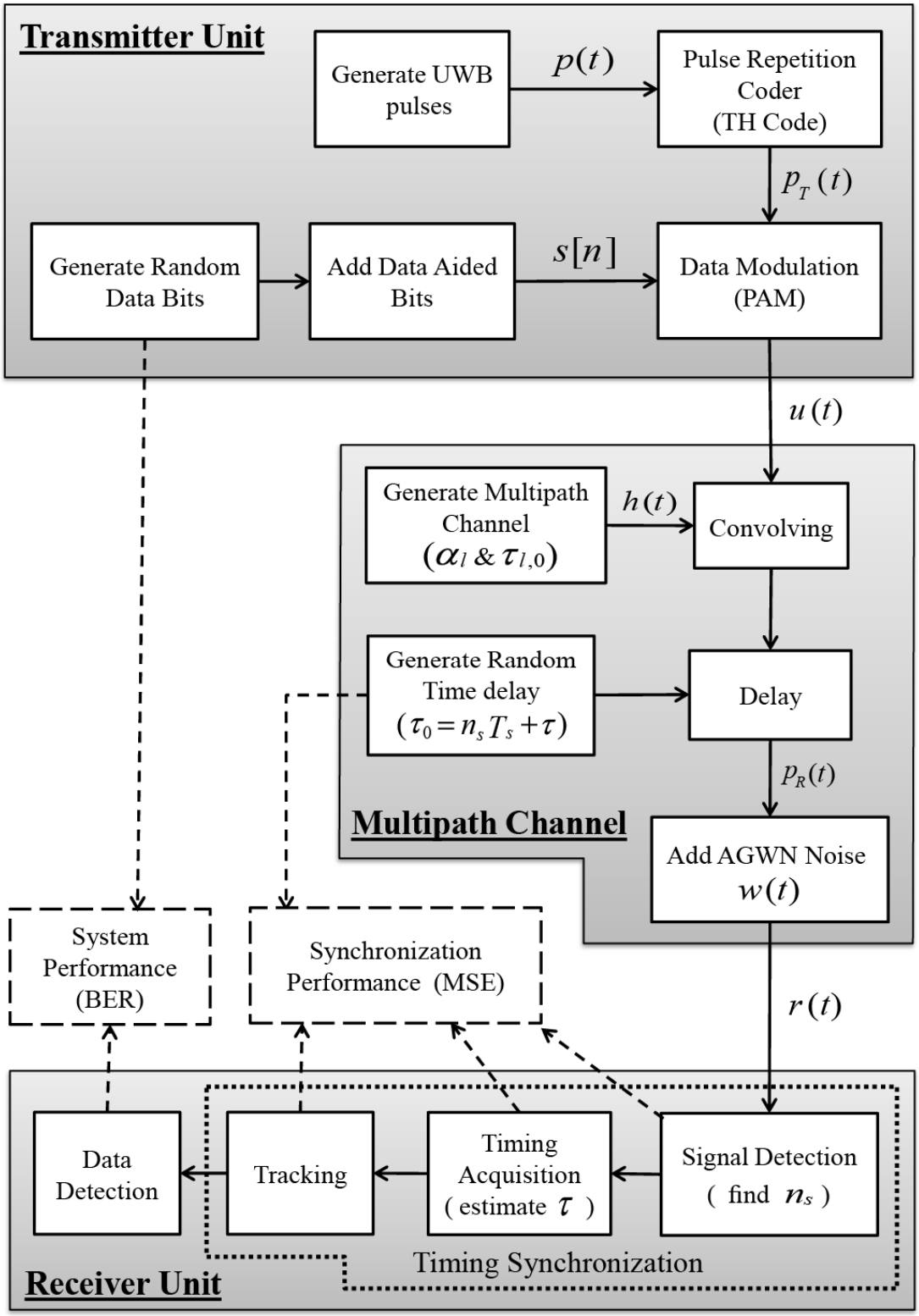


Figure 2.14 UWB system (transmitter, multipath channel and receiver)

Chapter 3

UWB Signal Detection

In UWB communication system, the transmission channels are polluted with AGWN noise, and the receiver antenna not only captures the energy of UWB waveforms, but also the ambient noise. This noise deteriorates the received signal and contributes to generate errors in the signal detection. Thus the detection problem for UWB system is defined as how to make a decision whether the received signal is a desired UWB waveform or noise only. In this chapter, we will first introduce the concepts of detection theory. Basically, the signal detection is achieved by comparing the received captured samples with a threshold, if the sample exceeds this threshold we say that the signal is present; otherwise, the signal is absent. We will find further that selecting the threshold value affects efficiently the detection performance. For determining the suitable threshold value, we suggest the Neyman-Pearson approach. Finally, we will apply the detection criteria on dirty template UWB system to establish a good detection circuit and to define optimal threshold values. The simulation results will then illustrate the detection performance for different values of signal to noise ratio (SNR), training sequence length and threshold.

3.1 Detection theory

For explaining the detection theory concepts, let's first start with a very simple signal detection problem. We consider first sending a constant signal with amplitude A corrupted by additive Gaussian noise $w[n]$ with zero mean and variance σ^2. To simplify the study we take only one received sample into consideration. Based on this observation sample we wish to decide between two cases (Hypotheses): the first hypothesis $\mathcal{H}_0$ indicates noise-only present case, whereas $\mathcal{H}_1$ denotes the signal present hypothesis.

$\mathcal{H}_0$: $x[0] = w[0]$

$$\mathcal{H}_1: \quad x[0] = A + w[0] \tag{3.1}$$

Since the noise $w[0]$ is assumed to be zero mean with σ^2 variance, the probability density function (PDF) of $x[0]$ under each hypothesis is denoted by [69]:

$$p(x[0];\mathcal{H}_0) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{1}{2\sigma^2}x^2[0]\right)$$
$$p(x[0];\mathcal{H}_1) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{1}{2\sigma^2}(x[0]-A)^2\right) \tag{3.2}$$

From Figure 3.1 the PDFs under $\mathcal{H}_0$ and $\mathcal{H}_1$ are plotted as dashed and solid curves respectively. The reasonable approach, might be used for making a decision if the signal is present or not, is to compare the observed sample $x[0]$ to a threshold η. In this example, the threshold could be intuitively set to $0.5A$. This means that the measured value of $x[0]$ falls below this threshold $x[0] < \eta$, it is considered as noise. In contrast, if the value exceeds the threshold $x[0] > \eta$, it is considered as signal. Basing on the selected threshold, there are four events (illustrated as darker areas in Figure 3.1) can occur. The probability of each event is done by:

$P(\mathcal{H}_1;\mathcal{H}_1)$: Probability of decide $\mathcal{H}_1$ when $\mathcal{H}_1$ is true = probability of detection (P_D)
$P(\mathcal{H}_0;\mathcal{H}_1)$: Probability of decide $\mathcal{H}_0$ when $\mathcal{H}_1$ is true = probability of miss (P_M)
$P(\mathcal{H}_1;\mathcal{H}_0)$: Probability of decide $\mathcal{H}_1$ when $\mathcal{H}_0$ is true = probability of false alarm (P_{FA})
$P(\mathcal{H}_0;\mathcal{H}_0)$: Probability of decide $\mathcal{H}_0$ when $\mathcal{H}_0$ is true = probability of no detection (P_{Null})

Hence, in the signal detection problem two correct types of decision could occur: P_D and P_{Null}, and two incorrect error types of decision could occur P_M and P_{FA}. It is evident that the probabilities P_M and P_{Null} are equal to $1 - P_D$ and $1 - P_{FA}$ respectively. Thus, to derive the optimal detector design in this example we have only two degrees of freedom P_D and P_{FA}. These two probabilities are computed, for our example, as follows:

$$\begin{aligned} P_{FA} &= P(\mathcal{H}_1;\mathcal{H}_0) \\ &= Pr\{x[0] > \eta;\mathcal{H}_0\} \\ &= \int_{\eta}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} exp\left[-\frac{1}{2\sigma^2}x^2\right] dx = Q\left(\frac{\eta}{\sqrt{\sigma^2}}\right) \end{aligned} \tag{3.3}$$

The function $Q(\eta)$ is called the *complementary cumulative distribution function*, and it is defined as the probability of exceeding a given value η [69]. Similarly, the probability of detection is derived, as:

$$P_D = P(\mathcal{H}_1;\mathcal{H}_1)$$

$$= Pr\{x[0] > \eta; \mathcal{H}_1\}$$

$$= \int_{\eta}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} exp\left[-\frac{1}{2\sigma^2}(x-A)^2\right] dx = Q\left(\frac{\eta - A}{\sqrt{\sigma^2}}\right) = Q\left(Q^{-1}(P_{FA}) - \sqrt{\frac{A^2}{\sigma^2}}\right) \quad (3.4)$$

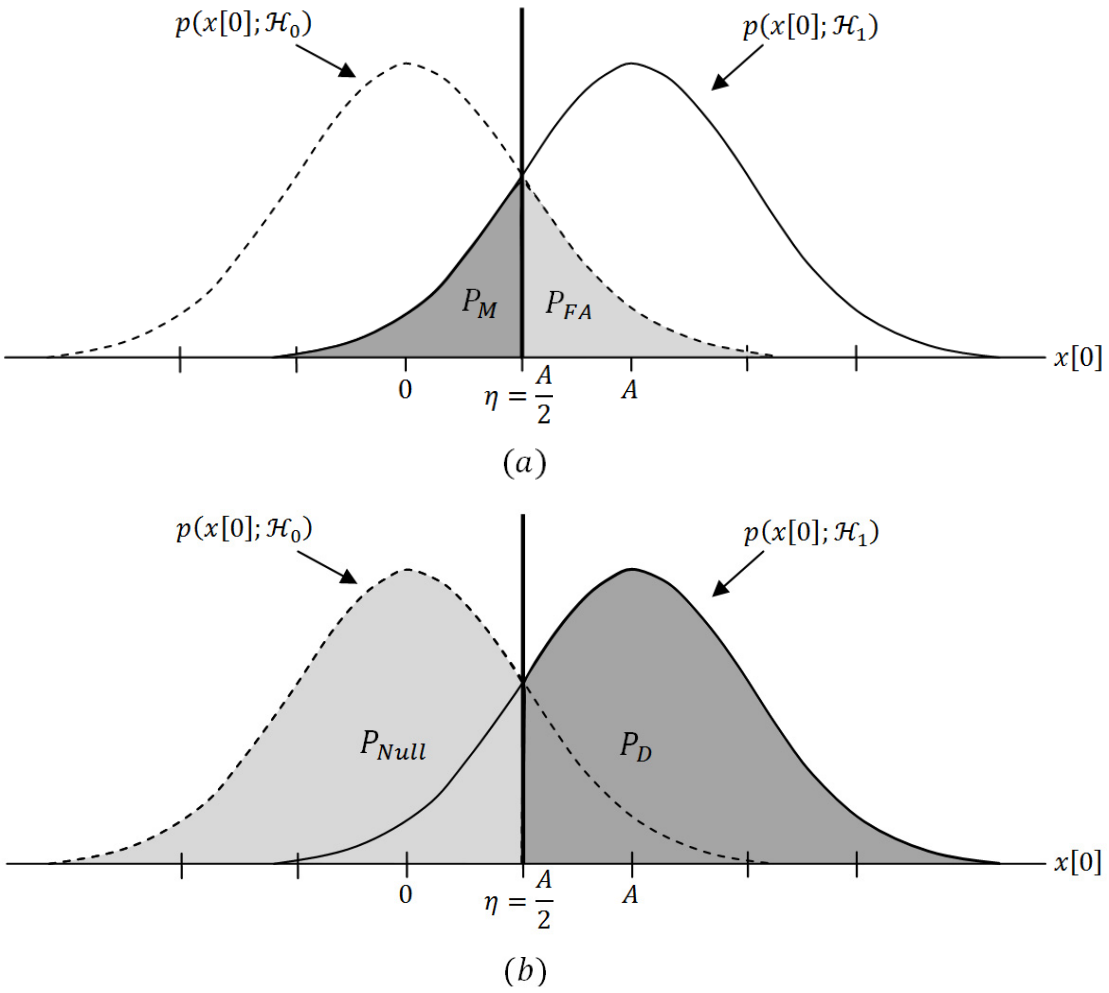


Figure 3.1 PDF of $x[0]$ for signal or noise only present (a) probability of testing errors P_M and P_{FA} (b) probability of correct decissions P_D and P_{Null}

In some applications of detection problems, the selection of a suitable threshold value would not be as evident as this one. Let's assume that the threshold value η in the previous example might also be set to A as mentioned in Figure 3.2. We can see that as the threshold value η increases, the probability of false alarm P_{FA} decreases, but the probability of detection P_D might be reduced simultaneously. We can see from (3.4) and from Figure 3.2 that these two probabilities are correlated, and in order to increase P_D we might also increase P_{FA}. We can

notice that detection performance depends upon threshold value. The objective of the next step is finding the optimal threshold value to improve detection performance.

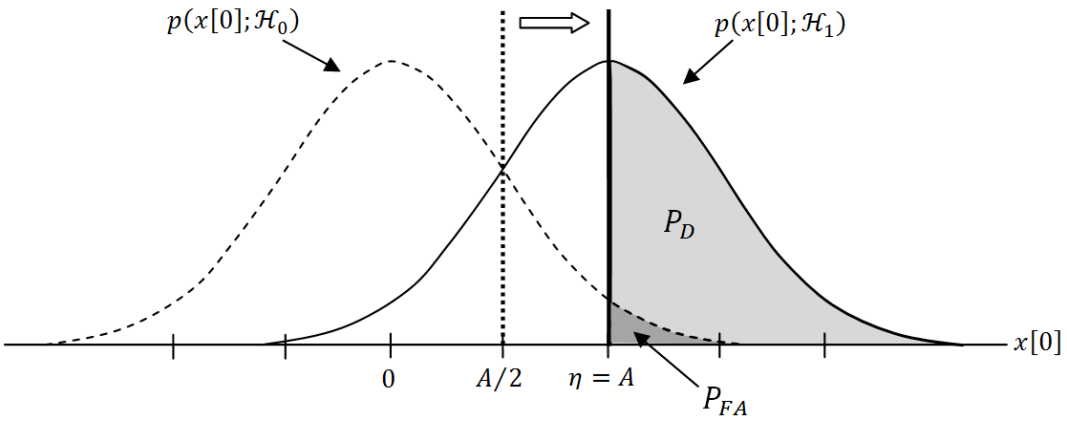


Figure 3.2 Decision and false alarm regions for thersould $\eta = A$

3.2 The Neyman-Pearson theorem

In general, there are two major theorems (criteria) used for designing good detection tests and defining optimal threshold values: the *Bayesian* approach and the *Neyman-Pearson* theorem. The Bayesian method is based on minimizing a risk function, but it requires a priori knowledge of probability of each hypothesis $\mathcal{H}_0$ and $\mathcal{H}_1$, as in our case, it requires the probabilities of the signal presence and signal absent, which are not known. On the other hand, the Neyman-Pearson criteria allows us choosing a suitable threshold value and maximizing the detection test quality.

The Neyman-Pearson approach provides us a way for maximizing P_D for a given $P_{FA} = \alpha$, by computing the ratio of the probability of the signal to the probability of the noise[69].

$$L(x) = \frac{p(x;\mathcal{H}_1)}{p(x;\mathcal{H}_0)} > \eta \tag{3.5}$$

where the function $L(x)$ is called the *likelihood ratio*, indicating the ratio between the likelihood of $\mathcal{H}_1$ and the likelihood of $\mathcal{H}_0$ for each value of x. The entire test of (3.5) is termed the *likelihood ratio test* (LRT). The threshold η is found from a give value of false alarm probability P_{FA}. Returning to the previous example, but instead of taking only one measured sample we take several observations $x[n]$ for $n = 0,1,\dots,N-1$ where N is the observed data length. The hypotheses are described as:

$$\mathcal{H}_0: \quad x[n] = w[n] \qquad\qquad n = 0,1,\dots,N-1$$

$$\mathcal{H}_1: \quad x[n] = A + w[n] \qquad n = 0,1,\dots,N-1 \tag{3.6}$$

The likelihood ratio is given by:

$$L(x) = \frac{p(x;\mathcal{H}_1)}{p(x;\mathcal{H}_0)} = \frac{\frac{1}{(2\pi\sigma^2)^{N/2}} exp\left[-\frac{1}{2\sigma^2}\sum_{n=0}^{N-1}(x[n]-A)^2\right]}{\frac{1}{(2\pi\sigma^2)^{N/2}} exp\left[-\frac{1}{2\sigma^2}\sum_{n=0}^{N-1}x^2[n]\right]} \tag{3.7}$$

By applying the Neyman-Pearson test, we decide $\mathcal{H}_1$ if

$$L(x) = \frac{p(x;\mathcal{H}_1)}{p(x;\mathcal{H}_0)} = \frac{exp\left[-\frac{1}{2\sigma^2}\sum_{n=0}^{N-1}(x[n]-A)^2\right]}{exp\left[-\frac{1}{2\sigma^2}\sum_{n=0}^{N-1}x^2[n]\right]} > \eta$$

Developing the inequality by cancelling the common terms:

$$exp\left[\frac{1}{2\sigma^2}(2A\sum_{n=0}^{N-1}x[n] + NA^2)\right] > \eta$$

Let's take the natural logarithm of the likelihood ratio and simplify it:

$$T(x) = \frac{1}{N}\sum_{n=0}^{N-1}x[n] > \underbrace{\frac{\sigma^2}{NA}\ln\eta + \frac{A}{2}}_{\xi} \tag{3.8}$$

If the test statistic $T(x)$ is higher than the threshold ξ we decide $\mathcal{H}_1$, otherwise we decide $\mathcal{H}_0$. To analyze the detection performance we should first study the statistical properties of $T(x)$. In our case, $T(x)$ has Gaussian distribution under each hypothesis with the following means and variances:

$$\mathcal{H}_0: \quad T(x) \sim \mathcal{N}\left(0, \frac{\sigma^2}{N}\right)$$

$$\mathcal{H}_1: \quad T(x) \sim \mathcal{N}\left(A, \frac{\sigma^2}{N}\right) \tag{3.9}$$

We could now compute the false alarm and detection probabilities, as follows:

$$P_{FA} = Pr\{T(x) > \xi; \mathcal{H}_0\} = Q\left(\frac{\xi}{\sqrt{\sigma^2/N}}\right) \tag{3.10}$$

$$P_D = Pr\{T(x) > \xi; \mathcal{H}_1\} = Q\left(\frac{\xi - A}{\sqrt{\sigma^2/N}}\right) = Q\left(Q^{-1}(P_{FA}) - \underbrace{\sqrt{\frac{NA^2}{\sigma^2}}}_{d}\right) \tag{3.11}$$

From (3.10), the threshold value, which yields the maximum P_D for the given P_{FA}, is found as follows:

$$\xi = \sqrt{\frac{\sigma^2}{N}} Q^{-1}(P_{FA}) \tag{3.12}$$

As we have said before that the detection performance depends on the threshold, there is another factor which impacts on (or measure which shows) the detection behavior, is called *detection index*, sometimes it is termed *signal energy to noise ratio* (ENR). This index is represented in (3.11) by the symbol $d^2 = NA^2/\sigma^2$. Figure 3.3 depicts the detection performance of the previous example versus the detection index d^2. It is seen that for a given value of P_{FA}, the detection probability increases when d^2 increases. We can also notice that for a given value of d^2, decreasing false alarm probability leads simultaneously to a reduction in the detection performance.

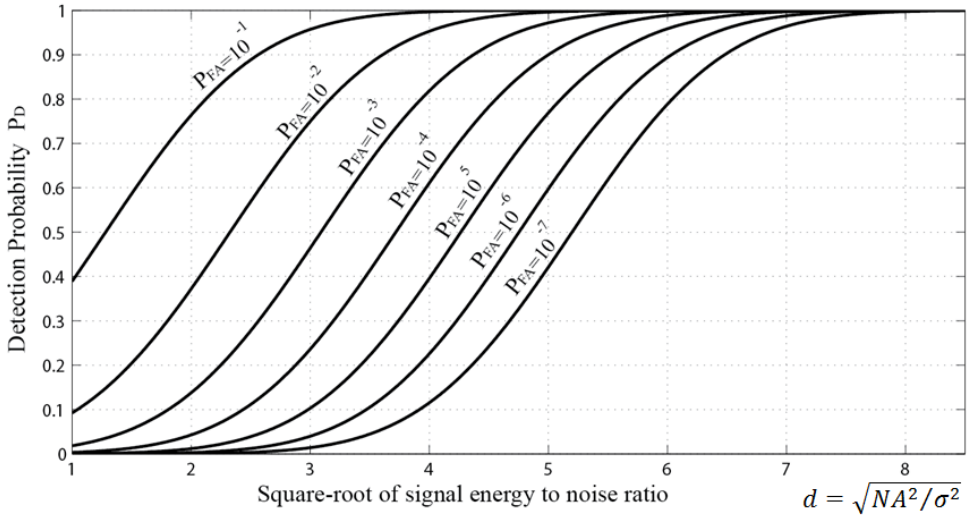


Figure 3.3 Detection performance for fix level signal in AWGN noise

3.2.1 Generalized likelihood ratio test (GLRT)

The likelihood ratio test (LRT) in (3.7) requires to know the value of each A and σ^2, but in the real life applications such knowledge is not available. Therefore these variables (A and σ^2) need to be estimated in order to obtain signal detector with good performance. A likelihood ratio test function which deals with one or more unknown parameters is called a *generalized likelihood ratio test* (GLRT) [69, 70]. The general idea of GLRT can be explained simply as follows: since the LRT is based on the likelihood functions $p(x;\theta,\mathcal{H}_1)$ and $p(x;\theta,\mathcal{H}_0)$ where θ denotes unknown parameters (in our case, θ represents A and σ^2), and it is evident that

maximizing the likelihood function leads to estimate the true (optimal) value of θ. Thus the generalized likelihood ratio test can be obtained as follows:

$$L_G(x) = \frac{\max_\theta \ p(x;\theta,\mathcal{H}_1)}{\max_\theta \ p(x;\theta,\mathcal{H}_0)} = \frac{p(x;\hat{\theta},\mathcal{H}_1)}{p(x;\hat{\theta},\mathcal{H}_0)} > \eta \tag{3.13}$$

where $\hat{\theta}$ is the maximum likelihood estimator MLE for an unknown parameter θ. Returning to the previous example and assuming that A and σ^2 are unknown parameters. Let's consider $\sigma^2 = \hat{\sigma}^2$ is known and $\mathcal{H}_1$ is correct, and we wish to find a desired value of A that maximizes the probability function:

$$\hat{A} = arg \max_A \ p(x;\hat{\sigma}^2,\mathcal{H}_1) = arg \max_A \frac{1}{(2\pi\hat{\sigma}^2)^{N/2}} exp\left[-\frac{1}{2\hat{\sigma}^2}\sum_{n=0}^{N-1}(x[n]-A)^2\right] \tag{3.14}$$

The MLE is derived by setting the derivative of the log-likelihood function to zero:

$$\mathcal{H}_1: \qquad \hat{A} = \frac{1}{N}\sum_{n=0}^{N-1} x[n]$$

In this example, the variance $\hat{\sigma}^2 = \hat{\sigma}_0^2 = \hat{\sigma}_1^2$ can be founded either under the hypothesis $\mathcal{H}_0$ or under the hypothesis $\mathcal{H}_1$that applying MLE:

$$\mathcal{H}_0: \qquad \hat{\sigma}_0^2 = \frac{1}{N}\sum_{n=0}^{N-1} x^2[n]$$

$$\mathcal{H}_1: \qquad \hat{\sigma}_1^2 = \frac{1}{N}\sum_{n=0}^{N-1}\left(x[n]-\hat{A}\right)^2$$

Using $\hat{A}$ and $\hat{\sigma}^2$ the GLRT in (3.7) and the test statistic in (3.8) become:

$$L_G(x) = \frac{\frac{1}{(2\pi\hat{\sigma}^2)^{N/2}} exp\left[-\frac{1}{2\hat{\sigma}^2}\sum_{n=0}^{N-1}(x[n]-\hat{A})^2\right]}{\frac{1}{(2\pi\hat{\sigma}^2)^{N/2}} exp\left[-\frac{1}{2\hat{\sigma}^2}\sum_{n=0}^{N-1} x^2[n]\right]} > \eta$$

$$T(x) = \frac{1}{N}\sum_{n=0}^{N-1} x[n] > \underbrace{\frac{\hat{\sigma}^2}{N\hat{A}}\ln\eta + \frac{\hat{A}}{2}}_{\xi}$$

Before applying the Neyman-Pearson theorem on dirty template UWB system to optimize the signal detection in the presence of noise, let us consider the detection problem concepts by few sentences: In wireless communication systems, the common problem is how to make a decision if the measured samples at receiver antenna output are classified as data embedded in noise or classified as noise-only. In our case, the detection problem could be achieved by respecting the following steps:

- Find first the statistical probabilities (means, variances, PDFs) for the observed signals under each hypothesis $\mathcal{H}_0$ and $\mathcal{H}_1$
- Apply then the Neyman-Pearson theorem to generate the test statistic $T(x)$ under each hypothesis and extract detection and false alarm probabilities P_D and P_{FA}
- Determine the optimal threshold value to maximize P_D for the given value of P_{FA}
- Improve the detection performance by increasing the detection index d^2, either by increasing SNR (A^2/σ^2) if that is possible, or by increasing the observed data length N. So for the communication systems whose SNR is low, we require N to be high enough for good detection performance.

3.3 DT-UWB signal detection

Considering the dirty template UWB model mentioned in (2.28), let's apply the Neyman-Pearson theorem to set a suitable threshold, to detect the presence of the received signal $r(t)$ and to identify the symbol-level offset n_s in (2.20) [33, 34]. Let's also assume that we send $M_1 + 1$ training symbols, have the same value $\{s[n] = 1\}_{n=0}^{M_1}$ (or equal -1), they are thus received during $[\tau_0, \tau_0 + (M_1 + 1)T_s]$. With the presence of the training symbols, we observe at the output of DT correlator (2.25) M_1 samples of $R_{x,x}[n]$. By looking on the Table 2.3 and taking the corresponding values of A and B, then substituting them in (2.28), $R_{x,x}[n]$ becomes:

$$R_{x,x}[n] = \mathcal{E}_s \int_0^{T_s} P_R^2(t)\, dt \; + \breve{w}_{d1}[n] = \underbrace{\mathcal{E}_s \mathcal{E}_{max}}_{\mathcal{E}_r} + \widetilde{\omega_{d1}}[n] = \mathcal{E}_r + \breve{w}_{d1}[n] \tag{3.15}$$

where $\breve{w}_{d1}[n]$ is the dirty template noise under the present of training symbols. Assume that under the noise-only hypothesis $\mathcal{H}_0$, we observe M_1 (i.i.d.) samples of $R_{x,x}[n] = \breve{w}_{do}[n]$ for $n = 0,1, \dots, M_1 - 1$, while under signal-present hypothesis $\mathcal{H}_1$, we observe $R_{x,x}[n] = \mathcal{E}_r + \breve{w}_{d1}[n]$ for $n = n_s, n_s + 1, \dots, n_s + M_1 - 1$. The detection problem is to distinguish between these two hypotheses:

$$\begin{cases} \mathcal{H}_0: \; R_{x,x}[n] = \breve{w}_{do}[n] & n = 0,1, \dots, M_1 - 1 \\ \mathcal{H}_1: \; R_{x,x}[n] = \mathcal{E}_r + \breve{w}_{d1}[n] & n = n_s, n_s + 1, \dots, n_s + M_1 - 1 \end{cases} \tag{3.16}$$

Appendix 2.B demonstrates that the DT noise component $\breve{\omega}[n]$ is composed of three terms: two of them are the result of correlation between the symbol and the noise, and the third term is between shifted noises. Under the hypothesis $\mathcal{H}_0$ the UWB signal is absent and the two signal-noise terms become zeros. Looking at (2.27), we can assert that the noise samples under both

hypotheses are independent Gaussian variables with zero mean. But their variance value is different under $\mathcal{H}_0$ and $\mathcal{H}_1$, as:

$$\begin{cases} \mathcal{H}_0: & \breve{w}_{do}[n] \sim \mathcal{N}\left(0, \sigma_0^2 = \frac{N_0^2}{2} BT_s\right) \\ \mathcal{H}_1: & \breve{w}_{d1}[n] \sim \mathcal{N}\left(0, \sigma_1^2 = N_0 \mathcal{E}_r + \frac{N_0^2}{2} BT_s\right) \end{cases} \tag{3.17}$$

We can consider that the parameters $(\sigma_1^2, \sigma_0^2, \mathcal{E}_r)$ in this detection problem are unknown constant. We need to find the correct value of these parameters before applying the signal detection system. Let's first assume that we have correctly estimated the value of all parameters and the GLRT test is employed to derive the test statistic inequality expression. We will explain later how to estimate the system parameter values by resorting to MLE estimator.

Thus let us achieve the Neyman-Pearson (NP) detector test proposed in (3.13) to decide $\mathcal{H}_1$ and declare the presence of $r(t)$ if the likelihood ratio exceeds a threshold: ($L_G(n) > \eta$), where η is set by the desired probability of false alarm (FA).

$$L_G(n) = \frac{p(R_{x,x}; \sigma_1^2, \mathcal{E}_r, \mathcal{H}_1)}{p(R_{x,x}; \sigma_0^2, \mathcal{H}_0)} > \eta \tag{3.18}$$

$$L_G(n) = \frac{\frac{1}{(2\pi\sigma_1^2)^{M_1/2}} exp\left[-\frac{1}{2\sigma_1^2}\sum_{m=n}^{n+M_1-1}\left(R_{x,x}[m] - \mathcal{E}_r\right)^2\right]}{\frac{1}{(2\pi\sigma_0^2)^{M_1/2}} exp\left[-\frac{1}{2\sigma_0^2}\sum_{m=n}^{n+M_1-1}\left(R_{x,x}[m]\right)^2\right]} > \eta$$

$$L_G(n) = \left(\frac{\sigma_0}{\sigma_1}\right)^{M_1} exp\left[-\frac{1}{2\sigma_1^2}\sum_{m=n}^{n+M_1-1}\left(R_{x,x}[m] - \mathcal{E}_r\right)^2 + \frac{1}{2\sigma_0^2}\sum_{m=n}^{n+M_1-1}\left(R_{x,x}[m]\right)^2\right] > \eta$$

$$L_G(n) = \left(\frac{\sigma_0}{\sigma_1}\right)^{M_1} exp\left[\sum_{m=n}^{n+M_1-1}\left\{\left(-\frac{1}{2\sigma_1^2} + \frac{1}{2\sigma_0^2}\right) R_{x,x}^2[m] + \frac{\mathcal{E}_r}{\sigma_1^2} R_{x,x}[m] - \frac{{\mathcal{E}_r}^2}{2\sigma_1^2}\right\}\right] > \eta$$

Taking the logarithm of both sides does not change the inequality, so that:

$$ln\left(L_G(n)\right) = M_1 ln\left(\frac{\sigma_0}{\sigma_1}\right) + \sum_{m=n}^{n+M_1-1}\left\{\left(-\frac{1}{2\sigma_1^2} + \frac{1}{2\sigma_0^2}\right) R_{x,x}^2[m] + \frac{\mathcal{E}_r}{\sigma_1^2} R_{x,x}[m] - \frac{{\mathcal{E}_r}^2}{2\sigma_1^2}\right\} > ln(\eta)$$

$$\sum_{m=n}^{n+M_1-1}\left\{\left(-\frac{1}{2\sigma_1^2} + \frac{1}{2\sigma_0^2}\right) R_{x,x}^2[m] + \frac{\mathcal{E}_r}{\sigma_1^2} R_{x,x}[m]\right\} > ln(\eta) - M_1 ln\left(\frac{\sigma_0}{\sigma_1}\right) + \frac{M_1 {\mathcal{E}_r}^2}{2\sigma_1^2}$$

$$\sum_{m=n}^{n+M_1-1}\left\{\frac{\sigma_1^2 - \sigma_0^2}{2\mathcal{E}_r\sigma_0^2} R_{x,x}^2[m] + R_{x,x}[m]\right\} > \left(ln(\eta) - M_1 ln\left(\frac{\sigma_0}{\sigma_1}\right) + \frac{M_1 {\mathcal{E}_r}^2}{2\sigma_1^2}\right)\frac{\sigma_1^2}{\mathcal{E}_r}$$

From (3.17) we have $(\sigma_1^2 - \sigma_0^2 = N_0\mathcal{E}_r)$, then

$$\sum_{m=n}^{n+M_1-1}\left\{\frac{N_0}{2\sigma_0^2} R_{x,x}^2[m] + R_{x,x}[m]\right\} > \left(ln(\eta) - M_1 ln\left(\frac{\sigma_0}{\sigma_1}\right) + \frac{M_1 {\mathcal{E}_r}^2}{2\sigma_1^2}\right)\frac{\sigma_1^2}{\mathcal{E}_r}$$

Multiplying each side of the inequality by $2\sigma_0^2/N_0$, the test statistic T becomes:

$$T(n) = \sum_{m=n}^{n+M_1-1}\left\{R_{x,x}^2[m] + \frac{2\sigma_0^2}{N_0}R_{x,x}[m]\right\} > \underbrace{\left(ln(\eta) - M_1 ln\left(\frac{\sigma_0}{\sigma_1}\right) + \frac{M_1 \mathcal{E}_r{}^2}{2\sigma_1^2}\right)\frac{2\sigma_0^2\sigma_1^2}{\mathcal{E}_r N_0}}_{\xi} \tag{3.19}$$

The test statistic in (3.19) decides $\mathcal{H}_1$ and declares the presence of signal if T exceeds the threshold $T > \xi$ .The architecture of the dirty template detector (3.19) is shown in Figure 3.4. The false alarm probability P_{FA} and the detection probability P_D are identified as follows:

$$P_{FA} = Pr\{T > \xi; \mathcal{H}_0\} = Pr\left\{\sum_{m=n}^{n+M_1-1}\left\{\breve{w}_{d0}^2[m] + \frac{2\sigma_0^2}{N_0}\breve{w}_{do}[m]\right\} > \xi\right\} \tag{3.20}$$

$$P_D = Pr\{T > \xi; \mathcal{H}_1\} \tag{3.21}$$

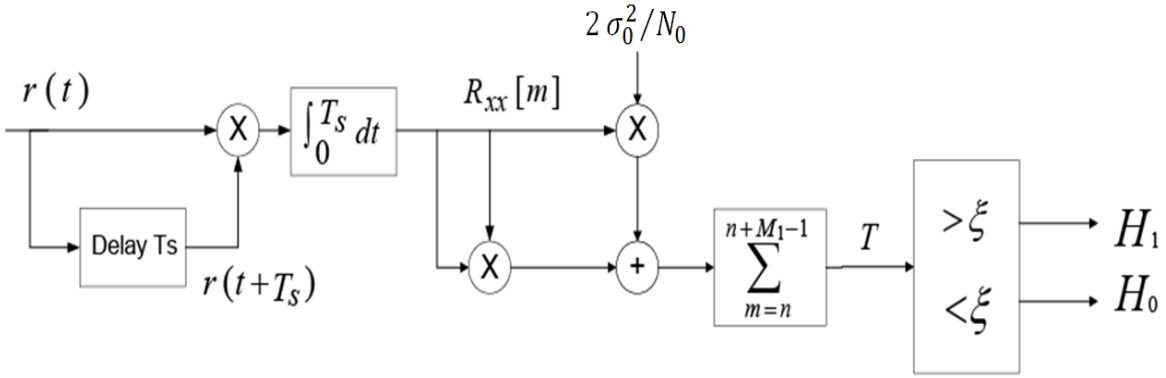


Figure 3.4 Block diagram of detection model for dirty template system

To find the false alarm and detection probabilities, we must first estimate the value of $(\sigma_1^2, \sigma_0^2, \mathcal{E}_r, N_0)$. Let's assume that the value of channel bandwidth B and symbol duration T_s are known and fixed at the receiver, where the bandwidth B is specified by the UWB standards approved by the FCC (e.g. $B \approx 5-7$ GHz for indoor applications) [1]. If we assume that the null hypothesis $\mathcal{H}_0$ is true, the value of (σ_0^2, N_0) can be estimated by applying the maximum likelihood estimation criteria, mentioned in the section 3.2.1, as follows:

$$\mathcal{H}_0: \qquad \hat{\sigma}_0^2 = \frac{1}{M_1}\sum_{m=n}^{n+M_1-1} R_{x,x}^2[m] = \frac{1}{M_1}\sum_{m=n}^{n+M_1-1}\breve{w}_{d0}^2[m] \tag{3.22}$$

Based on (3.17), the estimation of N_0 is:

$$\mathcal{H}_0: \qquad \hat{N}_0 = \sqrt{\frac{2\hat{\sigma}_0^2}{BT_s}} \tag{3.23}$$

Similarly, if we assume that the alternative hypothesis $\mathcal{H}_1$ is true, the value of $(\sigma_1^2, \mathcal{E}_r)$ can be estimated as follows:

$$\mathcal{H}_1: \quad \begin{cases} \widehat{\mathcal{E}}_r = \frac{1}{M_1}\sum_{m=n}^{n+M_1-1} R_{x,x}[m] \\ \hat{\sigma}_1^2 = \frac{1}{M_1}\sum_{m=n}^{n+M_1-1}\left(R_{x,x}[m] - \widehat{\mathcal{E}}_r\right)^2 \end{cases} \tag{3.24}$$

We have explained how to estimate each of $(\sigma_1^2, \sigma_0^2, \mathcal{E}_r, N_0)$. Thus, we can directly apply the GLRT test introduced in (3.19) and analyze the probabilities P_{FA} and P_D for different threshold values. Based on the analysis results, we will be able to create a table (or a curve) which represents the suitable threshold value ξ for each of a given P_{FA}.

3.3.1 Monte Carlo performance evaluation

In detection problem we wish to determine the probability that the test statistic T exceeds a given threshold. Unfortunately, the Neyman-Pearson test doesn't always produce a test statistic T with a know probability distribution. When we are unable to find the probability $Pr\{T > \xi\}$ by analytical ways, we have to find another way that permits us to derive the probability $Pr\{T > \xi\}$ under each hypothesis and to extract further the suitable threshold value ξ. For example, by returning to the dirty template detection problem and by looking at (3.19), we can notice that it is not easy to find the probability that the test statistic value T exceeds a threshold ξ ($Pr\{T > \xi\}$) by conventional analytical methods. For evaluating the properties of our detection system we resort an experimental method called *Monte Carlo computer simulation*. The Monte Carlo method, explained in [71], is used when it is impossible to solve the statistical and mathematical problems with using deterministic algorithms. Applying the Monte Carlo simulation to analyze the properties of the test statistic (3.19) which is achieved by respecting the following steps, the Matlab program for the algorithm is illustrated in Appendix 3.A:

1. Generate UWB pulse $p(t)$ (see section 2.4) and compute its variance $\{p(t)\}$.
2. Find the ambient noise variance $(N_0/2)$ for a selected value of SNR, which satisfies:

$$SNR = \frac{var\{p(t)\}}{N_0/2}$$

3. Generate two groups of independent random samples $R_{x,x}$ proposed in (3.16), the first group contains M_1 samples under the hypothesis $\mathcal{H}_0$, and the second contains M_1 samples under the hypothesis $\mathcal{H}_1$. In order to generate the samples $R_{x,x}$ we

assume that the system parameters T_s and B are known, and the captured energy $\mathcal{E}_r$ is normalized ($\mathcal{E}_r = 1$).

4. Compute the test statistic T under $\mathcal{H}_0$ and $\mathcal{H}_1$.
5. Repeat the steps (3 & 4) N times to yield $\{T_1, T_2, \dots, T_i \dots, T_N\}$ for each hypothesis.
6. Vary the threshold ξ and estimate for each value of ξ the probabilities of detection and false alarm, as follows:

$$\hat{P}_{FA} = \frac{\text{the number of } T_i \text{ under } \mathcal{H}_0 \text{ that exceed } \xi}{N}$$

$$\hat{P}_{D} = \frac{\text{the number of } T_i \text{ under } \mathcal{H}_1 \text{ that exceed } \xi}{N}$$

7. Repeat the procedure for different values of SNR and data-aided number M_1.

Actually, based on this experimental method, the generated probabilities of false alarm and detection are estimated probabilities $\{\hat{P}_{FA}, \hat{P}_D\}$. We can enhance the Monte Carlo simulation accuracy and reduce the estimation error by increasing the iteration number N. Hence, the choice of N will affect the simulation quality. For example, if the false alarm probability $P_{FA} = 10^{-6}$, it means that for 10^6 received samples we have the chance to produce a false alarm error about one only. In such a case the iteration number N have to be much larger than 10^6. So increasing N allows the estimated values of the results to converge to their correct values, but that will increase simultaneously computational complexity. For more details about how to determine the required value of N, see [69].

3.3.2 Simulation results

We plot the results of the Monte Carlo evaluation of $Pr\{T > \xi\}$ versus ξ in Figures 3.5, 3.6 and 3.7 for different values of SNR and data-aided number. The solid curve represents false alarm probability $Pr\{T > \xi; \mathcal{H}_0\}$ and the dashed curve represents detection probability $Pr\{T > \xi; \mathcal{H}_1\}$. From these figures we could intuitively determine the optimal value of the threshold ξ where the difference between the detection probability and the false probability is maximized ($max(P_D - P_{FA})|_\xi$). Furthermore, Figure 3.5 shows the interference area between the two curves, this area provides an indication on possible errors. When we reduce the false alarm probability by increasing ξ, the detection probability will be simultaneously reduced. Figure 3.6 and 3.7 show that by increasing the SNR or data aided number, the two curves are clearly separated and the overlap between the probabilities is vanished, leading to reduce the detection error probability and enhance the detection performance, but these improvements are

with the price of power efficiency or detector design simplification. Figure 3.7 shows that the area which separates between two curves, satisfies maximizing the difference $(P_D - P_{FA})$, and the optimal threshold value should be set inside this range. Figures 3.8, 3.9 and 3.10 depict histograms (which is equivalent to PDF within a scale) of the test statistic T, and respect the same example parameters proposed in Figures 3.5, 3.6 and 3.7 respectively. The detection performance improves when the distance between the histograms (PDF) increases. These figures assert that as SNR or M_1 increases the overlap between the histograms is reduced and the distance between PDF increases.

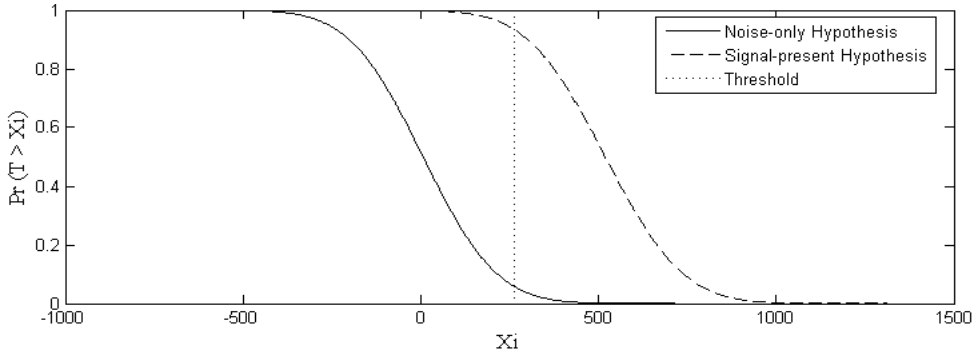


Figure 3.5 Monte Carlo simulation of $Pr\{T > \xi\}$ for SNR= -5 dB & $M_1 = 8$

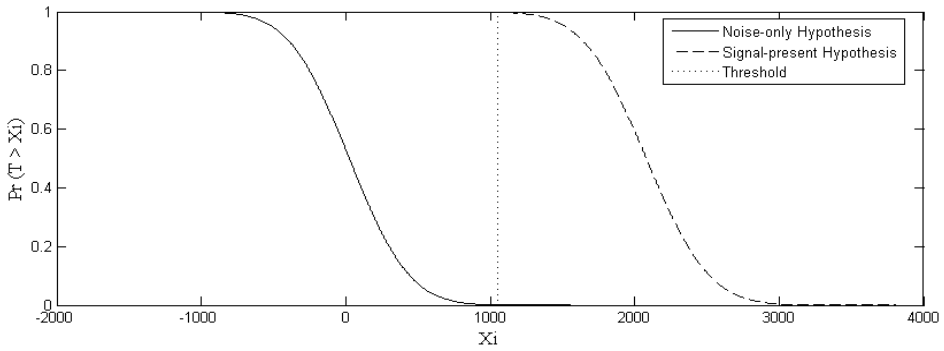


Figure 3.6 Monte Carlo simulation of $Pr\{T > \xi\}$ for SNR= -5 dB & $M_1 = 32$

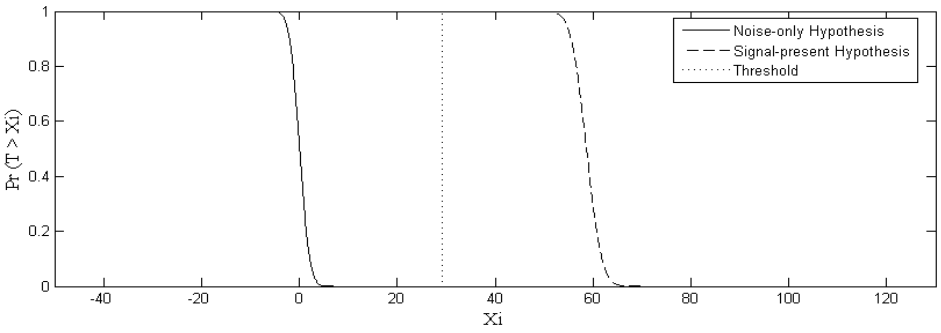


Figure 3.7 Monte Carlo simulation of $Pr\{T > \xi\}$ for SNR= 5 dB & $M_1 = 8$

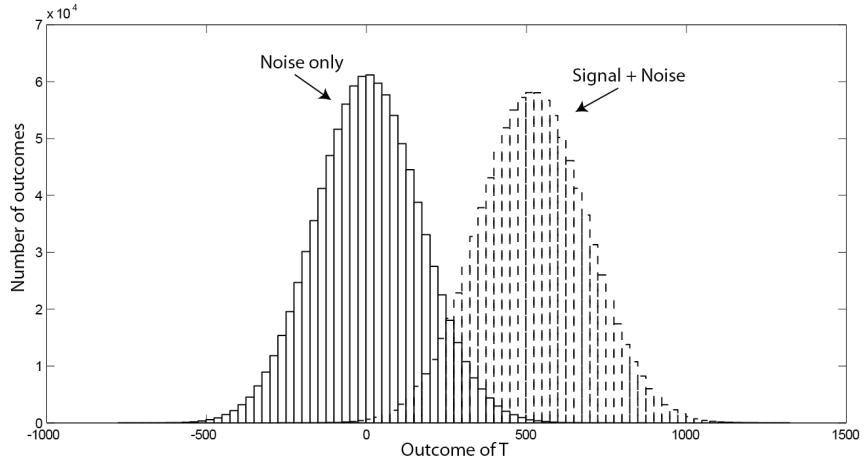


Figure 3.8 Histogram of T under two hypothese for SNR= -5 dB & $M_1 = 8$

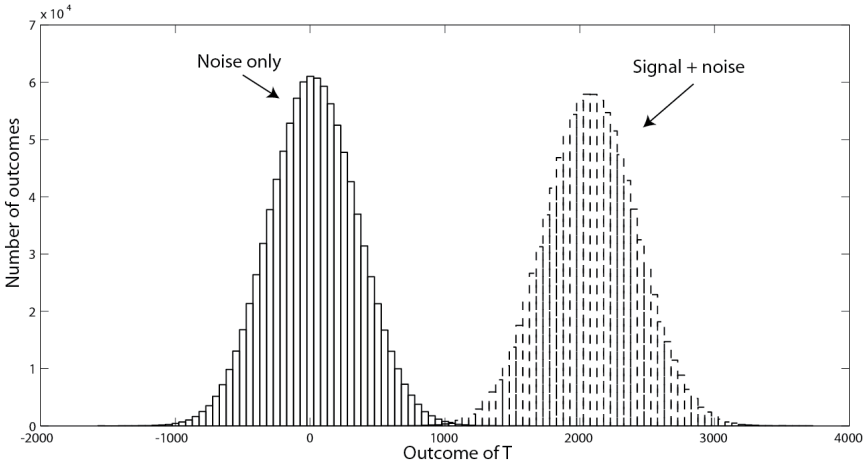


Figure 3.9 Histogram of T under two hypothese for SNR= -5 dB & $M_1 = 32$

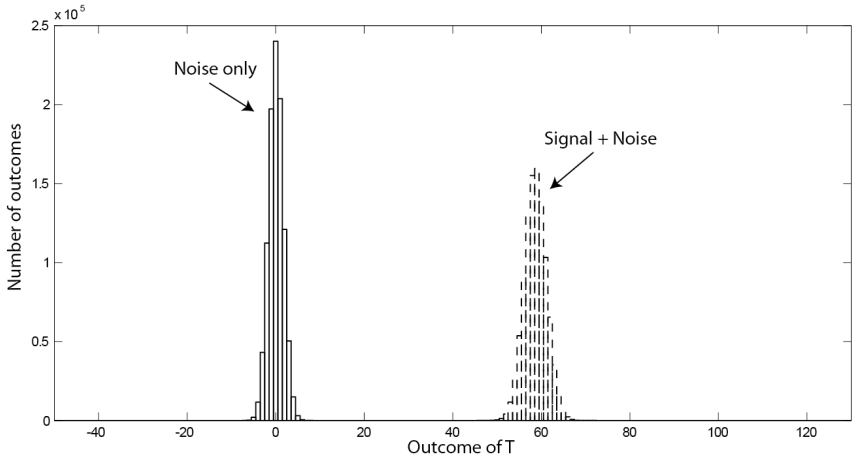


Figure 3.10 Histogram of T under two hypothese for SNR= 5 dB & $M_1 = 8$

Another way of representing the detection performance of a Neyman-Pearson detector is to plot the detection probability P_D versus the false alarm probability P_{FA}. Each point on the curve corresponds to a value of (P_{FA}, P_D) for a given threshold ξ. As we have already found when ξ increases, P_{FA} and P_D decrease. This type of plot is called *receiver operating characteristic* (ROC), shown in Figures 3.11 and 3.12. The ideal detector is when $P_D = 1$ for any value of P_{FA}. So it is clear that when the SNR or the data-aided number increases the ROC curve approaches to the ideal case and the detection performance is improved. In general, increasing the size of training symbols M_1 that will help to integrate the additive noise effects, and improve further detection performance, but that comes at the price of reduced channel capacity and power efficiency.

In this section, UWB signal detection has been carried out based on DT approach (Figure 3.4). Then, detection performance of the proposed method has been evaluated by Matlab simulation results (Figures 3.5 until 3.12). In this simulation, we select the pulse $p(t)$ as the third derivative of the Gaussian function with unit energy and duration $T_p \approx 1$ ns. Each symbol contains $N_f = 24$ frames with duration $T_f = 100$ ns. The Monte Carlo computer simulation repeats the experience $N = 1{,}000{,}000$ times.

Note: Based on the Monte Carlo simulation, we have been able to generate easily two groups which contain respectively million values for the test statistic T under the hypothesis $\mathcal{H}_0$ (for the first group) and the hypothesis $\mathcal{H}_1$ (for the first group). We don't have any idea about the probability distribution for each T group. For that, we decided to do a test to know if T samples have a standard normal distribution or not. *Kolmogorov-Smirnov* Test (K-S test) is one of the famous tests used for comparing result samples with a reference probability distribution [72, 73]. In Matlab, K-S test is easily done by using the following statement; for more details see Matlab help in [74]: H = kstest(X); where X is the input data vector which we want to know if its elements have a standard normal distribution or not. The output test H gives only two cases, either 1: it means that we reject the hypothesis that X has a standard normal distribution; or 0: it means that we accept the hypothesis that X has a standard normal distribution with risk of error equal 5%. For applying correctly K-S test, the input vector X must be standardized (with $\mu = 0$ and $\sigma^2 = 1$, where μ and σ^2 are the mean and the variance of X respectively). Let's assume that X isn't standard, to make X be standard we apply the following transformation: $Z = (X - \mu)/\sigma$. Let's now apply the K-S test by replacing X by standardized elements of T, as the following statement: H = kstest((T-mean(T)) / std(T)); where std denotes standard deviation function (σ). We have applied the K-S test using Matlab for each group of T, and the test result has been H=0 for both groups. Consequently, we could approximately consider that the test statistic T for $\mathcal{H}_0$ and $\mathcal{H}_1$has a normal distribution with an error risk less than 5%. The advantage of the K-S test results is: We can directly find the probabilities of false alarm and detection and set a suitable threshold based on mathematical

method (Gaussian properties) without the need to look up the results on the Monte Carlo tables or curves.

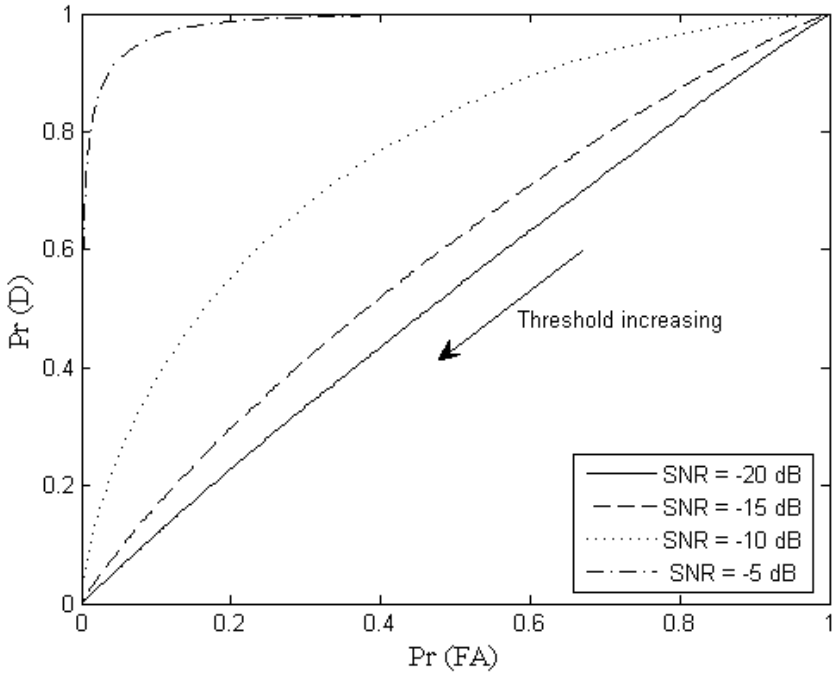


Figure 3.11 ROC for data aided number =8

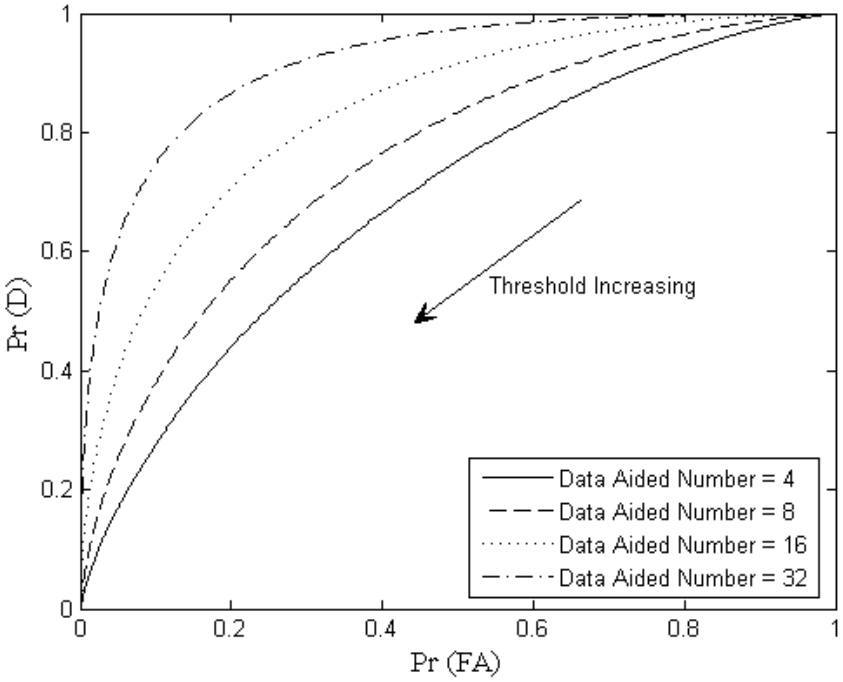


Figure 3.12 ROC for SNR= -10 dB

3.3.3 Symbol-level timing offset estimation

As we have said before, to make a decision that we receive an UWB signal, we send a training sequence composed of $M_1 + 1$ symbols equal 1. When the test statistic $T(n)$ goes over the threshold the signal detection occurs and we move on to the next step of the synchronization (timing acquisition) and to demodulate later information-bearing symbols.

From (3.20) and (3.21), we know the test statistic characters under the hypothesis $\mathcal{H}_0$ as well as under the hypothesis $\mathcal{H}_1$ respectively. Nevertheless there is a transition between these two hypotheses. During this transition from $\mathcal{H}_0$ to $\mathcal{H}_1$, the probability density function of $T(n)$ changes gradually from the first case (3.20) to the second (3.21), as shown in figure 3.13 [75].

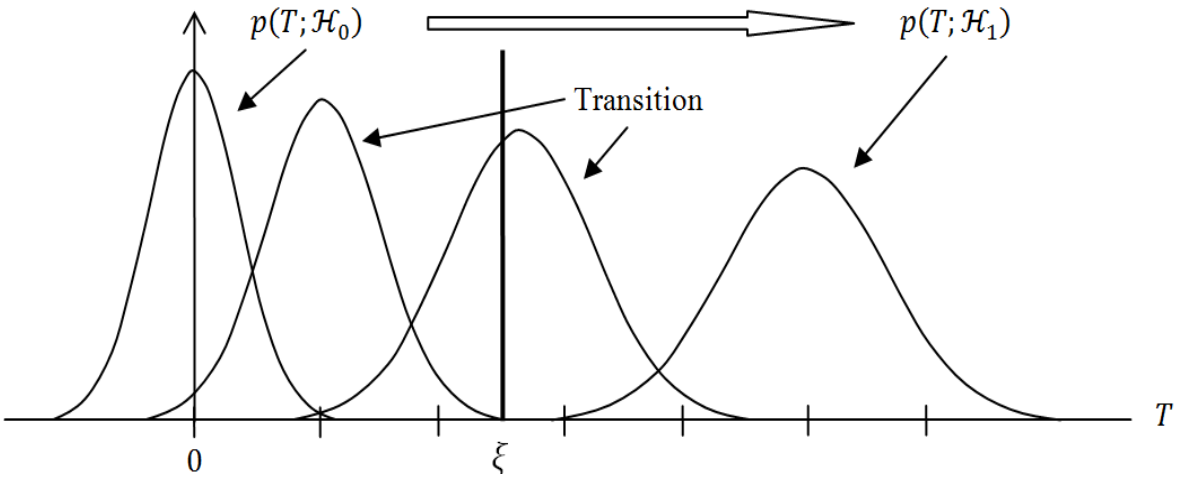


Figure 3.13 Transition of the PDF of T

What could happen if the decision occurs before the $M_1 + 1$ training sequence symbols are completely received? This means if we execute directly the timing acquisition algorithm and demodulate further the information symbols after the detection, we might commit errors and that will impact negatively system performance and increase bit error rate BER. Because there is an undesirable part of the detection training symbols which is used to carry out the synchronization as well as the demodulation. So what we must do to be sure that after the signal detection we will receive the entire detection training sequence before moving on to the next system stage. To avoid the influence of the detection data-aided bits on the subsequent stages, we separate between the detection data-aided symbols and the following information-bearing symbols (or acquisition training sequence symbols, see it later in the next section) by three-symbol pattern $(-1, -1, +1)$, at least. After the signal detection is done, we keep observing test statistic value $T(n)$ for the following symbols $R_{x,x}[n]$ until we arrive to the first maximum value of $T(n)$. At this point $n = \hat{n}_s$ we said that the $M_1 + 1$ data-aided sequence has

completely been received and the symbol-level timing offset $\hat{n}_s$ is determined. The three-symbol pattern permits to exhibit a unique maximum peak at $n = \hat{n}_s$ regardless of the values of subsequent information-bearing symbols. Thus, after detecting the signal, the symbol-level timing offset $\hat{n}_s$ mentioned in (2.19) can be estimated via a line search to maximize the test statistic $T(n)$, as show below:

$$\hat{n}_s = arg \max_n T(n); \qquad T(n) = \sum_{n=m}^{m+M_1-1} \left(R_{x,x}^2[m] + \frac{2\sigma_0^2}{N_0} R_{x,x}[m] \right) \tag{3.25}$$

Therefore, the detection problem for dirty template UWB systems can be devised into two steps. The first step applies the test statistic inequality expression proposed in (3.19). If the received sample goes over the threshold, the signal detection is declares and we move on to the second step which in its turn applies the line search algorithm proposed in (3.25) to estimate the symbol-level timing offset $\hat{n}_s$.

3.3.4 **Practical considerations**

In practice, to understand how we can carry out the detection stage and decide the present of the UWB signal. We must first estimate N_0 and σ_0^2, where they aren't used only for determining the threshold, but also for carrying out the detection test circuit in figure 3.4. To estimate them, we suppose that when the receiver is turned on, we don't transmit any signal for a certain period and keep the receiver listening to the environment during this noise-only period. Under this assumption, the system can estimate each of N_0 and σ_0^2 by applying the maximum likelihood estimation mentioned in (3.22 and 3.23). The variance of the noise samples is written: $\hat{\sigma}_0^2 = (1/N_l) \times \sum_{n=0}^{N_l-1} \breve{w}_{d0}^2[n]$, where N_l denotes the period where we don't transmit any signal since the receiver has been turned on. As N_l is large the variance estimation is more accurate.

To set the threshold ξ we suggest three approaches:

- In the first method, selecting the threshold value is based on the analysis results derived in the figures 3.5 until 3.12. The equation (3.20) asserts that the threshold value relays on P_{FA} as well as the variables N_0, σ_0^2 and M_1. Since the variables N_0 and σ_0^2 have been estimated, the threshold can be set for a give value of P_{FA} and M_1. The advantage of this method is: selecting threshold doesn't need to any knowledge of the probability P_D. So the initial noise-only period (listening period) is sufficient to effectively compute the suitable threshold for a given P_{FA}.

- The second way says that before the signal detection system runs on, for the first time, the system is passed through an initial period of adjustment, in which the user sends train of data-

aided symbols during this learning period and by utilizing temporarily a feedback loop, the threshold ξ is adjusted where the difference between the detection probability and the false alarm probability is maximized ($max(P_D - P_{FA})|_{\xi}$). So in this method we resort to the learning period for estimating the value of $(\sigma_1^2, \sigma_0^2, \mathcal{E}_r, N_0)$ and evaluating further the false alarm probability as well as the detection probability for various threshold values. The learning stage requires a relatively long time to be achieved, and it doesn't take into consideration the variation of channel parameters over time and its effects on $\mathcal{E}_r$ and P_D.

- The last method works efficiently when we are sure that there is the Line Of Sight (LOS) between the transmitter and the receiver. As we have seen until now, evaluating of the false alarm probability can easily occur during the listening period. But for evaluating the detection period, we must before know the captured energy samples $\mathcal{E}_r$. In fact, the value of $\mathcal{E}_r$ relays on the transmitted pulse $u(t)$, the communication channel $h(t)$ and also the antenna & receiver response. So to find $\mathcal{E}_r$ we must a priori estimate unknown channel parameters. Unfortunately, it is not efficient (and easy) to estimate the UWB multipath channel. The third method suggests that instead to calculate the detection probability P_D which is associated directly with the entire captured energy $\mathcal{E}_r$, we can find at least the minimum Detection Probability $P_{D\,min}$ which depends only upon the received energy value produced by the direct channel ray (LOS). Thus we need only to estimate the attenuation due to the direct channel path and neglect the effects of the other channel paths. Usually the attenuation factor is estimated for the maximum possible distance between the transmitter and the receiver (≈ 10m for indoor UWB applications). Therefore, this method employs also the listening period to estimate N_0 and σ_0^2. Mathematically, the maximum attenuation due to the direct channel path LOS is calculated, and its corresponding received energy sample is derived. Consequently, the threshold ξ is set where the difference between the minimum detection probability and the false alarm probability is maximized ($max(P_{D\,min} - P_{FA})|_{\xi}$). Even if the other channel paths are present, that would not deteriorate the signal detection operation; on the contrary, the channel paths would contribute to increase the received signal power and to move the detection probability curve far away from the false alarm probability area.

Once the threshold is set, we would be able to make the detection circuit in (3.19) work as follows: First we must collect M_1 observed samples of $R_{x,x}[n]$ and produce the corresponding test statistic $T(n)$. Next, we compare $T(n)$ with the selected threshold ξ, if T is bigger than ξ we declare that the signal is present; otherwise the signal is absent and we must update the observed sample pattern. Then we reproduce $T(n)$ and compare it again with ξ. We repeat the comparison operation each T_s period, until $T(n)$ goes over ξ. In this case we will be sure of the presence of signal and we go to estimate $\hat{n}_s$ based on (3.25). Figure 3.14 presents the flow chart of signal detection scheme.

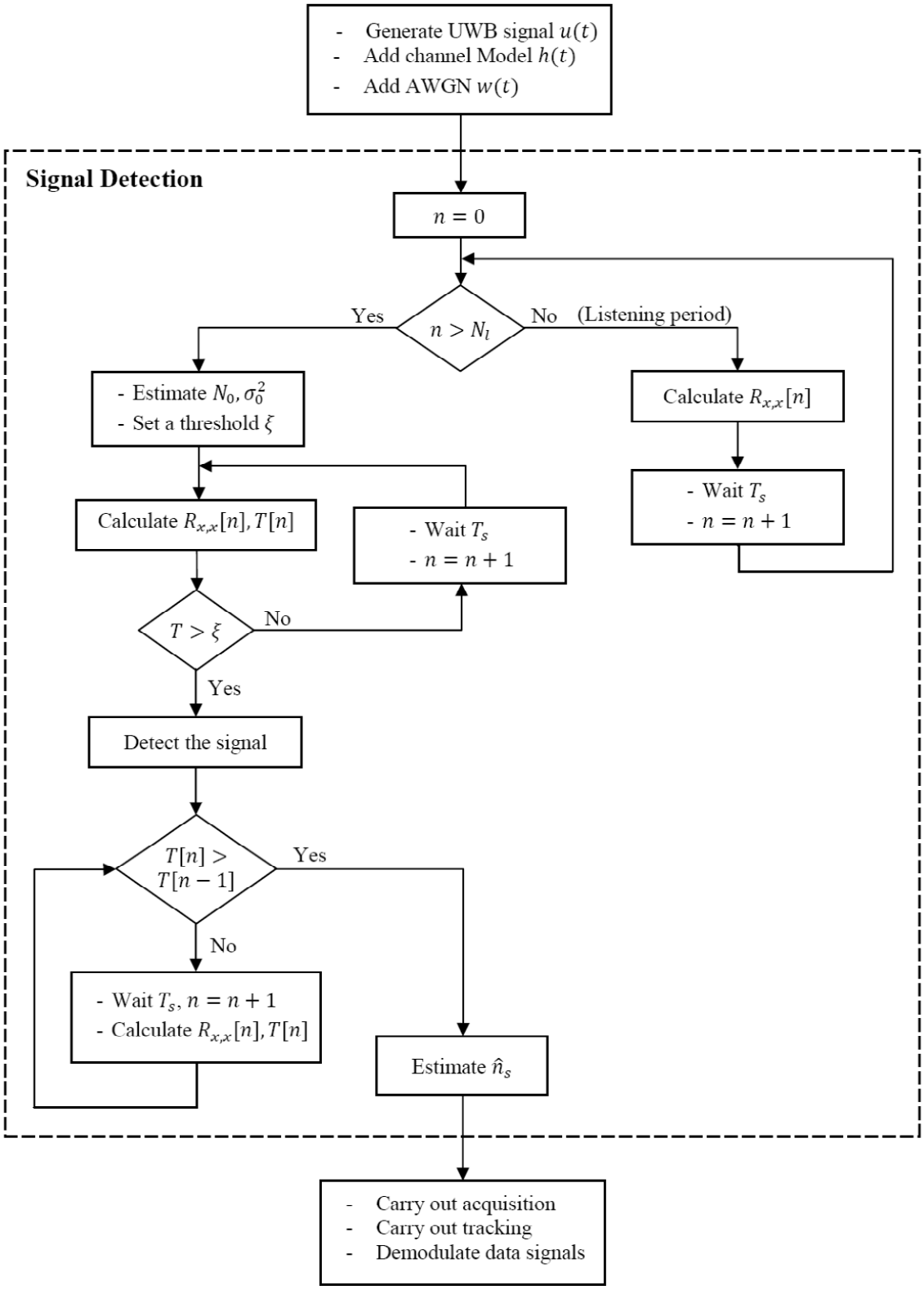


Figure 3.14 Flow chart of signal detection scheme

As a result, The dirty template detection can be applied to UWB receiver even in the presence of time hopping TH codes or Inter-Frame Interference (IFI), because the received segment and its dirty template contain the same TH codes and IFI properties regardless of the unknown channel characteristics (e.g. unknown time offset), but with the condition of the absence of ISI. Moreover, this method exploits the rich multipath diversity provided by UWB channels, and does not need to neither estimate the propagation channel nor generate clean correlation template at the receiver. Consequently, it reduces receiver complexity with high detection accuracy. Although the presence of "noise-cross-noise" term affects the detection performance, this limitation could be overcome either by using relatively large number of data-aided symbols, or by increasing the distance between the received signal and its dirty template, but that is at the price of bandwidth efficiency or receiver complexity.

3.4 Conclusion

In this chapter, we have presented a dirty template approach used for achieving rapid, accurate and low-complexity signal detection and symbol-level timing synchronization. The detection system is employed to determine if the desired UWB signal is present or not. The dirty template detector is derived by applying the Neyman-Pearson theory. Then, Monte Carlo simulations are performed to find the probabilities of false alarm (P_{FA}) and detection (P_D), in order to select the optimal threshold value. The results are analyzed in three ways; a graph of P_D & P_{FA} versus the threshold, histograms of test statistic under each hypothesis and a graph of the ROC curves. Analyses show that the detection performance of the dirty template approach is improved by increasing the signal-to-noise ratio (SNR) or the number of data-aided (DA). But these improvements reduce the power efficiency and complexity the receiver design. After detecting the UWB signal, the symbol-level timing offset estimation relies on searching the best test statistic value T.

The described technique can be applied on UWB systems even in the presence of time hopping TH codes, Inter-Frame Interference (IFI) and rich multipath environment, but Inter-Symbol Interference (ISI) is absent.

Chapter 4

Timing Acquisition

After detecting the desired UWB signal, and estimating the symbol-level timing offset, we move on to search suitable synchronizers that permit finer timing resolution for UWB systems. Timing synchronization is considered as one of the most stringent requirements in UWB transmission systems, because the transmitted pulses are narrow and have low power density under the noise floor. We propose the timing with dirty template (TDT) approach as a promising candidate for achieving rapid, accurate and low-complexity timing synchronization [68]. Some research on dirty template synchronization has been reported in [17, 27, 64, 68 and 76]. All these studies are based on Mean Square techniques for deriving the TDT estimator.

In this chapter, we describe the dirty template (DT) technique in order to develop and test timing algorithms in both modes: data aided (DA) and non-data aided (NDA) modes. For studying the quality and precision of timing estimators, it is important to first of all present fundamental performance limits. Therefore, we use the Cramer-Rao, described by [77, 78], which represents a lower bound of the error variance of these estimators. Next, the TDT acquisition estimator is achieved by employing maximum likelihood (ML) and mean-square algorithms. Then the performance of this estimator is improved by modifying the structure of the cross-correlation operation. Simulation results show timing estimation performance for DA and NDA modes, and confirm the high performance and fast timing acquisition of DA mode, compared to NDA mode, but with less bandwidth efficiency.

4.1 Timing modeling with dirty template

Assuming that the received signal is detected correctly and the optimal value of $\hat{n}_s$ is estimated. So the timing offset in (2.19) becomes: $\tau_0 = \tau \in [0, T_s)$. As we have said in (2.21), under mistiming ($\tau_0 \neq 0$), any T_s-long received segment of $r(t)$ can be represented by parts of two consecutive symbols, see Figure 4.1, as bellow:

$$x(t+nT_s) = w(t+nT_s) + \begin{cases} \sqrt{\mathcal{E}_s}s[n-1]p_R(t+T_s-\tau_0): t \in [0,\tau_0) \\ \sqrt{\mathcal{E}_s}s[n]p_R(t-\tau_0) \quad\quad\quad : t \in [\tau_0, T_s) \end{cases} \tag{4.1}$$

The timing offset τ_0 will be estimated using dirty template algorithms. The original timing with dirty template (TDT) method proposed in [17], is summarized as follows: in the absence of inter-symbol interference (ISI), the estimation of timing offset error (TOE) can be achieved for data-aided (DA) and non-data-aided (NDA) modes, depending on the cross-correlation between the pairs of successive received segments of duration T_s. The correlation output samples are done as shown below:

$$R_{x,x}[n] = \int_0^{T_s} x(t+nT_s)\cdot x(t+(n+1)T_s)\,dt \tag{4.2}$$

Substituting (4.1) in (4.2), we can get:

$$R_{x,x}[n] = \int_0^{T_s}\left\{w(t+nT_s) + \sqrt{\mathcal{E}_s}\sum_{m=0}^{1} s[n-m]p_R(t+mT_s-\tau_0)\right\}\times$$
$$\left\{w(t+(n+1)T_s) + \sqrt{\mathcal{E}_s}\sum_{m=0}^{1} s[n-m+1]p_R(t+mT_s-\tau_0)\right\}dt$$

$$R_{x,x}[n] = \breve{\omega}[n] + A\int_0^{T_s} p_R^2(t+T_s-\tau_0)\,dt + B\int_0^{T_s} p_R^2(t-\tau_0)\,dt$$

Hence:

$$R_{x,x}[n] = \breve{\omega}[n] + A\int_{T_s-\tau_0}^{T_s} p_R^2(t)\,dt + B\int_0^{T_s-\tau_0} p_R^2(t)\,dt \tag{4.3}$$

where $A = \mathcal{E}_s s[n-1].s[n]$, $B = \mathcal{E}_s s[n].s[n+1]$, the possible values of *A* and *B* are exhibited in Table 2.2. Appendix 2.B asserts that the sampled noise $\breve{\omega}[n]$ in (4.3) could be approximately treated as zero-mean Gaussian noise:

$$\breve{\omega}[n] \sim \mathcal{N}(0,\sigma_w^2) \quad : \sigma_w^2 = \frac{N_0^2}{2}BT_s + N_0\mathcal{E}_s\mathcal{E}_{max} \tag{4.4}$$

where $\mathcal{E}_{max} = \int_0^{T_s} p_R^2(t)\,dt$ represents T_s-long received segment energy. Considering the estimation of unknown deterministic τ_0 based upon M_2 independent and identically distributed (IID) observations $\{R_{x,x}[0,\tau_0], R_{x,x}[1,\tau_0], \ldots, R_{x,x}[M_2-1,\tau_0]\}$. These observation samples described in (4.3) could be represented as τ_0-conditioned variables, as the following:

$$R_{x,x}[n,\tau_0] = \breve{\omega}[n] + A_n\int_{T_s-\tau_0}^{T_s} p_R^2(t)\,dt + B_n\int_0^{T_s-\tau_0} p_R^2(t)\,dt \tag{4.5}$$

where $n \in [0, M_2-1)$. Since $\breve{\omega}[n]$ is approximated by a Gaussian signal with zero mean, the conditional density function of these M_2 observations would be given by:

$$p_{R|\tau_0}\left(R_{x,x}[\tau_0]|\tau_0\right) = \left(\frac{1}{\sqrt{2\pi\sigma_w^2}}\right)^{M_2} exp\left(\frac{-1}{2\sigma_w^2}\sum_{n=0}^{M_2-1}\left(R_{x,x}[n,\tau_0] - \left\{A_n\int_{T_s-\tau_0}^{T_s} p_R^2(t)\,dt + B_n\int_0^{T_s-\tau_0} p_R^2(t)\,dt\right\}\right)^2\right) \quad (4.6)$$

The purpose of this chapter is to derive an appropriate TDT estimator for DA and NDA modes. In order to study the quality and precision of any timing offset estimator, it is important to first of all present fundamental performance limits. Therefore, we use the Cramer-Rao evaluation, described by [77, 78], which represents a lower bound of the error variance of all estimators. The analysis of Cramer-Row lower bounds (CRLBs) for traditional systems is well founded in [79-87], but for dirty-template UWB systems, to our knowledge, there is no systematic work reported yet. Evaluating of the CRLB for DT-UWB is mathematically demonstrated in the next section.

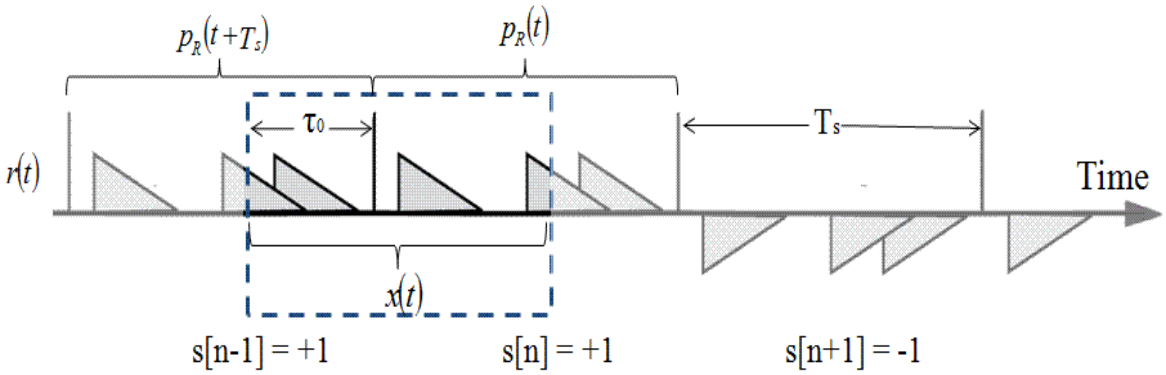


Figure 4.1 T_s-long observed received symbol $x(t)$

4.2 Cramer-Rao lower bounds

Suppose an estimate $\hat{\tau}_0$ of the time offset can be generated from (4.5). The estimation error variance is then bounded by the CRLB [77]:

$$var(\hat{\tau}_0) \geq CRLB(\tau_0) = \frac{\left(\frac{d}{d\tau_0}E(\hat{\tau}_0)\right)^2}{I(\tau_0)} \quad (4.7)$$

where $I(\tau_0)$ is the *Fisher Information* (FI) given by following:

$$I(\tau_0) = -E\left\{\frac{\partial^2}{\partial\tau_0^2} ln\left(p_{R|\tau_0}\left(R_{x,x}[\tau_0]|\tau_0\right)\right)\right\} \quad (4.8)$$

Assuming the estimate $\hat{\tau}_0$ of the time delay τ_0 is an unbiased, it means: $E\{\hat{\tau}_0\} = \tau_0 \Rrightarrow \frac{d}{d\tau_0} E\{\hat{\tau}_0\} = 1$, then the estimation error variance in (4.7) becomes:

$$var(\hat{\tau}_0) \geq CRLB(\tau_0) = \frac{1}{I(\tau_0)} \tag{4.9}$$

Substituting (4.6) in (4.8), we can get

$$I(\tau_0) = -E\left\{\frac{\partial^2}{\partial \tau_0^2}\left(-M_2.ln\left(\sqrt{2\pi\sigma_w^2}\right) - \frac{1}{2\sigma_w^2}\sum_{n=0}^{M_2-1}(f(n,\tau_0))^2\right)\right\} \tag{4.10}$$

where:

$$f(n,\tau_0) = R_{x,x}[n,\tau_0] - \left\{A_n \int_{T_s-\tau_0}^{T_s} p_R^2(t)\,dt + B_n \int_0^{T_s-\tau_0} p_R^2(t)\,dt\right\} \tag{4.11}$$

Hence:

$$I(\tau_0) = -E\left\{\frac{\partial}{\partial \tau_0}\left(\frac{1}{\sigma_w^2}\sum_{n=0}^{M_2-1}\left(f(n,\tau_0)\frac{\partial}{\partial \tau_0}\left(A_n \int_{T_s-\tau_0}^{T_s} p_R^2(t)\,dt + B_n \int_0^{T_s-\tau_0} p_R^2(t)\,dt\right)\right)\right)\right\} \tag{4.12}$$

By using the following expression: $\frac{d}{dx}\int_0^x g(t)\,dt = g(x)$, the two integration operations in (4.12) are given as follows:

- $\frac{\partial}{\partial \tau_0}\int_0^{T_s-\tau_0} p_R^2(t)\,dt = -p_R^2(T_s-\tau_0)$ (4.13)
- $\frac{\partial}{\partial \tau_0}\int_{T_s-\tau_0}^{T_s} p_R^2(t)\,dt = \frac{\partial}{\partial \tau_0}\left(\mathcal{E}_{max} - \int_0^{T_s-\tau_0} p_R^2(t)\,dt\right) = p_R^2(T_s-\tau_0)$ (4.14)

Substituting (4.13) and (4.14) in (4.12), we can get

$$I(\tau_0) = -E\left\{\frac{\partial}{\partial \tau_0}\underbrace{\left(\frac{1}{\sigma_w^2}\sum_{n=0}^{M_2-1} f(n,\tau_0)(A_n - B_n)p_R^2(T_s-\tau_0)\right)}_{\frac{\partial}{\partial \tau_0} ln\left(p_{R|\tau_0}(R_{x,x}[\tau_0]|\tau_0)\right)}\right\} \tag{4.15}$$

$$I(\tau_0) = -E\left\{\frac{1}{\sigma_w^2}\sum_{n=0}^{M_2-1}\left(-(A_n-B_n)^2 p_R^4(T_s-\tau_0) + 2(A_n-B_n)p_R(T_s-\tau_0)\dot{p}_R(T_s-\tau_0) \times f(n,\tau_0)\right)\right\}$$

where $\dot{p}_R(T_s-\tau_0)$ denotes partial differentiation with respect to τ_0.

$$I(\tau_0) = \frac{1}{\sigma_w^2}\sum_{n=0}^{M_2-1}\Bigg(p_R^4(T_s-\tau_0)E\{(A_n-B_n)^2\} - 2p_R(T_s-\tau_0)\dot{p}_R(T_s-\tau_0)E\{A_n-B_n\}\times E\Bigg\{\underbrace{R_{x,x}[n,\tau_0] - \Big\{A_n\int_{T_s-\tau_0}^{T_s} p_R^2(t)\,dt + B_n\int_0^{T_s-\tau_0} p_R^2(t)\,dt\Big\}}_{=\breve{\omega}[n]\ \ Gaussien\ Noise\sim\mathcal{N}(0,\sigma_w^2)}\Bigg\}\Bigg)$$

where $E\{\breve{\omega}[n]\} = 0$. Hence:

$$I(\tau_0) = \frac{1}{\sigma_w^2}p_R^4(T_s-\tau_0)\cdot\sum_{n=0}^{M_2-1}E\{(A_n-B_n)^2\} \tag{4.16}$$

$$I(\tau_0) = \frac{1}{\sigma_w^2}p_R^4(T_s-\tau_0)\cdot\sum_{n=0}^{M_2-1}E\{{A_n}^2 + {B_n}^2 - 2A_nB_n\} \tag{4.17}$$

From (4.3) and (4.17), we can get:

$$I(\tau_0) = \frac{1}{\sigma_w^2}p_R^4(T_s-\tau_0)\cdot\sum_{n=0}^{M_2-1}(2\mathcal{E}_s - 2E\{A_nB_n\}) \tag{4.18}$$

Assuming $p_R^4(T_s-\tau_0)\neq 0$, and substituting (4.18) in (4.9), we can compute the Cramer-Rao lower bound as follows:

$$var(\hat{\tau}_0) \geq CRLB(\tau_0) = \frac{\sigma_w^2}{2p_R^4(T_s-\tau_0)\cdot\sum_{n=0}^{M_2-1}(\mathcal{E}_s-E\{A_nB_n\})} \tag{4.19}$$

Based on the inequality expression (4.19), let's then develop the Cramer-Rao Lower Bound (CRLB) for Data Aided (DA) and Non-Data Aided (NDA) modes.

4.2.1 CRLB for data-aided mode

In DA mode, the best form of training symbols used for achieving fast acquisition is as follows [17]:

$$s[n] = \begin{cases} -1, & if(n \bmod 4) = 0, or, 1 \\ +1, & if(n \bmod 4) = 2, or, 3 \end{cases} \quad : n \in [0, M_2] \tag{4.20}$$

From (4.20), we find clearly that $s[n-1] = -s[n+1]$. By looking at (4.3) and Table 2.2, we find $A_n = -B_n$. And (4.19) becomes:

$$\text{DA}: \quad var(\hat{\tau}_0) \geq CRLB(\tau_0) = \frac{\sigma_w^2}{4M_2\mathcal{E}_s p_R^4(T_s-\tau_0)} \tag{4.21}$$

4.2.2 CRLB for non-data-aided mode

In NDA mode, the received signal doesn't contain the training symbols. So the information-bearing symbols $\{s[n]: n \in [0, M_2]\}$ are considered as independent random variables. From (4.3), we have $E\{A_n B_n\} = E\{A_n\}.E\{B_n\} = 0$ (see Table 2.B.1 in Appendix 2.B). Thus, the equation (4.19) is developed as follows:

NDA: $$var(\hat{\tau}_0) \geq CRLB(\tau_0) = \frac{\sigma_w^2}{2M_2 \mathcal{E}_s p_R^4(T_s - \tau_0)} \quad (4.22)$$

Comparing between (4.20) and (4.22), we find that $CRLB(DA) < CRLB(NDA)$. Figure 4.2 presents the influence of the signal-to-noise ratio (SNR) on the CRLBs for DA (*solid lines*) and NDA (*dashed lines*) modes. To simplify, the simulation results are normalized by $\mathcal{E}_s p_R^4(T_s)$ and calculated at timing offset $\tau_0 = 0$. Clearly, the CRLBs are inversely proportional to SNR. We can see also that the increase of the observation number M_2 leads to decrease CRLB curves and enhance synchronization performance (trading-off with computational complexity). These results confirm also that the CRLBs for DA synchronizers are generally better (fewer) than those for NDA mode. Furthermore, other factors that may affect CRLBs have been neglected, for example: TH codes, inter-symbol interference (ISI), UWB pulse shape and the channel parameters.

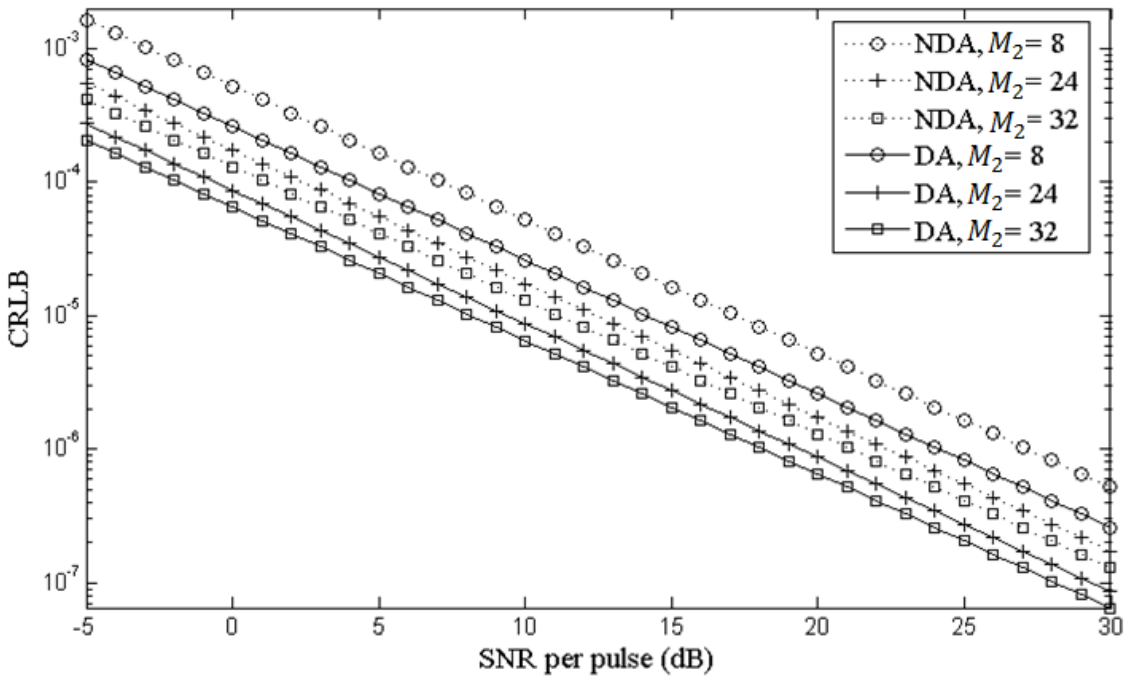


Figure 4.2 CRLB vs. SNR per pulse for DA and NDA modes

4.3 Timing offset estimator using ML approach

In this section, we will use Maximum Likelihood (ML) approach for estimating $\tau_0 \in [0, T_s)$ in both data-aided and non-data-aided modes. Maximum Likelihood estimator is the most popular approach for obtaining practical estimator which could be considered approximately as the minimum variance unbiased (MVU) estimator [77]. The Maximum Likelihood estimator (MLE) is defined to be the value of τ_0 that maximizes $p_{R|\tau_0}\big(R_{x,x}[\tau_0]|\tau_0\big)$. To extract the MLE algorithm for a DT system, we first rewrite $R_{x,x}[n,\tau_0]$ in (4.5) as:

$$R_{x,x}[n,\tau_0] = \breve{\omega}[n] + A_n \int_{T_s-\tau_0}^{T_s} p_R^2(t)\,dt + B_n\left(\mathcal{E}_{max} - \int_{T_s-\tau_0}^{T_s} p_R^2(t)\,dt\right) \tag{4.23}$$

$$R_{x,x}[n,\tau_0] = \breve{\omega}[n] + (A_n - B_n)\gamma(\tau_0) + B_n\mathcal{E}_{max} \text{ : where } \gamma(\tau_0) = \int_{T_s-\tau_0}^{T_s} p_R^2(t)\,dt \tag{4.24}$$

And $f(n,\tau_0)$ in (4.11) could be also rewritten as:

$$f(n,\tau_0) = R_{x,x}[n,\tau_0] - (A_n - B_n)\gamma(\tau_0) - B_n\mathcal{E}_{max} \tag{4.25}$$

Considering the likelihood function $p_{R|\tau_0}\big(R_{x,x}[\tau_0]|\tau_0\big)$ given in (4.6), we try to find an estimate $\hat{\tau}_0$ which maximizes $p_{R|\tau_0}\big(R_{x,x}[\tau_0]|\tau_0\big)$, by satisfying the following equation:

$$\frac{\partial}{\partial\tau_0} ln\left(p_{R|\tau_0}\big(R_{x,x}[\tau_0]|\tau_0\big)\right) = 0 \tag{4.26}$$

This derivation has already been derived in (4.15), as:

$$\frac{\partial}{\partial\tau_0} ln\left(p_{R|\tau_0}\big(R_{x,x}[n,\tau_0]|\tau_0\big)\right) = \frac{1}{\sigma_w^2}\sum_{n=0}^{M_2-1} f(n,\tau_0)(A_n - B_n)p_R^2(T_s - \tau_0) = 0 \tag{4.27}$$

Assuming $\frac{1}{\sigma_w^2}p_R^2(T_s - \tau_0) \neq 0$, so the estimate $\hat{\tau}_0$ have to satisfy the following condition:

$$\sum_{n=0}^{M_2-1} f(n,\tau_0)(A_n - B_n) = 0 \tag{4.28}$$

Substituting (4.25) in (4.28):

$$\sum_{n=0}^{M_2-1}\{(A_n - B_n)R_{x,x}[n,\tau_0] - (A_n - B_n)^2\gamma(\tau_0) - (A_n - B_n)B_n\mathcal{E}_{max}\} = 0 \tag{4.29}$$

Let's now find the ML estimator for DA and NDA modes.

4.3.1 Maximum likelihood estimator for data-aided mode

Appendixes 4.A and 4.B derive two forms of timing offset estimator for DA mode by applying ML algorithms. These estimators have likely the same estimation performance (estimator error variances). Therefore, we will present both estimator versions, but we will analyze the performance of only one of them. The maximum likelihood estimator (MLE) which is derived by Appendix 4.A is given as:

$$\hat{\tau}_0(\mathrm{DA}) = \frac{\mathcal{E}_{max}}{\left(p_R^2(T_s)+p_R^2(T_s-\tau_0)\right)}\left(1 - \frac{1}{\mathcal{E}_s\mathcal{E}_{max}}\sqrt{-\frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}[n,\tau_0]\cdot R_{x,x}[n+1,\tau_0]}\right) \tag{4.30}$$

Figure 4.3 presents the comparison between the normalized variance of ML estimator (*dashed lines*) extracted from (4.30) and CRLB (*solid lines*) extracted from (4.21). Supposing $\tau_0 = 0$, $\mathcal{E}_{max} = 1$ and normalizing the simulation according to $p_R^2(T_s)$. Figure 4.3 shows that CRLB is always lower than the estimation error variance for different values of SNR or M_2. Clearly, the estimation variance is inversely proportional to SNR, and converges on the corresponding CRLB curve with increasing SNR. We can also see that an increase in the number of observations (M_2) leads to a decrease in the variance and an improvement in the estimation performance.

But the parameters $\mathcal{E}_{max}$ and $p_R^2(T_s-\tau_0)$ in (4.30) are unknown. Intuitively, the $\hat{\tau}_0$ estimation occurs at the minimum point of the equation (4.30) for different values of $\tau \in [0, T_s)$, it means: for finding the optimal estimated value of $\hat{\tau}_0$ we vary the controllable variable τ inside the range $[0, T_s)$. The value of τ which allows minimizing the following expression is considered as the desired estimate of timing offset:

$$\hat{\tau}_0(\mathrm{DA}) = \underset{\tau}{arg\,min}\ \frac{\mathcal{E}_{max}}{\left(p_R^2(T_s)+p_R^2(T_s-\tau_0)\right)} \times \left(1 - \frac{1}{\mathcal{E}_s\mathcal{E}_{max}}\sqrt{-\frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}[n,\tau_0-\tau\,]\cdot R_{x,x}[n+1,\tau_0-\tau\,]}\right) \tag{4.31}$$

where $\tau = m\Delta_t \in [0, T_s)$; Δ_t represents the size of the increment, and m denotes the number of increments. Since reducing Δ_t can improve the synchronization accuracy and decrease the error floor (described later in the simulations section), but with higher estimation time and computational complexity. In this work, we assume that $\Delta_t = T_c/2$.The estimation $\hat{\tau}_0(DA)$ could be developed as:

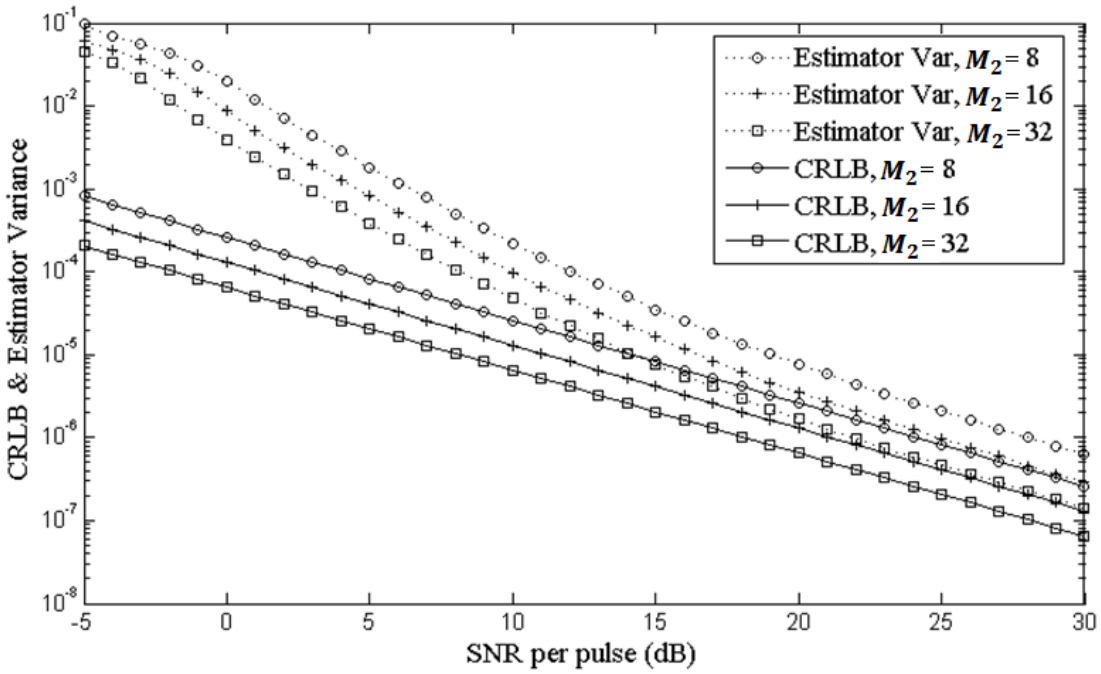


Figure 4.3 Comparison of normalized estimation variance and CRLB in DA mode

$$\hat{\tau}_0(\mathrm{DA}) = \underset{m}{arg\,max} \left(\sqrt{-\frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}[n, \tau_0 - m\Delta_t] \cdot R_{x,x}[n+1, \tau_0 - m\Delta_t]} \right)$$

$$\hat{\tau}_0(\mathrm{DA}) = \underset{m}{arg\,min} \frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}[n, \tau_0 - m\Delta_t] \cdot R_{x,x}[n+1, \tau_0 - m\Delta_t] \quad (4.32)$$

The architecture of the maximum likelihood estimator (4.32) is shown in Figure 4.4.

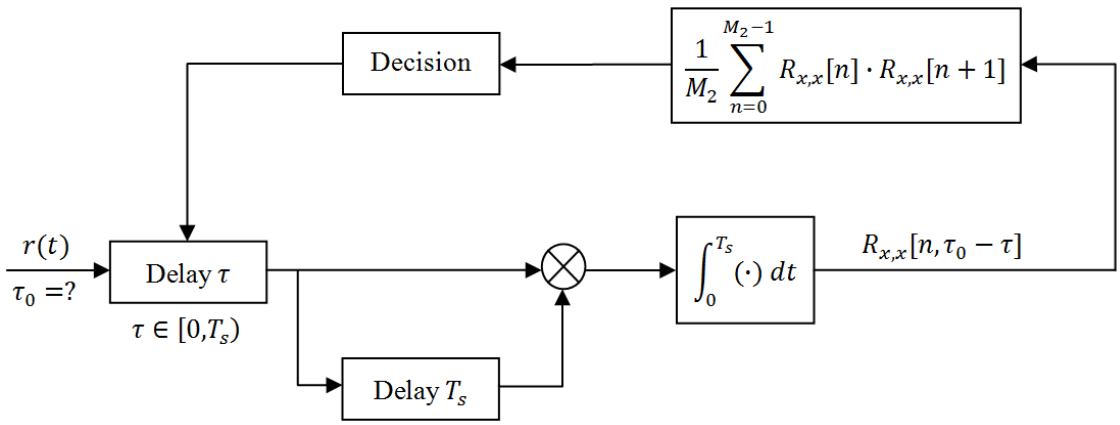


Figure 4.4 Block diagram of MLE for DA mode

However, we can notice from Figure 4.5, in the UWB received signal, the presence of gaps between the received symbol and its dirty template. In (4.32) above, these gaps may cause multiple maxima points around the peak (optimal point) inside the ML estimator range, so the estimation error of the timing synchronization may be increased [17].

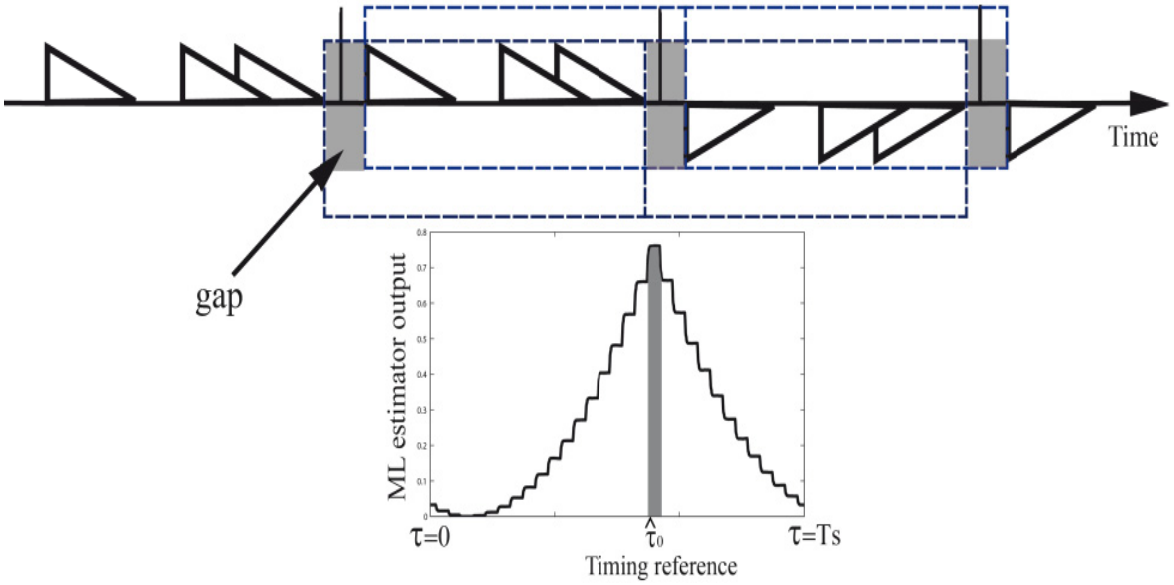


Figure 4.5 Timing reference illustration

To avoid this problem, we modify the structure of the cross-correlation operation, mentioned in (4.2), by adding the suitable window filter. So the modified cross-correlation becomes:

$$\acute{R}_{xx}[n, \tau_0 - m\Delta_t] = \int_{nT_s}^{(n+1)T_s} x(t + m\Delta_t).x(t + T_s + m\Delta_t).W(t)dt \quad (4.33)$$

where the window $W(t)$ contains the information about TH codes, which are known at the receiver. The window $W(t)$ is expressed by the following expression:

$$W(t) = \sum_{j=0}^{N_f-1} p_w(t - jT_f - c_jT_c)\,;\; p_w(t) = \begin{cases} 1 & si\ 0 \le t \le T_p + \tau_{L-1,0} \\ 0 & otherwise\ . \end{cases} \quad (4.34)$$

As the time of channel delay spread $\tau_{L-1,0}$ is unknown, we assume that the window width could be practically inside the range $\{T_c, 2T_c\}$. Figure 4.6 illustrates the structure of this window filter. We can notice that this window filter reduces the effect of these gaps at the UWB receiver. However, the window may slightly reduce the energy capture at the output of cross-correlation operation, but fortunately that doesn't impact noticeably on the estimation accuracy.

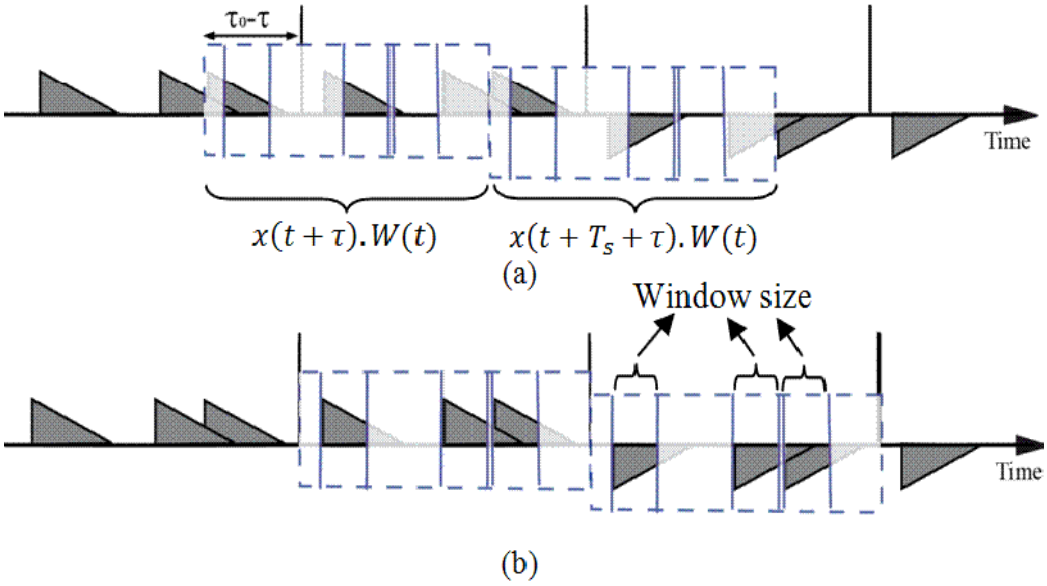


Figure 4.6 Window filter illustration; (a) $\tau \neq \tau_0$, (b) $\tau = \tau_0$

Therefore, by replacing $R_{xx}[n, \tau_0 - m\Delta_t]$ by $\acute{R}_{xx}[n, \tau_0 - m\Delta_t]$ in (4.32), the timing offset estimator is performed as the following statement and its architecture is shown in Figure 4.7.

$$\hat{\tau}_0(\text{DA}) = \underset{m}{arg\,min}\ \frac{1}{M_2}\sum_{n=0}^{M_2-1} \acute{R}_{xx}[n, \tau_0 - m\Delta_t] \cdot \acute{R}_{xx}[n+1, \tau_0 - m\Delta_t] \tag{4.35}$$

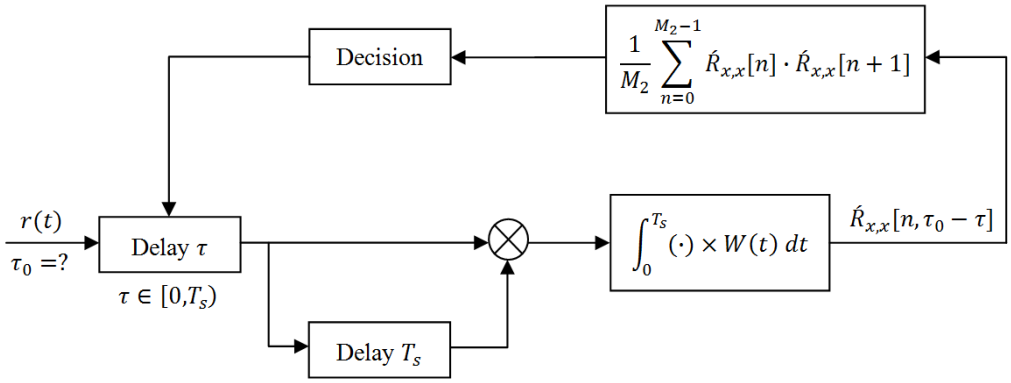


Figure 4.7 Block diagram of MLE with window for DA mode

Figures 4.8-4.10 show the captured energy at the ML estimator output versus the increment value of $\tau = m\Delta_t \in [0, T_s)$. Each of these figures has three graphs, one (*dashed line*) represents the ML estimator without any window, and the others (*dotted & solid lines*) represent the ML estimator with window size equal to $6T_c$ and $2T_c$ respectively. We have studied the UWB

system without and with TH codes and with SNR= (20 or 3) dB. These figures show that the ML estimator output without window in (4.32) has multiple maxima points around the peak. Besides, they exhibit that decreasing the window size in (4.35) leads to decrease the maxima point number, until we reach a single maxima point (at $m\Delta_t \approx \tau_0$).Therefore, the estimation accuracy is improved.

Moreover, the window filter also contributes to reduce the unwanted noise effect on the timing estimation. When window size equals $2T_c$, the received noise energy will be equal to (N_0) multiplied by the normalized window size ($2T_c/T_f$). Figure 4.11 shows that the modified ML estimator (MLE with windows) has higher estimation performance (CRLB & estimation variance) than the standard estimator (MLE without any windows), and the estimator variance lines converge together with increasing SNR, while the noise effect on the estimator will be reduced. Figure 4.12 presents the comparison between the normalized variance of the proposed ML estimator (*dashed lines*) extracted from (4.30) and its CRLB (*solid lines*) extracted from (4.21) with replacing the standard cross-correlation $R_{x,x}$ with the modified correlation $\acute{R}_{xx}$, and calculating the new value of σ_w^2 from (4.4) by using $\acute{N}_0 = 2N_0T_c/T_f$ instead of N_0. We assume window size $= 2T_c$, $\tau_0 = 0$, $\mathcal{E}_{max} = 1$ and normalizing the simulation according to $p_R^2(T_s)$.

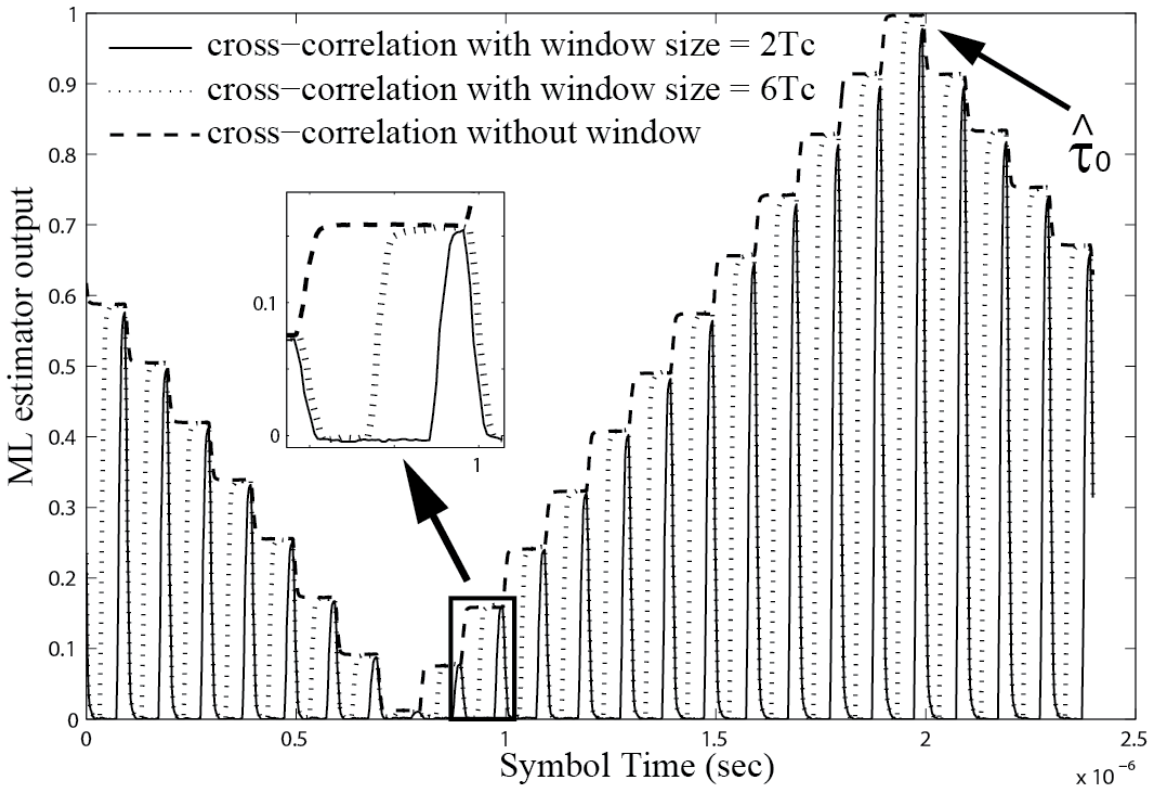


Figure 4.8 ML estimator output (without TH codes, SNR=20 & M_2=16)

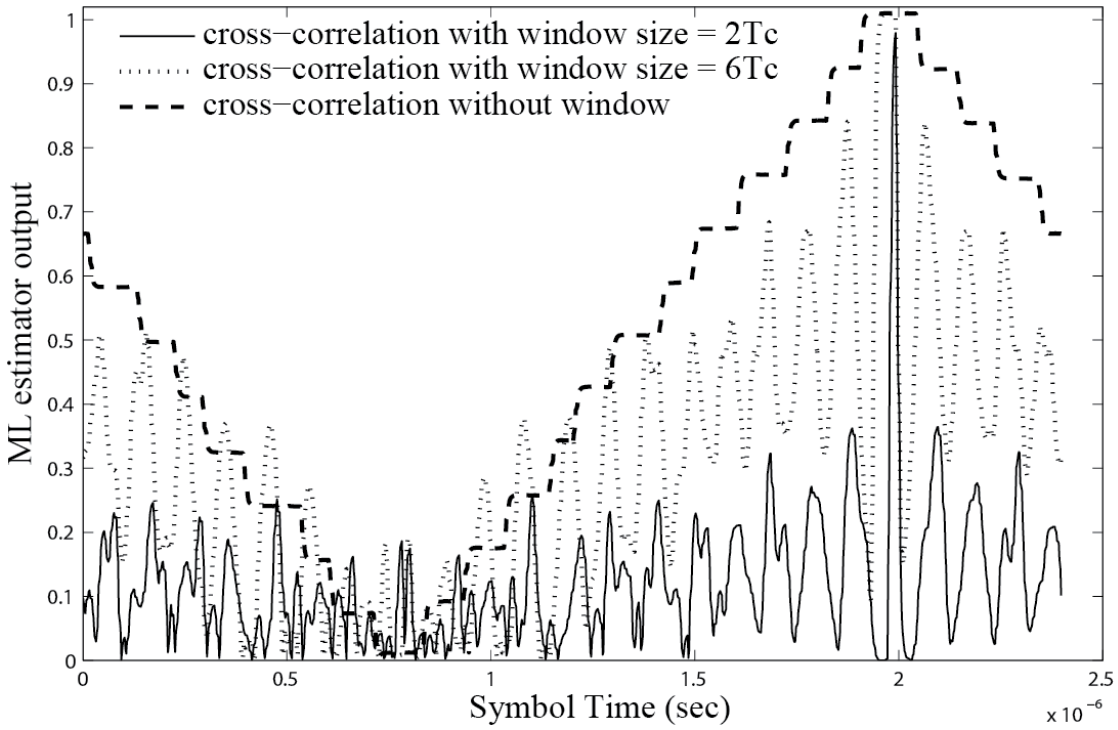


Figure 4.9 ML estimator output (TH codes, SNR=20 & M_2=16)

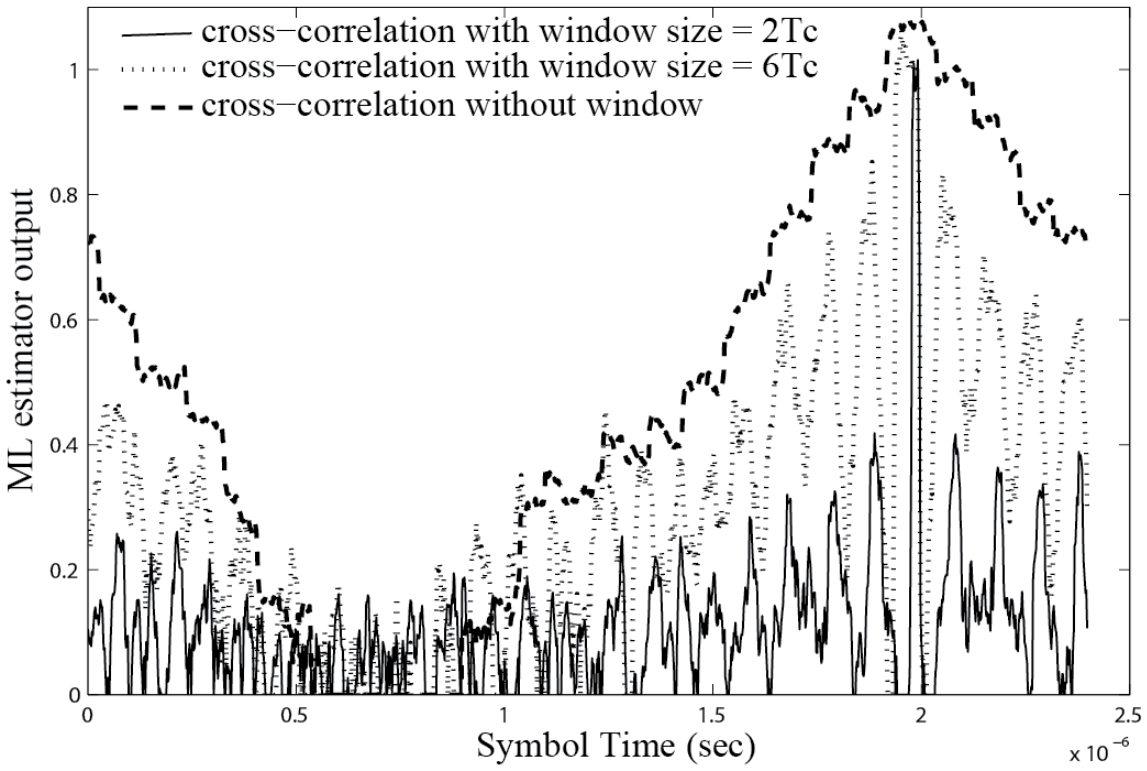


Figure 4.10 ML estimator output (TH codes, SNR=3 & M_2=16)

In particular, the window $W(t)$ simplifies the implementation of timing operation, because we don't need to take all sampled points of the symbol to calculate the cross-correlation operation, the sample points inside the windows are sufficient. Moreover, this method can reduce multiple-user interference MUI, by using orthogonal TH codes for different users.

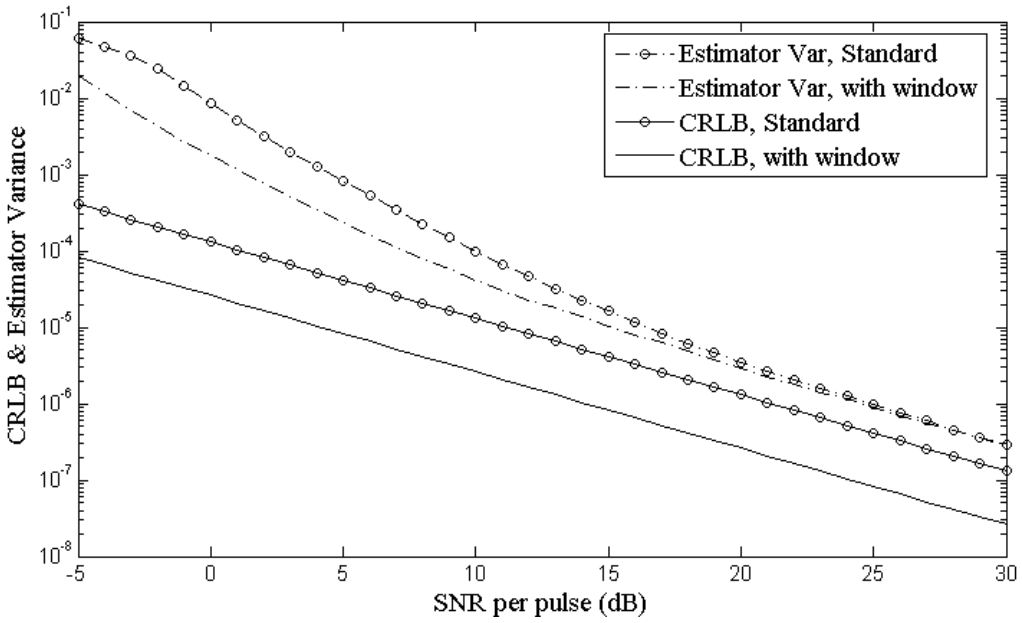


Figure 4.11 Comparison between normalized estimation variance and CRLB in DA mode for the MLE without any windows (standard) and MLE with windows (modified) for M_2=16

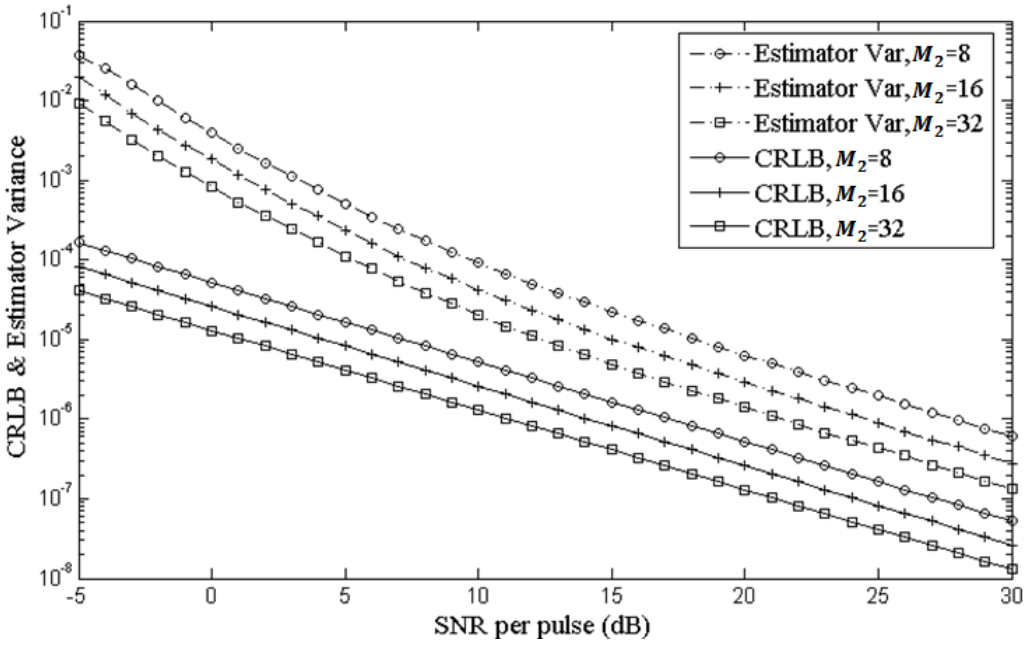


Figure 4.12 Comparison of normalized estimation variance and CRLB for modified MLE in DA mode

After we have studied the performance of the MLE which is derived by Appendix 4.A, let us briefly present the second expression of MLE for DA mode which is derived by Appendix 4.B. According to Appendix 4.B, the MLE without windows is given as follows, see Figure 4.13:

$$\hat{\tau}_0(\mathrm{DA}) = \underset{m}{arg\,max}\ \frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n, \tau_0 - m\Delta_t] \tag{4.36}$$

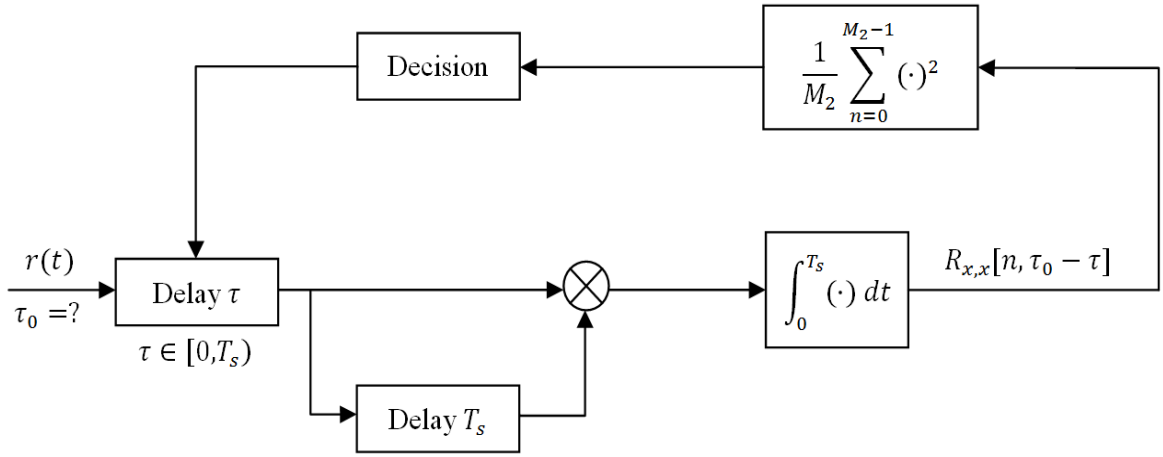


Figure 4.13 Block diagram of MLE without windows

The MLE with windows is given as follows, see Figure 4.14:

$$\hat{\tau}_0(\mathrm{DA}) = \underset{m}{arg\,max}\ \frac{1}{M_2}\sum_{n=0}^{M_2-1} \acute{R}_{x,x}^2[n, \tau_0 - m\Delta_t] \tag{4.37}$$

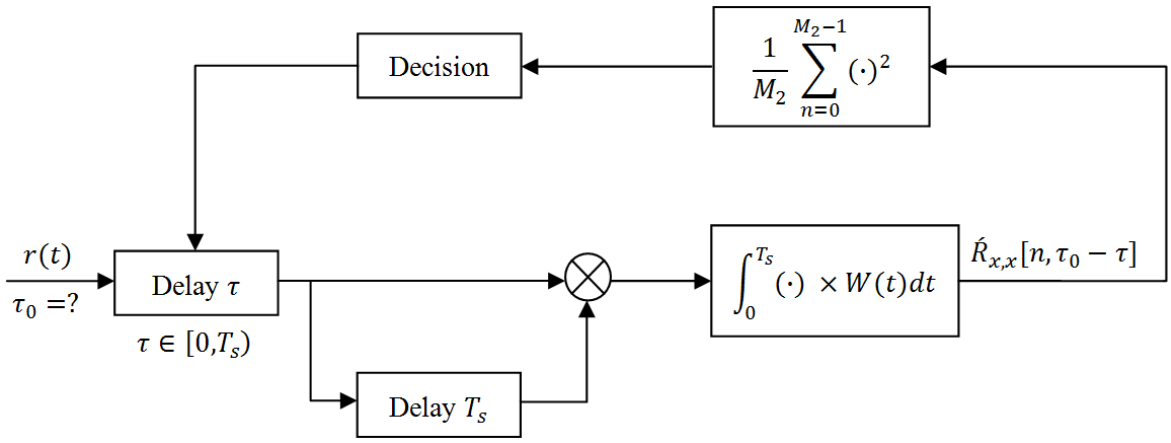


Figure 4.14 Block diagram of MLE with windows

4.3.2 Maximum likelihood estimator for non-data-aided mode

In this section, we will seek to minimize the timing offset τ_0 for NDA mode. where A_n and B_n are considered as independent random variables. According to Appendix 4.C, the MLE for NDA mode is given as follows:

$$\hat{\tau}_0(\text{NDA}) = \frac{\mathcal{E}_{max}}{\left(p_R^2(T_s)+p_R^2(T_s-\tau_0)\right)}\left(1-\frac{1}{\mathcal{E}_s\mathcal{E}_{max}}\sqrt{\frac{2}{M_2}\sum_{n=0}^{M_2-1}R_{x,x}^2[n,\tau_0]-2\sigma_w^2-\mathcal{E}_s^2\mathcal{E}_{max}^2}\right) \tag{4.38}$$

But the variables $\mathcal{E}_{max}$ and $p_R^2(T_s-\tau_0)$ are unknown. Similar to the previous section, we can deduce the estimate of timing offset for NDA mode by searching the minimum point of the following expression for different values of $\tau = m\Delta_t \in [0, T_s)$:

$$\hat{\tau}_0(\text{NDA}) = \underset{m}{arg\,min}\ \frac{\mathcal{E}_{max}}{\left(p_R^2(T_s)+p_R^2(T_s-\tau_0)\right)}\left(1-\frac{1}{\mathcal{E}_s\mathcal{E}_{max}}\sqrt{\frac{2}{M_2}\sum_{n=0}^{M_2-1}R_{x,x}^2[n,\tau_0-m\Delta_t]-2\sigma_w^2-\mathcal{E}_s^2\mathcal{E}_{max}^2}\right)$$

Considering the variables σ_w^2, $p_R^2(T_s-\tau_0)$, ε_s and ε_{max} independent of the change of τ, the estimator can be developed as follows, see Figure 4.13:

$$\hat{\tau}_0(NDA) = \underset{m}{arg\,max}\ \frac{1}{M_2}\sum_{n=0}^{M_2-1}R_{x,x}^2[n,\tau_0-m\Delta_t\,] \tag{4.39}$$

To avoid the presence of multiple maxima points around the optimal point, we use the modified structure of the cross-correlation operation $\acute{R}_{xx}[n,\tau_0-m\Delta_t]$ mentioned in (4.33). So the NDA estimator is given as, see Figure 4.14:

$$\hat{\tau}_0(NDA) = \underset{m}{arg\,max}\ \frac{1}{M_2}\sum_{n=0}^{M_2-1}\acute{R}_{x,x}^2[n,\tau_0-m\Delta_t\,] \tag{4.40}$$

Figures 4.15-4.17 study the timing performance for the ML estimator without and with the window filter for NDA mode. Figure 4.15 presents the comparison between the normalized variance of ML estimator (*dashed lines*) extracted from (4.38) and CRLB (*solid lines*) extracted from (4.22) for NDA mode. Assuming $\tau_0 = 0$, $\mathcal{E}_{max} = 1$ and normalizing the simulation according to $p_R^2(T_s)$. Figure 4.15 shows clearly that the estimation variance is inversely proportional to SNR, and converges on CRLB with increasing SNR. We can also see that an increase in the number of observations (M_2) improves the estimation performance. Figure 4.16 shows that the modified MLE (with windows) has higher estimation performance (CRLB & estimation variance) than the standard estimator (without any windows), and the

estimator variance lines converge together with increasing SNR, where the noise effects on the estimator will be reduced. Figure 4.17 presents the comparison between the normalized variance of the proposed ML estimator (*dashed lines*) extracted from (4.38) and its CRLB (*solid lines*) extracted from (4.22) with replacing the standard cross-correlation $R_{x,x}$ with the modified correlation $\acute{R}_{xx}$, and calculating the new value of σ_w^2 from (4.4) by using $\acute{N}_0 = 2N_0T_c/T_f$ instead of N_0. We assume window size $= 2T_c$, $\tau_0 = 0$, $\mathcal{E}_{max} = 1$ and normalizing the simulation according to $p_R^2(T_s)$.

Summary: In this section, we seek to generate a suitable timing estimator based on the received samples $R_{x,x}[n, \tau_0]$ which is mentioned in (4.24) as follows:

$$R_{x,x}[n, \tau_0] = \breve{\omega}[n] + (A_n - B_n)\gamma(\tau_0) + B_n\mathcal{E}_{max} \quad : n = 0,1, \dots, M_2 - 1 \tag{4.41}$$

where $\mathcal{E}_{max} = \int_0^{T_s} p_R^2(t)\, dt$ indicates the maximum useful energy of the received waveform $r(t)$ which is captured at the correlator output. $\gamma(\tau_0) = \int_{T_s-\tau_0}^{T_s} p_R^2(t)\, dt$ represents the fraction of total energy $\mathcal{E}_{max}$ captured by the interfering symbol $\mathrm{s[n-1]}$, for more details go back to (4.3) and Figure 4.1. Hence, in the case of mistiming ($\tau_0 \neq 0 \ \leq T_s$), the interfering effect of the symbol $\mathrm{s[n-1]}$ presents in $R_{x,x}[n, \tau_0]$. In contrast, without mistiming $\tau_0 = 0$, it leads to cancel this interference and to make $\gamma(\tau_0)$ equal to zero. In order to estimate the optimal value of timing offset τ_0 we proposed two approaches: the first one, is presented in Appendixes 4.A, 4.B and 4.C, and based on maximum likelihood algorithms. The second, is presented in Appendix 4.D, and based on mean-square sense algorithms. Appendixes 4.A, 4.B, 4.C and 4.D explain deeply with mathematical demonstration how to estimate τ_0 in both DA and NDA modes. Additionally, Appendixes 4.B, 4.C and 4.D assert that the sample square average in (4.36), see Figure 13, can be employed for estimating timing offset in either a DA mode or an NDA mode. So TDT synchronization relies on searching a peak in the output of the sliding correlation between the received signal and its dirty template. However, we have found multiple maxima points around the peak at the output of the correlator, which may degrade the estimation performance of timing offset error (TOE). To avoid this problem, we add a suitable window filter $W(t)$ to the structure of the cross-correlator as in (4.37), see Figure 14. This modified approach guarantees to obtain a single maximal peak inside the estimator range and improve the estimation error performance. This window filter also contributes to reduce the unwanted noise effect on the timing estimation. Moreover, the theoretical analysis confirm that for the same size of correlation sequence pattern M_2 the DA mode has the high performance compared to NDA mode, but that is at the price of bandwidth efficiency.

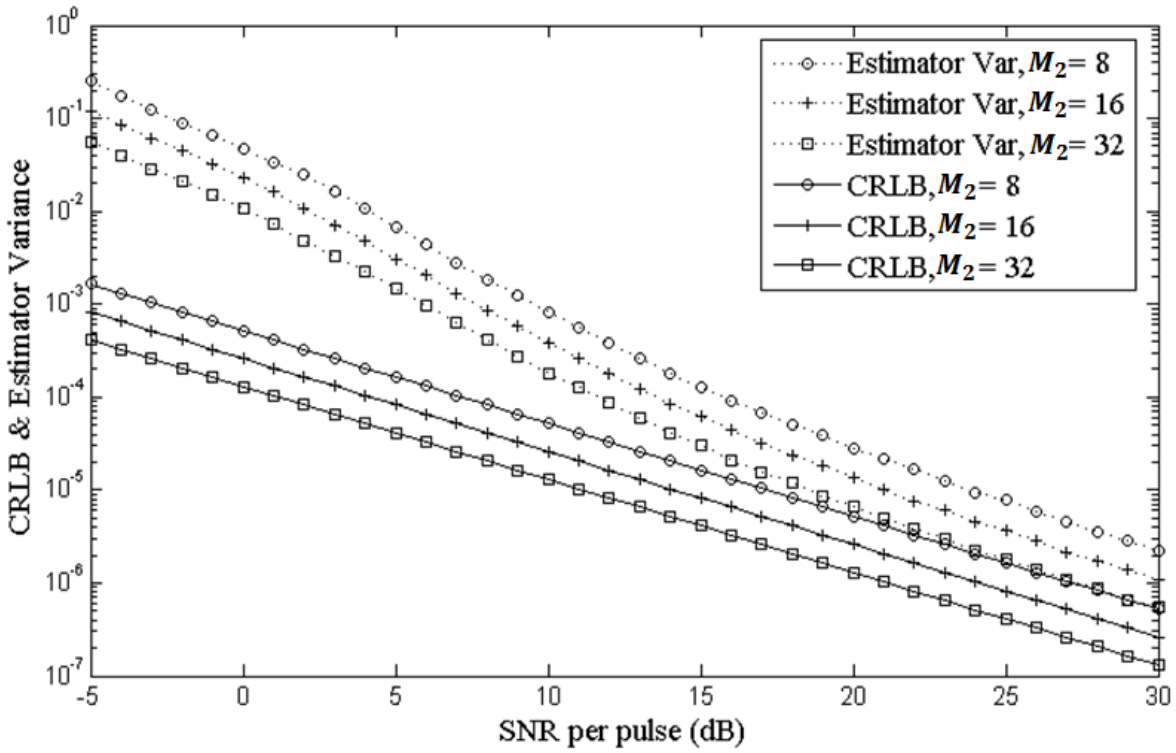


Figure 4.15 Comparison between normalized estimation variance and CRLB in NDA mode

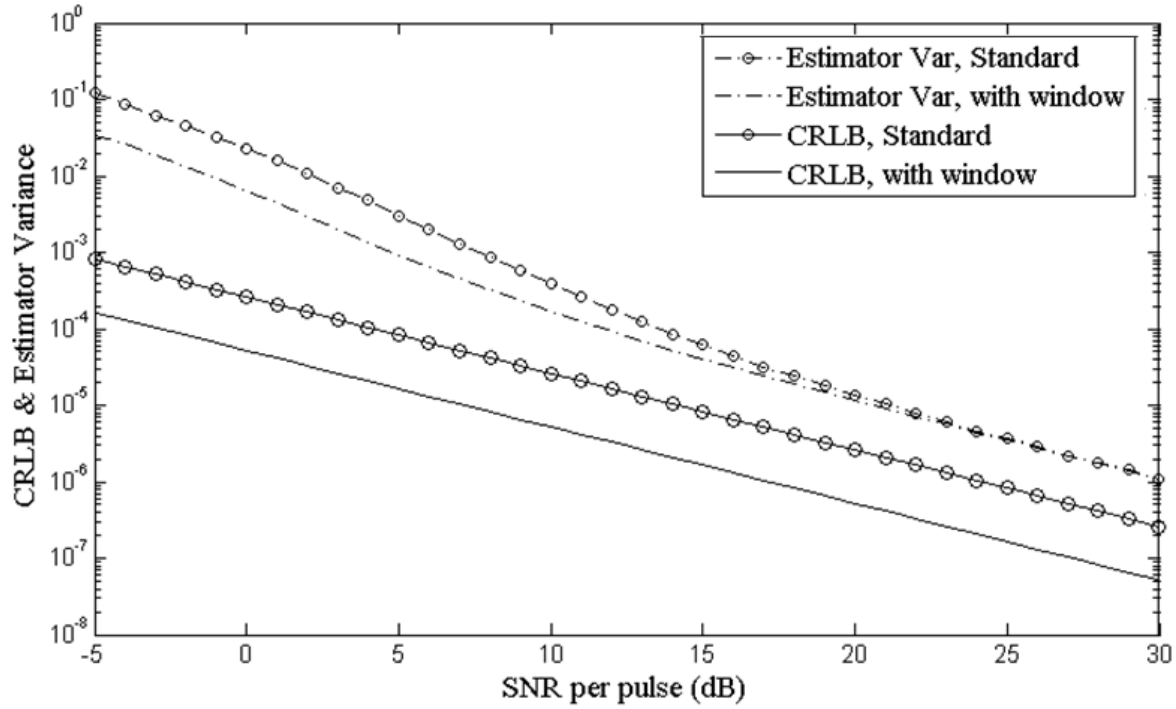


Figure 4.16 Comparison of normalized estimation variance and CRLB in NDA mode for the MLE without any windows (standard) and MLE with windows (modified) for $M_2 = 16$

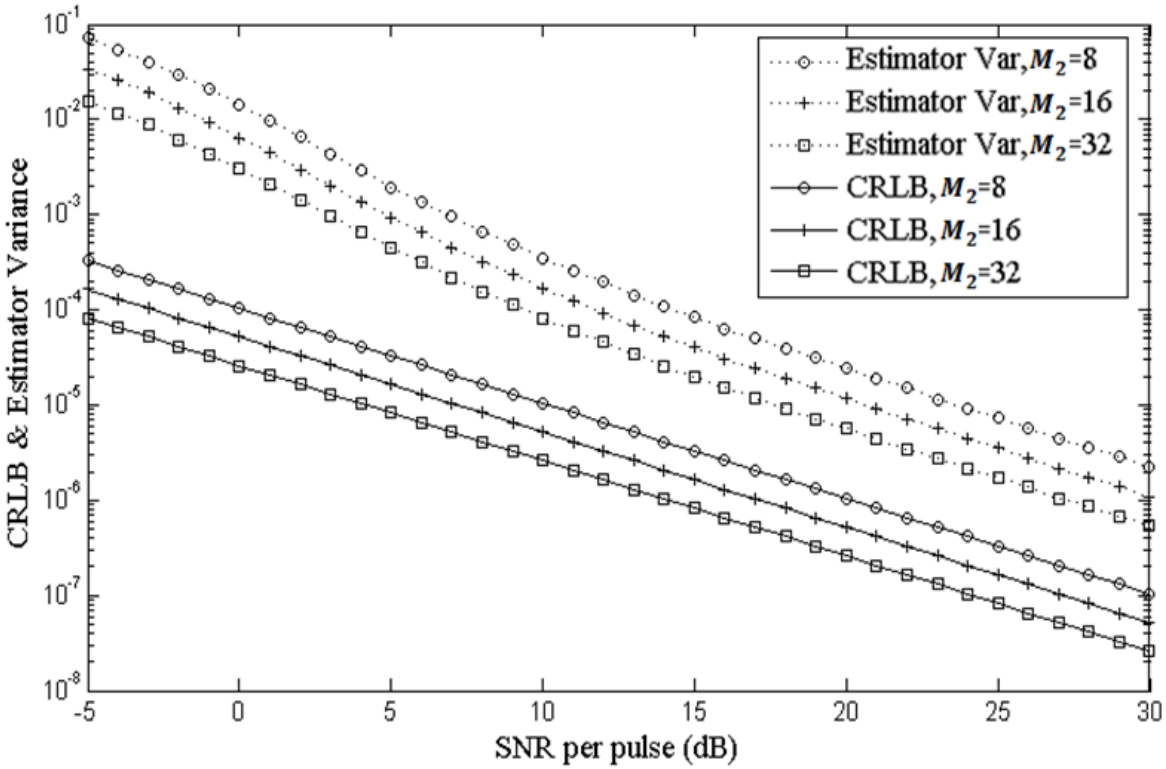


Figure 4.17 Comparison of normalized estimation variance and CRLB for modified MLE in NDA mode

4.4 **Simulations**

In this section, we will evaluate the performance of the proposed ML synchronizers for DA & NDA modes using Matlab, Figure 4.18 presents the flow chart of timing acquisition scheme. We select the pulse $p(t)$ as the third derivative of the Gaussian function with unit energy and duration $T_p \approx 1$ns. Each symbol contains $N_f = 24$ frames with duration $T_f = 100$ns. The simulations are performed in a Saleh-Valenzuela channel [61]. The parameters of this channel are chosen with (1/Λ, 1/λ, Γ, γ) = (2, 0.5, 30, 5) ns. The maximum channel delay spread of the channel is about 99 ns, and the inter-symbol interference (ISI) is negligible. We compare the accurate acquisition between the ML estimator without any window (4.32, 4.39) and the modified ML estimator with window size equal to $2T_c$ (4.35, 4.40).

In all simulations, we supposed that signal detection has been carried out and that n_s has been identified successfully. We generate τ_0 randomly from a uniform distribution over $[0, T_s)$. We employ TH spreading codes of period N_f, which is generated from a uniform distribution over $[0, N_c - 1]$, with $N_c = 6$, and $T_c = 10$ns. We supposed the size of the increment Δ_t mentioned in (4.31) equal to $T_c/2$.

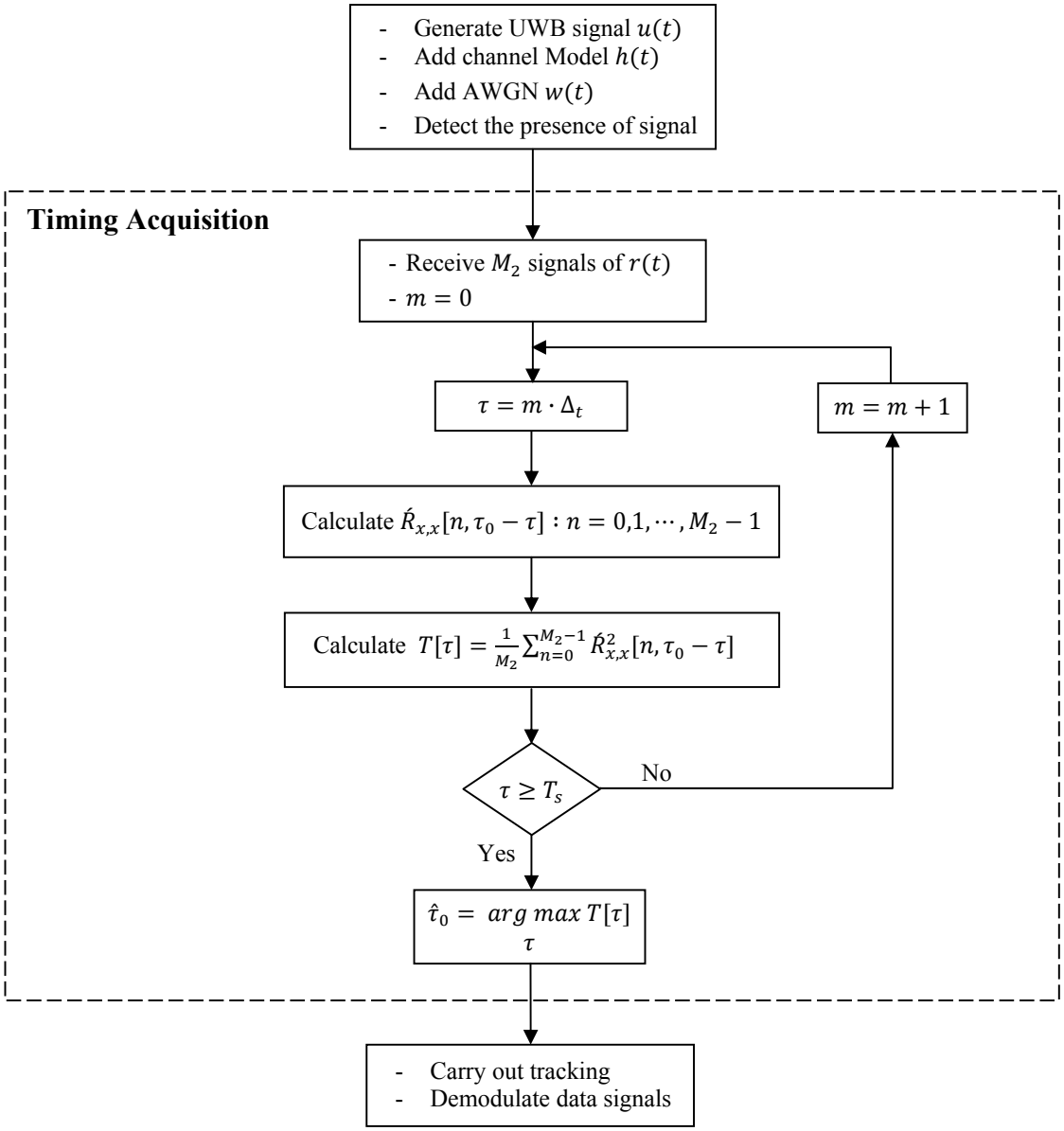


Figure 4.18 Flow chart of timing acquisition scheme

Figures 4.19 and 4.20 show the comparison of mean square error (MSE) performance in DA & NDA modes with dirty templates for various values of M_2. In these figures, the MSE results are normalized by T_s^2, and plotted versus the signal-to-noise ratio (SNR) per pulse. The simulations confirm that the ML estimator with window (*solid lines*) has higher timing estimation performance than the ML estimator without window (*dashed lines*). In general, as M_2 increases the normalized MSE decreases. Increasing SNR also helps to reduce the MSE. That means, when the observation number M_2 or the signal-to-noise ratio SNR is increased, the

mean square error MSE is reduced and the estimation performance is improved. In addition, we can clearly notice that all curves with high SNR reach an error floor, which depends on the synchronization accuracy. This error floor can be reduced by decreasing the size of Δ_t, but with increasing simulation time and computational complexity. The results also show that the timing estimator for DA mode in Figure 4.19 has smaller MSE values and more accurate timing simulation than NDA mode in Figure 4.20 for the same SNR & M_2, but with less bandwidth efficiency. Moreover, Figure 4.19 shows that the DA estimator could be used with small training pattern size such as $M_2 = 4$, that helps to reduce the number of operations performed at the receiver as well as the synchronization time and to provide more efficient use of bandwidth.

4.5 Conclusion

In this paper, we use dirty template algorithms for carrying out low-complexity high-performance timing acquisition, which constitutes a major challenge in realizing the UWB communications. We derive first the Cramer-Rao lower bound for our system, which is used as a fundamental performance limit for any timing estimator.

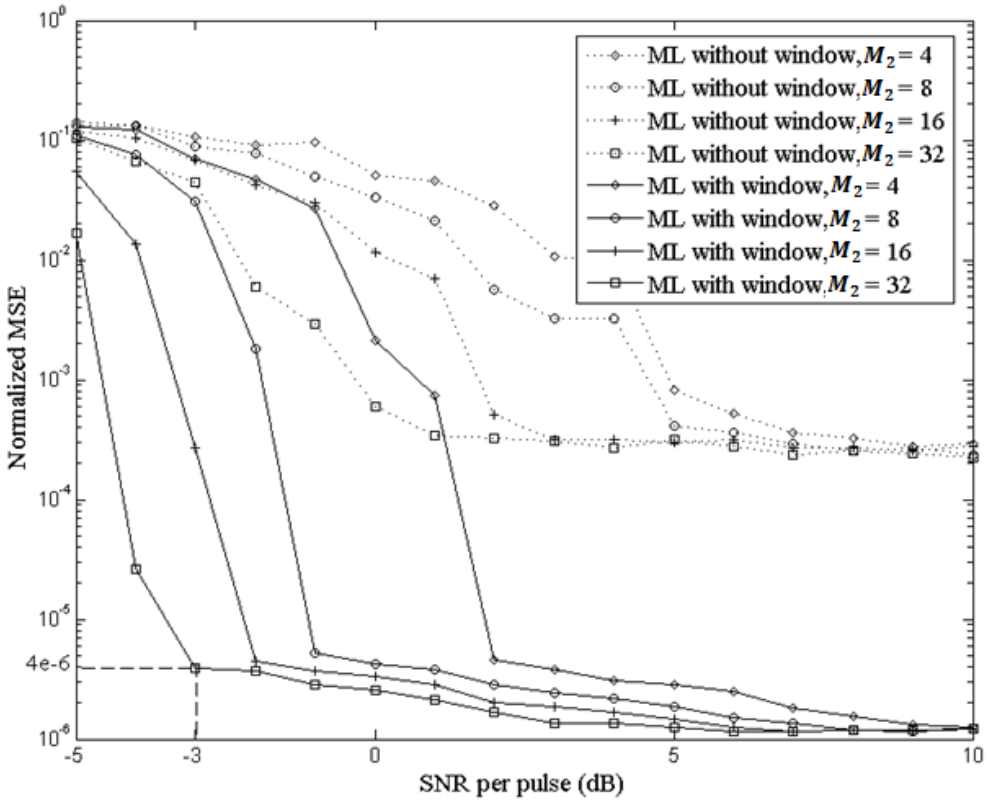


Figure 4.19 Normalized MSE vs. SNR per pulse for MLE in DA mode

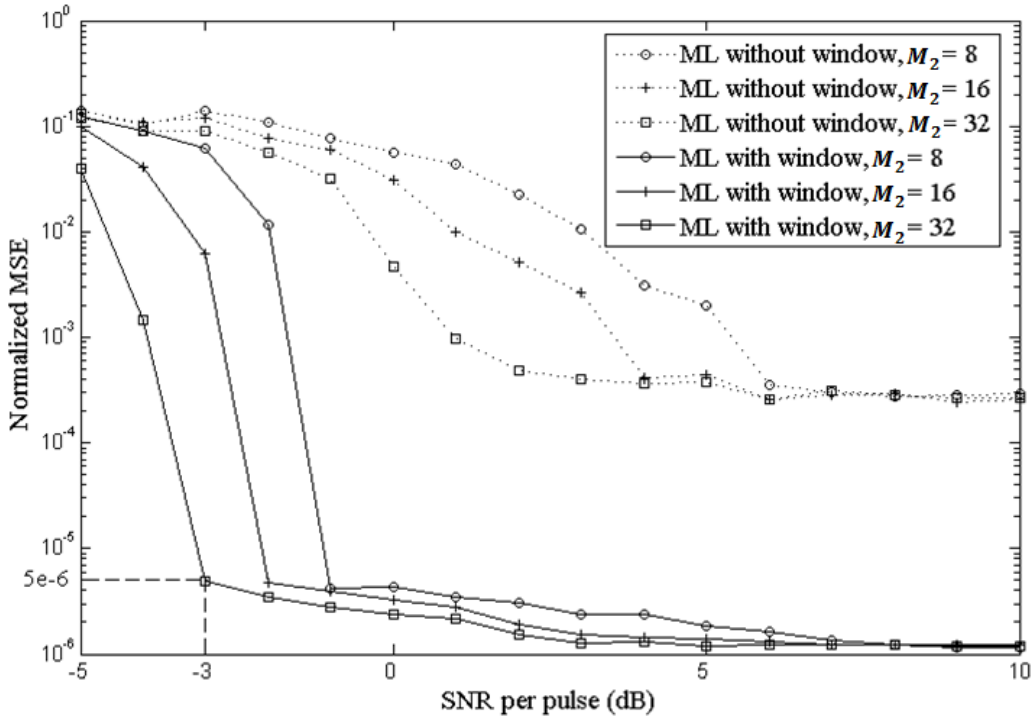


Figure 4.20 Normalized MSE vs. SNR per pulse for MLE in NDA mode

Next, we develop the TDT acquisition estimator with and without training symbols by using maximum likelihood and mean-square algorithms. We assert that TDT synchronization relies on searching a peak in the output of the sliding correlation between the received signal and its dirty template. However, we have found multiple maxima points around the peak at the output of the correlator, which may degrade the estimation performance of timing offset error (TOE). To avoid this problem, we add a suitable window filter to the structure of the cross-correlator. This modified approach guarantees to obtain a single maximal peak inside the estimator range and improve the estimation error performance. This window filter also contributes to reduce the unwanted noise effect on the timing estimation. Both the theoretical analysis and Matlab simulation results show the performance of the proposed timing estimator, and confirm that for the same size of correlation sequence pattern M_2 the DA mode has the high performance and fast timing compared to NDA mode, but that is at the price of bandwidth efficiency.

Note: As we have said in this chapter and Chapter 3, we send two parts of training sequence: the first part of long M_1 is used for detecting UWB signals, and the second part of long M_2 is used for fulfilling timing acquisition. There is an alternative way to detect the presence of UWB signal. Instead of sending two parts of training sequence, it is enough to send M_2 data aided sequence. The idea is simply to apply the timing algorithms as in (4.37) even if we don't know if the desired signal is present or not. In the same time, we compare the value of sample square average in (4.37) for each value of τ with a threshold, which is set following the

Neyman-Person theorem. If the sample output value in (4.37) is bigger than the threshold we say that the signal is present. Simultaneously, the estimation of timing offset τ_0 relies on searching a peak in the output of (4.37). However, the problem of this method is that the timing algorithms are applied even if we don't receive the signal. So this method always executes a lot of calculations and consumes a lot of energy even when we don't yet receive the signal.

Chapter 5

Tracking Loop

After the timing acquisition task has successfully been completed and the timing offset error is confined within a fraction of UWB pulse duration. This chapter will focus on the third step of synchronisation system: tracking. The objective of this step is to maintain and lock the satisfactory synchronization throughout the transmission period between receiver and transmitter, even if timing offset variations in the received waveform occur as a result of relative transmitter-receiver motion (Doppler Effect) [79, 88]. The delay locked loop (DLL) is the fundamental technique used for tracking purpose in UWB and spread-spectrum systems [89, 90]. According to [91], M. Proakis described that DLL technique is considered as an approximation to the maximum-likelihood estimate of the timing offset. Several DLL schemes have been proposed to improve the tracking performance in UWB systems [76, 92-94]. So, the goal which we hope to reach in this chapter is to design a suitable tracking unit for DT-UWB systems, which contributes to alleviate the effects of timing offset variations in the received waveform, and to enhance the transmission performance (BER).

This chapter is organized as follows: First, we will explain deeply how to design an open-loop structure of DLL and how to generate a suitable reference signal based on DT technique. Next, an appropriate first-order closed-loop will be derived by employing a simple iterative method. Then, the essential parameters of the proposed DLL will be found and the tracking performance will be analyzed in term of output noise variance (Jitter) and transient behavior. We will see later that in the presence of dynamic Doppler Effect, the first-order DLL is no longer able to track the timing variations of input signal. For tracking efficiently the input UWB waveform in the presence of ambient noise and Doppler effects, we suggest DLL with second-order closed-loop structure. The parameters of suggested DLL will be selected by applying Wiener-filter theory, in order to improve transient behavior as well as noise rejecting ability.

5.1 Open-loop DLL model

Assuming that detecting the presence of UWB signal and estimating the timing offset (acquisition) have been correctly achieved, so the timing error is confined within a fraction of UWB pulse duration and the tracking loop is "*in lock*" at the beginning of its operation. Considering the received signal rewritten as follows:

$$r(t) = \sqrt{\mathcal{E}_s}\sum_{n=0}^{\infty} s[n]p_R(t - nT_s) + w(t) \tag{5.1}$$

To maintain the satisfactory synchronization between the received and the reference signals, we employ a Delay-Locked Loop (DLL) which is widely used for tracking purpose. Figure 5.1 shows the proposed DLL method, known also as Early-Late Gate (ELG), consisting of two correlator branches: The first branch is called *early correlator* because the received signal is cross-correlated with the advanced version of the reference signal $y(t + \Delta)$. And the second is called *late correlator*, where the received signal is correlated with the retarded version of the reference signal $y(t - \Delta)$. In other word, the generated reference $y(t)$ is advanced and delayed in time by Δ (some fraction of a UWB pulse). The cross-correlation outputs are given by:

$$\begin{cases} R_e = \int_0^{T_s} r(t)y(t - \hat{\tau} + \Delta)\, dt \\ R_l = \int_0^{T_s} r(t)y(t - \hat{\tau} - \Delta)\, dt \end{cases} \tag{5.2}$$

where the reference signal $y(t)$ could be extracted from the received waveform $r(t)$ based on training sequence. We can exploit the same data-aided samples which had already been used to detect the UWB signal and to acquire the timing (see chapters 3 and 4). Thus, we will use again this data-aided sequence to design the reference signal $y(t)$. Assuming that M transmitted training bits equal one $\{s[n] = 1\}_{n=1}^{M}$, the noise-free part of $r(t)$ in (5.1) consists of $p_R(t)$ replicas with spacing T_s. Hence, $y(t)$ can be produced from $r(t)$ by averaging over these M received waveform replicas, as follows:

$$y(t) = \frac{1}{M}\sum_{n=1}^{M}\left(\sqrt{\mathcal{E}_s}p_R(t - nT_s) + w_n(t)\right); \quad t \in [0, T_s) \tag{5.3}$$

Then:

$$y(t) \cong \sqrt{\mathcal{E}_s}p_R(t) + \bar{w}(t): \qquad \bar{w}(t) = \frac{1}{M}\sum_{n=1}^{M} w_n(t) \sim \mathcal{N}(0, \bar{\sigma}^2) \tag{5.4}$$

where $\bar{\sigma}^2 = N_0/2M$. The template $y(t)$ is called dirty or noisy, because it is distorted by the unknown channel and by the ambient noise. We can notice that an increase in the number of data-aided bits M leads to reduce the noise variance $\bar{\sigma}^2$ and improve the tracking performance. Another way to decrease the amount of noise term $\bar{w}(t)$ inside the reference template $y(t)$ in

(5.4) is to pass $y(t)$ through the window filter $W(t)$, mentioned in (4.34). Assume that the window size equals $1.5T_c$, the noise energy $\bar{w}(t)$ will be divided by $(1.5\,T_c/T_f)$.

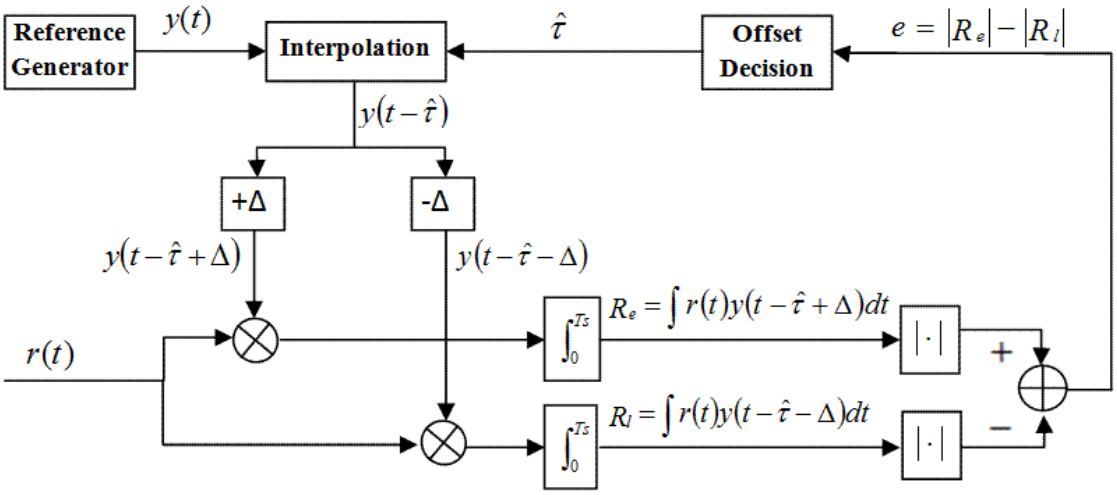


Figure 5.1 Delay-locked loop diagram

In the presence of sampling clock offsets (oscillator drifts) or Doppler Effect, this will produce the mistiming ($\tau \neq 0$) between the reference and received signals. Figure 5.1 shows that DLL tracking is performed by estimating the time offset τ between $r(t)$ and $y(t)$. Then, it compensates this offset by shifting the reference signal position to $y(t-\hat{\tau})$, where $\hat{\tau}$ denotes the DLL estimate of τ. In other word, The DLL tracking is used to minimize the time offset error $\epsilon = \hat{\tau} - \tau$.

Supposing the time offset error $\epsilon = \hat{\tau} - \tau$ is initially within UWB pulse duration; the tracking loop is "*in lock*" state. Supposing also ϵ is essentially constant over the correlation time T_s. Substituting (5.1) and (5.4) in (5.2), the cross-correlation outputs $R_{e,l}[\epsilon, n]$ become:

$$R_{e,l}[\epsilon,n] = \int_0^{T_s}\{\sqrt{\mathcal{E}_s}s[n]p_R(t) + w_n(t)\} \times \{\sqrt{\mathcal{E}_s}p_R(t-\epsilon \pm \Delta) + \bar{w}(t)\}dt \tag{5.5}$$

where $t \in [0,T_s)$. Thus:

$$R_{e,l}[\epsilon,n] = s[n]\Gamma_{e,l}[\epsilon] + n_{e,l}[\epsilon,n] \tag{5.6}$$

where: $\Gamma_{e,l}[\epsilon] = \mathcal{E}_s \int_0^{T_s} p_R(t)p_R(t-\epsilon \pm \Delta)\,dt$, and the sample noise factors $n_{e,l}[\epsilon,n]$, explained in Appendix 5.A, is composed of three terms, two of them are the result of correlation between the symbol and the noise, and the third term is between shifted noises, as follows:

$$n_{e,l}[\epsilon, n] = \xi_1 + \xi_2 + \xi_3 \tag{5.7}$$

where: $n_{e,l}[\epsilon, n]$: $\begin{cases} \xi_1 = \int_0^{T_s} \sqrt{\mathcal{E}_s} p_R(t - \epsilon \pm \Delta) w_n(t)\, dt \\ \xi_2 = \int_0^{T_s} \sqrt{\mathcal{E}_s} s[n] p_R(t) \overline{w}(t)\, dt \\ \xi_3 = \int_0^{T_s} w_n(t) \overline{w}(t)\, dt \end{cases}$

These terms could be treated approximately as mutually uncorrelated zero mean Gaussian noises. So $n_{e,l}[\epsilon, n]$ could be represented as zero-mean Gaussian noises with statistic properties given as (see Appendix 5.A):

$$n_{e,l} \sim \mathcal{N}\left(0, \sigma_{e,l}^2\right) \quad : \sigma_{e,l}^2 = \frac{{N_0}^2}{2M} B T_s + \frac{N_0 (M+1)}{2M} \mathcal{E}_s \mathcal{E}_{max} \tag{5.8}$$

where: $N_0/2$ and B represent power spectral density (PSD) and bandwidth of the propagation channel noise respectively, M denotes number of the training symbols used for generating the reference signal $y(t)$ and $\mathcal{E}_{max} = \int_0^{T_s} p_R(t)^2\, dt$ represents T_s-long received segment energy.

The difference value (discriminator) between the ELG correlator outputs (see Figure 5.1) serves as a timing error indicator. For example, if the discriminator (e) is close to zero, then $y(t)$ and $r(t)$ are synchronized perfectly ($\hat{\tau} \approx \tau$). Otherwise, $y(t)$ is too late or too early. The discriminator error e at the output of DLL tracking is expressed by the following equation:

$$e[\epsilon, n] = R_e[\epsilon, n] - R_l[\epsilon, n] \tag{5.9}$$

Substituting (5.6) in (5.9), we get:

$$e[\epsilon, n] = s[n] \underbrace{(\Gamma_e[\epsilon] - \Gamma_l[\epsilon])}_{\mathbb{S}[\epsilon]} + \underbrace{n_e[\epsilon, n] - n_l[\epsilon, n]}_{\eta[n]} = s[n] \cdot \mathbb{S}[\epsilon] + \eta[n] \tag{5.10}$$

where $\eta[n] = n_e[\epsilon, n] - n_l[\epsilon, n]$ is the equivalent additive noise for DLL tracking system, and $\mathbb{S}[\cdot]$ is the loop discriminator characteristic (S-curve), considered as the useful term for tracking τ. From (5.10), we notice that the noise-free part of discriminator error (e) depends on the time offset ϵ and also on the data sign $s[n]$. To avoid the influence of the unknown received data $s[n] \in \{\pm 1\}$ on the tracking decision, we add an absolute value block inside the DLL diagram (in such case, our tracking technique is called non-decision-direct loop [91]). The absolute value block could be executed by different ways. Three methods are here proposed to avoid the data sign effect, as follows:

- $e_1[\epsilon, n] = |R_e[\epsilon, n]| - |R_l[\epsilon, n]|$
- $e_2[\epsilon, n] = R_e^2[\epsilon, n] - R_l^2[\epsilon, n]$

- $e_3[\epsilon, n] = (R_e[\epsilon, n] - R_l[\epsilon, n]) \times sign(R_e[\epsilon, n] + R_l[\epsilon, n])$

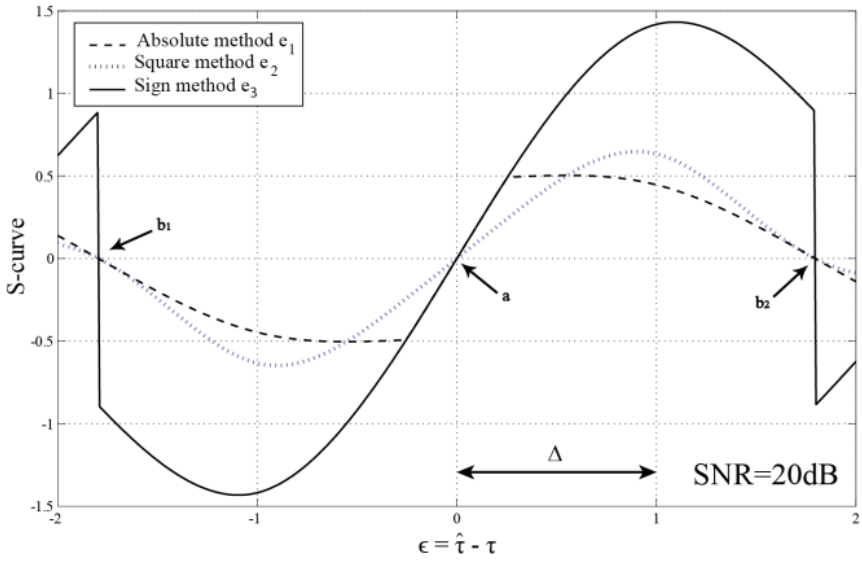
The S-curve for the proposed methods (absolute, square and sign methods corresponding) is plotted in Figure 5.2. We notice clearly that, all these three methods have the same lock range $[b_1, b_2]$. The first method (e_1) has a small linear tracking range within $[-\Delta, +\Delta]$. The second method (e_2) has a good properties (linearity) for SNR=20dB. But when SNR decreases, the linearity performance is reduced and the energy capture at the correlator output is rapidly augmented, since the square operation increases usually the noise energy effects. For that, the square method (e_2) is not advised to be employed for the tracking. The third method (e_3) has the best linearity performance among them for difference value of SNR. In addition, the S-curve of e_3 has the highest slope near the origin, so the sign method is most sensitive to small timing error. Thus, the third approach is chosen to make the tracking error indicator independent of the unknown data sign. Hence, the output error e in (5.10) becomes as follows:

$$e[\epsilon, n] = \mathbb{S}[\epsilon] + \eta[n] \tag{5.11}$$

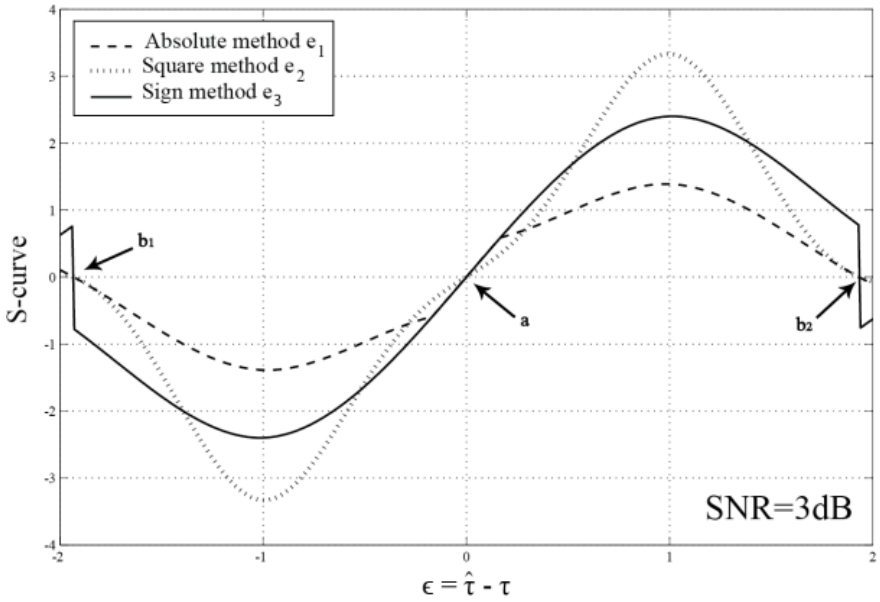
Explicitly, the term $\mathbb{S}[\cdot]$ is not dependent on time, depends only on the offset ϵ. The DLL noise $\eta[n]$, explained in Appendix 5.B, could be treated as a zero mean Gaussian noise:

$$\eta[n] \sim \mathcal{N}\left(0, \sigma_\eta^2\right) \qquad : \sigma_\eta^2 = \frac{N_0(M+1)}{M}(\mathcal{E}_s \mathcal{E}_{max} - \Gamma_{rr}(2\Delta)) \tag{5.12}$$

where $\Gamma_{rr}(2\Delta) = \mathcal{E}_s \int_0^{T_s} p_R(t)\, p_R(t - 2\Delta) dt$. Comparing (5.12) to (5.8), we observe that the ELG approach has helped to cancel the influence of the noise-cross-noise component at the tracking output and to alleviate the noise effect.



(a)



(b)

Figure 5.2 S-curve for absolute, square and sign methods: (a) SNR=20dB, (b) SNR=3dB

5.2 First-order DLL tracking

5.2.1 Stucture & Model

To simplify the DLL tracking system, it could be represented by the equivalent model shown in Figure 5.3. We focus now on designing a fit offset decision block, which is used to adjust and update the current estimate of time offset $\hat{\tau}$. For achieving the offset decision block, we employ an iterative method, as in [94]:

$$\hat{\tau}_{k+1} = \hat{\tau}_k + \gamma e_k[\epsilon] \quad : \gamma \in [0,1] \tag{5.13}$$

where k represents the iterative step number, and γ is a forgetting factor that is selected as a trade off between tracking speed and accuracy (tracking performance). We could notice that this equation represents a digital integrator, as such, has a lowpass action on $e_k[\epsilon]$. Moving to z-domain:

$$\hat{\tau}(z) = \hat{\tau}(z-1) + \gamma e_k(z-1)$$

$$\hat{\tau}(z) - z^{-1}\hat{\tau}(z) = \gamma z^{-1} e_k(z)$$

$$\hat{\tau}(z)(z^{-1}-1) = -\gamma z^{-1} e_k(z)$$

Transfer function for the offset decision block is:

$$\frac{\hat{\tau}(z)}{e_k(z)} = H(z) = \frac{\gamma z^{-1}}{1-z^{-1}} \tag{5.14}$$

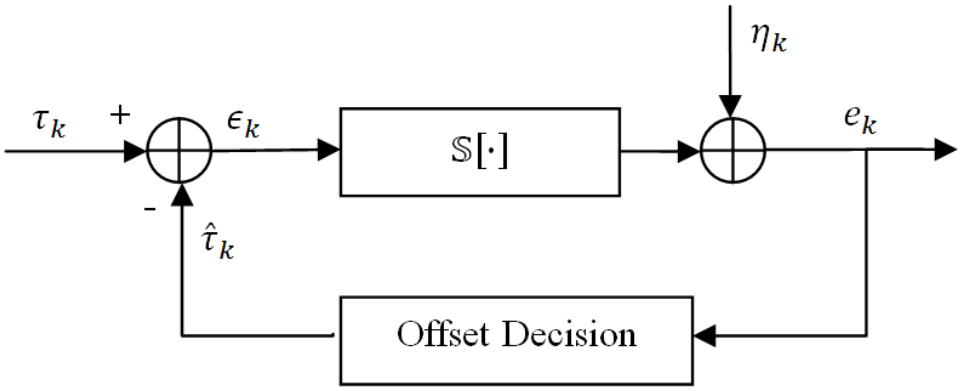


Figure 5.3 Equivalent timing model for the DLL tracking

After establishing the behavior of the dirty template DLL in UWB systems, the equivalent timing model of a DLL using iterative loop is shown in Figure 5.4. The following points are proposed for developing the tracking performance (accuracy), but that will be at the price of the tracking time and the implementation complexity:

- Increasing the memory size (the filter order), for example: $\hat{\tau}_{k+1} = \hat{\tau}_k - \gamma_1 e_k[\epsilon] - \gamma_2 e_{k-1}[\epsilon]$.
- Reducing the forgetting factor value γ.
- Increasing the iteration times for each T_s-long symbol.

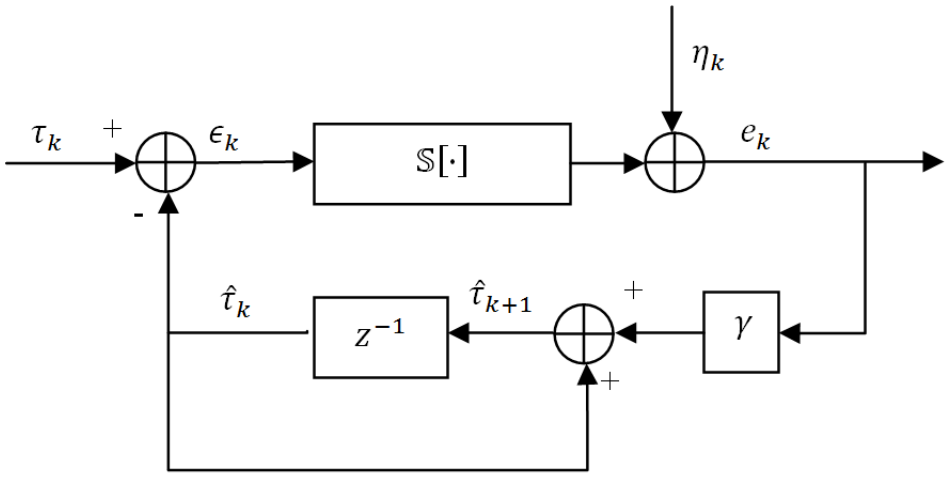


Figure 5.4 Block diagram of DLL using a first-order closed-loop transfer function

Let's now derive the loop response for the input noise η_k, we have:

$$\hat{\tau}_{k+1} = \hat{\tau}_k + \gamma e_k[\epsilon_k]$$
$$\tau - \hat{\tau}_{k+1} = \tau - (\hat{\tau}_k + \gamma e_k[\epsilon_k])$$

Considering $\epsilon_k = \tau - \hat{\tau}_k$, thus:

$$\epsilon_{k+1} = \epsilon_k - \gamma e_k[\epsilon_k]$$
$$\epsilon_{k+1} = \epsilon_k - \gamma(\mathbb{S}[\epsilon_k] + \eta_k)$$

Assuming that the time offset ϵ_k is small to be inside the linear range of the S-curve $[\epsilon_k]$ (see Figure 5.2) and the statistical properties for the equivalent DLL noise η_k in (5.12) do not depend on the timing error ϵ_k. In this case, the DLL behaves approximately like a linear filter ($\mathbb{S}[\epsilon_k] = A\epsilon_k$, where $A = \dot{\mathbb{S}}[0]$ is the gradient of the discriminator characteristic at $\epsilon_k = 0$) [79]. Thus:

$$\epsilon_{k+1} = \epsilon_k - \gamma(A\epsilon_k + \eta_k)$$
$$\epsilon(z) \cdot (z - 1 + \gamma A) = -\gamma . \eta(z)$$
$$\frac{\epsilon(z)}{\eta(z)} = H_N(z) = \frac{-\gamma}{z-(1-\gamma A)}; \text{ where } z = e^{2\pi j f T_s} \quad (5.15)$$

To analyze the loop tracking performance, we derive the tracking jitter σ_ϵ^2, which is defined as the variance of the time offset error ($\sigma_\epsilon^2 = E\{\epsilon^2\}$), and is expressed by:

$$\sigma_\epsilon^2 = \int_{-1/(2T_s)}^{1/(2T_s)} S_\eta(f)\,|H_N(f)|^2 df \quad (5.16)$$

For ease of analysis, we assume that the power spectral density of η_k, denoted by $S_\eta(f)$, is nearly flat over the frequency range where $H_N(f)$ takes significant values. Thus, the tracking jitter becomes:

$$\sigma_\epsilon^2 = S_\eta(0) \int_{-1/(2T_s)}^{1/(2T_s)} |H_N(f)|^2 df \quad (5.17)$$

After series of calculations, mentioned in [79], the integration part in (5.17) is given as:

$$\int_{-1/(2T_s)}^{1/(2T_s)} |H_N(f)|^2 df = \frac{\gamma}{(2-\gamma A)AT_s} \quad (5.18)$$

Assuming that $S_\eta(0) = \sigma_\eta^2$, and substituting (5.12) and (5.18) in (5.17),

$$\sigma_\epsilon^2 = \frac{\gamma \sigma_\eta^2}{(2-\gamma A)AT_s} \quad (5.19)$$

On the other hand, the closed loop function $H_N(f)$ behaves like low pass filter with noise equivalent bandwidth $B_L \ll 1/T_s$ defined as follows,

$$B_L = \frac{\int_{-\infty}^{\infty}|H_N(f)|^2 df}{2|H_N(0)|^2} \quad Hz \tag{5.20}$$

From (5.15), we have $H_N(0) = A^{-1}$. Substituting (5.18) in (5.20),

$$B_L = \frac{\gamma A}{2(2-\gamma A)T_s} \ll 1/T_s \tag{5.21}$$

Substituting (5.21) in (5.19):

$$\sigma_\epsilon^2 = \frac{2B_L\sigma_\eta^2}{A^2} \tag{5.22}$$

The tracking jitter variance is one of the most important criteria to evaluate the DLL performance. Since, the tracking jitter σ_ϵ^2 should be as small as possible. The equations (5.19 and 5.22) indicate that the variance σ_ϵ^2 can be made as small as desired by increasing the slope parameter A or by reducing the step size γ (or its corresponding B_L). However, this may make the tracking time too long. Thus, a trade-off is required between tracking performance and its length.

5.2.2 Performance

In this section, we will evaluate tracking (Jitter) performance for the proposed first-order DLL. The performance will be enhanced by modifying the DT reference generator structure inside the DLL system. Further, we will study the proposed DLL system behavior, taking Doppler Effect into account. For Matlab simulation, we select the pulse $p(t)$ as the second derivative of the Gaussian function with unit energy and duration $T_p \approx 1$ns. Each symbol contains $N_f = 10$ frames with duration $T_f = 100$ns, the symbol duration is $T_s = N_f T_f = 1\mu s$. We employ TH spreading codes of period N_f, which is generated from a uniform distribution over $[0, N_c - 1]$, with $N_c = 9$, and $T_c = 10$ ns. The simulations are performed in a Saleh-Valenzuela channel [61].

For various values of data-aided bit number M and step size γ, the tracking jitter results normalized by T_p^2 are shown in Figure 5.5. The results show that the jitter is inversely proportional to SNR. We can also see that an increase in the number of data-aided bits M or a decrease in the step size γ lead to reduce the jitter and improve the tracking performance. For large values of SNR the different values of M have rather similar effect on the tracking jitter and the corresponding curves are approximately matched.

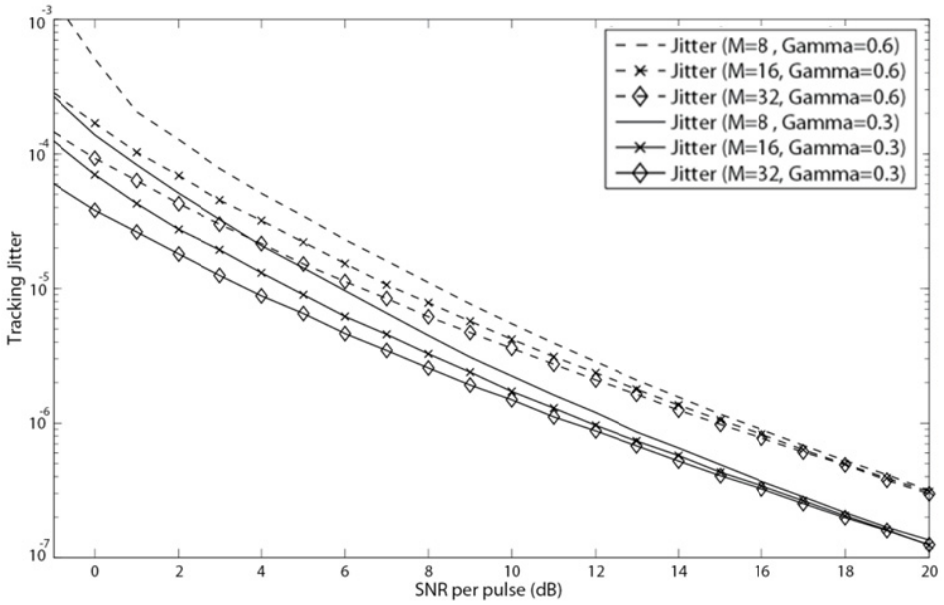


Figure 5.5 Tracking jitter for DLL

As we have mentioned before that dirty template tracking is based on the cross-correlation operation between the received signal and dirty reference template. We have also seen that the dirty reference template and its statistical properties are derived in (5.3 and 5.4). Another way to reduce the tracking jitter is to decrease the amount of noise term $\bar{w}(t)$ inside the template reference $y(t)$, this could be realized by passing $y(t)$ through a suitable window filter, such as in (4.34). So the modified dirty reference template becomes:

$$\acute{y}(t) = y(t) \cdot W(t) \tag{5.23}$$

Figure 5.6 illustrates the structure of this window filter. When the window size equals $1.5T_c$, the noise energy $\bar{w}(t)$ will be approximately divided by $(1.5\,T_c/T_f)$, and $\acute{y}(t)$ is given as:

$$\acute{y}(t) \cong \sqrt{\mathcal{E}_s}p_R(t) + \bar{w}(t): \quad \bar{w}(t) \sim \mathcal{N}\left(0, \bar{\sigma}^2 = \frac{N_0}{2M} \cdot \left(\frac{1.5T_c}{T_f}\right)^2\right) \tag{5.24}$$

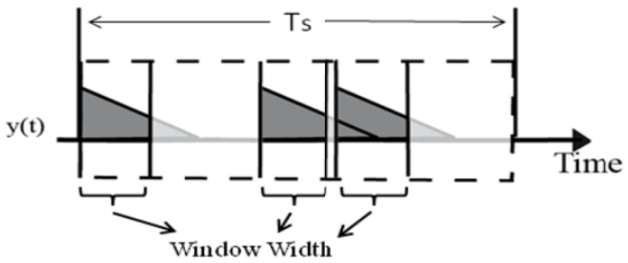


Figure 5.6 Window filter illustration

Figure 5.7 presents the comparison between the tracking jitter for DLL without windows (*dashed lines*) and DLL with windows (*solid lines*). It shows that the DLL with modified reference structure has higher tracking performance (smaller output noise variance) than the standard structure, and the jitter lines converge together with increasing SNR. In Figure 5.8, the slop of the s-curve A is plotted versus SNR for DLL with/without windows and for different M. Generally, we can notice from Figure 5.8 that the S-curve slop A is dependent on the noise part. For high SNR values or for DLL with windows, the noise effect on the S-curve function is relatively small and A is approximately fixed ($A \approx 1$). Cleary, for standard DLL model (without windows), the slope value A decreases when either SNR or M increases.

Note: In the rest of the chapter, to simplify the theoretical analysis, we employ the classical reference template $y(t)$ in (5.4). But Matlab simulation results will be based on the modified reference template $\acute{y}(t)$ in (5.24).

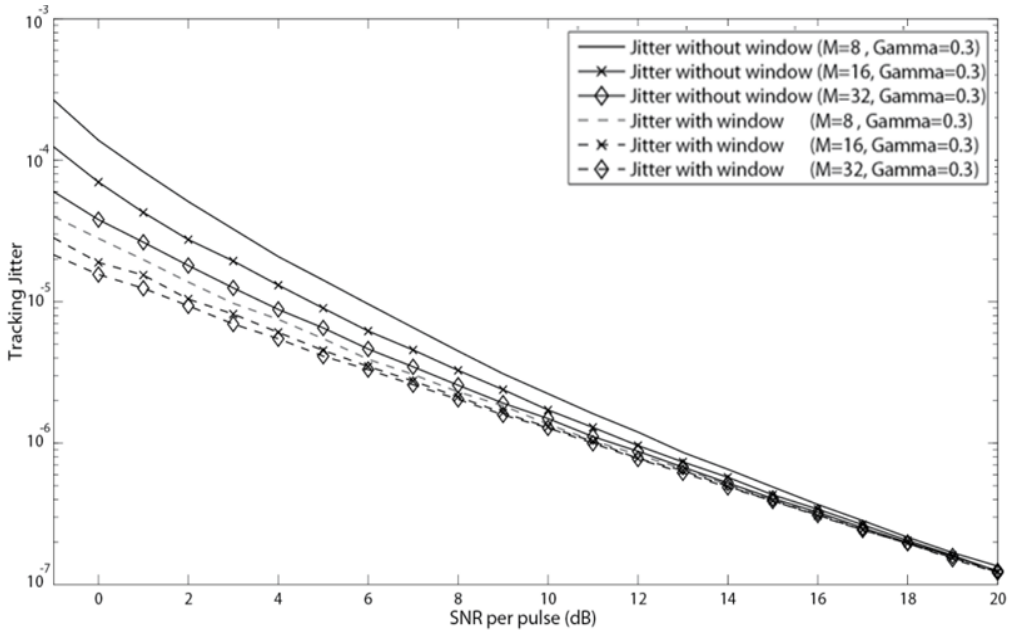


Figure 5.7 Tracking jitter for DLL without any window and DLL with window size 1.5Tc (γ = 0.3)

So far, we have studied the DLL behavior for different values of signal to noise ratio. The tracking jitter is one of an interest criterion to value the DLL performance with the presence of noise. On the other hand, to determine the optimal value of the loop parameter (γ), or more generally to select the loop structure order, we must take into account transient behavior of the loop as well as noise handling ability. For figuring out the transient behavior, we need to study the DLL system in the S-domain. To move from Z-domain to S-domain, we assume that $2\pi f T_s = \omega T_s \ll 1$, as in [95]. Then $z = e^{2\pi j f T_s} = e^{j\omega T_s} \approx 1 + j\omega T_s = 1 + sT_s$ where $s = j\omega$. The loop function $H_N(z)$ in (5.15) can be written in S-domain as:

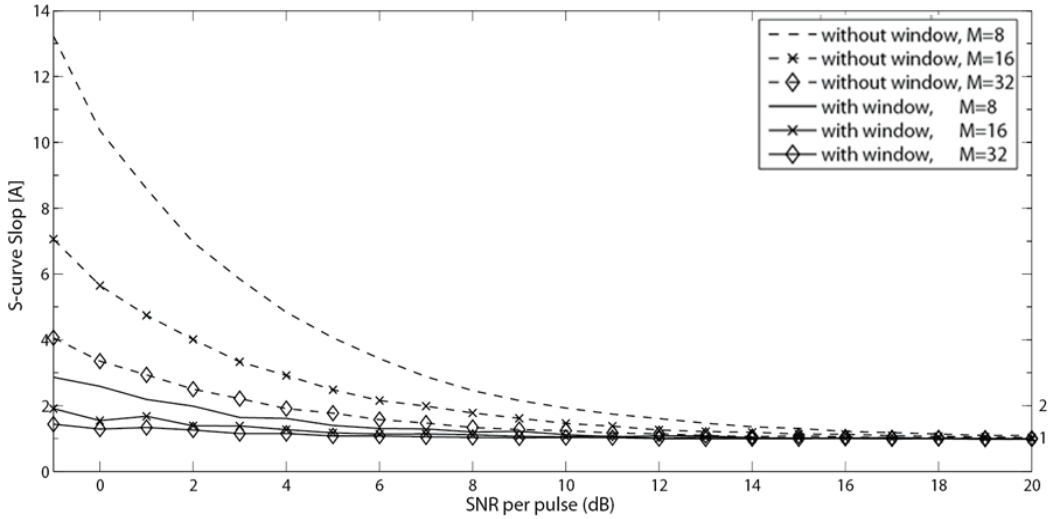


Figure 5.8 The slop of the discriminator characteristic vs. SNR

$$H_N(s) = \frac{-\gamma_0}{s+\gamma_0 A} \tag{5.25}$$

where $\gamma_0 = \gamma/T_s$. The offset decision function in (5.14) becomes $H(s) = \gamma/(T_s s) = \gamma_0/s$. Figure 5.9 shows the equivalent linear model of the tracking system in the S-domain. Let us consider the case where the dynamic input $\tau(t)$ has a ramp form, as:

$$\tau(t) = \tau_0 + mt \tag{5.26}$$

where τ_0 represents some constant time delay and m is a Doppler shift which is produced by a constant velocity relative to transmitter and receiver. The steady-state offset error is another criterion, which should be evaluated. It must be close to zero. In the absence of the noise loop $\eta(t)$, the offset error in the S-domain is given as follows, see Figure 5.9:

$$\epsilon(s) = \tau(s) - \hat{\tau}(s) = \tau(s) - A\gamma_0\epsilon(s)s^{-1} \tag{5.27}$$

$$\epsilon(s) = \frac{s}{s+A\gamma_0}\,\tau(s) \tag{5.28}$$

The steady-state error can be evaluated by means the final value theorem in Laplace transformation [88]:

$$\epsilon(t \to \infty) = \lim_{s\to 0} s\epsilon(s) = \lim_{s\to 0} \frac{s^2}{s+A\gamma_0}\tau(s) \tag{5.29}$$

The steady-state error depends on the loop structure and on the input signal $\tau(s)$ [88]. In our case, the Laplace transform of the dynamic input signal proposed in (5.26) is:

$$\tau(s) = \frac{\tau_0}{s} + \frac{m}{s^2} \tag{5.30}$$

Substituting (5.30) in (5.29), the steady-state offset error becomes:

$$\epsilon(t \to \infty) = \frac{m}{A\gamma_0} \neq 0 \tag{5.31}$$

Thus, if the dynamic input $\tau(t)$ has a ramp form, then the first-order loop in (5.28) is no longer able to track the input signal. To completely eliminate any steady-state errors resulting from the Doppler shift, the order of the DLL must be at least second-order [88].

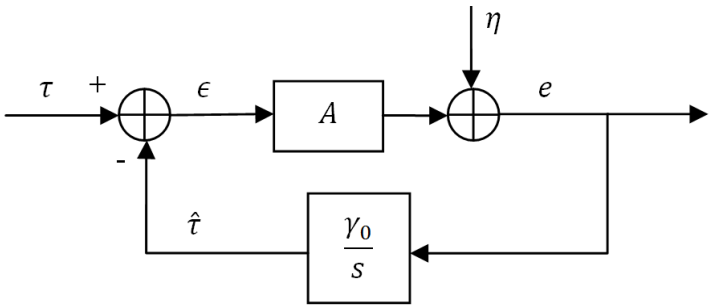


Figure 5.9 Linearized model of the DLL

5.3 Second-order DLL tracking

5.3.1 Structure & Model

For tracking a ramp input with no steady-state error remains $\epsilon(t \to \infty) = 0$, the second-order DLL is selected. Figure 5.10 depicts the suggested DLL structure with filter $F(s) = (1 + k_2 s)/(k_1 s)$. Assuming that the time offset is small to be inside the linear range of the S-curve. In line with the work in (5.27), the offset error for the second order DLL is written:

$$\epsilon(s) = \frac{s}{s + AF(s)}\,\tau(s); \tag{5.32}$$

Constants $\{k_1, k_2\}$ are loop parameters which will be found to optimize the tracking performance. If the dynamic input $\tau(s)$ satisfies (5.30), the evaluation of steady-state error is:

$$\epsilon(t \to \infty) = \lim_{s \to 0} s\epsilon(s) = \lim_{s \to 0} \frac{k_1 s^3}{k_1 s^2 + A(1 + k_2 s)} \left(\frac{\tau_0}{s} + \frac{m}{s^2}\right) = 0 \tag{5.33}$$

The closed loop transfer function and the loop noise bandwidth for the second-order DLL, plotted in Figure 5.10, can be found as (see [88: Table 3.3.1]):

$$H_N(s) = \frac{\epsilon(s)}{\eta(s)} = \frac{-1}{A} \times \frac{1+2\xi(s/\omega_n)}{(s/\omega_n)^2+2\xi(s/\omega_n)+1} \tag{5.34}$$

$$B_L = \frac{\xi\omega_n}{2}\left(1+\frac{1}{4\xi^2}\right) \ll 1/T_s \tag{5.35}$$

where the natural frequency $\omega_n = \sqrt{A/k_1}$ and the loop damping ratio $\xi = (k_2\omega_n)/2$. In the presence of the loop noise, the timing error $\epsilon(t)$ fluctuates around a steady-state point. The jitter variance for the second-loop DLL is derived by replacing the loop noise bandwidth B_L in (5.21) with (5.35). In addition to the jitter effect, the loop noise also contributes to the steady-state error; this phenomenon is called noise-induced drift [79: section 2.3.3]. Fortunately, this phenomenon vanishes in our tracking system and the steady-state remains always close to zero. The reason is that the statistical properties of the loop noise in (5.12) do not depend on the timing error ϵ (see [79]).

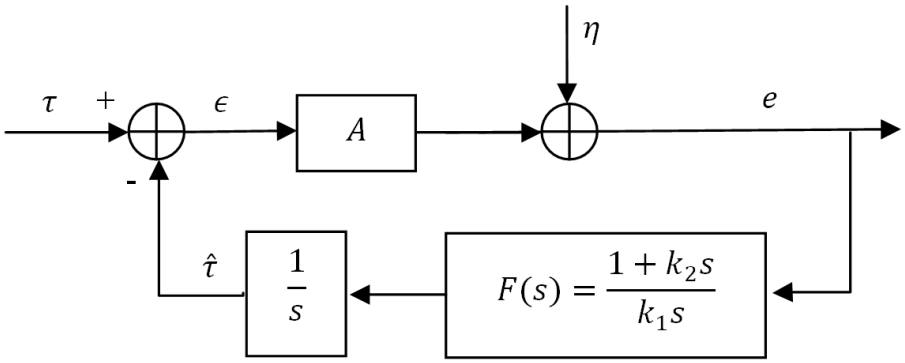


Figure 5.10 Model of the second-order DLL

With Doppler shifted input signal as in (5.26), DLL tracking is subject to two error sources: additive noise error and transient error. So, we are seeking to determine suitable values of the loop parameters $\{\omega_n, \xi\}$ for optimizing the performance criterion: minimizing noise jitter effect as well as maintaining specified transient error due to transmitter-receiver motion or oscillator drifts. For selecting $\{\omega_n, \xi\}$, we apply Wiener-filter theory [96]. This optimization method allows us to minimize the following design criterion (objective function): $\sigma_\epsilon^2 + \lambda^2\epsilon_T^2$, where σ_ϵ^2 is the error energy (variance) due to ambient noise, $\epsilon_T^2 = \int_0^\infty [\tau_0(t) - \tau(t)]^2 dt$ is the error energy due to transients, and λ is the Lagrange multiplier (considered as relative weight between noise and transient error energies). Based on Wiener-filter theory, the optimal second-order loop parameters are given as follows (see [88: Table 3.7.1]):

$$\omega_n^2 = m\,\lambda\,\sigma_\eta^{-1}, \quad \xi = \sqrt{2}/2 \tag{5.36}$$

Substituting (5.36) in (5.35), we get:

$$B_L = 0.53\omega_n \ll 1/T_s\ , \quad k_2 = \sqrt{2}/\omega_n\ , \quad k_1 = A/\omega_n^2 \tag{5.37}$$

Relaying on the above studies, we are now able to design a suitable second-order DLL and to select its optimal parameters in order to maintain the satisfactory synchronization, even when Doppler Effect is present and/or channel variations occur. Returning back to Figure 5.1, it shows that for each T_s period the tracking loop is achieved one time. Leading to estimate timing offset τ between $r(t)$ and $y(t)$, and compensating then this offset by shifting the reference signal position to $y(t-\hat{\tau})$, where $\hat{\tau}$ denotes the DLL estimate of τ. The received symbols $\hat{s}[n]$ can be eventually demodulated, using:

$$\hat{s}[n] = \text{ sign } \left[\int_0^{T_s} r(t)y(t-\hat{\tau})dt\right] \tag{5.38}$$

5.3.2 **Performance**

Let's now evaluate the proposed second-order DLL system behavior, taking into the consideration Doppler Effect. For Matlab simulation, we select the same UWB signal model which is proposed in the section 5.2.2, the dirty reference template is generated base on (5.24) with window filter size $= 1.5T_c$. Figures 5.11 and 5.12 show the mean square error (MSE) for various values of natural frequency ω_n and data-aided bit number M.

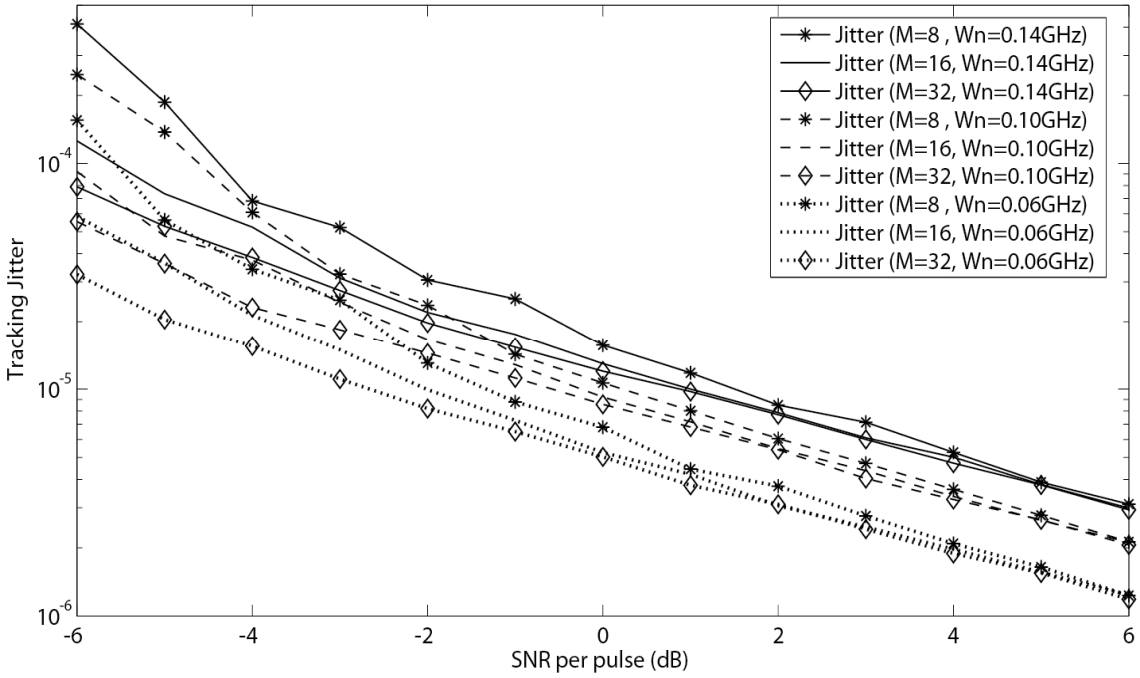


Figure 5.11 Tracking jitter for second-order DLL corresponding to small SNR values

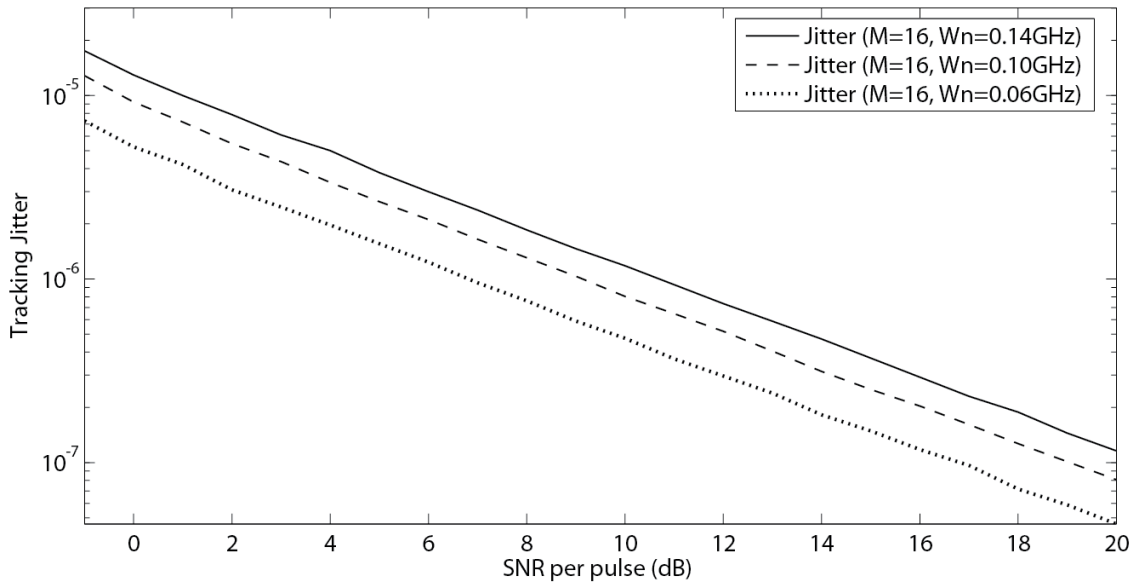


Figure 5.12 Tracking jitter for second-order DLL corresponding to large SNR values

In the previous figures, the MSE (jitter) results are normalized by T_p^2, and plotted versus the signal-to-noise ratio (SNR) per pulse. Doppler Effect is neglected $m = 0$. The simulations confirm that an increase in M or a decrease in ω_n leads to reduce noise energy effect and improves tracking performance. We can notice from Figure 5.11 that for SNR $\geq$ 5dB the jitter lines at various M are matched. For this reason, it is sufficient in Figure 5.12 to draw the curves only at a single value of M.

In order to evaluate the transient performance, the input signal time varies as follows $\tau(t) = mt$ with $m = T_p/100T_s$. We select $\lambda = 10^7$, and $\sigma_\eta^2 = 10^{-16}$; thus, the optimum natural frequency in (5.36) is $\omega_n = 0.1$ GHz. Since DLL operates digitally with sampling rate T_s, we have $\tau(n) = m \cdot nT_s = 0.01nT_p$. Figure 5.13 presents DLL behavior in the presence of input time variations $\tau(n)$ for different ω_n. It is seen that DLL follows the input signal $\tau(n)$ rapidly when the natural frequency increases. Another way of representing DLL transient performance is exhibited in Figure 5.14, which shows how many symbols (iteration steps) DLL requires to match efficiently $\tau(n)$. It is clear that when ω_n increases, the required number of symbols would decrease and transient response would be better.

The BER comparisons for various ω_n are depicted in Figure 5.15. The simulation has been executed two times: In Figure 5.15.a the BER comparisons occur while the input signal time is changing with the following form $\tau(n) = 0.01nT_p$. While in Figure 5.15.b the time is changing according to $\tau(n) = 0.1nT_p$. We observe that for high SNR values, an increase in ω_n (Simultaneously, B_L augments) leads to speed up the tracking operation and to improve BER performance.

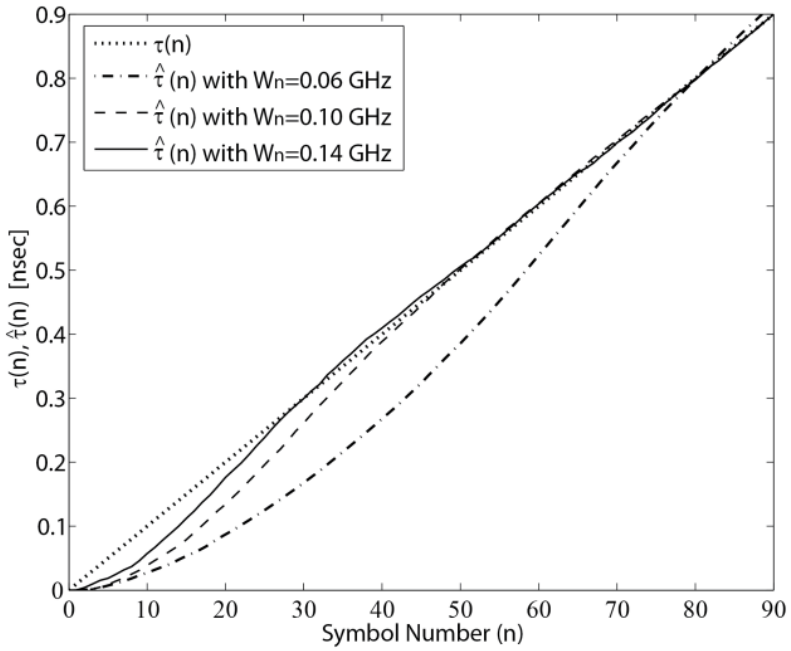


Figure 5.13 DLL transient response for ramp input (SNR=3 dB)

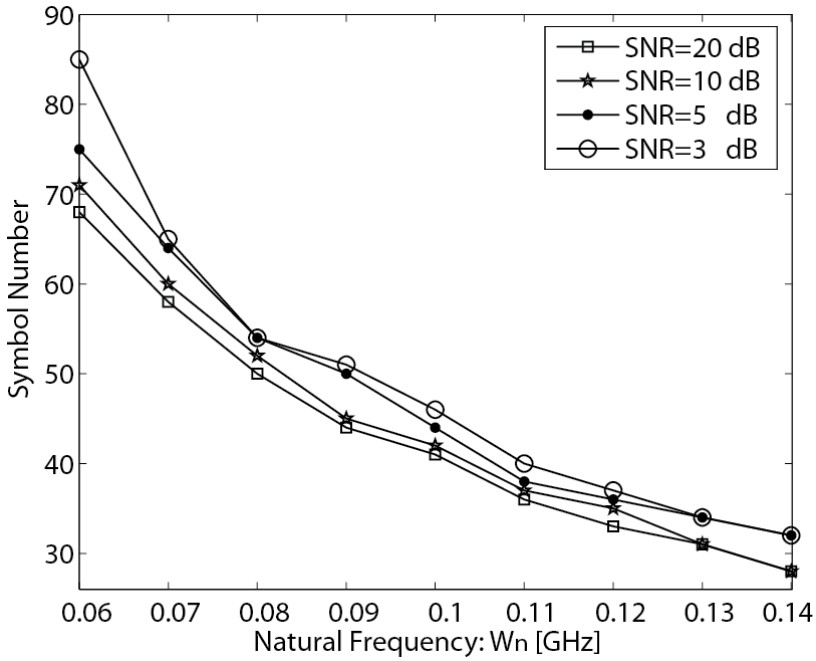
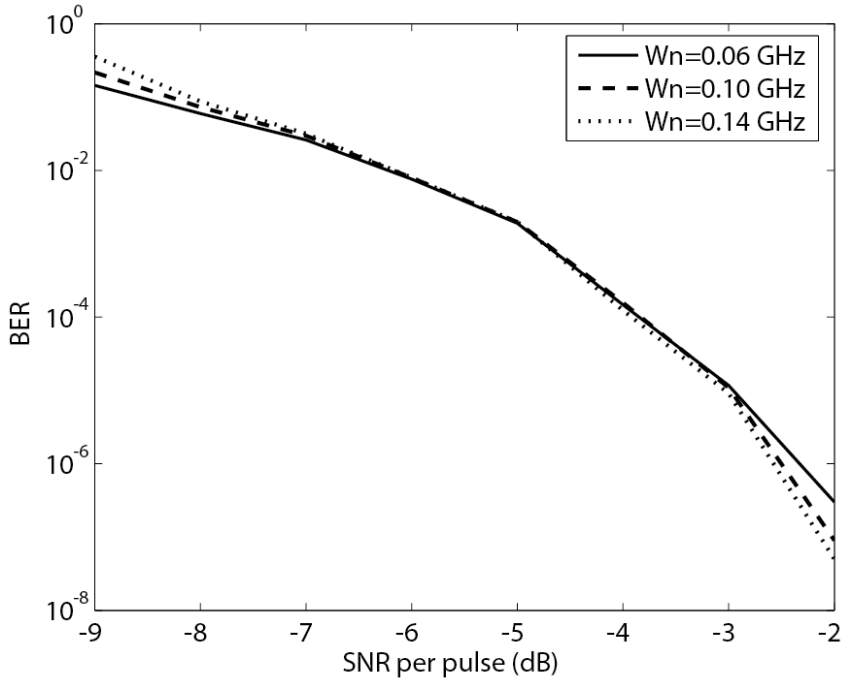


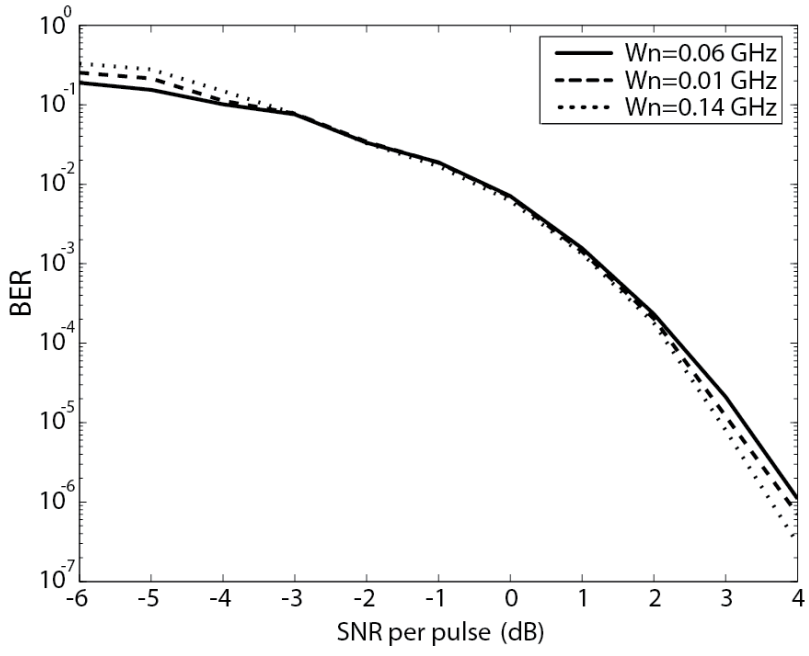
Figure 5.14 DLL transient performance (symbol number vs. natural frequency)

On the other hand, for low SNR values, the bandwidth B_L should be decreased for alleviating the noise effects, but that degrades the transient performance. Consequently, increasing ω_n helps to reduce the transient error effect, but in the price of noise handling ability. We also

notice that increasing the velocity of variations of the input signal leads to significant tracking performance decrease.



(a)



(b)

Figure 5.15 BER for the proposed DLL with $M = 16$: (a) $\tau(n) = 0.01nT_p$, (b) $\tau(n) = 0.1nT_p$

5.4 Practical considerations

Figure 5.16 summarizes the synchronization system based on dirty template technique, which consists of three main blocks: signal detector, timing acquisition, and tracking. In the presence of the noise and with the channel variations, the DLL might be derived out of lock. We have mentioned in (2.17 and 2.18) that the received UWB signal $r(t)$ consists of a large number of channel paths. So when the DLL falls out of lock, it begins to track with a wrong path instead of the first one. It has been found in [95] that dirty template demodulator is able to perform efficiently symbol detecting, even if DLL is tracking another path close to the first one. In Figure 5.16, a lock detector is employed to measure capture energy at the correlator output and to compare it versus a threshold, for deciding if the DLL remains in lock or it is out of lock. Once the lose-lock signal is generated, the receiver turns back to the acquisition mode. After executing successfully the acquisition phase, the receiver enters again the tracking phase and DLL is brought back in lock.

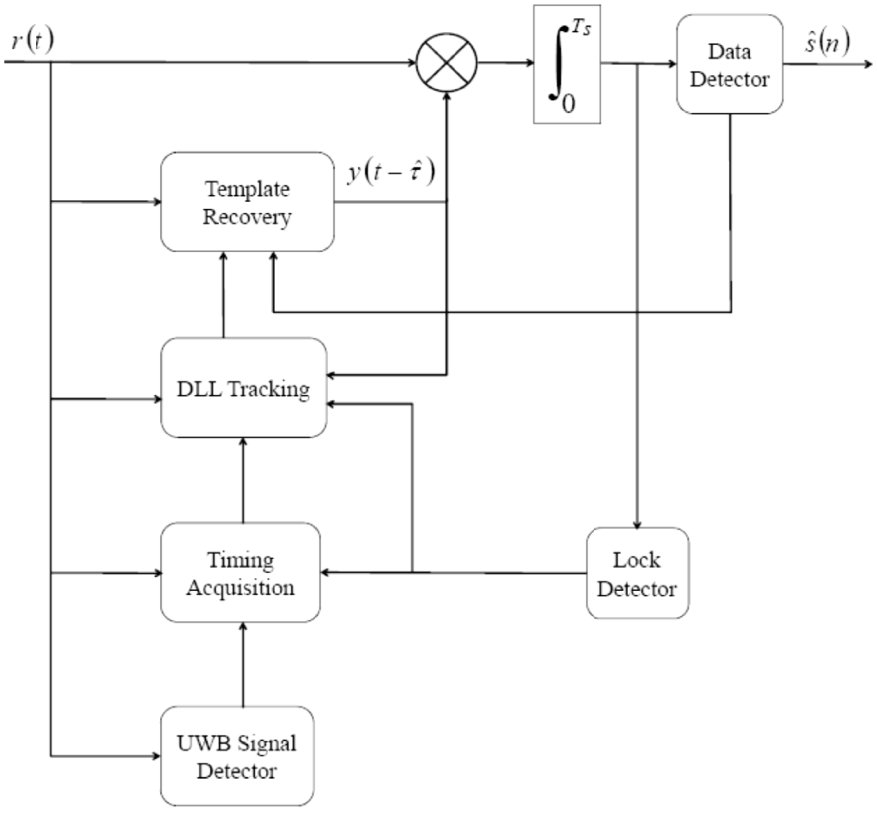


Figure 5.16 Block diagram of the synchronization system based on dirty template technique

In UWB channel, due to the relative movement between transmitter and receiver (or object movement inside the signal propagation field), the multipath channel parameters vary slowly

with time. Consequently, the reference signal $y(t)$, extracted in (5.4), becomes outdated and is not efficiently similar to T_s-long received segments. This leads to degrade gradually synchronization performance and increase BER. To avoid this problem, The dirty template $y(t)$, which is used for tracking the received signal and for estimating $\hat{s}[n]$, should be updated, as follows:

$$y(t) = \frac{1}{K}\sum_{n=1}^{K} \hat{s}[n]r(t + nT_s)\,; \quad t \in [0,T_s) \tag{5.39}$$

where K denotes the averaging size. Update of the template $y(t)$, does not need to be continuous, only periodic with a frequency $(1/T_y)$, where update period T_y must be less than the channel coherence time ($T_y \ll T_{coh}$). On the other hand, by considering that UWB pulses are ideal candidates for indoor communication applications, where the expected speed of objects or users is typically in the order of $v = 1\ m/s$. With maximum carrier frequency for UWB-IR $f_c = 6$ GHz, the maximum Doppler spread is given as: $f_D = v/\lambda = (v/c) \times f_c = 20$ Hz, where c is the speed of light, and λ is the wavelength. As discussed in [97], the channel coherence time is about $T_{coh} = 0.423/f_D \approx 21$ msec. Thus, updating the DT reference $y(t)$ must be done periodically with T_y less than 21 msec.

5.5 Conclusion

In this chapter, we derive first-order and second-order DLL designs used for maintaining satisfactory synchronization between transmitter and receiver. We combine DLL with TDT, which enables enhancing tracking behaviour and reducing receiver structure even when the multipath channel is unknown. The reference template is further modified by adding an appropriate window filter that allows reducing the ambient noise effects on the tracking operation. The theoretical analysis confirms that in the presence of Doppler Effect, the first-order DLL isn't able to track the timing variations of input signal without any steady-state error remains. For that the second-order DLL is suggested as an efficient alternative structure. We apply further Wiener-filter theory to optimize the suggested DLL parameters. Simulation results show the performance of the proposed DLL and confirm that increasing natural frequency ω_n helps to reduce the transient error effect, but at the price of noise handling ability.

While in this chapter we have designed the closed-loop controller (offset decision block) using conventional methods: iterative method and Winner filter. In chapters 6 and 7, we are seeking to design the control unit of the DLL by employing supervisory control concepts, based on a novel and promising technique for radio and telecommunication applications which called: Internal Model Control (IMC).

Chapter 6

Internal Model Control

We have explained in chapter 5 how DLL tracking system can efficiently be designed by employing conventional methods. In this chapter we will focus on the design of DLL tracking system, in particular, the offset decision block structure, using supervisory control concepts. The problem we encounter with the classical approach is that the DLL system is non-linear and it is so difficult to design an appropriate closed-loop filter without taking into account stability and robustness issues. To avoid the complexity of such a study, we present an original DLL structure used to achieve satisfactory and accurate tracking. The proposed DLL scheme is based on Internal Model Control (IMC). This technique is a well-known and widely used in various automatic control areas, offering high performance in robustness for disturbance rejection, as well as offering global stability for both linear and non-linear systems. The main restriction on the IMC control is that the system under control needs to be open loop stable [98-101]. Design procedure and implementation form for IMC controller are quite extensive, including single-input single output (SISO) and multi-input multi-output (MIMO) structures, feedback and feedforward controller forms, design procedure for continuous-time and discrete-time applications, etc [102]. In this thesis we will focus only on how to design feedback SISO IMC controller for discrete-time systems.

This chapter is organized as follows: First, we will give an overview of Internal Model Control concept used for discrete-time applications. Then, we will explain in depth how to integrate IMC concept in DLL tracking system. We will see that IMC controller is composed of two blocks (models): an inverse model (control model) connected in series with the DLL system and a forward model (system model) connected in parallel with the DLL system. This structure has a good performance of overcoming disturbance and deviations of model parameters. Next, we will apply Least-Squares (LS) estimation algorithm methods in order to determine the optimal coefficients for the system and control models. This chapter focuses on designing IMC DLL system and analyzing the tracking behavior of the proposed system in the

absence of Doppler Effect. Simulation results study proposed IMC tracking performance across various values of data-aided bit number and signal-to-noise ratio.

6.1 Internal model control basic principle

IMC controller usually consists of two blocks: the first (Process Model Block) should match perfectly the tracking system and the second (Control Block) represents the inverse model of the system. The IMC strategy has the general structure depicted in Figure 6.1.

In the Figure 6.1 H_c is the control (inverse) model, H_p is the system (process), $\tilde{H}_p$ is process (direct) model, r is the (set-point) input signal, d is an unknown disturbance affecting the system, and $[y, u]$ are the (process) system output and input. The process output y is compared with the output of the model $\hat{y}$, resulting in a feedback signal $\hat{d}$. That is:

$$\hat{d}(z) = \left[H_p(z) - \tilde{H}_p(z)\right] \cdot u(z) + d(z) \tag{6.1}$$

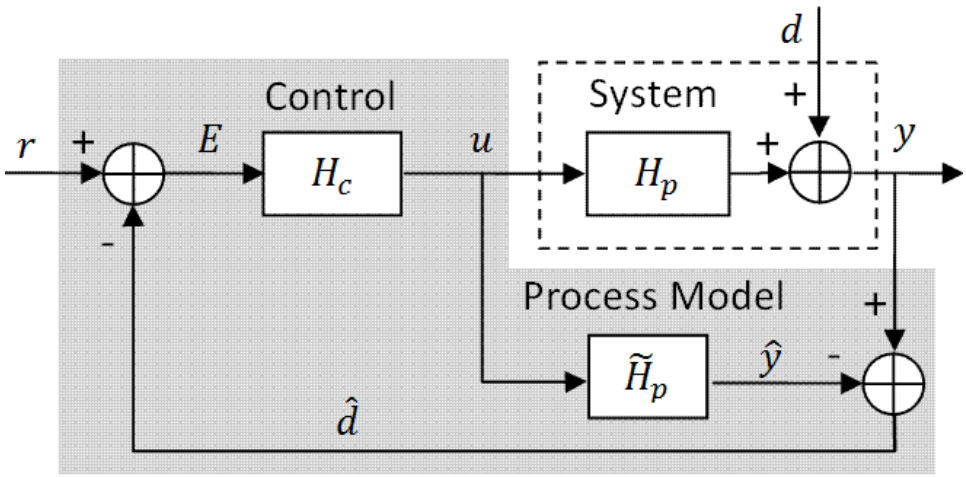


Figure 6.1 Internal model control system

Assuming $d(z) = 0$, then $\hat{d}(z)$ is considered as a measure of the difference in behavior between the process $H_p(z)$ and its model $\tilde{H}_p(z)$. On the other hand, if $H_p(z) = \tilde{H}_p(z)$, then $\hat{d}(z)$ represents the unknown disturbance. Thus $\hat{d}(z)$ can be regarded as the information missing in the model $\tilde{H}_p(z)$, and can therefore be used to improve control. This improvement could be achieved by subtracting $\hat{d}(z)$ from the setpoint $r(z)$. The resulting control signal $u(z)$ is given as follows:

$$u(z) = \left[r(z) - \hat{d}(z)\right] \cdot H_c(z) = \left\{r(z) - \left[H_p(z) - \tilde{H}_p(z)\right] \cdot u(z) + d(z)\right\} \cdot H_c(z)$$

Thus, $u(z) = \frac{[r(z)-d(z)]\cdot H_c(z)}{1+[H_p(z)-\tilde{H}_p(z)]\cdot H_c(z)}$ (6.2)

Since $y(z) = u(z) \cdot H_p(z) + d(z)$. Therefore, the system output could be expressed as:

$$y(z) = \frac{H_c(z)\cdot H_p(z)}{1+[H_p(z)-\tilde{H}_p(z)]\cdot H_c(z)} r(z) + \frac{[1-\tilde{H}_p(z)\cdot H_c(z)]}{1+[H_p(z)-\tilde{H}_p(z)]\cdot H_c(z)} d(z) \quad (6.3)$$

We can see clearly from (6.3) that if the model matches exactly the process $\tilde{H}_p(z) = H_p(z)$, and the control matches the inverse of the model $H_c(z) = \tilde{H}_p(z)^{-1}$, then perfect set-point tracking and disturbance rejecting are achieved. As a result, the IMC system is often used to make the output y track set-point r perfectly, and to reject or minimize the effects of external disturbances d. We notice also from (6.3), even if $\tilde{H}_p(z) \neq H_p(z)$, the unknown disturbance can still be rejected perfectly by keeping $H_c(z) = \tilde{H}_p(z)^{-1}$.

6.2 Modified DLL system

As we have demonstrated in chapter 5 the discriminator error e at the output of DLL tracking is plotted in Figure 6.2 and expressed by the following equation, see (5.10 and 5.11):

$$e[\epsilon, k] = \underbrace{\Gamma_e[\epsilon] - \Gamma_l[\epsilon]}_{\mathbb{S}[\epsilon]} + \eta[k] = \mathbb{S}[\epsilon] + \eta[k] \quad (6.4)$$

where $\eta[k]$ is the equivalent additive noise for DLL tracking system, $\Gamma_e[\epsilon]$ & $\Gamma_l[\epsilon]$ are noise-free early and late correlator outputs, and $\mathbb{S}[\epsilon]$ is the loop discriminator characteristic (S-curve) drawn in Figure 6.3.

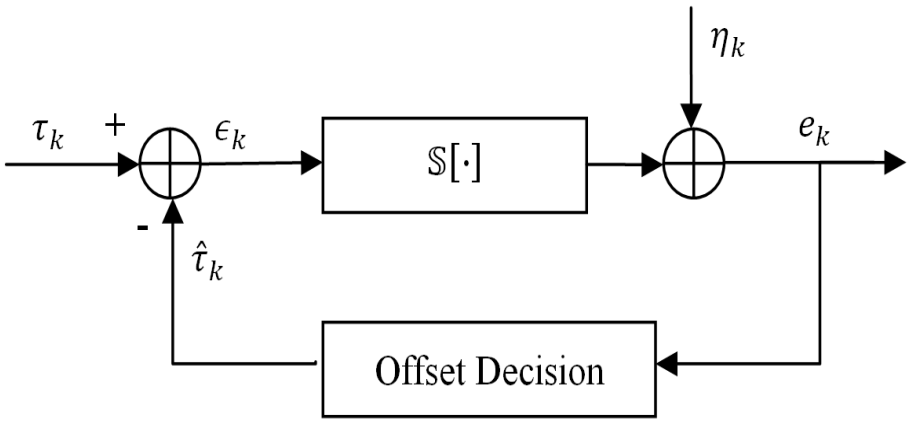


Figure 6.2 Equivalent timing model for the DLL tracking

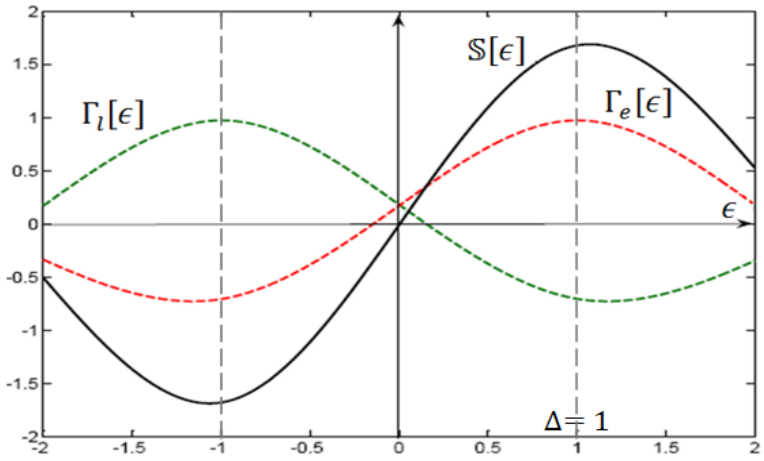


Figure 6.3 Loop discriminator characteristics (S-curve) for the DLL

The objective of IMC technique here is to design an adequate offset decision block, which is derived by the output of DLL system $e(\epsilon)$. The output of the decision block gives the time offset estimate $\hat{\tau}$ which is regarded as a controllable parameter, is used to reduce the offset error $\epsilon = \tau - \hat{\tau}$ between the received and the reference signals and to therefore improve the system performance.

As we have seen in the previous section, IMC controller allows good set-point tracking as well as perfect disturbance rejecting. In this section, we will carry out IMC strategy in DLL system to efficiently control on the system input parameter $\hat{\tau}$, for reducing the offset ϵ and making the system output error e be close to zero regardless of the ambient noise effects. Figure 6.4 illustrates the equivalent DLL tracking diagram including the IMC regulation loop. We have mentioned in the last section that IMC method is used for making the output signal e track the set-point input r. So in our DLL system the desired value of the set-point input r, for making the system output error e be close to zero, must intuitively be equal to zero.

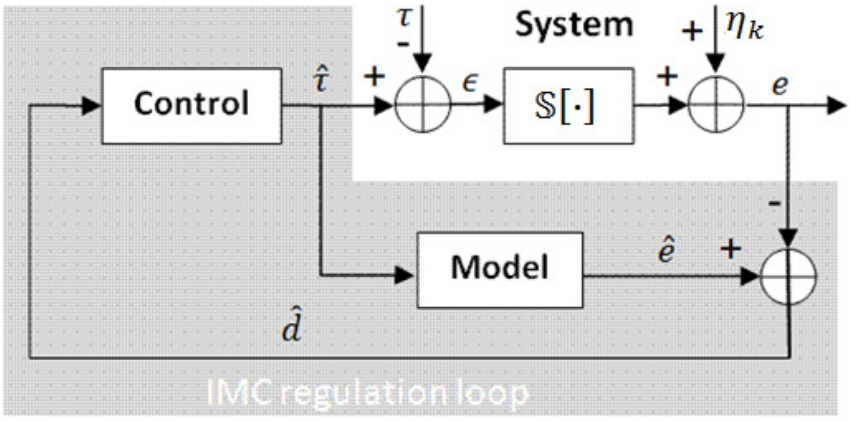


Figure 6.4 Equivalent timing model for modified tracking system

We will focus now on defining a suitable model for the system (Direct model) and control (Inverse model). This is achieved by selecting firstly relevant model structure, and computing then optimal model coefficients. Above all, We should emphasize that we are seeking a convenient model for DLL process which has an S-curve form mentioned in Figure 6.3 and for its inverse. However, for tracking application we are interested to study system behaviour only within the range $\epsilon \in [-\Delta, \Delta]$ (see Figure 6.3). The model design could be established by using the empirical procedure, which has the following outline (shown Figure 6.5) [30]:

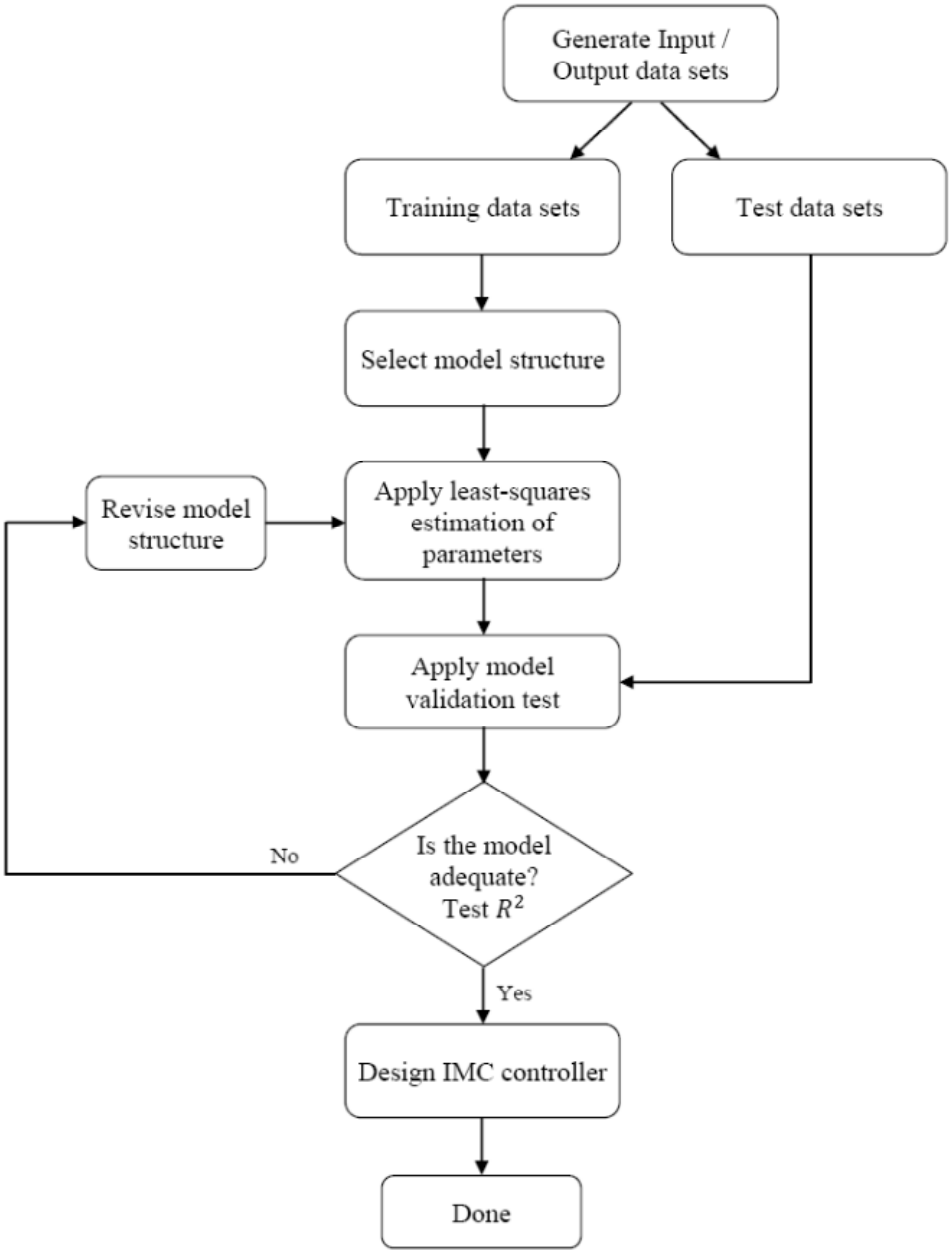


Figure 6.5 Process identification procedure

6.2.1 Input/output data set generation

The first step for IMC synthesis is to generate set of Input/output data. The Early-Late Gate (ELG) scheme plotted in Figure 6.6 is used to establish the training data set, by changing the input signal $\hat{\tau}_i$ from $-\Delta$ to $+\Delta$ with a small step, and observing the corresponding e_i at the output. This set of input/output data (U, V) will later be used for computing the model parameters.

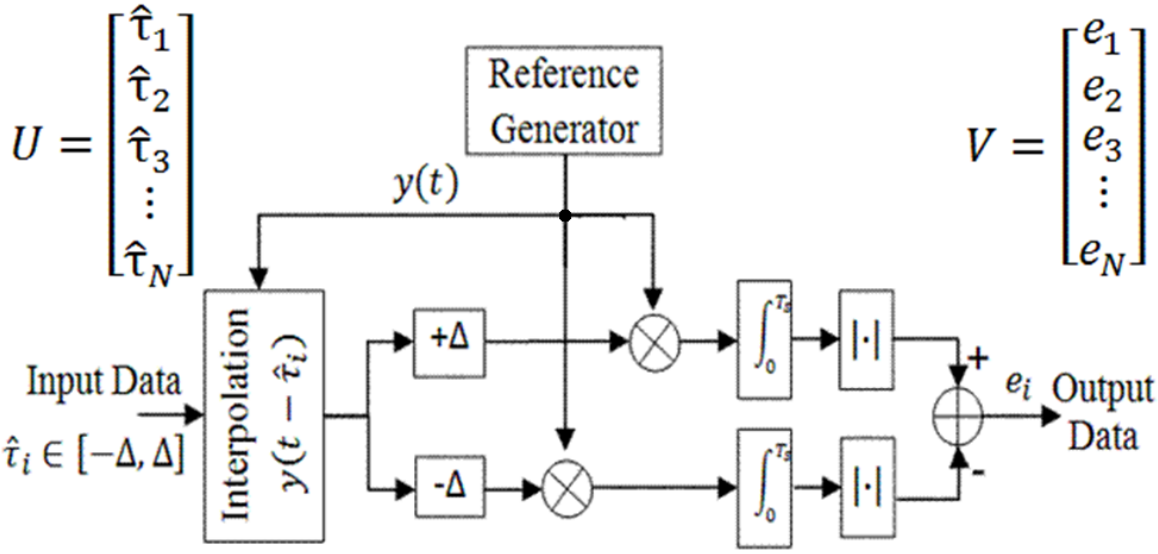


Figure 6.6 Data sets generator

6.2.2 Model structure selection for the system and the control

In this phase, we are seeking the polynomial function which could effectively express the S-curve system and its inverse, drawn in Figure 6.7. Although the higher-order polynomial models can provide a better fit for the data than the lower-order models do, it is recommended to select a model with a low order which could sufficiently fit the system. The reason is that higher-order models contain large number of coefficients (parameters), which require much more input/output system information and more complex mathematical operations in order to determine accurately the value of these parameters [103].

The following expression of order 3 could efficiently be employed to express the structure of system and control models:

$$y(t) = \theta_1 x(t) + \theta_3 x^3(t) \tag{6.5}$$

where $\{\theta_1, \theta_3\}$ are the polynomial function coefficients (parameters). The main reason of selecting $y(t)$ function as a model structure is that the DLL system, shown in Figure 6.3, has always odd function form: $f(-x) = -f(x)$ and it is rotational symmetry with respect to the origin (zero). Therefore, we have suggested a polynomial function with odd-degree terms $(x, x^3, \dots)$. After a model structure is selected, the parameters are computed to minimize the error between model prediction and actual observations. Standard Least-Squares (LS) method could be employed here for extracting the optimal function parameters [104].

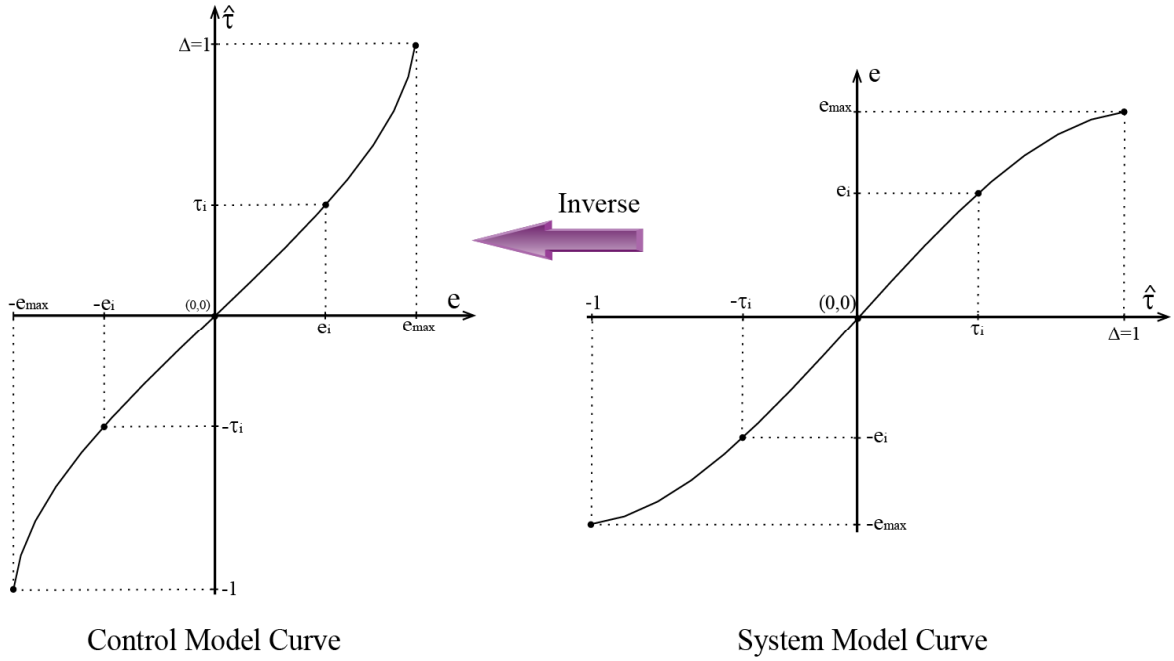


Figure 6.7 System and control model curves

6.2.3 **Least-squares estimation method**

In this phase the identification problem will be formulated as a least-squares parameter estimation problem, we will then derive the Least-Squares (LS) parameter solution based on generated data. Initially, we have:

$$\varepsilon_i = \hat{y}_i - y_i \ : i = 1, \dots, N \tag{6.6}$$

where ε_i represents the measurement error, y_i represents a variable that can be modeled, $\hat{y}_i$ is the estimated value of y_i given by the LS model and N denotes the set number of input/output data (points). Let us say that due to noise in the measurements, these set of points (U, V) do not exactly fit the model of (6.5) perfectly. So we replace it by the following:

$$y_i = \theta_1 x_i + \theta_3 x_i^3 + \varepsilon_i \quad : i = 1, \dots, N \tag{6.7}$$

We can write the expression (6.7) in expanded form, as:

$$y_1 = \theta_1 x_1 + \theta_3 x_1^3 + \varepsilon_1$$
$$y_2 = \theta_1 x_2 + \theta_3 x_2^3 + \varepsilon_2$$
$$\cdot$$
$$y_i = \theta_1 x_i + \theta_3 x_i^3 + \varepsilon_i$$
$$\vdots$$
$$y_N = \theta_1 x_N + \theta_3 x_N^3 + \varepsilon_N$$

This can be written compactly in matrix notation, as:

$$Y = X \cdot \theta + E \tag{6.8}$$

where: $Y = \begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ y_i \\ \vdots \\ y_N \end{bmatrix}$; $X = \begin{bmatrix} x_1 & x_1^3 \\ x_2 & x_2^3 \\ \cdot & \cdot \\ x_i & x_i^3 \\ \vdots & \vdots \\ x_N & x_N^3 \end{bmatrix}$; $\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$; $E = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \cdot \\ \varepsilon_i \\ \vdots \\ \varepsilon_N \end{bmatrix}$

We seek the least-squares solution that minimizes the sum of the square of the measurement errors. So LS objective function is:

$$J = \sum_{i=1}^{N} \varepsilon_i^2 = \|E\|^2 = E^T \cdot E \tag{6.9}$$

where ($.^T$) denotes the transpose of the matrix. Substituting (6.8) in (6.9), we get:

$$J = (Y - X\theta)^T \cdot (Y - X\theta) \tag{6.10}$$
$$= Y^T Y + \theta^T X^T X\theta - Y^T X\theta - \theta^T X^T Y$$

The optimal value of θ that minimizes the LS error is defined by follows:

$$\hat{\theta} = \arg\min_{\theta} J \Leftrightarrow \left.\frac{\partial J}{\partial \theta}\right|_{\theta=\hat{\theta}} = 2X^T X\theta - 2X^T Y = 0 \tag{6.11}$$

Thus, the least-squares solution is:

$$\hat{\theta} = (X^T \cdot X)^{-1} X^T \cdot Y \tag{6.12}$$

6.2.4 Direct & Inverse model synthesis

The purpose of this phase is to compute the optimal parameters for Direct & Inverse models. First, we employ the least square method to estimate the optimal parameters for the Direct (system) model $\widetilde{H}_p$, where the input data for this model is $\hat{\tau}_i$, the output data is e_i, and the model parameters are $\{a_1, a_3\}$. Thus:

$$x_i = \hat{\tau}_i; \quad y_i = e_i; \quad \theta = \begin{bmatrix} a_1 \\ a_3 \end{bmatrix} \tag{6.13}$$

Then, we repeat the LS algorithm to determine the parameters of the Inverse (control) model H_c, which is the inverse of the Direct (system) model. Therefore, e_i denotes the input data and $\hat{\tau}_i$ denotes the output data in (6.12). $\{b_1, b_3\}$ represent the control model parameters.

$$x_i = e_i; \quad y_i = \hat{\tau}_i\,; \quad \theta = \begin{bmatrix} b_1 \\ b_3 \end{bmatrix} \tag{6.14}$$

6.2.5 Identification & System validation

After selecting a simple adequate model structure and determining relevant parameters for both process and control blocks. We should then verify the accuracy of the parameters estimated as well as the accuracy of the proposed model itself. There are several statistical measures that we can use to evaluate the goodness of fit of a suggested model such as Kolmogorov-Smirnov test, Pearson's chi-squared test and F-test [72, 105]. For evaluating identification accuracy of third-order polynomial model we suggest one of common measures called *determination coefficient* R^2 [102, 106]. It is defined as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2} \tag{6.15}$$

where $\hat{y}_i = X_i \cdot \theta = \theta_1 x_i + \theta_3 x_i^3$ and $\bar{y}_i = \sum_{i=1}^{N} y_i / N$. In our example R^2 ranges from 0 to 1, small values of R^2 (close to zero) indicate poor fit, whereas values close to one indicate better fit, $R^2 = 1$ indicates that the model perfectly fits the data. If we assume that the calculated value of R^2 is relatively small, we must in this case change the model structure, in particular, increase the order of the polynomial, so that we may be able to reduce the regression error and enhance R^2 value but at the expense of increasing parameter number, leading to increase the size of matrix X and raise the complexity of least square regression operation in (6.12).

For properly evaluating identification performance it is not recommended to use the training data sets which have already been employed for designing control system model, while fit

evaluation will be much better when we apply to new data sets [102]. Thus the available date can be divided into two group. The first called the training data group is used to compute the adequate least-squares parameters, while the second group is used to determine model validation (estimate R^2).

To sum up, we have found that the system and control models could be estimated using an identification process. Both models are next used to design the IMC controller for DLL application as shown in Figure 6.4.

Practical consideration: In UWB indoor applications, the multipath channel parameters vary slowly with time. As we have mentioned in section 5.4 that (dirty template) DT-DLL system extracts a reference signal from the received dirty signal. So the reference signal must be periodically updated to avoid transmission quality degradation. Since DLL process is extremely associated with propagation channel and ambient noise energy, so it changes also with time, leading sometimes to degrade control system performance. Therefore, it is better to apply the process identification procedure and update the control system parameters each time after updating the dirty reference signal. On the other side, by looking at Figure 5.16, lock detector block can help us to constantly monitor control system performance in order to determine whether model update is really required or not.

6.3 Numerical application

Let's take a simple numerical case to provide more clarification on the identification steps. Consider time offset input $\hat{\tau}$ varied gradually from $+\Delta = +1$ to $-\Delta = -1$ with a step equal to 0.25, and assume SNR=3dB and data-aided bits $M = 16$.. Observing the output of the generator circuit showed in Figure 6.6 we get the following data:

TABLE 6.1 Input / output training data sets

i	**Input,** $\hat{\tau}_i$	**Output,** e_i
1	1	0.95681
2	0.75	0.8524
3	0.5	0.63246
4	0.25	0.33548
5	0	0

i	**Input,** $\hat{\tau}_i$	**Output,** e_i
6	-0.25	- 0.33548
7	-0.5	- 0.63246
8	-0.75	- 0.8524
9	-1	- 0.95681

− Let's first extract the system model. On applying a third order polynomial model presented in (6.5) to this data, where:

$$Y = \begin{bmatrix} 0.95681 \\ 0.8524 \\ 0.63246 \\ 0.33548 \\ 0 \end{bmatrix};\ X = \begin{bmatrix} 1 & 1 \\ 0.75 & 0.75^3 \\ 0.5 & 0.5^3 \\ 0.25 & 0.25^3 \\ 0 & 0 \end{bmatrix};\ \theta = \begin{bmatrix} a_1 \\ a_3 \end{bmatrix}$$

It is important to mention that it is sufficient to utilize only positive sets of generated data from Table 6.1, because negative data sets have exactly the same (information) values but with opposite sign. It is normal that the generated output data are symmetric, since the generator, mentioned in Figure 6.6, correlates of the signal $y(t)$ with itself (auto-correlation). As a result, using only data sets of one side (either positive or negative) will help to simplify least-squares operation and avoid generating an ill-conditioned matrix (X) [102, 107]. Briefly, we say a certain matrix is ill-conditioned when that matrix is so difficult to inverse accurately, so that the least-squares solution represented in (6.12) may not be efficiently correct. The main reason of having an ill-conditioned matrix is when training data is insufficient or redundant. The least-squares solution is given by:

$$\begin{bmatrix} a_1 \\ a_3 \end{bmatrix} = (X^T \cdot X)^{-1} X^T \cdot Y = \begin{bmatrix} 1.3676 \\ -0.4108 \end{bmatrix} \Rightarrow a_1 = 1.3676,\ a_3 = -0.4108$$

Hence the process model is done by:

$$\hat{e} = 1.3676 \cdot \hat{\tau} - 0.4108 \cdot \hat{\tau}^3 \tag{6.16}$$

− Repeating the same procedure for designing the control model, so we apply again (6.5) but the variables are given as:

$$Y = \begin{bmatrix} 1 \\ 0.75 \\ 0.5 \\ 0.25 \\ 0 \end{bmatrix};\ X = \begin{bmatrix} 0.95681 & 0.95681^3 \\ 0.8524 & 0.8524^3 \\ 0.63246 & 0.63246^3 \\ 0.33548 & 0.33548^3 \\ 0 & 0 \end{bmatrix};\ \theta = \begin{bmatrix} b_1 \\ b_3 \end{bmatrix}$$

The least-squares solution is given by:

$$\begin{bmatrix} b_1 \\ b_3 \end{bmatrix} = (X^T \cdot X)^{-1} X^T \cdot Y = \begin{bmatrix} 0.61752 \\ 0.43262 \end{bmatrix} \Rightarrow b_1 = 0.61752,\ b_3 = 0.43262$$

Hence the control model is done by (where $\hat{d}$ is presented in Figure 6.4):

$$\hat{\tau} = 0.61752 \cdot \hat{d} + 0.43262 \cdot \hat{d}^3 \tag{6.17}$$

− System & Control model validation: Consider the data used to evaluate identification accuracy given in Table 6.2. We apply the equations (6.16) and (6.17) to estimate e_i and $\hat{\tau}_i$ correspondingly, where $\hat{d}_i = e_i$. Then, the determination coefficients are calculated by applying (6.15).

TABLE 6.2 Input / output test data sets

i	Input, $\hat{\tau}_i$	Output, e_i
1	0.875	0.90759
2	0.625	0.74617
3	0.375	0.48686
4	0.125	0.16881
5	-0.125	-0.16881
6	-0.375	-0.48686
7	-0.625	-0.74617
8	-0.875	-0.90759

Applying (6.16 &6.17)

Estimate $\hat{\tau}_i$	Estimate e_i
0.88388	0.92145
0.6405	0.75446
0.35057	0.49119
0.10633	0.17015
-0.10633	-0.17015
-0.35057	-0.49119
-0.6405	-0.75446
-0.88388	-0.92145

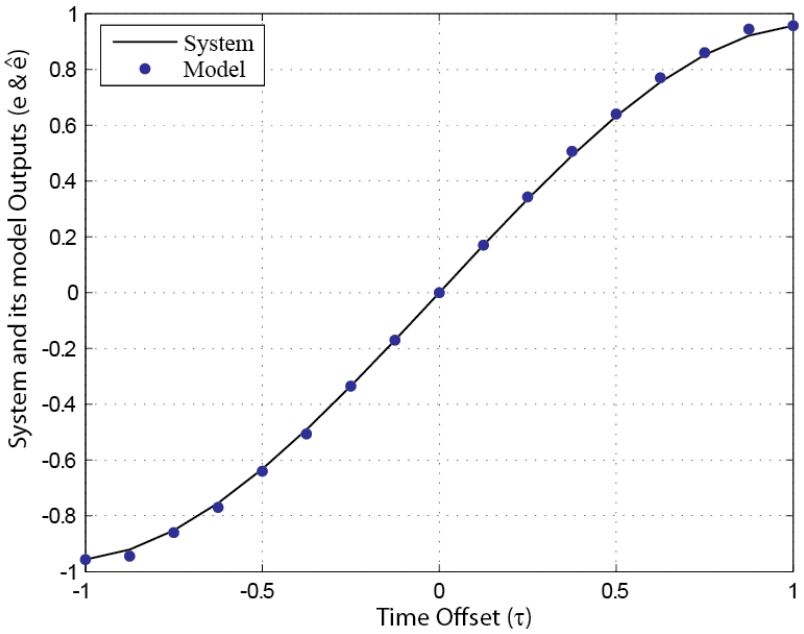
Applying (6.15)

$R^2_{inverse}$	R^2_{direct}
0.999	0.9998

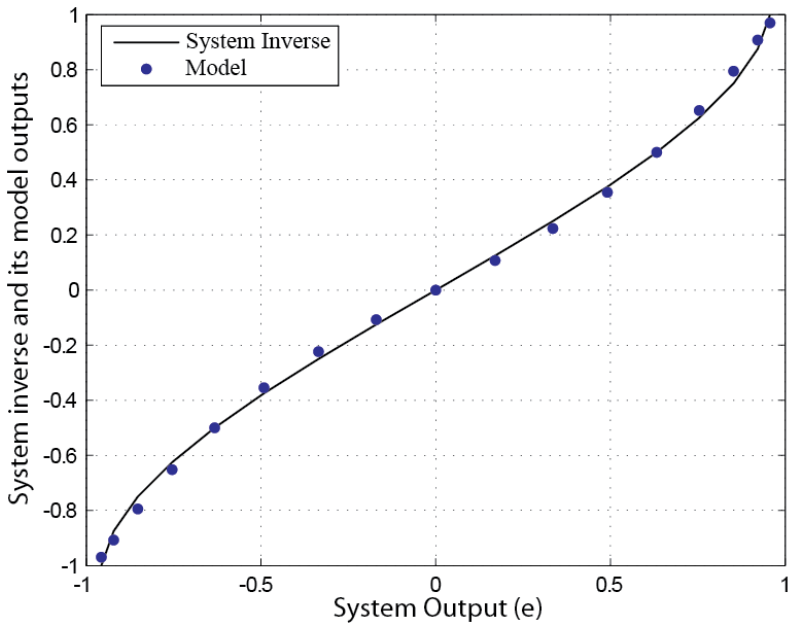
Eventually, the direct & inverse system models are used to design the IMC system for DLL tracking application. Figure 6.8 depicts S-curve function and its direct & inverse models. The numerical example confirms that sufficient identification accuracy can be obtained for a polynomial model of order three with a determination coefficient R^2 high enough ($R^2 > 0.99$). The Least-Squares estimation algorithm method is applied for seeking the optimal function coefficients for the system & control models. Table 6.3 shows two identification process examples for SNR equal to 20db and 3db. This table emphasizes that the system and control models should be modified whenever the ambient noise energy varies.

Usually, identification process is considered as the most time-consuming and expensive part of IMC operation [108]. By contrast, the numerical application section has proven that we just need five input/output data points to effectively execute the identification process. Moreover,

the number of coefficients which are required to be determined is only two for each model (direct & inverse models). As a result, the identification process in DLL tracking application could in fact be achieved more rapidly and much easier than expected at the beginning of this research.



(a)



(b)

Figure 6.8 System and its models: (a) direct model, (b) inverse model

TABLE 6.3 Model and R^2 for system & control blocks

SNR	Model	R^2	Coefficients
20db	System	1	$\hat{e} = 0.98286 \cdot \hat{\tau} - 0.27497 \cdot \hat{\tau}^3$
	Control	0.999	$\hat{\tau} = 0.90639 \cdot \hat{d} + 0.9005 \cdot \hat{d}^3$
3db	System	1	$\hat{e} = 1.3676 \cdot \hat{\tau} - 0.4108 \cdot \hat{\tau}^3$
	Control	0.999	$\hat{\tau} = 0.61752 \cdot \hat{d} + 0.43262 \cdot \hat{d}^3$

6.4 Simulation results

In this section, we will evaluate the performance of the modified DLL tracking with Matlab simulation results. We select the UWB pulse $p(t)$ as the second derivative of the Gaussian function with unit energy and duration $T_p \approx 1$. Each symbol contains $N_f = 10$ frames with duration $T_f = 100$ns, symbol duration is $T_s = N_f T_f = 1\mu s$. We employ TH spreading codes of period N_f, which is generated from a uniform distribution over $[0, N_c - 1]$, with $N_c = 9$, and $T_c = 10$ ns. The simulations are performed in a Saleh-Valenzuela propagation channel with the following parameters: (1/Λ, 1/λ, Γ, γ) = (2, 0.5, 30, 5) ns [61]. The inter-symbol interference (ISI) is negligible. In all simulations, we supposed that signal detection and timing acquisition have successfully been carried out, and the initial timing error is confined within a fraction of UWB pulse duration $\epsilon \in [-\Delta, \Delta]$ where $\Delta = 0.1T_p$. Thus, the tracking loop is "*in lock*" at the beginning of simulation process. Resulted variance and mean values are normalized by T_p^2 and T_p correspondingly.

Figure 6.9 & 6.10 plot the estimated time offset $\hat{\tau}$ and discriminator error e for SNR=20dB and 3dB, the initial value of time offset is $\tau = 0.5$. The simulation results confirm that as SNR increases the discriminator error decreases and the tracking performance is enhanced. For example, in the case of SNR=20dB, the normalized variance and mean of time offset error $\epsilon = \tau - \hat{\tau}$ are 1.5498e-006 and 0.00007 correspondingly, and the normalized variance and mean of discriminator output error are 1.2471e-006 and 1.2e-005. While the estimation performance reduces for SNR=3dB: the offset error variance and mean are 2.1316e-005 and 0.00094, the variance and mean of discriminator output error are 6.4365e-005 and 0.00014 correspondingly.

Figure 6.11 studies tracking performance for various values of data-aided bit number M and SNR, The results show that the offset error variance is inversely proportional to SNR. We can also see that an increase in the number of data-aided bits M leads to reduce the offset error

variance and improve tracking performance. With large values of SNR, the error variance lines converge together, in such case, the different M values have approximately similar effect on the tracking behavior.

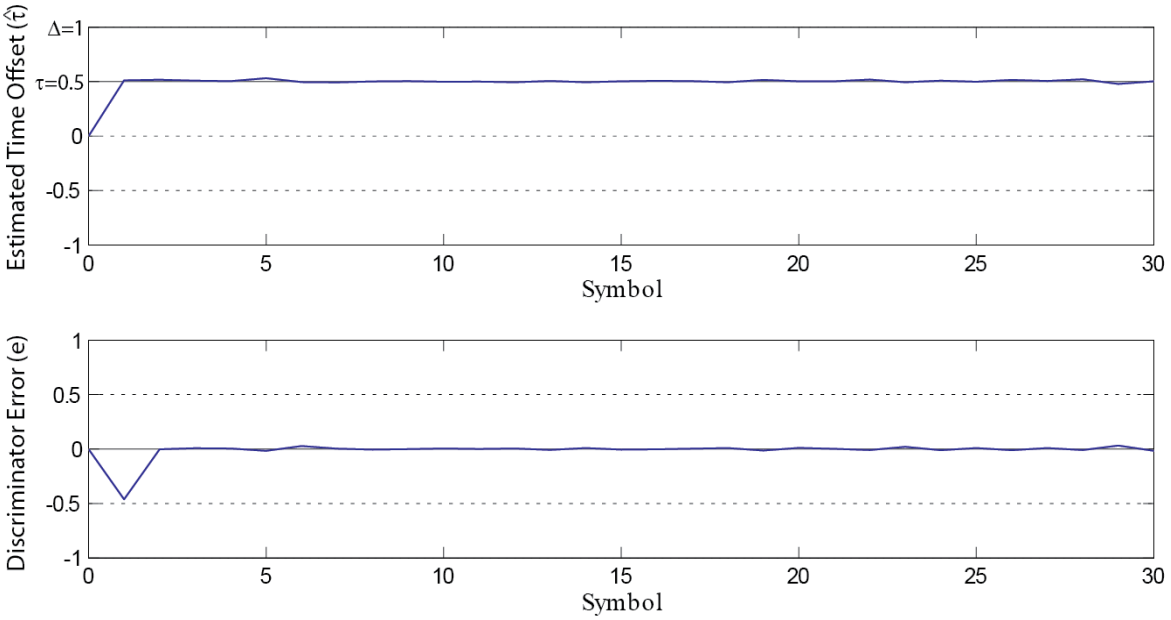


Figure 6.9 Evolution of the IMC DLL tracking for (SNR=20dB and τ = 0.5)

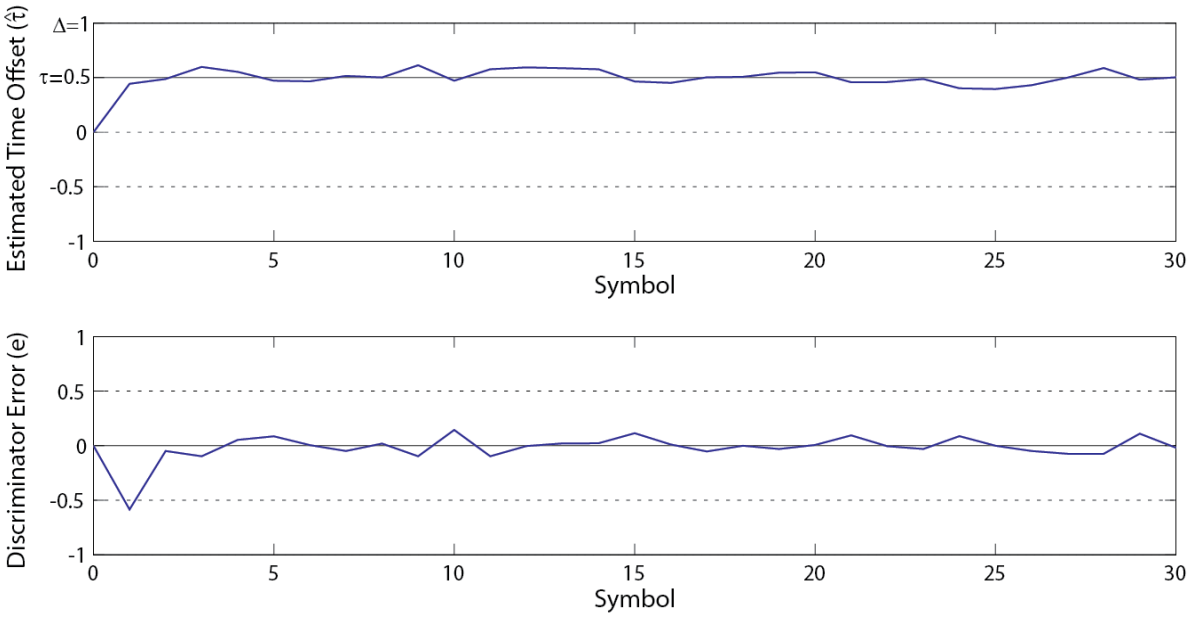


Figure 6.10 Evolution of the IMC DLL tracking for (SNR=3dB and τ = 0.5)

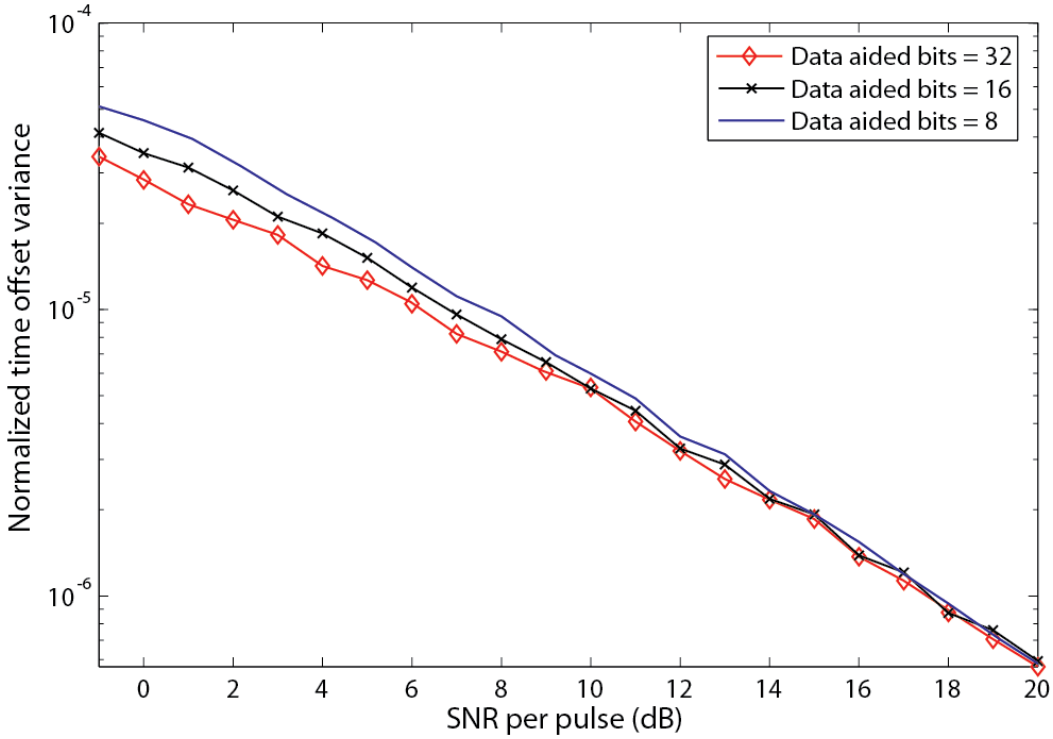


Figure 6.11 Normalized time offset variance for IMC tracking

6.5 Conclusion

In this chapter, we presented a novel DLL structure used to achieve satisfactory and accurate tracking. The proposed DLL scheme is based on Internal Model Control (IMC), which has an excellent performance in many control areas. But it is considered as a novel control strategy in the communication systems. IMC controller structure is composed of two blocks: an inverse model connected in series with the DLL system and a forward model connected in parallel with the DLL system.

The proposed DLL scheme is achieved by using the empirical procedure, which has the following outline: first, the system and control models are estimated using an identification process. Then, the Least-Squares estimation algorithm method is applied for seeking the optimal function coefficients for the internal model of the system and the control blocks. Simulation results confirm that proposed DLL tracking performance is improved by increasing data aided bits or SNR.

In the next chapter, we will study the tracking performance of UWB system, taking into account Doppler Effect. In such case, the polynomial function coefficients for the system and control models must be on-line changing by applying adequate adaptive methods on internal model control system [101, 109].

Chapter 7

Multi-Model Based IMC

So far, we have explained how the IMC structure could be combined with DLL system. Then the proposed tracking system has been evaluated based on Matlab simulation results, where the modeling errors (R^2) for both system and control blocks have been calculated, and the tracking performance has also been analyzed, including discrimination error estimation (e) and timing offset estimation (tracking jitter). On the other hand, the previous simulation results have been performed without taking into account Doppler Effect, which is the result of the relative movement between transmitter and receiver.

In the chapter, we will study the behavior of IMC tracking system taking into account Doppler Effect. We will find through the simulation that the transient performance for the proposed IMC structure is disappointing, and the tracking extremely suffers from Doppler Effect. This chapter seeks to develop more perfectly the IMC behavior in the presence of Doppler Effect. To solve Doppler problem, the multi-model concepts are applied; where the tracking region is divided into several sub-regions, the DLL system within each sub-region is represented by a private local-model. Switching between these local-models occurs at each time when the operating point exceeds certain threshold. Further, we will notice that the transient performance of the modified IMC system is improved as the threshold value decreases. Therefore, instead of switching between local-models with dramatic jumps, we will suggest achieving the model switches smoothly by using adaptive control technique. At the end of the chapter, we will compare the tracking performance between the classical DLL, which has been proposed in Chapter 5, and the DLL combined with IMC technique.

7.1 Tracking degradation with Doppler shift

Let's now evaluate the proposed IMC tracking behavior, taking into the consideration Doppler Effect. For Matlab simulation, we select the same UWB signal model which is proposed in the section 6.4 and consider 16 data-aided symbols used to generate the reference

template signal. We also suppose the input signal time varies as follows $\tau(n) = 0.01nT_p$. Figure 7.1 plots the estimated timing offset ($\hat{\tau}$), discriminator error (e) and system model output ($\hat{e}$). We can notice that in the presence of Doppler shift, the tracking system can track input changes in a certain timing offset range, but the tracking would be lost out of this range. To find an appropriate explication for IMC behavior in the Doppler presence, we repeat Matlab simulation, as in Figure 7.2, considering the input signal time $\tau(n)$ has a step form where τ values are constant for each ten successive symbols then τ increases by $0.01T_p$.

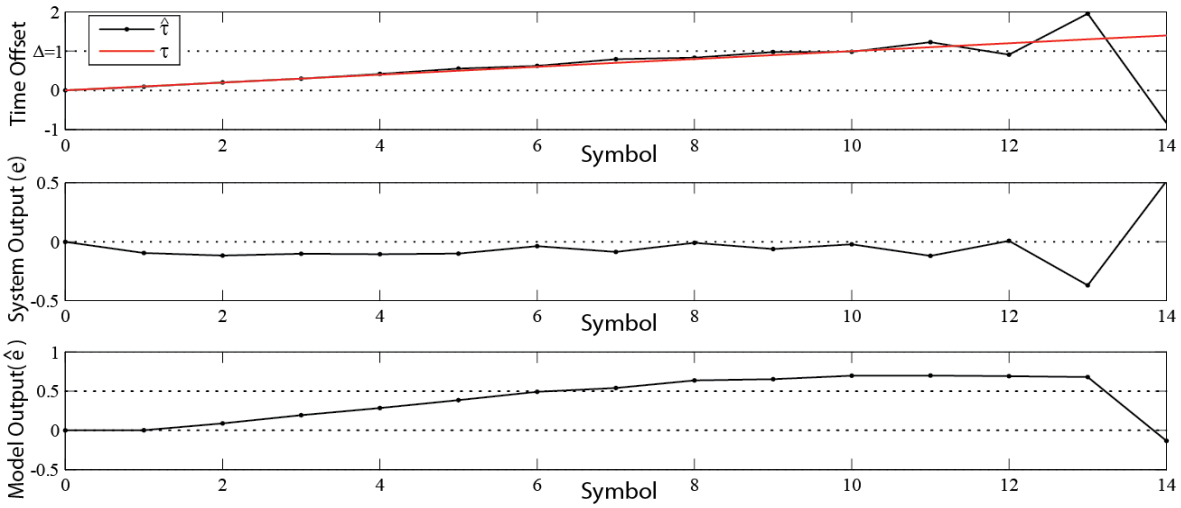


Figure 7.1 Evolution of the IMC DLL tracking (SNR=20dB and $\tau(n) = 0.01nT_p$)

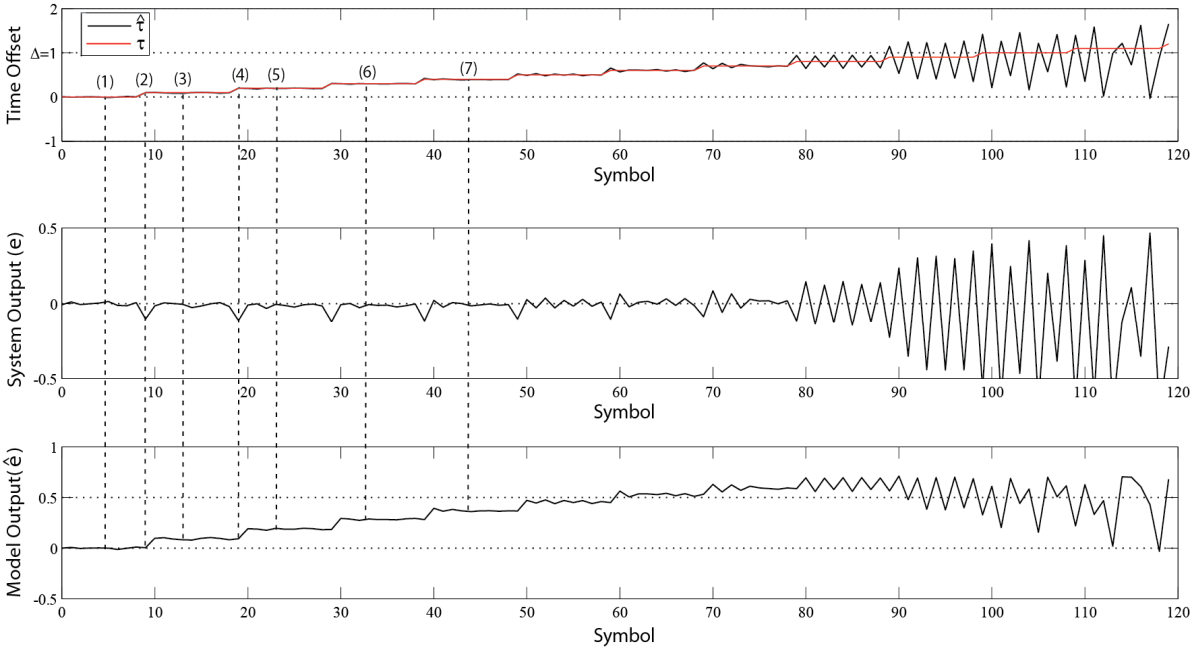


Figure 7.2 Evolution of the IMC DLL tracking in the presence of Doppler Effect (SNR=20dB)

Figures 7.2 and 7.3 depict IMC tracking and its trajectory in the presence of Doppler Effect respectively. Figures confirm that at the beginning of tracking operation, IMC system could be effectively following input signal variations. Whenever the signal input increases, the IMC loop augments properly the signal value at the output of each system & control blocks, leading to keep the error at the system output (e) close to zero.

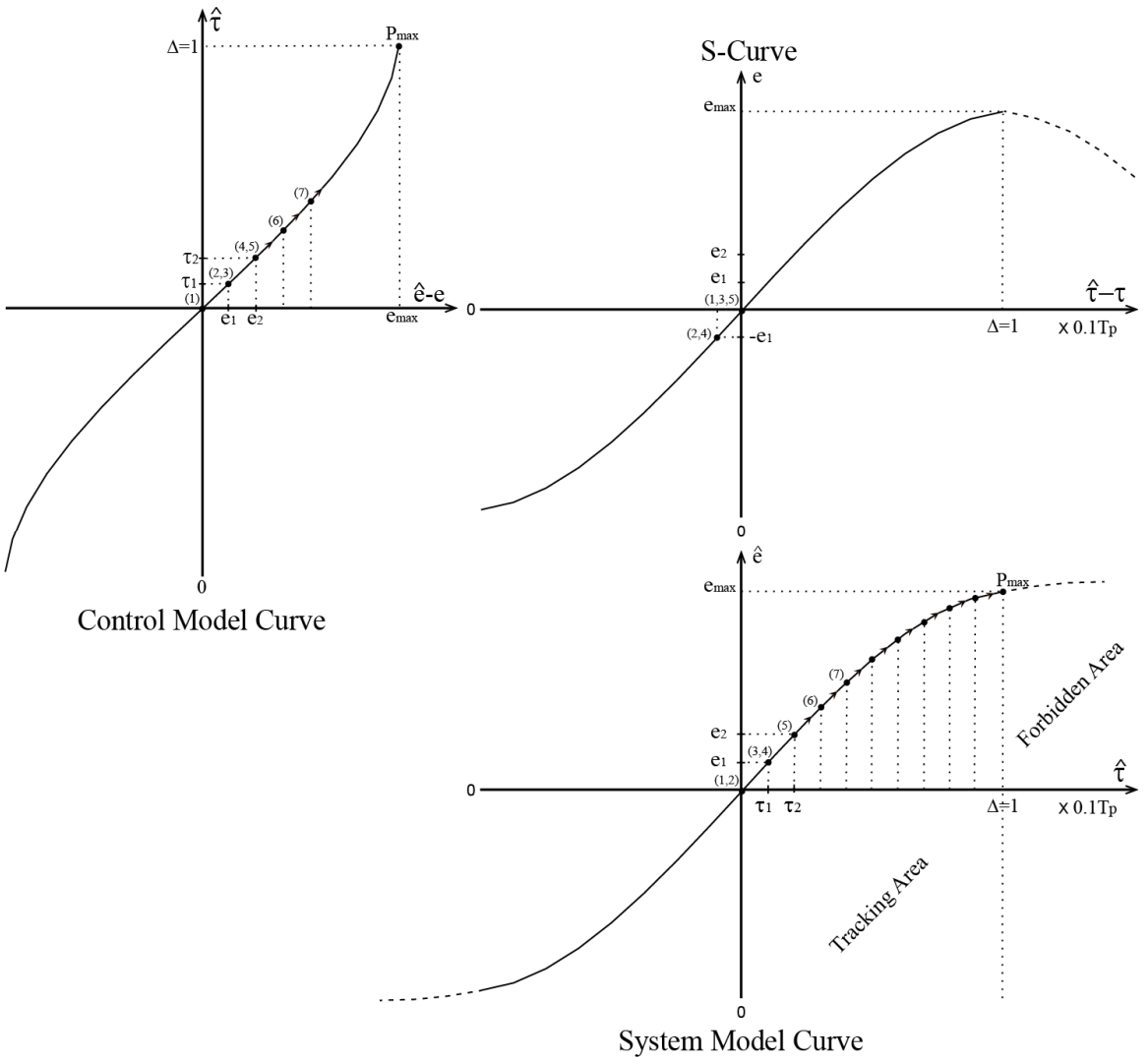


Figure 7.3 Trajectory of the IMC DLL tracking in the presence of Doppler Effect

The IMC system is able to compensate the input signal increment until the tracking operation reaches approximately the maximum point P_{max}, which is consider as a limit separating tracking region from forbidden region as shown in Figure 7.3. Unfortunately, after that, the timing offset estimation and discrimination error estimation will not correctly work and the tracking will completely be lost.

7.2 Multi-model based concept

To resolve the tracking problem in the Doppler presence, the IMC structure will be developed using the multi-model approach. This approach is a powerful and an efficient way used to resolve problem of modeling complex and non-linear processes [110-115]. Generally, the multi-model based IMC, depicted in Figure 7.4, can be summarizing in three steps: The first step consists of dividing the systematic space in some subspaces. In the second step, a fit local (system & control) model is built for every subspace, including selection of an appropriate structural and identification of optimal parameters for each local model. Finally, these local models will be combined with the IMC structure [116]. In this dissertation, we term this new IMC structure as Internal Multi-Model Control (IMMC). In Figure 7.4, the symbol C denotes the number of local base models which are used to represent the uncertain dynamic system across the entire operating range. The Classifier block is employed to switch between different local models, selecting the appropriate local (system & control) model for every moment. Several validation methods used for choosing the appropriate local mode were proposed in the literature [117-119]. One well-known validity criteria is to choose the model which leads to the least modeling error.

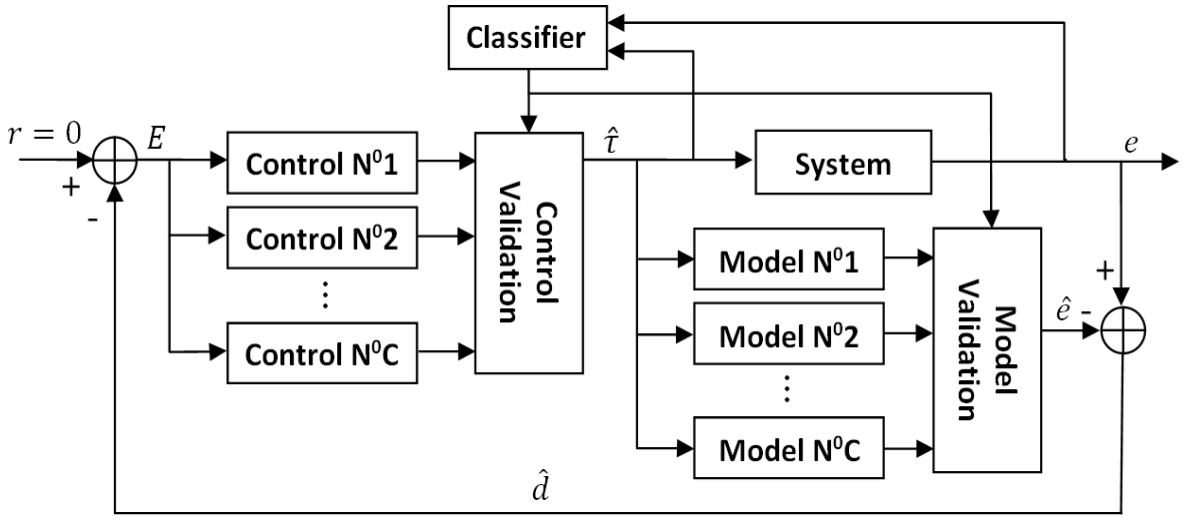


Figure 7.4 Structure of internal multi-model control system

After explaining briefly about IMMC concept, we focus now on combining the multi-mode strategy with the IMC tracking system. The objective is to efficiently control on the system input parameter $\hat{\tau}$ for reducing the timing offset error ϵ in the presence of Doppler Effect. The IMMC tracking system can be applied in the following steps:

7.2.1 **Partitioning of the full operating range**

The first step of IMMC synthesis is to divide the full operating range of DLL system into multiple smaller operating ranges (subranges). From Figure 7.3 we have seen that the DLL system must be working within the range $\epsilon = \hat{\tau} - \tau \in [-\Delta, \Delta]$ and outside of this range the tracking will be lost. While its model is working across $\hat{\tau} \in [-\Delta, \Delta]$, and outside this range the modelling error will noticeably increase.

In the case of absence of Doppler Effect and with assuming the value of τ is constant and close to zero throughout tracking operation, As we have seen in the previous chapter, the DLL system could sufficiently be represented by only single local model across the full operating range. So the systematic space consists of only one subspace ($\mathcal{A}_0$). This subspace has following parameters: $c_0 = 0$ is the center of the region where the index 0 denotes the sub-region number, the operating range is $[v_{min,0}, v_{max,0}] = [c_0 - d_{max}, c_0 + d_{max}]$ where $v_{min,0}$ & $v_{max,0}$ are the minimum and maximum thresholds respectively and d_{max} is the maximum distance between actual input value $\hat{\tau}_i$ and the center of the sub-region c_0, d_{max} is intuitively equal to Δ. If the distance between $\hat{\tau}_i$ and c_0 is bigger than d_{max} ($d_{i,0} > d_{max}$) the valid sub-model must be changed otherwise the operation will be outside the tracking zone.

In the presence of Doppler Effect and with assuming that τ_i increases linearly, Figure 7.1 shows that the tracking operation is working correctly until the input signal $\hat{\tau}_i$ exceeds the maximum threshold $v_{max,0} = \Delta$ where the tracking is lost. To avoid this problem we suggest changing the (system & control) local models each time the input signal $\hat{\tau}_i$ exceeds one of local thresholds. Hence, the full operating range could be divided into an infinite number of subranges $\mathcal{A}_j = [v_{min,j}, v_{max,j}]$ where $j \in \{\cdots, -2, -1, 0, 1, 2, \cdots\}$. Empirically, we found that d_{max} is same for all subranges. As we have mentioned in the previous example (Doppler absence) the maximum value of d_{max} could be equal to Δ. Returning to Figure 7.2 we can also notice that the tracking performance starts to degrade as soon as $\hat{\tau}_i$ exceeds 0.8, so initially we consider that the maximum value of d_{max} in the case of Doppler Effect is about 0.8. Later, in this chapter we will select the optimal value of d_{max} which will be the compromise between the tracking quality and model-change frequency. Finally, the distance between the centers c_j of each two neighborhood subranges is always constant and equal to d_{max}.

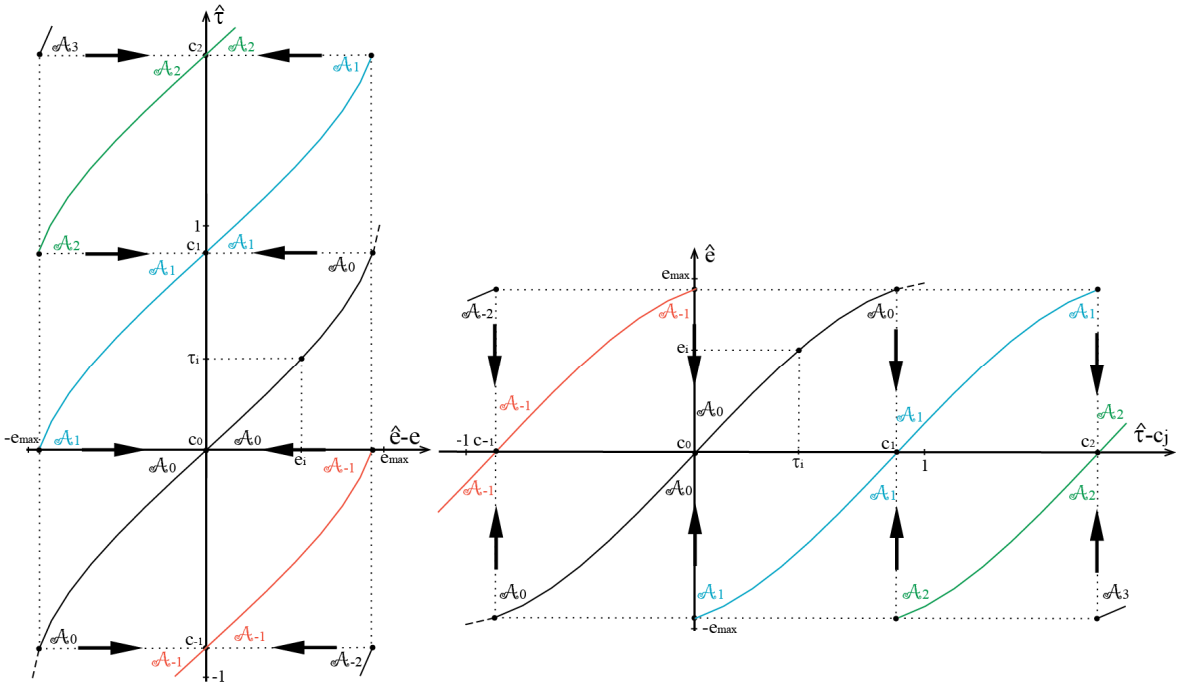
7.2.2 **Development of local models**

The objective of this step is to determine adequate local models (structural and parametric identification) which express the DLL system and its inverse across each subrange. So, each local system & control model is valid only around its corresponding operating local range.

From Figure 7.3, we can notice that when the input signal τ_i increases linearly, the DLL system (S-curve) has always the same form. Initially, we could say that the model of the system and its inverse must keep the same form, even if Doppler shift is present. On the other hand, with an increase in τ_i, the point $(\hat{\tau}_i, \hat{e}_i)$ moves compatibly up toward the maximum point P_{max} (see Figure 7.3). It is right that this movement assures to keep the difference $\hat{\tau}_i - \tau_i$ close to zero, but after certain moment it leads to lose the control. To avoid the tracking loss and to keep the same mode form of the system and its inverse simultaneously, each time $\hat{\tau}_i$ exceeds the local threshold: the system model must be slipped to the right (or to the left depending on the movement direction) and the control model must be slipped too upward (or downward) respectively. The model change helps the operating point $(\hat{\tau}_i, \hat{e}_i)$ to come back around the center of the new local region for each of the system and its inverse models. These shifts are mathematically represented as follows:

$$\mathcal{A}_j: \begin{cases} \text{System Model:} & \hat{e}_i = a_1(\hat{\tau}_i - c_j) + a_3(\hat{\tau}_i - c_j)^3 \\ \text{Control Model:} & \hat{\tau}_{i+1} = b_1(\hat{e}_i - e_i) + b_3(\hat{e}_i - e_i)^3 + c_j \end{cases} \tag{7.1}$$

where c_j is the center of the valid local region $\mathcal{A}_j$. Therefore, it is sufficient for switching between the models to change the value of c_j in (7.1). Since the distance between the centers of neighborhood subranges is d_{max}, so the switching is easily achieved by: $c_{j\pm1} = c_j \pm d_{max}$.



Multiple Models for Inverse Block Multiple Models for Direct Block

Figure 7.5 Transition between different local models

Figure 7.5 shows the curve form of neighbourhood local models for both direct and inverse blocks, and illustrates also how the transition between these models is achieved.

7.2.3 Implementation of IMMC system

The structure of IMMC for DLL tracking application is depicted in Figure 7.6. The multiple models design is so simple and consists only of a classifier block and two addition operations, we could replace these two addition operations by only one, as shown in Figure 7.7, but we prefer to keep the IMMC design as it stands in Figure 7.6 only in order to be compatible with Figure 7.4. The classifier block in its turn is composed of: two comparators in order to compare the actual input $\hat{\tau}_i$ with the minimum and maximum local thresholds, and three addition/subtraction operations for slipping the thresholds and centre, as shown in Figure 7.8.

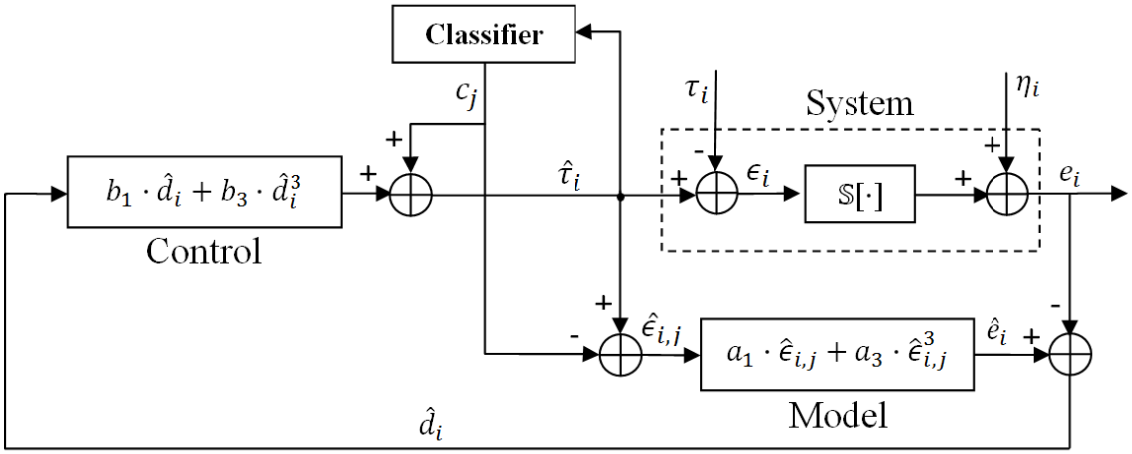


Figure 7.6 IMMC for DLL tracking system

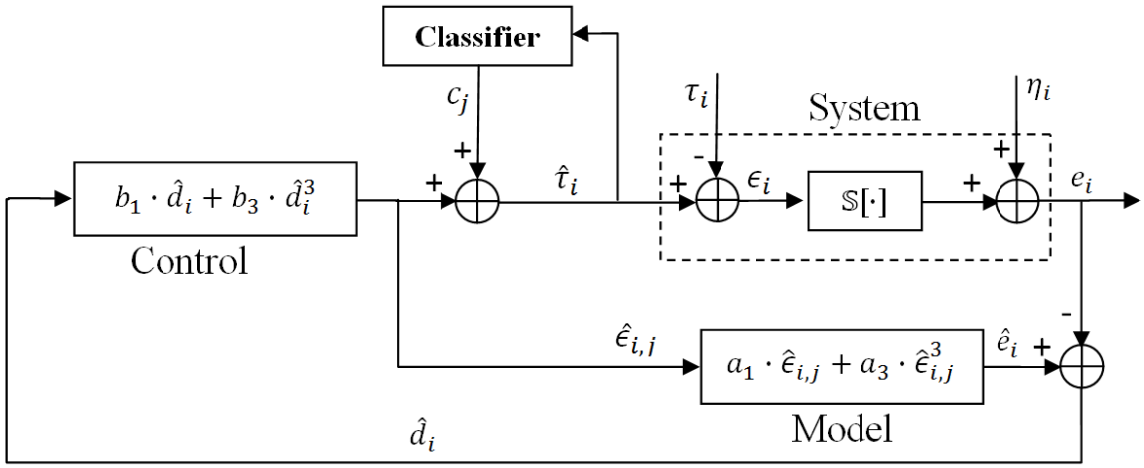


Figure 7.7 Modified IMMC for DLL tracking system

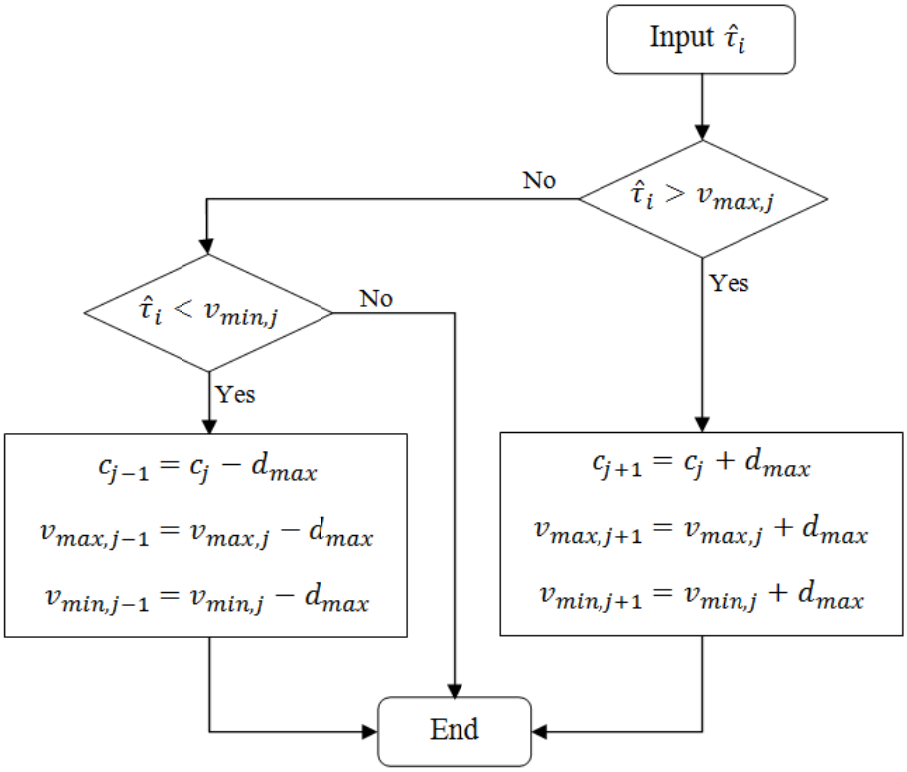


Figure 7.8 Classifier block procedure

7.3 Evaluation of IMMC

In order to evaluate the performance of the proposed approach with the presence of Doppler Effect, two examples will be considered. The first one illustrates the tracking behavior when the input signal increases in step form, represented by Figures 7.9 and 7.10. The second application concerns the tracking of linear-increasing signal, represented in Figure 7.11 and 7.12. These examples are applied two times for SNR equal to 20dB and then for SNR equal to 3 dB. For Matlab simulation, we select the same UWB signal model which is proposed in the section 7.1. All these figures show the behavior of each of the estimated timing offset ($\hat{\tau}$), discriminator error (e) and system model output ($\hat{e}$) with Doppler shift.

Figures 7.9 and 7.10 illustrate that with an increase in input signal τ the proposed approach permits $\hat{\tau}$ to efficiently follow the input signal, keeping measurable error factor e close to zero. These figures show also that the output model signal $\hat{e}$ increases harmoniously with the increase of $\hat{\tau}$ until the moment that $\hat{\tau}$ exceeds the local threshold. At this moment the local model is changed and the operating point is set to the center of the next local region. That justifies why $\hat{e}$ returns back to zero at each transition moments.

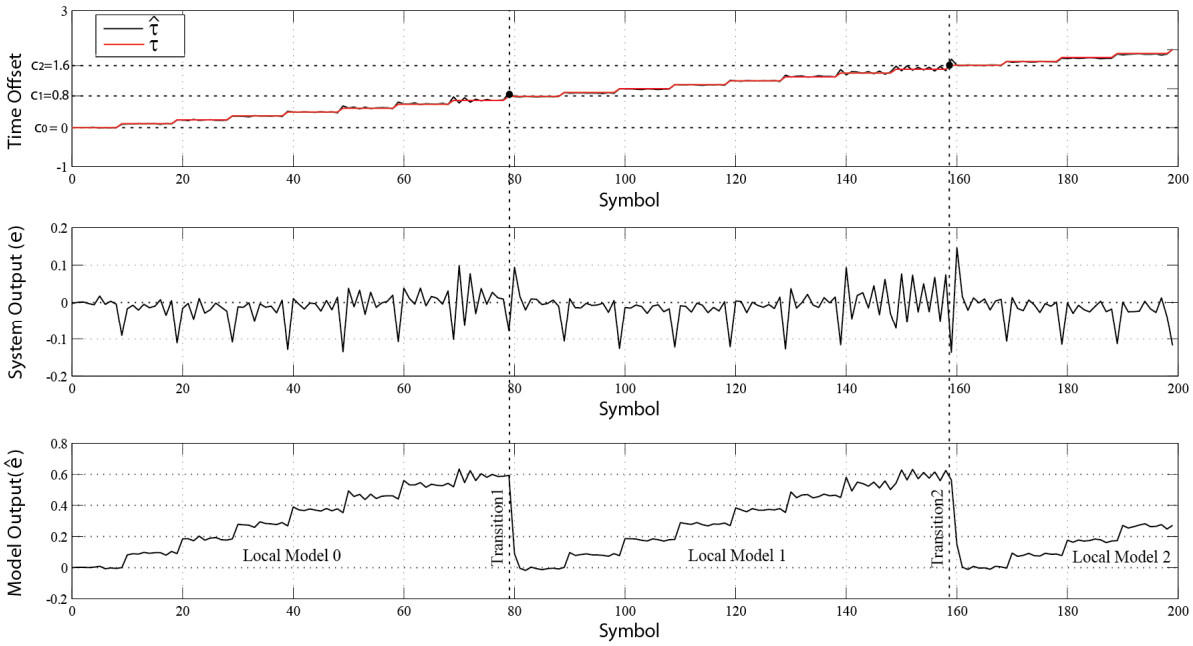


Figure 7.9 Evolution of the IMMC DLL tracking (SNR=20dB and $\tau(n)$ increases in step form)

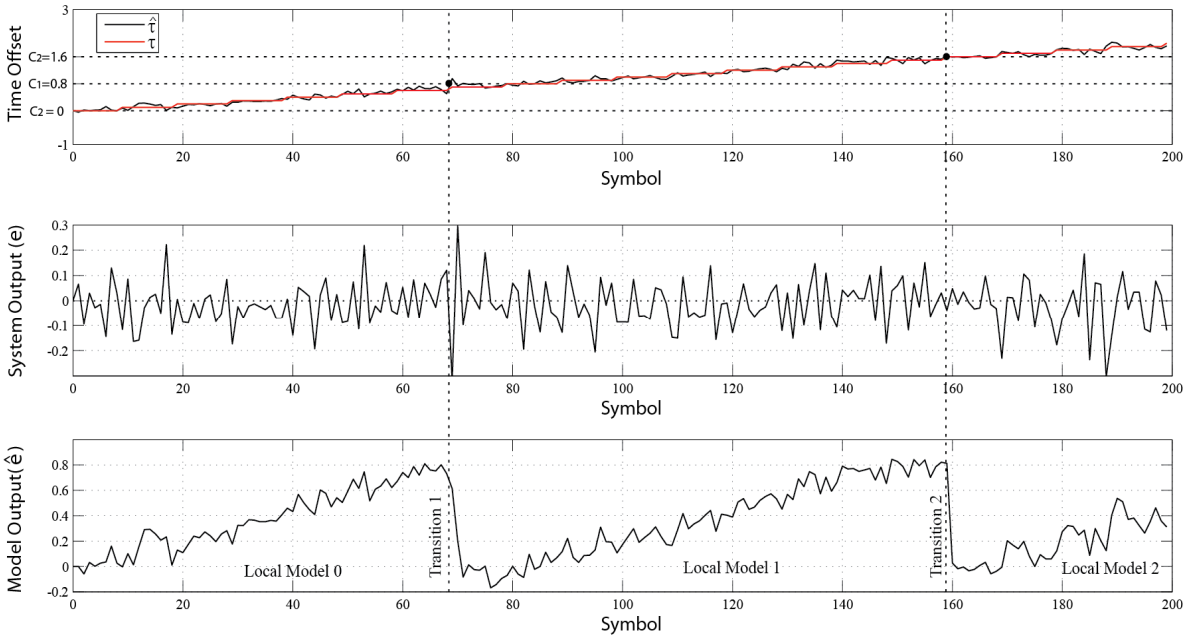


Figure 7.10 Evolution of the IMMC DLL tracking (SNR=3dB and $\tau(n)$ increases in step form)

Similarly, Figures 7.11 and 7.12 conform that IMMC approach is also capable to follow the linear variation of input signal. These figures show also that as SNR decreases the variance of the measurable output signal e increases, leading to degrade slightly the tracking performance.

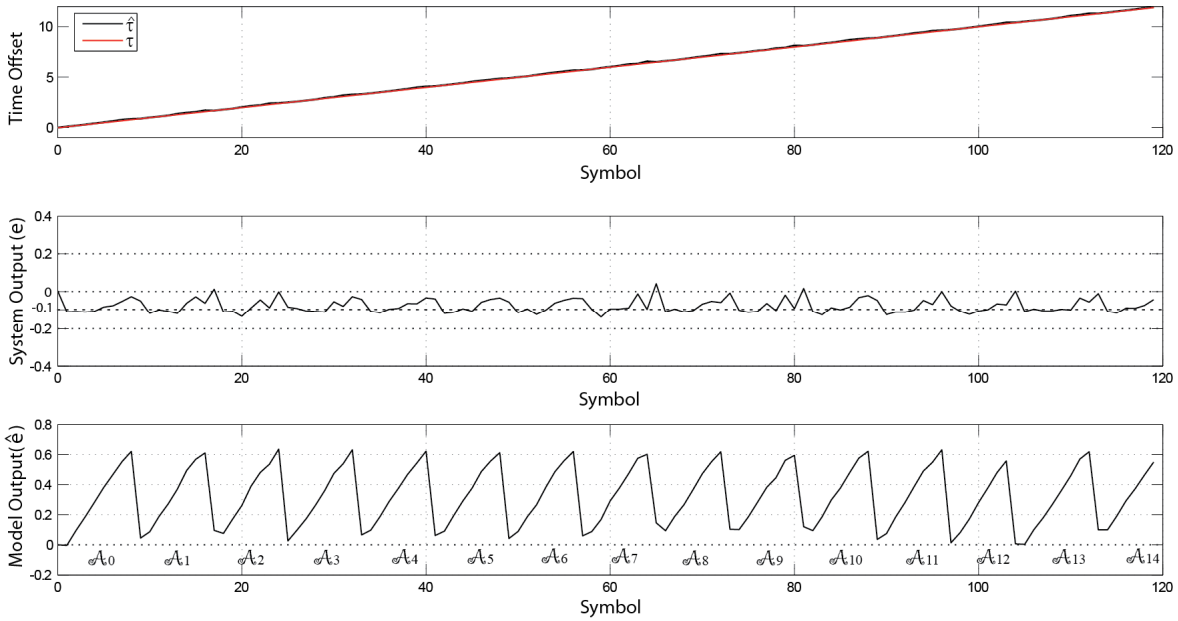


Figure 7.11 Evolution of the IMMC DLL tracking (SNR=20dB and $\tau(n) = 0.01nT_p$)

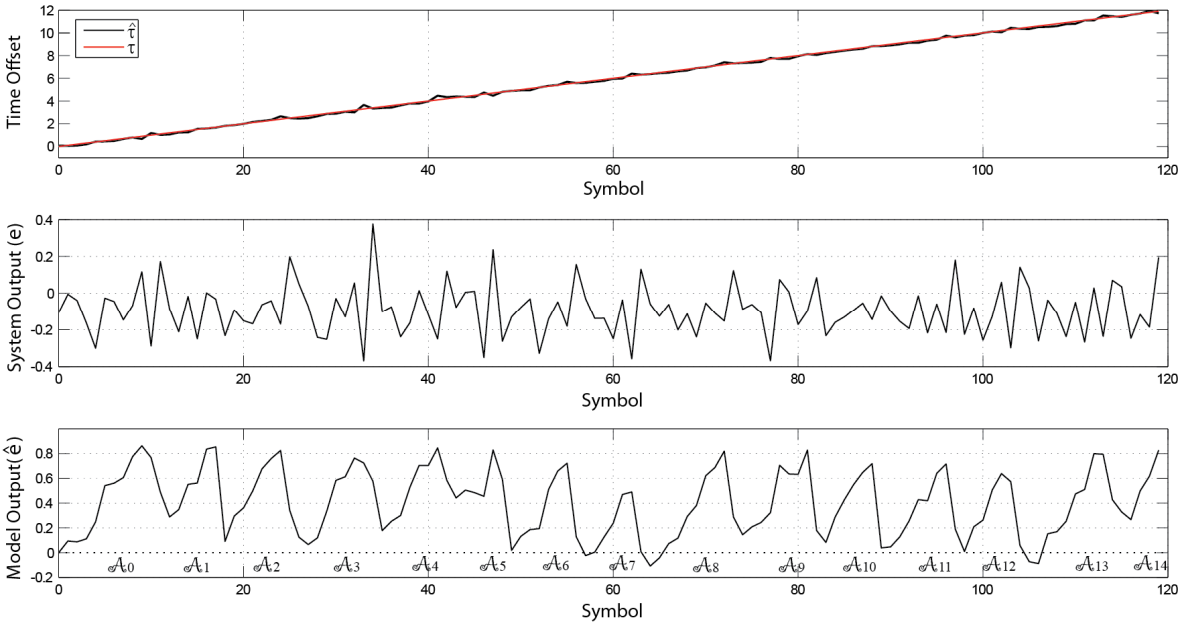


Figure 7.12 Evolution of the IMMC DLL tracking (SNR=3dB and $\tau(n) = 0.01nT_p$)

In general, the simulation results confirm that the IMMC has strong robustness against disturbances, and it can effectively track the input signal variations due to Doppler Effect. On the other hand, looking at Figure 7.11, in particular, at the curve of discrimination error e we notice that the mean of e is not zero. This bias is due to the linear change of Doppler Effect, and the value of this bias depends on the velocity of relative transmitter-receiver motion. This

bias is usually undesired and it impacts somewhat on the system performance. Thus, the objective of the next step is to study how to make e unbiased even if Doppler Effect arises.

7.4 Reduction of the residual steady-state error

As we have found in the previous section that the discrimination error output e has non-zero mean. Usually, to mitigate a steady-state offset at the output we should add an adequate low pass filter (integral filter). Based on Trapezoidal approximation, the integral in a discrete-time control system is defined as follows [120]:

$$y_i = \frac{K_I}{N} \sum_{k=0}^{N-1} e_{i-k} \tag{7.2}$$

Where K_I is the integral gain and N is the observation symbol numbers (integral window). To simplify we suppose $K_I = 1$, the integral is thus define as the sum of the instantaneous error e_i over time (has the same definition of average function).

On the other side, the basic objective of IMC technology, as mentioned in Chapter 6, is to make the output signal follow the setpoint signal. In our application the setpoint signal is set to zero in order to assure that the discrimination error output stays close to zero. Unfortunately, in the presence of Doppler Effect this measure is not sufficient to make output error close to zero. To correct the residual offset we suggest adding the opposite value of this offset (bias) to the setpoint for compensating this bias at the system output.

As a result, to guarantee the mean of e close to zero with or without Doppler Effect an integral filter is combined into the IMC system. There are different forms and techniques for implementing an integral filter [120]. In this thesis, the proposed integral is carried out by taking the average of last N-samples of e and subtracting it from the setpoint value, the subtraction result then becomes the new value of the setpoint; the setpoint is updated once each N-symbols. The only parameter we use to calibrate the integral block quality is the integral window size N. Intuitively, decreasing N leads to compensate the residual steady-state error much faster whatever Doppler shift form is like but at the price of the stability and noise robustness. Figure 7.13 depicts the modified structure of IMMC system, which permits to on-line tune the setpoint in order to get zero-mean system output.

The proposed integral filter, called moving average (MA) filter [121, 122], is the most common filter in digital signal processing DSP applications, because it is the easiest digital filter to understand and use and its common task is reducing random noise. Typically, the standard moving average approach consists of an unweighted average of the sequence of N values $e_i, \cdots, e_{i-N+1}$. In literature, the moving average can also be represented by a weighted average structure such as linearly weighted moving average LWMA and exponentially

weighted moving average EWMA [122, 123]. Here, we focus only on the simple moving average filter. One advantage of the MA filter is that can be calculated by recursive equation, as follows:

$$y_i = y_{i-1} + \frac{e_i - e_{i-N}}{N} \tag{7.3}$$

Thanks the recursive property, the MA filter algorithm is faster than other digital filters, because it requires for each output point only three computations (an addition, subtraction and division) regardless of the length of the filter (N). Applying z-transform theories on (7.3), the transfer function of the filter is given:

$$H(z) = \frac{Y(z)}{e(z)} = \frac{1}{N} \cdot \frac{1-z^{-N}}{1-z^{-1}} = \frac{1}{N} \cdot \frac{z^N - 1}{z^{N-1}(z-1)} \tag{7.4}$$

Based on the impulse response of the filter in (7.4), the frequency response curve can be found, as shown in Figure 7.14. The first null occurs at frequency $\hat{\omega}_0 = \frac{2\pi}{N}$ which is considered as cut-off frequency for this low pass filter. Therefore, an increase of the average length N leads to decrease the filter bandwidth and to reduce the noise effects.

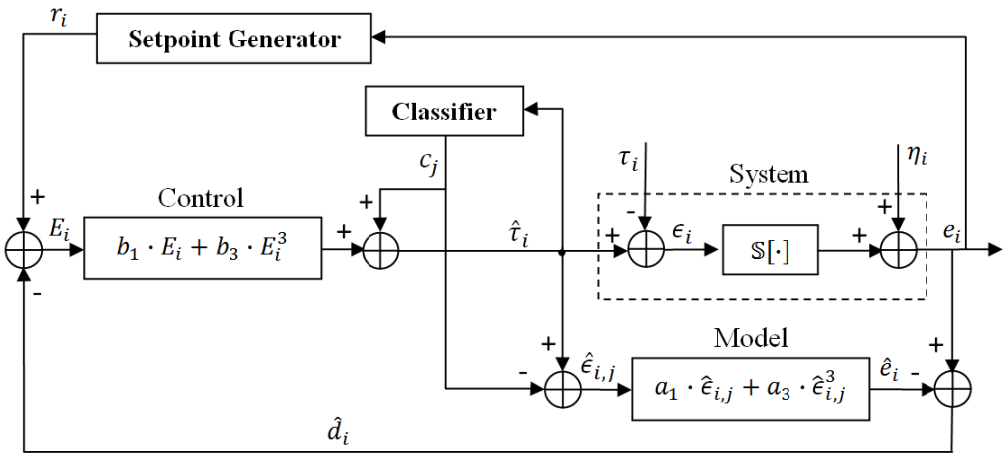


Figure 7.13 Enhanced IMMC structure

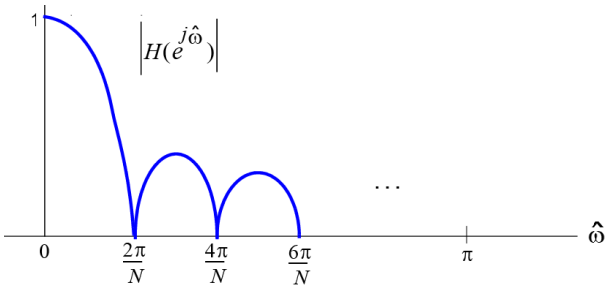


Figure 7.14 Frequency response of the moving average filter

The structure of the setpoint generator is illustrated in Figure 7.15. We use the buffer to hold the setpoint r_i during the interval of N-symbols, that means: at each NT_s, the proposed system mitigates the residual steady-state error by subtracting the average of last N-outputs e_i from the setpoint. Then it initializes the moving average filter memory (reset the states variables: $e_{i-1} = e_{i-2} = \cdots = e_{i-N+1} = 0$). We would like to note that during our research to resolve the static error and before we proposed the average filter with buffer as a suitable solution, we had evaluated the moving average filter without buffer (simple integral filter with moving window); we had found that this simple filter has slightly reduced the output bias but not completely eliminated it. That prompted us to improve the moving average filter structure, and to propose the actual filter with buffer as an adequate and simple solution for eliminating the static error.

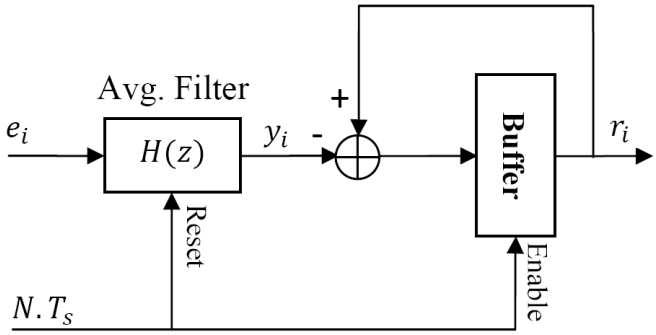


Figure 7.15 Structure of setpoint generator block

To simplify the writing, the enhanced IMMC with moving average filter is abbreviated by EIMMC. The behavior of the EIMMC system in the presence of Doppler Effect is depicted in Figure 7.16. In this example we take the average of discrimination error over $N = 5$ symbols and used then it to on-line adjust the setpoint value. Figure 7.16 shows that the moving average filter contributes to keep the movement of e around zero (eliminate its bias). Figure 7.17 shows the comparison between standard IMMC and EIMMC with average range N equal to 2 then 10. This figure illustrates that standard IMMC produces output signal e with non-zero bias, while the bias problem is completely treated by integrating an average filter within IMMC system. We notice also that for a small value of average window range $N = 2$ the EIMMC has a high response to the input signal movement (rise time is smaller). On the other hand, for $N = 2$ the variance of discrimination error increases and the tracking stability slightly degrades. As a result, setting a value of N is often a tradeoff between high response to relatively fast alterations in input signal state and system stability and noise robustness. Finally, the system response to Doppler Effect which is depicted in Figure 7.17 is somewhat oscillatory and it perhaps impacts the tracking stability. The response could be improved and be more stabilized by adding, for instance, a derivative term parallel to the integral, which permits to improve the stability but at expense of the transient response and noise robustness [124, 125].

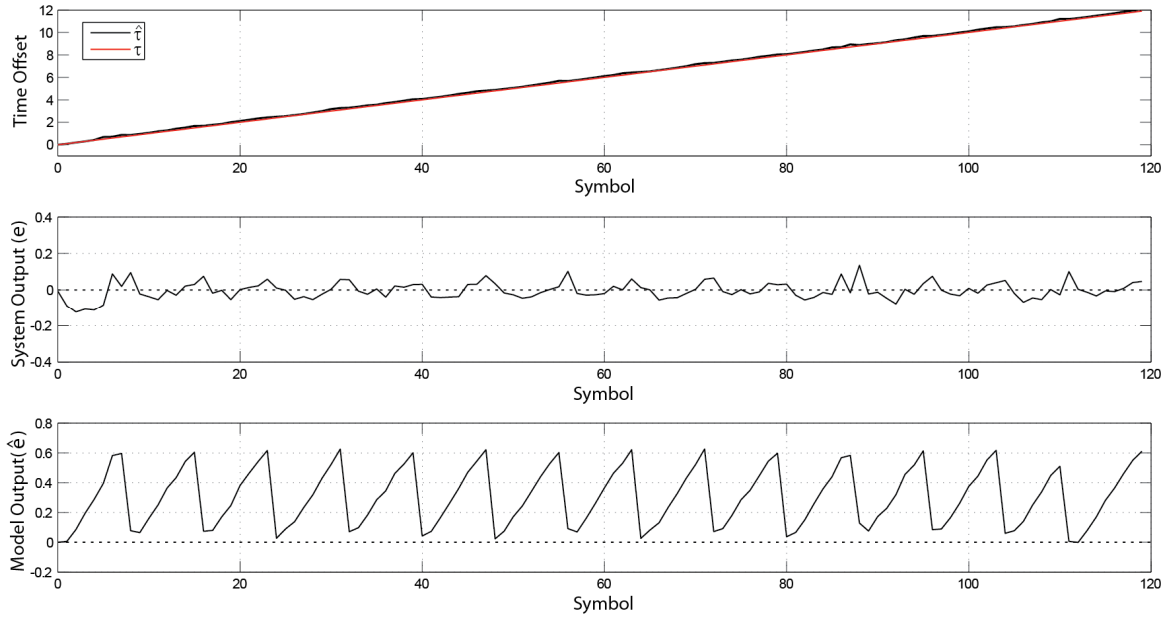


Figure 7.16 Enhanced IMMC DLL tracking (SNR=20dB, N=5 and $\tau(n) = 0.01nT_p$)

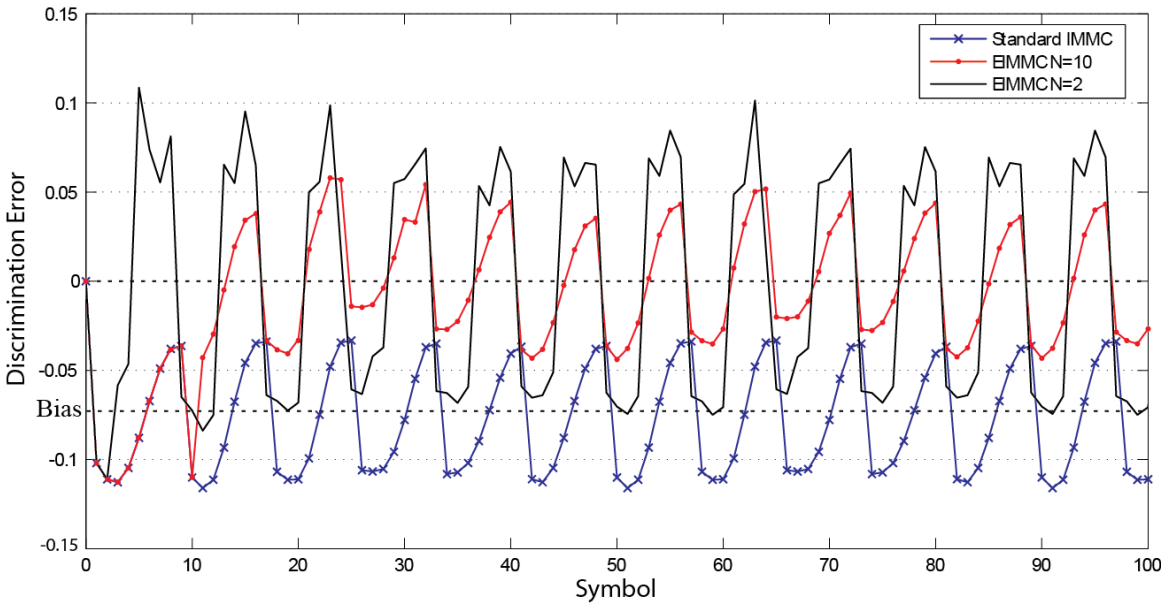


Figure 7.17 Comparison between standard IMMC and enhanced IMMC (without noise)

Return back to Figure 7.17 and compare the curve of Standard IMMC and the curve of EIMMC with average range $N = 10$, we notice that those two curves are matching at the first 10 points. The reason is that the setpoint generator in Figure 7.15 waits N-symbols before updating the setpoint value. To avoid this bidding state each N-symbols and to make the average filter change smoothly without degrade its quality, we change the position of the buffer inside the filter to be, as shown in Figure 7.18:

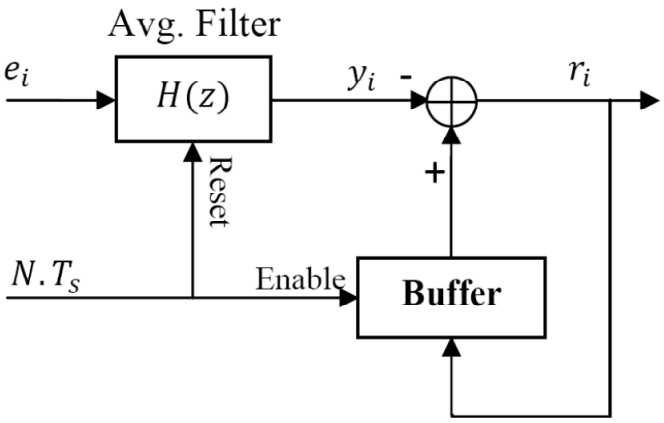


Figure 7.18 Structure of modified setpoint generator block

Let us term the IMMC with the modified moving average filter mentioned in Figure 7.18 as modified EIMMC. Figure 7.19 shows the comparison among standard IMMC, EIMMC and modified EIMMC with average range N equal to 10. This figure illustrates that the modified EIMMC enables eliminating efficiently the residual steady-state error without the need to wait N-symbols for updating the setpoint value. In addition, this modification enables us to change easily the position of the average filter inside the circuit and to replace it by loop filter placed at the output of the system as mentioned in Figures 7.20. The structure of this loop filter, shown in Figure 7.21, consists of two terms: proportional term with gain equal one and moving window integral term with transfer term $H(z)$.

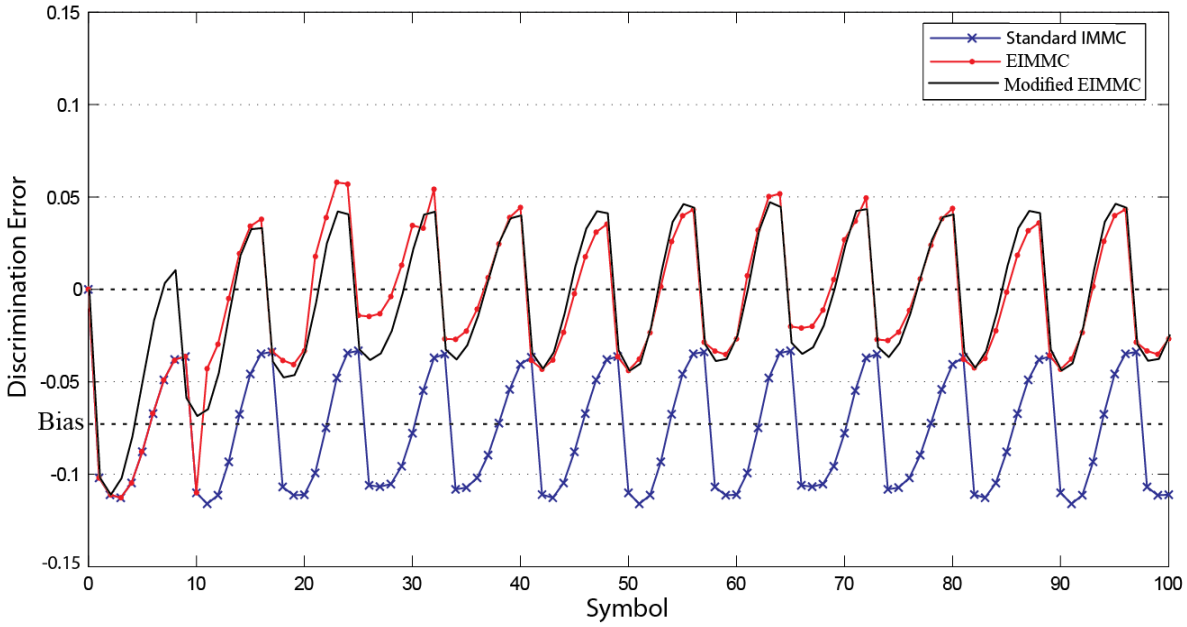


Figure 7.19 Comparison among standard IMMC, EIMMC and modified EIMMC ($N = 10$)

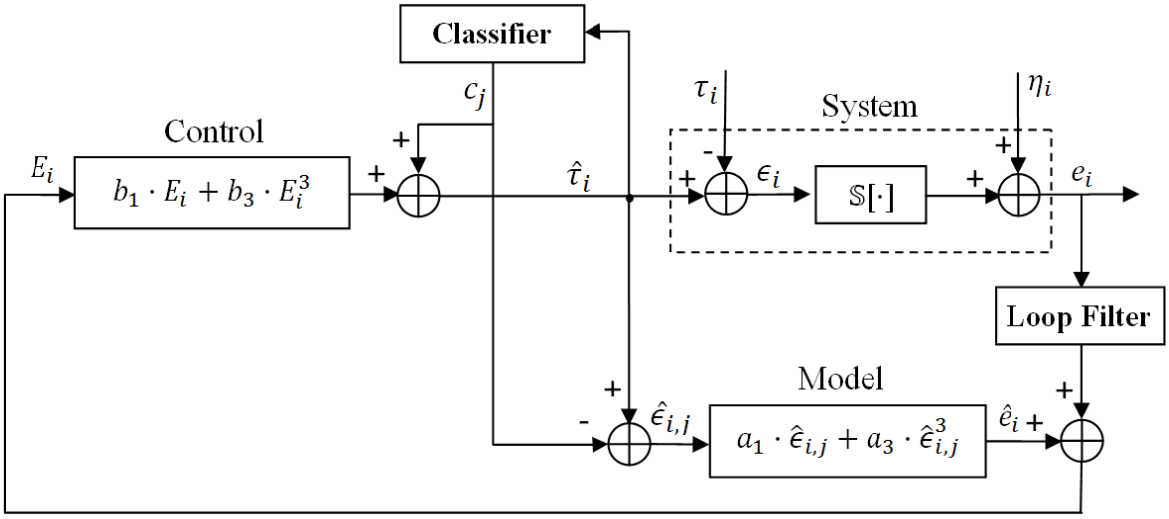


Figure 7.20 IMMC structure with loop filter

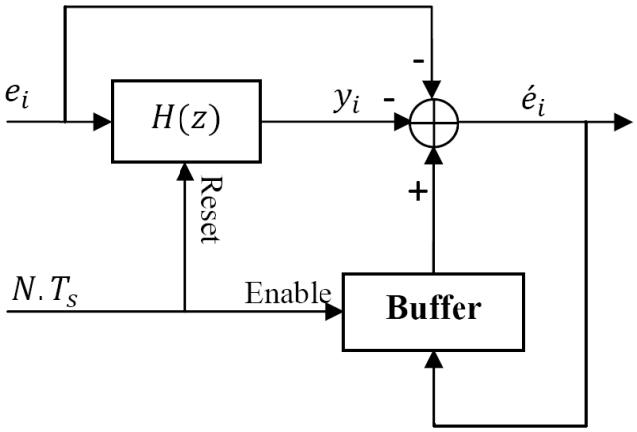


Figure 7.21 Loop filter structure

Figures 7.22 and 7.23 present the IMC performance (variance and bias of output error) for two cases: the first case (*solid lines*) is for the average filter mentioned in Figure 7.15, and the second case (*dashed lines*) is for the average filter mentioned in Figure 7.18. The variance and bias are plotted for different value of filter rank N. From Figure 7.15 we can notice that when N converges to infinity the setpoint stays always zero and the IMC system is isolated from the average filter. While for the second structure of average filter shown in Figure 7.18, when N is so big, only the buffer stays out of service and the filter is transferred to classic moving window integral filter. Figures 7.22 and 7.23 confirm that when $N \to \infty$ the classic moving window integral filter contributes to reduce the bias but not sufficiently. The figures confirm also that the output bias is reduced as N decreases but it accompanies an increase in the error variance. In addition, for a large value of average rank ($N > 5$) the variance can be considered

approximately constant. Finally, as the Doppler shift, in our work, changes linearly, the first method produces smaller error variance than that achieved by the second method. It seems interesting for the future work to analyse the performance of these two methods for other forms of Doppler shift, maybe the performance of the second method would be much better. Henceforth, for the rest of the PhD research work, we decide to use only the first moving average structure with window size N equal to 6.

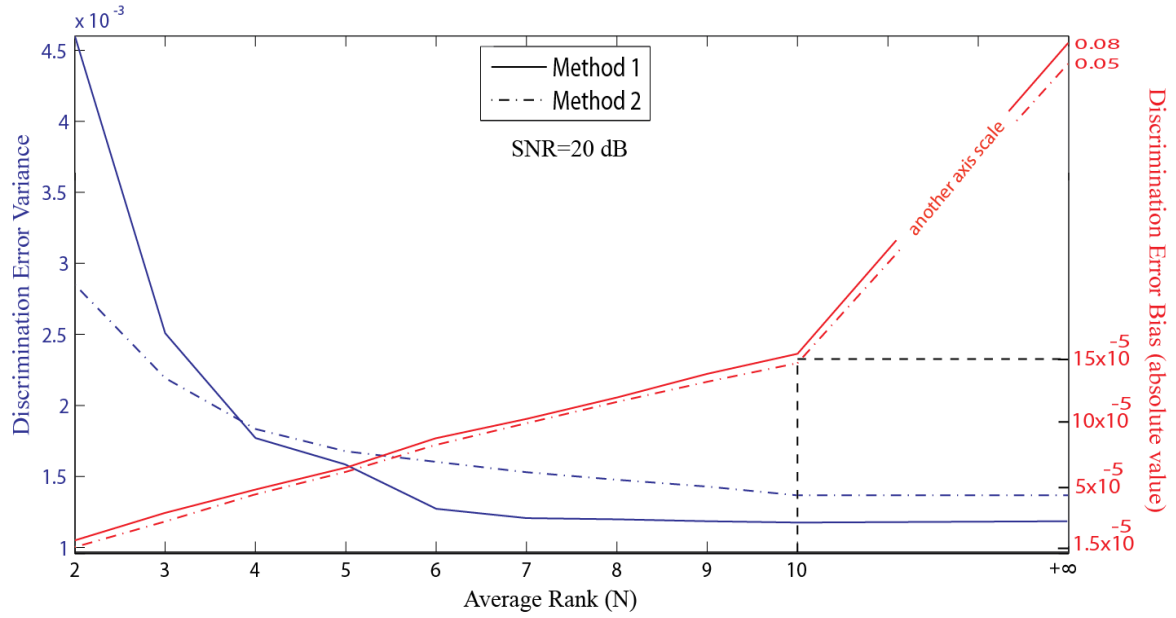


Figure 7.22 Discrimination error properties verse average window size (SNR=20 dB and d_{max}=0.8)

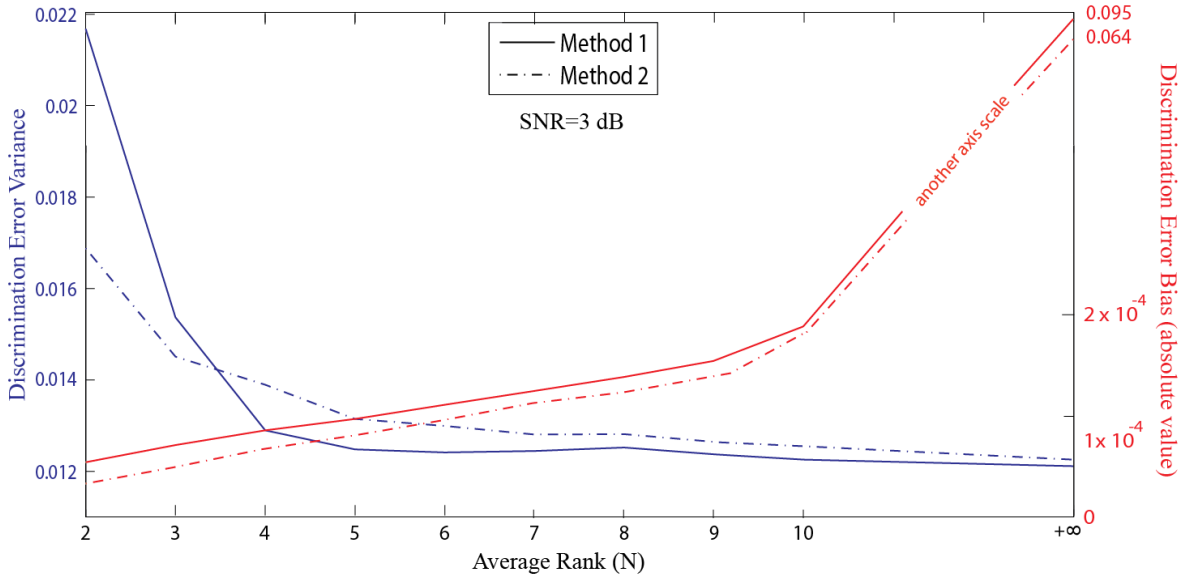


Figure 7.23 Discrimination error properties verse average window size (SNR=3 dB and d_{max}=0.8)

Practical consideration: We have noticed that the Doppler shift produces the bias at the output of the IMC system. We can briefly justify the bias appearance that the proposed IMC structure for DLL system has a static design characters and it only uses the current measurement e_i to estimate the current value of $\hat{\tau}_i$ without any exploitation of the previous measuring samples ($e_{i-1}, e_{i-2}, \cdots$). The static model allows us to directly estimate the timing offset τ_i at the moment i with high precision. According to the estimate $\hat{\tau}_i$, the DLL system will displace the reference signal $y(t)$ to reduce the offset with the received signal $r(t)$. But at the moment $i+1$, due to the linear Doppler shift the offset will linearly increase: $\tau_{i+1} = \tau_i + \Delta t$; where the constant parameter Δt related to the Doppler shift velocity (assuming the noise and other channel effects neglected). Consequently, the deference between τ_{i+1} and $\hat{\tau}_i$ is significantly exist and tends almost to Δt. Thus this constant deference justifies the bias at the output of the system. On the other side, the moving average filter which is built within the IMC system offers dynamic characters to our system, since the filter exploits the previous output measurements ($e_{i-1}, e_{i-2}, \cdots$) to eliminate the output bias. Furthermore, this filter could be used as a way to predict the future values of timing offset. In addition, the generation setpoint signal at the MA filter output could contribute to estimate the Doppler shift velocity (estimate a model of the relative transmitter-receiver motion) where r_i is almost proportional to Δt and the sign of r_i indicates the movement direction.

7.5 Selction of the threshold of local models

As we have found in section 7.2.2, the maximum value of local distance must be $d_{max} \leq 1$ and it is better to be equal or smaller than 0.8. In this section, we focus on selecting empirically the optimal value of d_{max} which is trade-off between the control quality and model-change frequency. We are interested in studying the impact of d_{max} on model-change frequency, because the switching procedure, from a point of view of circuit implementation, usually requires more computation time and more consummated energy. Figures 7.24-7.29 show the EIMMC tracking behaviour for different value of d_{max} and SNR (d_{max}=1, 0.5 and 0.1, SNR=20 and 3dB). While Figures 7.30 and 7.31 depict the trade off between discrimination error variance and model switching number, d_{max} here changes from 0.1 to 1 and the simulation results have been extracted in the case of transmitting 1000 symbols.

All these figures confirm that increasing d_{max} leads to decreases model switching number, but at expense of IMC controller performance which is represented by the output error variance. The figures show also that as SNR increases the output error variance decreases. In addition, by comparing the $\hat{e}$ curves for different d_{max}, it is evident that the maximum reached value of $\hat{e}$ is decreased when d_{max} decreases. For example, when d_{max} decreases from 1 via

0.5 to 0.1 as shown respectively in figures 7.24, 7.25 and 7.26, the maximum reached value of $\hat{e}$ decreases also from 0.7 via 0.4 to 0.2 respectively.

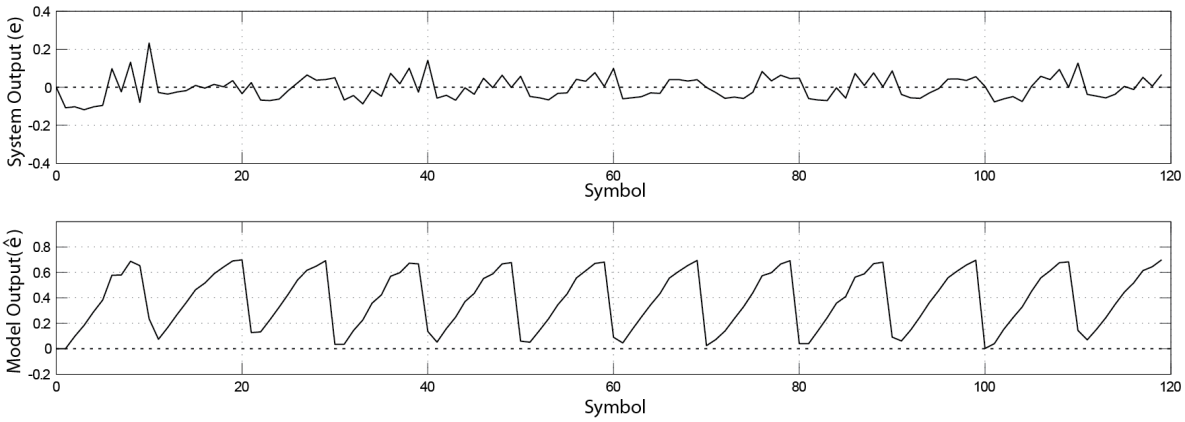


Figure 7.24 EIMMC behavior for $d_{max} = 1$ and SNR=20 dB

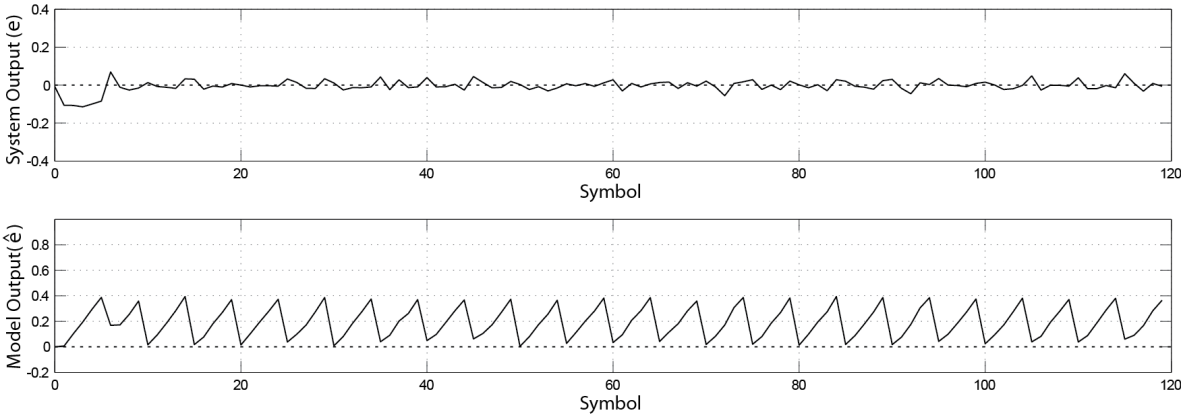


Figure 7.25 EIMMC behavior for $d_{max} = 0.5$ and SNR=20 dB

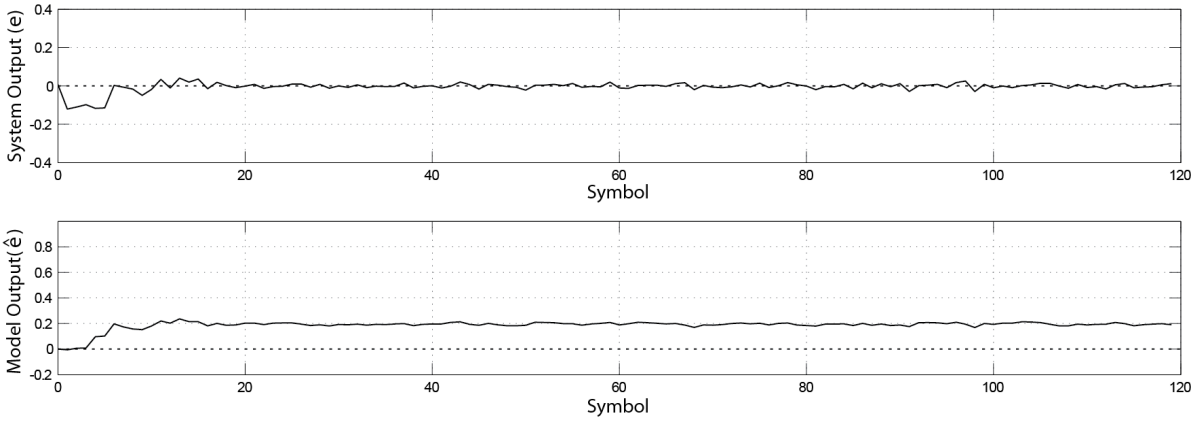


Figure 7.26 EIMMC behavior for $d_{max} = 0.1$ and SNR=20 dB

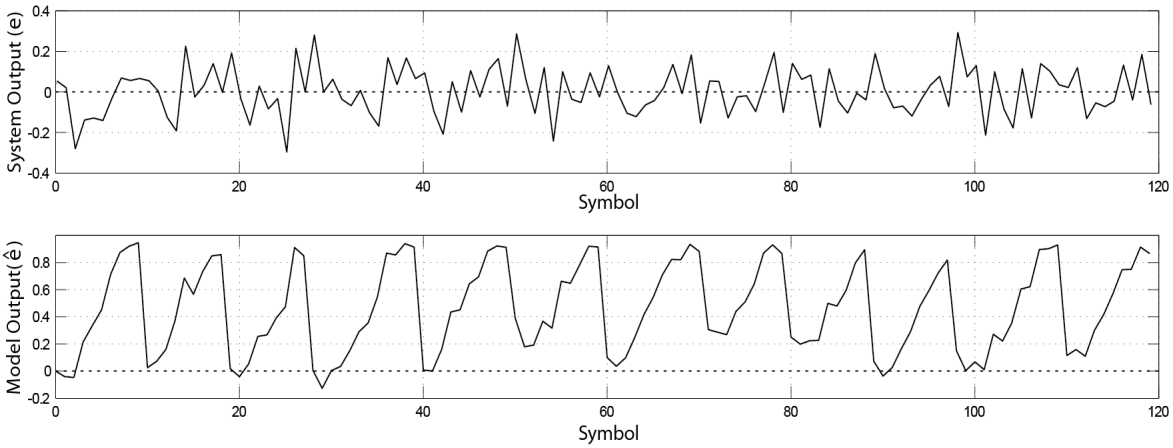


Figure 7.27 EIMMC behavior for $d_{max} = 1$ and SNR=3 dB

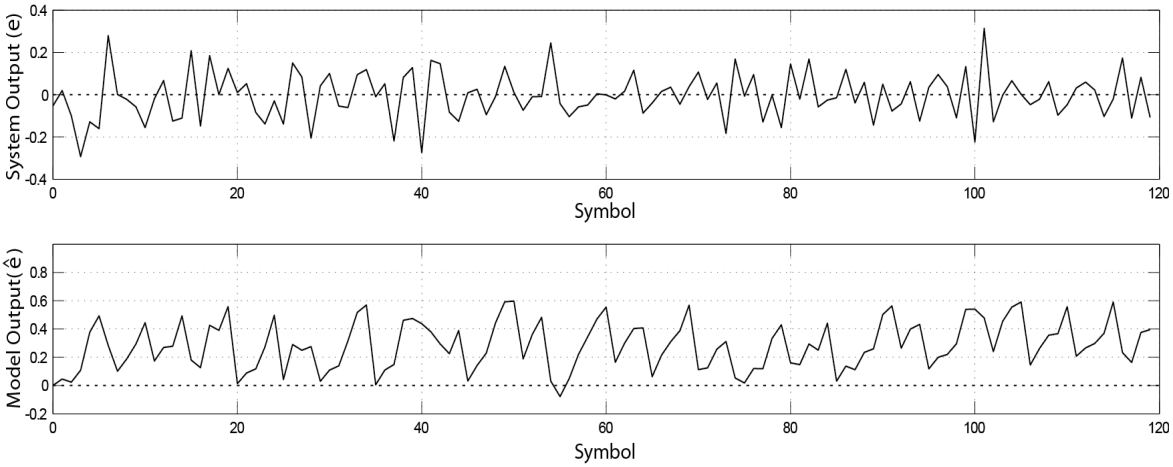


Figure 7.28 EIMMC behavior for $d_{max} = 0.5$ and SNR=3 dB

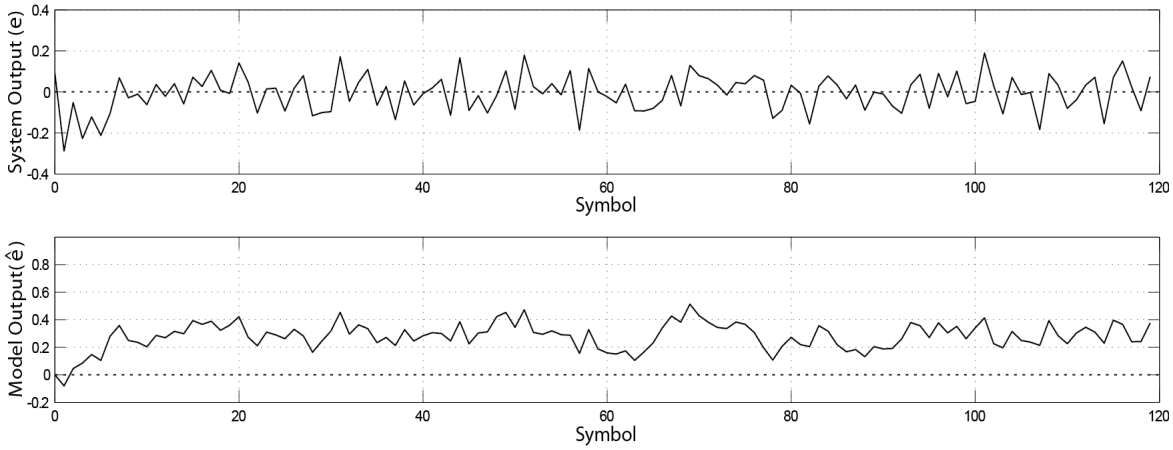


Figure 7.29 EIMMC behavior for $d_{max} = 0.1$ and SNR=3 dB

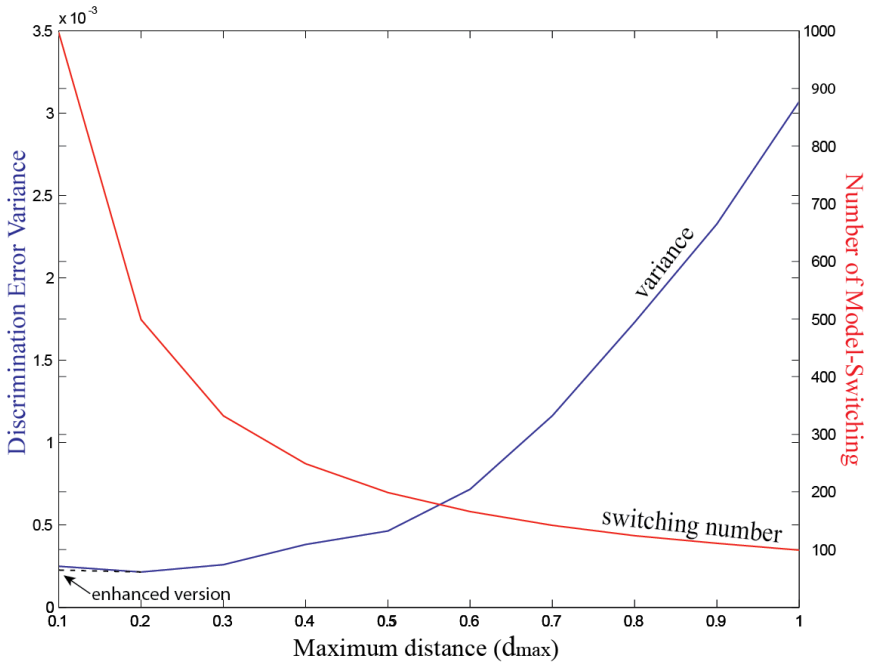


Figure 7.30 Trade-off between control quality and number of model-switching (SNR=20dB)

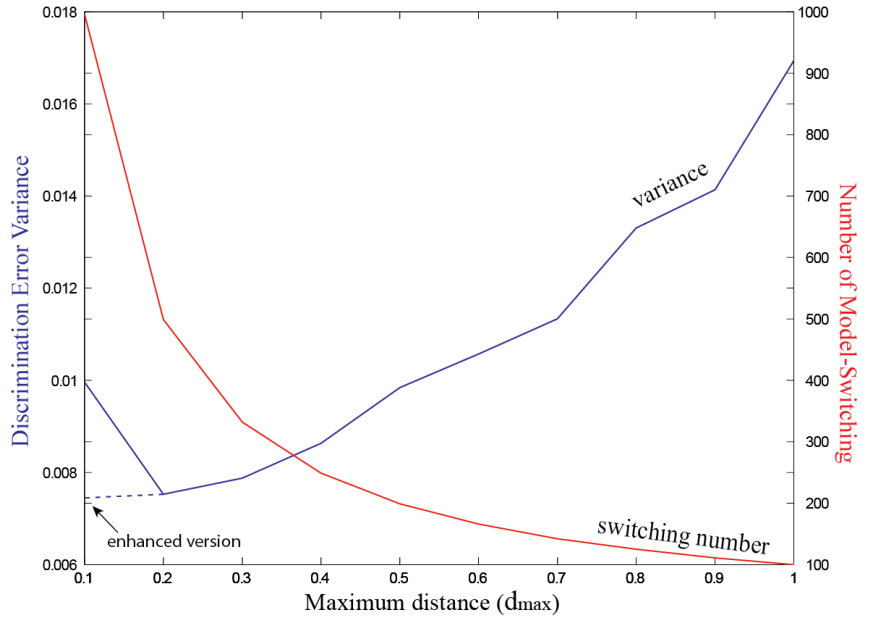


Figure 7.31 Trade-off between the control quality and number of model-switching (SNR=3dB)

From Figure 7.31 we notice an expected degradation in the control quality for a small value of $d_{max} = 0.1$ especially for small SNR case. This issue could be justified as follows: When d_{max} is set to so small value and in the presence of noise or Doppler Effect it is not sufficient to switch from the actual local model to its direct neighbour for conserving the tracking quality,

the operating point should jump over the neighbourhood sub-region directly to the next one (or the third ...), that means that the model classifier block must sometimes switch directly from $\mathcal{A}_j$ to $\mathcal{A}_{j+2}$ without passing through $\mathcal{A}_{j+1}$. Therefore we have developed the classifier scheme as mentioned in Figure 7.32 in order to be able to do switching between the local models, taking into account the noted statement, this modification permits to improve the control quality, as mentioned in Figures 7.30 and 7.31, where the modified system is represented by dashed lines.

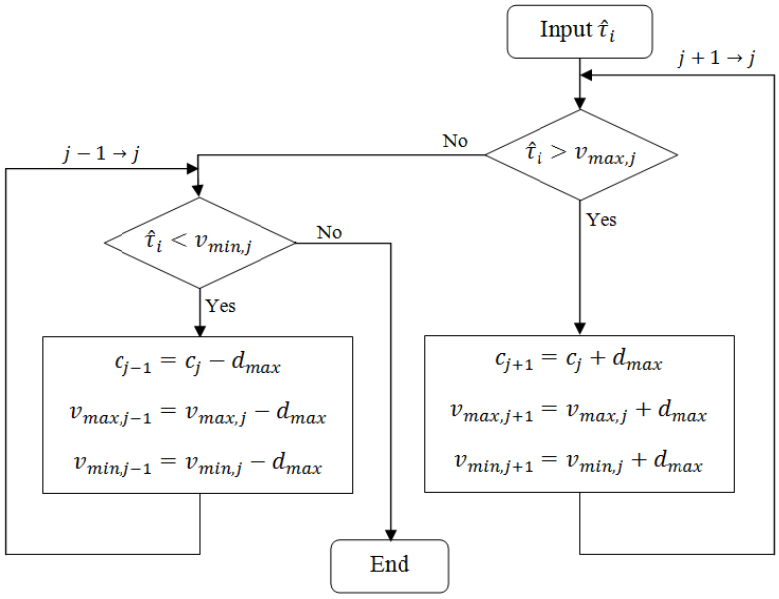


Figure 7.32 Modified scheme of classifier block

Figures 7.33 and 7.34 depict the system behavior after developing the classifier system. We would like to point out that sometimes for a small d_{max}, the ambient noise can sufficiently provoke switching between the local models even if Doppler effect is absent.

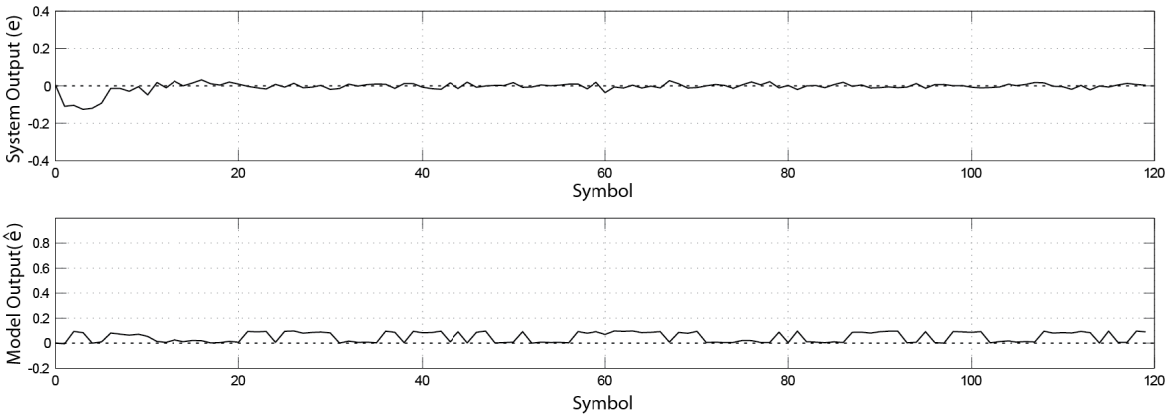


Figure 7.33 Behavior of EIMMC with modified classifier block ($d_{max} = 0.1$ and SNR=20 dB)

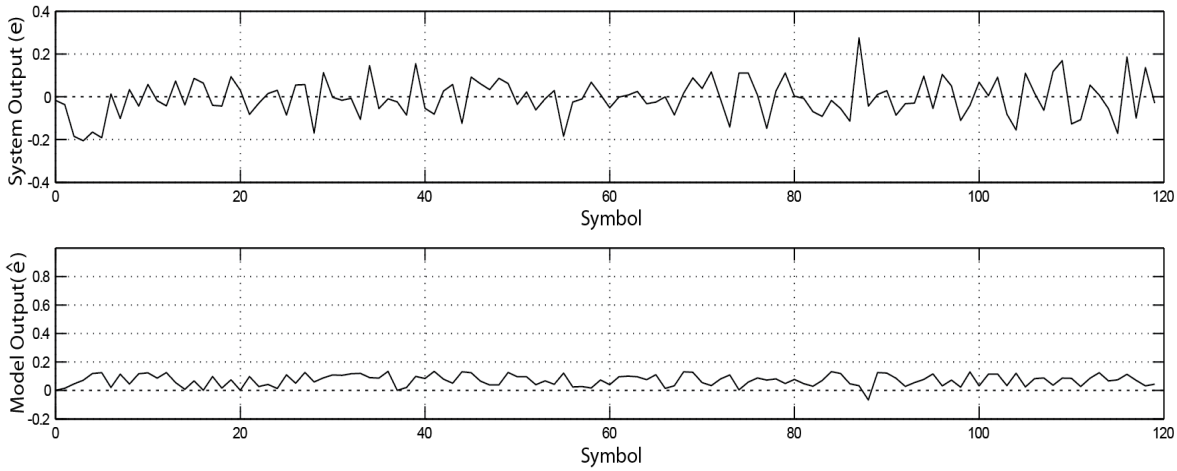


Figure 7.34 Behavior of EIMMC with modified classifier block ($d_{max} = 0.1$ and SNR=3 dB)

Practical consideration: Figures 7.30 and 7.31 confirm that the control quality can be improved by decreasing d_{max}. Returning back to Figure 7.32, we notice that decreasing d_{max} leads to reduce the sub-region range and to make the classifier output signal c_j change smoothly. Besides, the classifier will need to repeat the loop procedure several times until it places $\hat{\tau}_i$ inside an appropriate sub-region, all that requires more computation time and more consummated energy. Furthermore, when d_{max} converges to zero, c_j tends approximately to $\hat{\tau}_i$. Consequently, c_j and $\hat{\tau}_i$ will cancel each other out at the input of the system model (see Figure 7.13) and produce isolating the direct model block from the total IMC system, menacing the stability of the system.

On the other hand, if we find an approach to make c_j tends approximately to the unknown input τ_i (instead of converging to $\hat{\tau}_i$), the system and its model will be significantly identical even in Doppler presence (see Figure 7.13), and the IMC tracking system will be perfect. Therefore the objective of the next section is to replace the classifier block by another block which works for observing τ_i, the signal c_j is next set to the observed value of τ_i. To observe τ_i we suggest an adaptive filter as a proper promising method [126]. Figure 7.35 depicts the structure of EIMMC included adaptive filter.

7.6 Adaptive filter design

The essential objective of this section is to generate an appropriate control parameter c_j based on adaptive filter algorithms. The general abstract of this section is as following: We will first attempt to express the DLL system with unknown Doppler Effect (unknown τ_i) by a suitable model at each time instant. The approach proposed for realizing the system identification is adaptive filter illustrated in Figure 7.36 [126, 127]. In this configuration, the

DLL system is in parallel with an adaptive filter and both are exited with the same signal $\hat{\tau}_i$. The adaptive filter self-adjusts the changing parameter $\tilde{\tau}_i$ according to an optimization algorithm. One of the most popular adaptive algorithms used today is the Least Mean Squares (LMS) algorithm. Essentially, this algorithm refines $\tilde{\tau}_i$ to minimize the error (ε_i) that occurs between the desired signal (e_i) and the estimated value of the signal ($\tilde{e}_i$), producing the best match between $\tilde{\tau}_i$ and unknown signal τ_i. Consequently, thanks this filter, we are able to effectively observe $\tilde{\tau}_i$ at every time instant. Finnaly, c_j will be set to $\tilde{\tau}_i$ ($c_j = \tilde{\tau}_i$), and will be further inserted into our system in order to self-tune the direct and invers models without the need to dramatically change the local model from time to time.

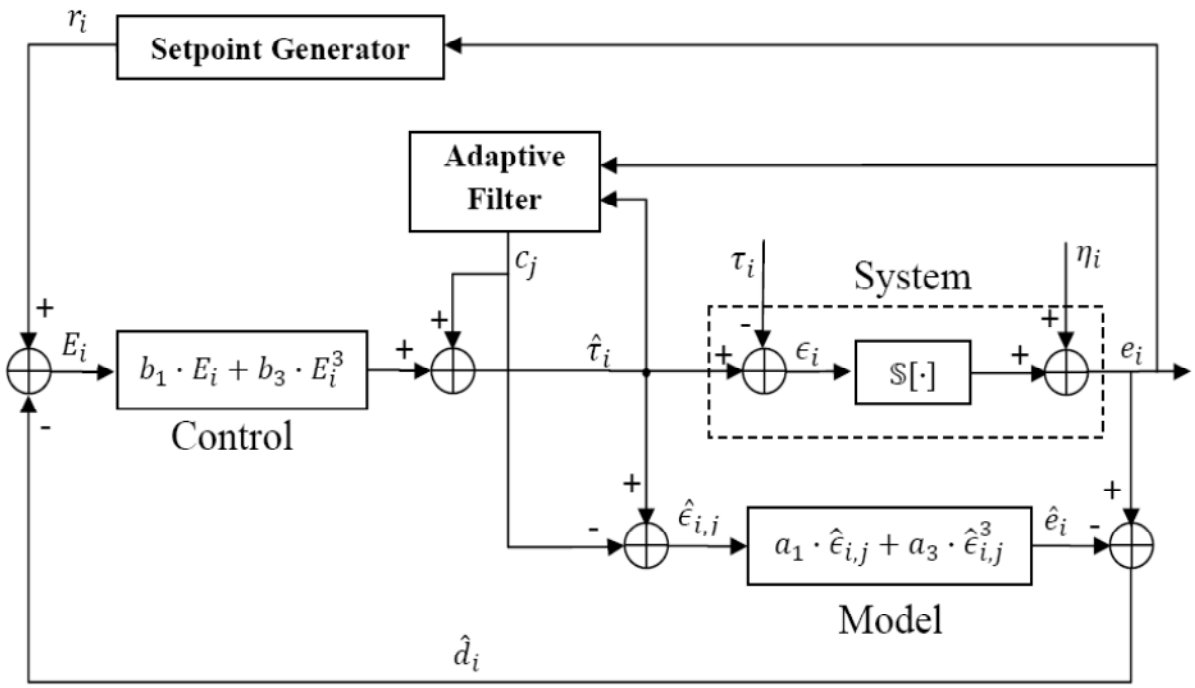


Figure 7.35 EIMMC structure included adaptive filter

To start the discussion of the adaptive filter we take the following assumption: the ambient noise effects are neglected and the DLL system with Doppler Effect has the following model, provided that the DLL system works within the tracking operation area:

$$e_i = a_1(\hat{\tau}_i - \tau_i) + a_3(\hat{\tau}_i - \tau_i)^3 : \quad i = 0,1,2,3,\cdots \tag{7.5}$$

The adaptive filter model is:

$$\tilde{e}_i = a_1(\hat{\tau}_i - \tilde{\tau}_i) + a_3(\hat{\tau}_i - \tilde{\tau}_i)^3 \tag{7.6}$$

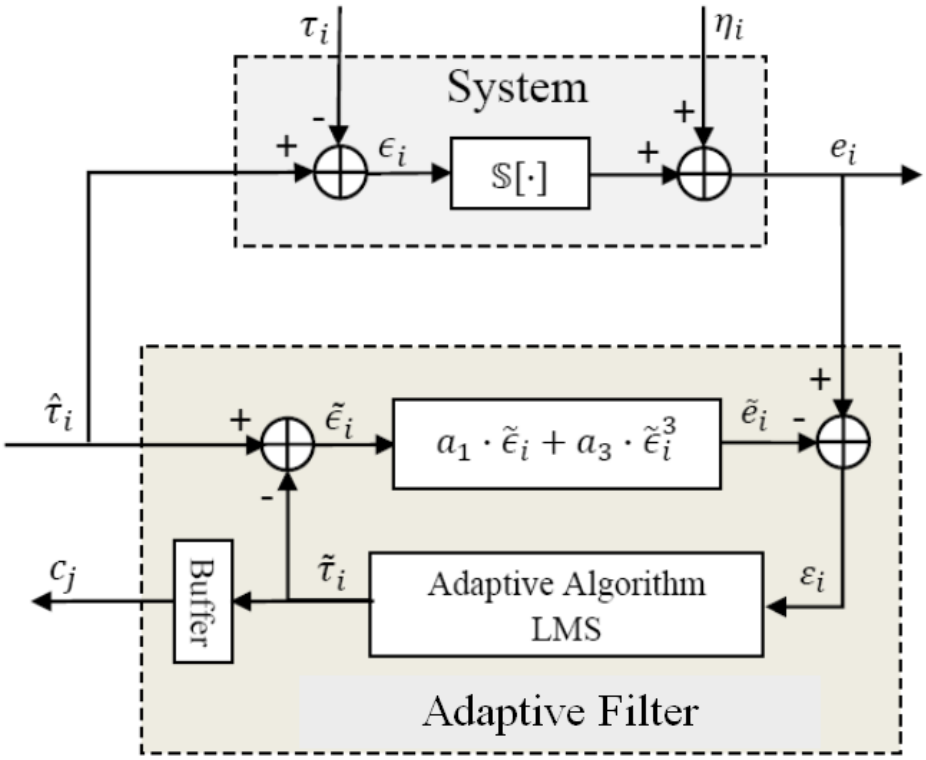


Figure 7.36 Adaptive filter structure for DLL system identification

According to this filter, the input signal $\hat{\tau}_i$ and the parameters $\{a_1, a_3\}$ have already been determined (known and fixed during the adaptive filter process). Thus, the only adjustable coefficient is $\tilde{\tau}_i$. Once the signal e_i and its approximation $\tilde{e}_i$ are found, the error is just the difference between the two signals (e_i & $\tilde{e}_i$) at the current point in time:

$$\varepsilon_i = e_i - \tilde{e}_i = e_i - [a_1(\hat{\tau}_i - \tilde{\tau}_i) + a_3(\hat{\tau}_i - \tilde{\tau}_i)^3] \tag{7.7}$$

Let's now develop this equation, as follows:

$$\varepsilon_i = e_i - [a_1\hat{\tau}_i - a_1\tilde{\tau}_i + a_3\hat{\tau}_i^3 - 3a_3\hat{\tau}_i^2\tilde{\tau}_i + 3a_3\hat{\tau}_i\tilde{\tau}_i^2 - a_3\tilde{\tau}_i^3]$$

$$\varepsilon_i = \underbrace{\frac{e_i - a_3\hat{\tau}_i^3 - a_1\hat{\tau}_i}{a_3}}_{\tilde{y}_i} + \tilde{\tau}_i^3 - 3\hat{\tau}_i \cdot \tilde{\tau}_i^2 + \left(3\hat{\tau}_i^2 + \frac{a_1}{a_3}\right) \cdot \tilde{\tau}_i \tag{7.8}$$

We would like to select $\tilde{\tau}_i$ that makes the cost function $J_i = \varepsilon_i^2$ as small as possible. Thus, to find $\tilde{\tau}_i$ we apply the LMS algorithm [128]. In this example, The LMS algorithm takes the derivative of the cost function J_i with respect to $\tilde{\tau}_i$ (steepest descent concept) in order to find filter coefficient $\tilde{\tau}_i$ which minimizes the cost function. Thus, to find $\tilde{\tau}_i$ we seek the following derivative process:

$$c_j = \arg\min_{\tilde{\tau}_i} J_i \Leftrightarrow \frac{\partial J_i}{\tilde{\tau}_i}\Big|_{c_j=\tilde{\tau}_i} = 0 \tag{7.9}$$

The derivative is given by:

$$2\left\{y_i + \tilde{\tau}_i^3 - 3\hat{\tau}_i \cdot \tilde{\tau}_i^2 + \left(3\hat{\tau}_i^2 + \frac{a_1}{a_3}\right) \cdot \tilde{\tau}_i\right\} \times \left\{3\tilde{\tau}_i^2 - 6\hat{\tau}_i \cdot \tilde{\tau}_i + 3\hat{\tau}_i^2 + \frac{a_1}{a_3}\right\} = 0$$

Rearranging the LMS equation yields:

$$f_{LMS}(\tilde{\tau}_i) = 3\tilde{\tau}_i^5 - 15\hat{\tau}_i \cdot \tilde{\tau}_i^4 + \left(30\hat{\tau}_i^2 + 4\frac{a_1}{a_3}\right) \cdot \tilde{\tau}_i^3 + \left(3y_i - 27\hat{\tau}_i^3 - 9\frac{a_1}{a_3}\hat{\tau}_i\right) \cdot \tilde{\tau}_i^2 + \left(-6y_i\hat{\tau}_i + \left(3\hat{\tau}_i^2 + \frac{a_1}{a_3}\right)^2\right) \cdot \tilde{\tau}_i + \left(3\hat{\tau}_i^2 + \frac{a_1}{a_3}\right) y_i = 0 \tag{7.10}$$

The LMS equation could be written in vector form as follows:

$$f_{LMS}(\tilde{\tau}_i) = \begin{bmatrix}\tilde{\tau}_i^5 & \tilde{\tau}_i^4 & \tilde{\tau}_i^3 & \tilde{\tau}_i^2 & \tilde{\tau}_i & 1\end{bmatrix} \times \begin{bmatrix}\theta_5\\ \theta_4\\ \theta_3\\ \theta_2\\ \theta_1\\ \theta_0\end{bmatrix} = 0; \quad \begin{bmatrix}\theta_5\\ \theta_4\\ \theta_3\\ \theta_2\\ \theta_1\\ \theta_0\end{bmatrix} = \begin{bmatrix}3\\ -15\hat{\tau}_i\\ 30\hat{\tau}_i^2 + 4\frac{a_1}{a_3}\\ 3y_i - 27\hat{\tau}_i^3 - 9\frac{a_1}{a_3}\hat{\tau}_i\\ -6y_i\hat{\tau}_i + \left(3\hat{\tau}_i^2 + \frac{a_1}{a_3}\right)^2\\ \left(3\hat{\tau}_i^2 + \frac{a_1}{a_3}\right) y_i\end{bmatrix} \tag{7.11}$$

The polynomial function in (7.11) has five roots. Thanks Matlab functions it is so easy to find the roots of any polynomial function, even if they are complex. We would like to study and understand the 5-root behavior of the LMS equation for different value of τ_i, in order to find later a suitable and simple way for calculating and selecting the specific LMS root. We start the study with the initial case ($\tau_i = 0$) and without considering the noise effects. Figure 7.37 draws the LMS function curve in the case: $\tau_i = 0$. It shows that the LMS function has 5 real roots: $\tilde{\tau}_i \in \{0, \pm 1.1, \pm 1.9\}$, where one root is our desired solution and the rest 4 roots are undesired. Since the desired LMS solution must be equal (or so close to) the unknown input value τ_i, so the optimal LMS equation root here is $\tilde{\tau}_i = 0$. Next step is to change τ_i linearly (taking Doppler shift into account) and study the 5-root trajectory corresponding to different τ_i. Figure 7.38 illustrates that as τ_i increases with linear form, the LMS roots shift incrementally with keeping the same difference between them. Besides, the desired root has always real value and tracks τ_i.

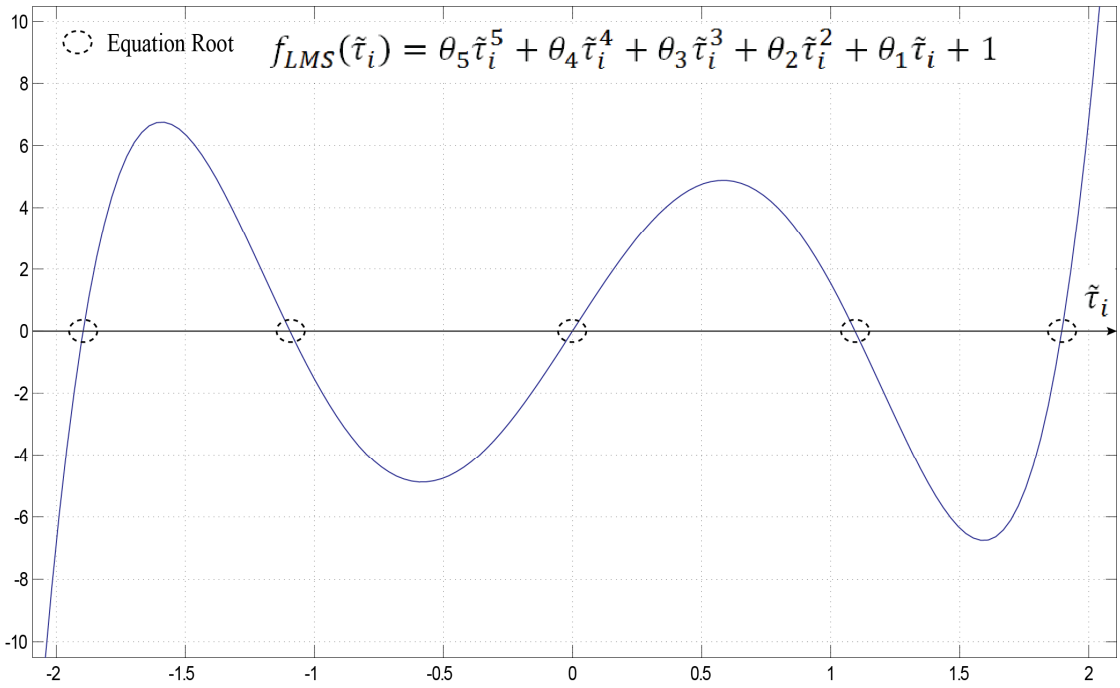


Figure 7.37 The graph of the polynomial function of degree 5 mentioned in (7.10) for $\tau_i = 0$

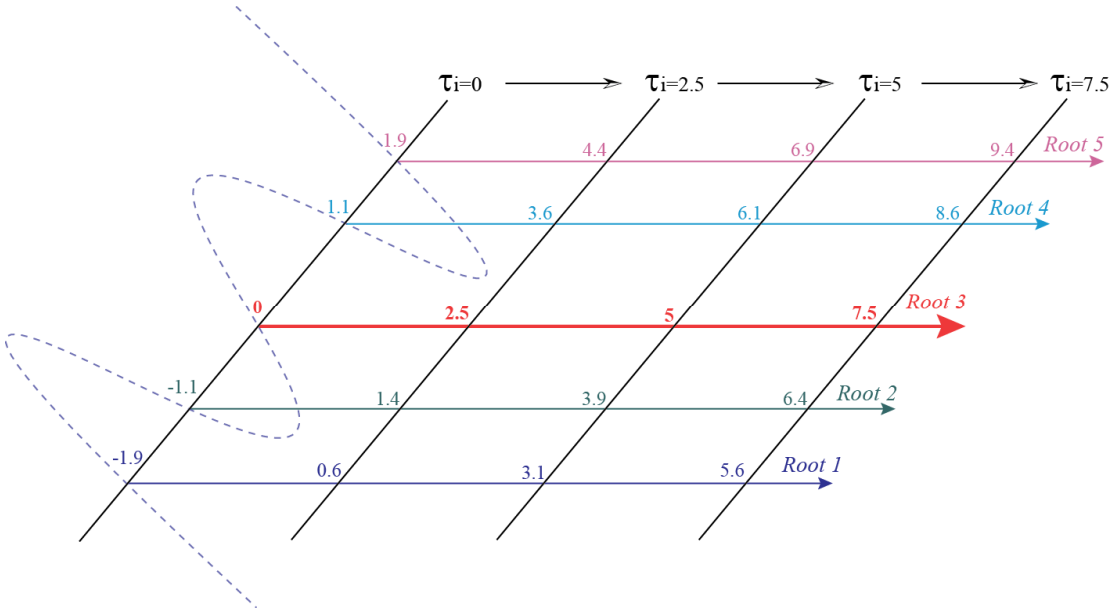


Figure 7.38 LMS root trajectory (locus)

After studying the LMS root behavior based on Matlab simulation results, we focus now on seeking the practical way to find and select the desired root for the LMS function $f_{LMS}(\tilde{\tau}_i)$ among these 5-roots. Generally speaking, methods used to solve such problems are called *root-finding* or *zero-finding* methods [129]. It is not possible to solve the LMS polynomial

equation with order 5 using analytical root-finding methods, while such problem could be approximately solved using numerical methods. Common root finding approaches for functions with one independent variable are: Bisection Method, Secant Method, False Position Method and Newton's Method [129]. The method proposed here in order to find LMS solution is Newton's Method [129, 130].

Newton's method is a popular numerical root finding algorithm, it is based on the first-order Taylor series expansion theory [131]. The idea of the method, shown in Figure 7.39, is as follows: to start with an initial guess point ($\tilde{\tau}_{i,0}$) which should be relatively close to true root. Then, find the slope (the tangent line) of the function at this initial point and use the zero of the tangent line as the next reference point ($\tilde{\tau}_{i,1}$). We see that the next point $\tilde{\tau}_{i,1}$ is a better approximation than the actual point $\tilde{\tau}_{i,0}$ for the desired root of the LMS function. The process is repeated until the root is converged. To sum up, The Newton's way to resolve the hard nonlinear problem is to replace it by a succession of linear problem whose solutions converge effectively to the solution of the nonlinear problem.

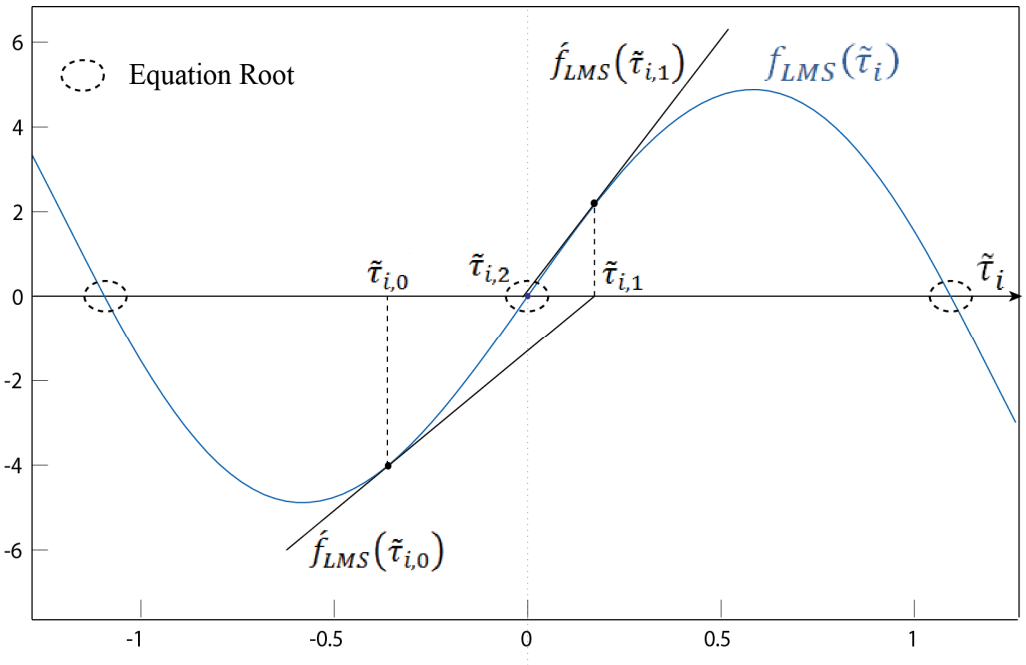


Figure 7.39 Applying Newton's root-finding method on LMS function $f_{LMS}(\tilde{\tau}_i)$

Hence, to find a root of LMS function $f_{LMS}(\tilde{\tau}_i) = 0$ with the initial guess $\tilde{\tau}_{i,0}$ (the initial reference point), the Newton's method is applied as follows:

$$\tilde{\tau}_{i,1} = \tilde{\tau}_{i,0} - \frac{f_{LMS}(\tilde{\tau}_{i,0})}{\acute{f}_{LMS}(\tilde{\tau}_{i,0})} \tag{7.12}$$

where $\acute{f}_{LMS}(\tilde{\tau}_{i,0})$ is the derivation of the function $f_{LMS}(\tilde{\tau}_i)$ at the point $\tilde{\tau}_{i,0}$. The process is repeated, as in (7.13):

$$\tilde{\tau}_{i,n+1} = \tilde{\tau}_{i,n} - \frac{f_{LMS}(\tilde{\tau}_{i,n})}{\acute{f}_{LMS}(\tilde{\tau}_{i,n})} \tag{7.13}$$

until the difference between the two successive point becomes less than a predetermined error tolerance δ. On the other word: if $\left|\tilde{\tau}_{i,n+1} - \tilde{\tau}_{i,n}\right| < \delta$ the process stops, otherwise $n = n + 1$ and it repeats applying (7.13), and so on.

Generally speaking, The Newton's method is much more efficient and converges faster than other iterative numerical root-finding methods such as the bisection method [129, 130]. However, the Newton's method requires the calculation of the derivative of a function at the reference points. Sometimes the function is not differentiable or so difficult to compute its derivative. It is possible to replace the hard analytical derivative operation by the approximate numerical one, but at the expense of the convergence rate and solution accuracy. Besides, the reference point is required to not be stationary, which means the function derivation at this point must not be zero, otherwise the division by zero will occur in (7.13) and the method will catastrophically terminate. Furthermore, Newton's method requires also a good initial guess in order to converge correctly and rapidly to the desired root. Otherwise it may not converge if started too far away from a root, or it may converge to another undesired root when more than one root exists [129].

In addition of the previous advantages (very fast convergence properties with high accuracy close to solution), the Newton's method is a promising iterative algorithm here in order to find the desired root related to the LMS function. Because, the derivation of $f_{LMS}(\tilde{\tau}_i)$, mentioned in (7.11), is always existing and carried out easily:

$$\acute{f}_{LMS}(\tilde{\tau}_i) = 5\theta_5\tilde{\tau}_i^4 + 4\theta_4\tilde{\tau}_i^3 + 3\theta_3\tilde{\tau}_i^2 + 2\theta_2\tilde{\tau}_i + \theta_1 \tag{7.14}$$

Figure 7.40 plots the derivative equation in (7.20) in the case $\tau_i = 0$. It has 4 roots at $\{\pm 0.6, \pm 1.6\}$. Thus, to converge to the desired root $\tilde{\tau}_{i=0}$ the first initial point should be inside the following interval: $\tilde{\tau}_{i,0} \in]-0.6, 0.6[$. The additional raison to select Newton's method concerns the facility to choose the good initial point which guarantees to efficiently converge to the specific desired root and without passing via any stationary point. Since, the timing acquisition stage in the synchronization application assures to set the timing offset $\tau_{i=0}$ close to zero, so the first initial point is proposed to be $\tilde{\tau}_{0,0} = 0$. Then, for observing τ_i at each time instant we consider that the solution of the previous period $\tilde{\tau}_{i-1}$ as the initial point for the actual period $\tilde{\tau}_{i,0}$ ($\tilde{\tau}_{i-1} \rightarrow \tilde{\tau}_{i,0}$), or we can consider the input signal $\hat{\tau}_i$ as the initial point ($\hat{\tau}_i \rightarrow \tilde{\tau}_{i,0}$).

We present numerical example which illustrates the efficiency and rapidity of the Newton's method to find the root solution of LMS equation. Assuming that $\tau_i = 0$, so the resulted optimal observed solution must be zero. Consider that the initial guess equal $\tilde{\tau}_{i,0} = -0.1$, the sequence given by Newton's method is as follows where n denotes the iteration number and δ_n refers to the distance between the previous and actual points:

$$n = 1:\ \tilde{\tau}_{i,1} = \tilde{\tau}_{i,0} - \frac{f_{LMS}(\tilde{\tau}_{i,0})}{\acute{f}_{LMS}(\tilde{\tau}_{i,0})} = -0.1 - \frac{-1.277}{12.484} = \underline{0.00}22932 \qquad \Longrightarrow \delta_1 = 0.10229$$

$$n = 2:\ \tilde{\tau}_{i,2} = \tilde{\tau}_{i,1} - \frac{f_{LMS}(\tilde{\tau}_{i,1})}{\acute{f}_{LMS}(\tilde{\tau}_{i,1})} = 0.0022932 - \frac{0.029614}{12.913} = -\underline{0.000000026}848 \Longrightarrow \delta_2 = 0.0022932$$

$$n = 3:\ \tilde{\tau}_{i,3} = \tilde{\tau}_{i,2} - \frac{f_{LMS}(\tilde{\tau}_{i,2})}{\acute{f}_{LMS}(\tilde{\tau}_{i,2})} = \underline{0.0000000000000000000000}43013 \qquad \Longrightarrow \delta_3 = 2.6848e-8$$

The correct digits are underlined in the above example. We see that the number of correct digits after the decimal point increases from 2 (for $n = 1$) to 7 and 22 for $n = 2\ \&\ 3$ respectively, illustrating the high convergence rate.

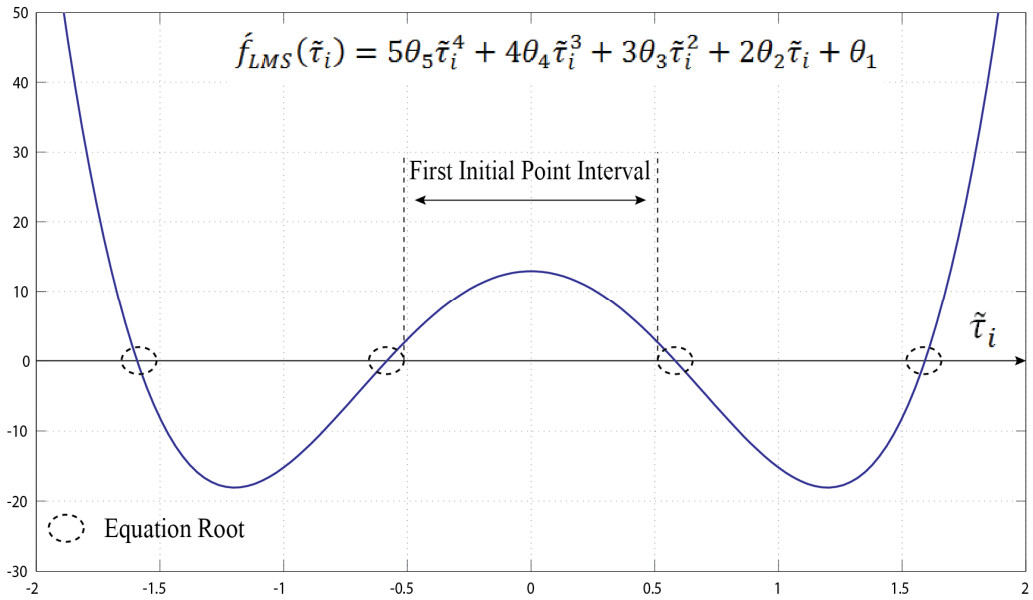


Figure 7.40 The derivative of the LMS function for $\tau_i = 0$

The iteration process continues until $\left|\tilde{\tau}_{i,n+1} - \tilde{\tau}_{i,n}\right| < \delta$, where δ is set, for instance, to 0.0001. As shown in the above numerical example that Newton's method usually requires two or three iterations for achieving the criterion: $\left|\tilde{\tau}_{i,n+1} - \tilde{\tau}_{i,n}\right| < \delta = 0.0001$. Finally, after finding the specific root of the LMS equation, which is considered as the observed value for τ_i. The output signal c_j will be updated as $c_j = \tilde{\tau}_i$, and will be further inserted into our system in order to self-tune the direct and inverse models. The adaptive filter procedure is illustrated

in Figure 7.41. Figures 7.42 and 7.43 depict the performance of EIMMC system combined with adaptive filter block. We notice that the system and its model outputs $(e, \hat{e})$ have extremely the same curve form in spite of Doppler presence. It means that due to the specific modifications in the structure of the internal model control, the DLL system is able to predict and to track the variations of the input signal efficiency and with high robustness to the distribution and with high stability, even in the presence of Doppler Effect or ambient noise. These modifications consist of three essential concepts: multi-model approach, moving average filter and adaptive filter. Furthermore, the proposed modifications have also provided to the IMC system with the predictive features.

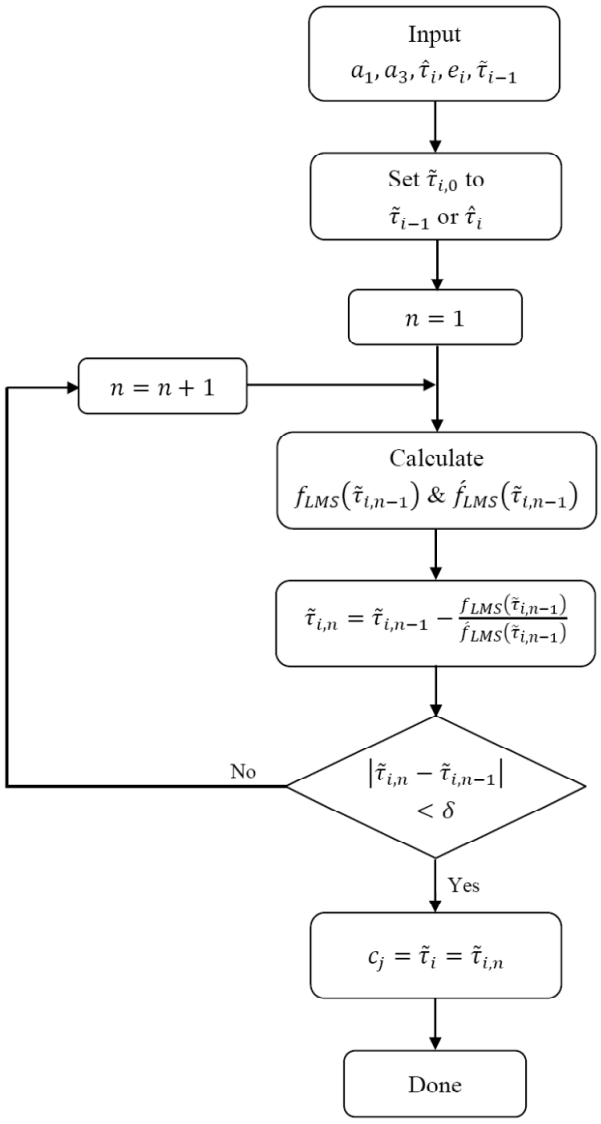


Figure 7.41 Adaptive filter scheme

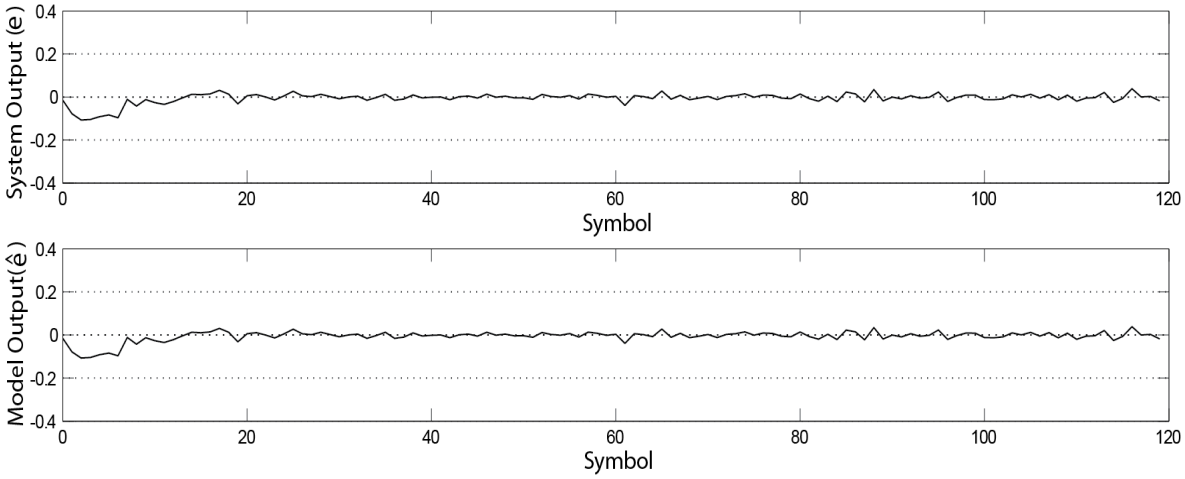


Figure 7.42 Behavior of EIMMC with adaptive filter block for $\delta = 0.0001$ and SNR=20 dB

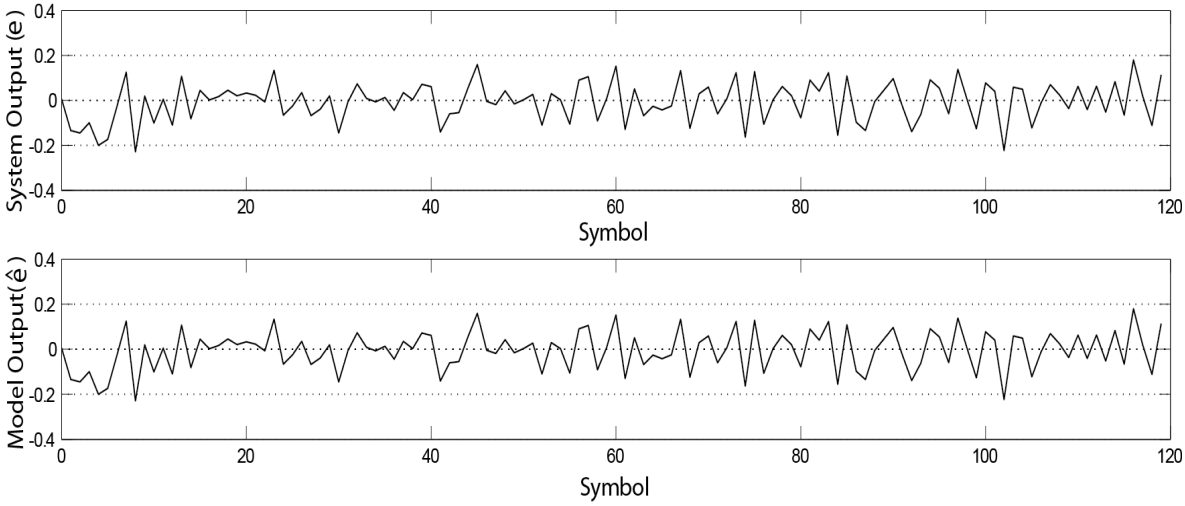


Figure 7.43 Behavior of EIMMC with adaptive filter block for $\delta = 0.0001$ and SNR=3 dB

7.7 Performance comparaision

The aim of this section is to compare the tracking performance between the classical DLL system, proposed in Chapter 5, and IMC DLL system. For Matlab simulation, the natural frequency ω_n for the classical DLL system is 0.1 GHz and data-aided bit M is 16. Figure 7.44 shows the mean square error (tracking jitter) for the two systems. In this figure, the MSE results are normalized by T_p^2, and plotted versus the signal-to-noise ratio (SNR) per pulse. Doppler Effect is neglected $m = 0$. The simulations confirm that in the absence of Doppler Effect the classical DLL approach has much better tracking behavior than the IMC DLL approach.

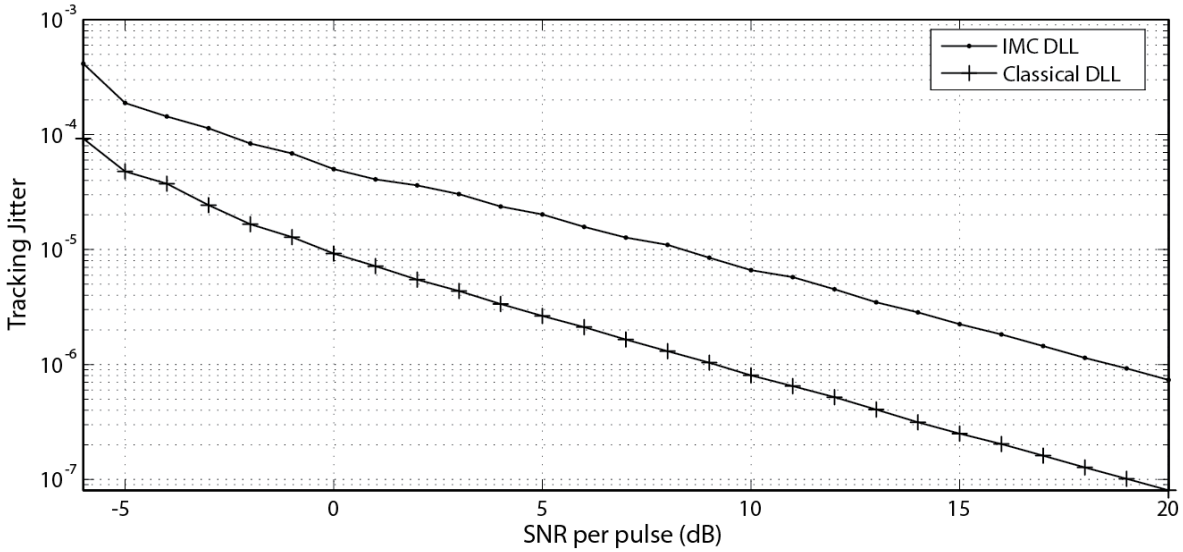


Figure 7.44 Tracking jitter for DLL

In order to evaluate the transient response performance, the input signal time varies as follows $\tau(n) = m \cdot nT_s = 0.01nT_p$. Figure 7.45 presents the behavior of both systems in the presence of input time variations $\tau(n)$. It is evident that IMC DLL follows the input signal $\tau(n)$ much faster than the classical DLL. Thus it has significantly so low rise time (high transient response). However, after the equilibrium is attained, the tracking error variance of the classical DLL approach is better than the IMC DLL.

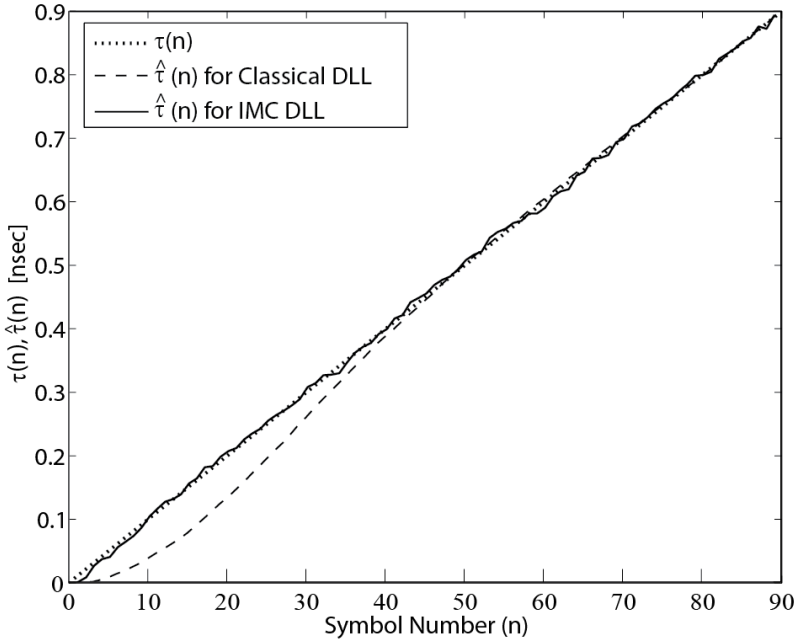


Figure 7.45 DLL transient response for ramp input (SNR=3 dB)

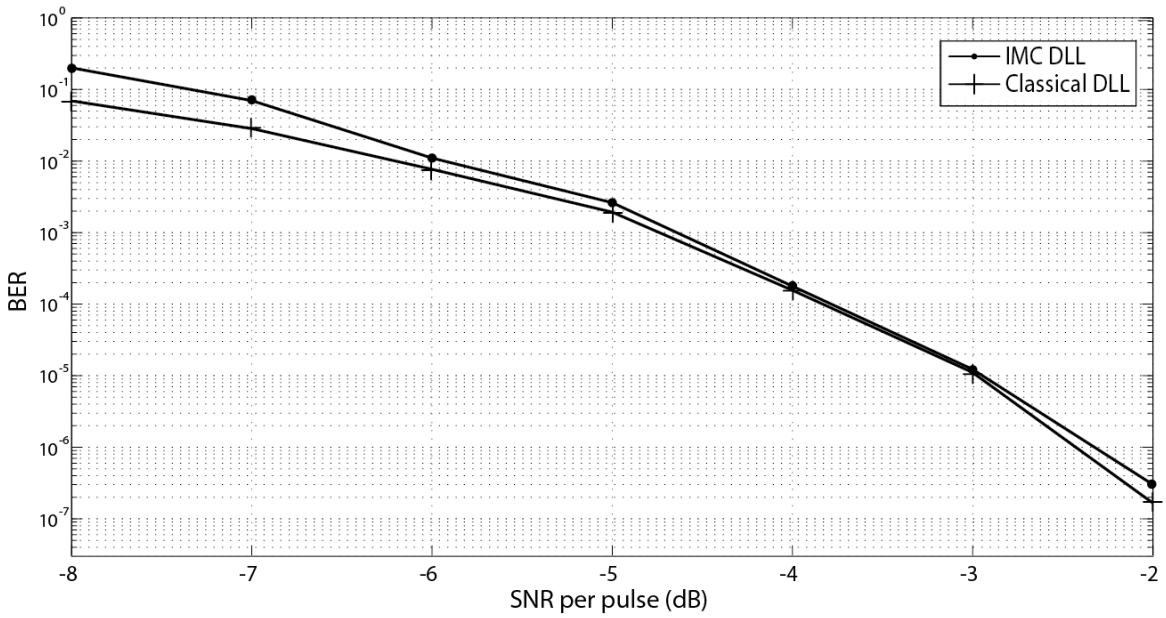
The BER comparisons for both systems are depicted in Figure 7.46. The simulation has been executed two times: In Figure 7.46.a the BER comparisons occur while the input signal time has being varied as follows $\tau(n) = 0.01nT_p$. While in Figure 7.46.b the time has being varied according to $\tau(n) = 0.1nT_p$. We observe that the classical DLL system has better BER performance.

Consequently, as the relative velocity between transmitter and receiver is constant (m is constant) the tracking performance of the classical DLL system is always better even for different value of SNR and m. However, when the relative motion accelerates or decelerates (m is changed) the IMC DLL system has so fast transient response and is capable to effectively follow the input variation, while the classical DLL system has high opportunity to loss the tracking; we have met this problem during executing the Matlab simulations. So, for the future work we can for instance combine both systems to get a hybrid DLL system which has the potential advantages of the two systems.

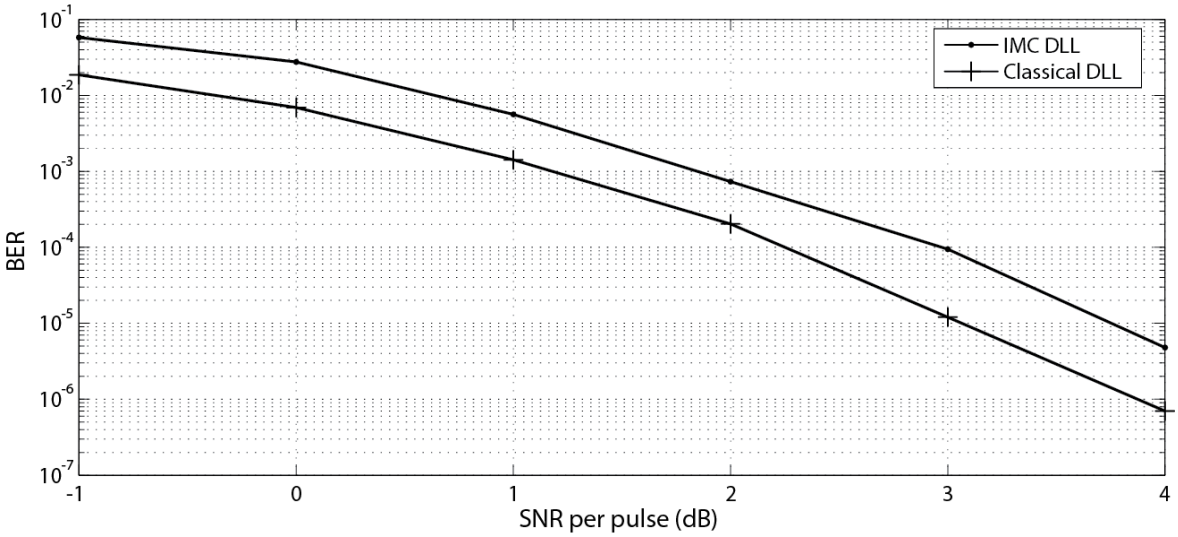
7.8 Conclusion

In the chapter, we studied the behavior of IMC tracking system taking into account Doppler Effect. We found that the standard IMC controller does not correctly work in Doppler presence, and the tracking system is exposed to the high possibility of being injured. Thus, the objective of this chapter was to develop more perfectly the IMC behavior even in the presence of Doppler Effect. This chapter has consisted of three main points: **First**, we proposed the multi-model approach as a potential solution for Doppler problem; where the tracking region is divided into several sub-regions, the DLL system within each sub-region is represented by an appropriate local-model. Switching between these local-models occurs at each time when the operating point exceeds certain threshold. **Second**, we noticed that Doppler shift produces also the residual steady-state error at the system output, and this undesired static error may impact on the system quality. To eliminate the residual steady-state error, the moving average filter has been built within the IMMC system. This modification does not only permit to mitigate the static error, but permits also to predict the future value of timing offset. **Third**, we found that the transient performance of the modified IMC system is improved as the threshold value decreases. So, instead of switching between local-models with dramatic jumps, the model switches are smoothly achieved by using adaptive control technique. To sum up, the IMC structure in this chapter is developed by adding three essential concepts: multiple models, moving average filter and adaptive filter. Thanks of these modifications the IMC tracking system is able to efficiently follow the variation of the input signal with high robustness and high stability. Finally, we have compared the tracking performance between the classical DLL, which has been proposed in Chapter 5, and the DLL combined with IMC technique. The results have shown that as the relative velocity between transmitter and receiver is constant the

tracking performance of the classical DLL system is better. However, when the relative motion accelerates or decelerates the IMC DLL system has so fast transient response and is capable to effectively follow the input variation, while the classical DLL system has high opportunity to loss the tracking.



(a)



(b)

Figure 7.46 BER for both DLL: (a) $\tau(n) = 0.01nT_p$, (b) $\tau(n) = 0.1nT_p$

Conclusion and Perspectives

This thesis has demonstrated that IR-UWB is a feasible technology for various wireless communication applications in short-range such as dynamic wireless sensor networks. However, timing synchronization is the most critical challenge in enabling the potential benefits of UWB transmissions. Therefore, the thesis goal has been to evaluate and improve timing synchronization stage in UWB communication systems.

In Chapter 2, we have started with general overview on dynamic wireless sensor networks and feasibility of UWB for it as a wireless communication system, including the advantages, characteristics and regulations of UWB technology. The second part has introduced the timing synchronization challenge in UWB systems and proposed the timing with dirty template approach as an adequate solution for UWB systems, in order to achieve rapid, accurate and low-complexity timing synchronization. The proposed timing approach is based on correlating the received signal with dirty template extracted from the received waveforms. Finally, we considered that timing synchronization system consists of three main blocks: signal detection, timing acquisition and tracking. Each block of them has been explained in a separated chapter of this dissertation.

In Chapter 3, we have focused on the first stage of timing synchronization system: signal detection. In the first part of this chapter, we have introduced the concepts of detection theory. The signal detection is initially achieved by comparing the received captured samples with a threshold, if the sample exceeds this threshold we say that the signal is present; otherwise, we consider that the signal is absent. The analytic study has shown that selecting the threshold value affects detection performance. The Neyman-Pearson theorem has been suggested in order to set the suitable threshold value. In the second part, the detection criteria have been applied on dirty template UWB system to establish a good detector structure and to define optimal threshold values. Finally, the results have been analyzed in three ways: a graph of the probabilities of false alarm and detection versus the threshold, histograms of test statistic under each hypothesis and a graph of the ROC curves. The simulation results have showed that the detection performance of the dirty template approach is improved by increasing the signal-to-

noise ratio or data-aided number. But these improvements reduce the power efficiency and complicate the receiver design.

In Chapter 4, we have moved on to the second stage of synchronization: timing acquisition. First, we have derived the Cramer-Rao lower bound for DT-UWB system, which is used as a fundamental performance limit for any timing estimator. Next, we have developed the TDT acquisition estimator with and without training symbols by using maximum likelihood and mean-square algorithms. The proposed TDT synchronization relies on searching a peak in the output of the sliding correlation between the received signal and its dirty template. However, we found multiple maxima points around the peak at the output of the correlator, which may degrade the estimation performance of timing offset error (TOE). To avoid this problem, we have added a suitable window filter to the structure of the cross-correlator. This modified approach guarantees to obtain a single maximal peak inside the estimator range and improve the estimation error performance. This window filter also contributes to reduce the unwanted noise effects on the timing estimation. Both the theoretical analysis and Matlab simulation results have shown the performance of the proposed timing estimator, and confirmed that the timing estimator with training symbols has high performance and fast execution compared to that without data-aided symbols, but at the expense of bandwidth efficiency.

After the signal detection and timing acquisition tasks had successfully been completed and the timing offset error had been confined within a fraction of UWB pulse duration. In Chapters 5, 6 and 7, we have focused on the last step of synchronization system: tracking. Thus, the objective of these chapters has been to design a suitable tracking unit for DT-UWB systems, which contributes to maintain the satisfactory synchronization throughout the transmission period between receiver and transmitter, and alleviate timing offset variations due to the relative transmitter-receiver motion. According to the literature, the delay locked loop is the fundamental technique used for tracking purpose in UWB systems. We considered that DLL system consists of two main blocks: open-loop system (early-late gate) and closed-loop controller (offset decision). The first part of Chapter 5 has studied the specifications of open-loop DLL for DT-UWB systems. This dissertation has provided two efficient approaches for designing the closed-loop controller. The first approach has been based on conventional algorithms, such as iterative method and Winner filter, while the second approach has been based on Internal Model Control, which is a novel and promising technique for radio and telecommunication applications.

The second part of Chapter 5 has employed the conventional algorithms for deriving the closed-loop control structure. We have started with designing an appropriate first-order closed-loop controller based on a simple iterative method. However, theoretical analyses have confirmed that in the presence of dynamic Doppler effects, the first-order DLL is no longer able to track the timing variations in received signal. For efficiently tracking the input UWB

waveform in the presence of ambient noise and Doppler effects, we suggested DLL with second-order closed-loop controller as an efficient alternative structure. The parameters of suggested DLL have been selected by applying Wiener-filter theories, in order to improve transient behavior as well as noise handling ability.

Chapters 6 and 7 have employed IMC for deriving the closed-loop control structure. Chapter 6 has focused on designing IMC DLL system and on analyzing the tracking behavior in the absence of Doppler effects. On the other hand, Chapter 7 has developed the tracking performance of IMC DLL system taking into account Doppler effects. So in Chapter 6, we have given an overview of Internal Model Control concept and combined it with DLL tracking system. The chapter has shown that IMC controller is composed of two blocks (models): an inverse model (control model) connected in series with the open-loop DLL system and a forward model (system model) connected in parallel with the open-loop DLL system. We have proved that this structure has a good performance of overcoming disturbance and deviations of model parameters. Next, we have applied Least-Squares (LS) estimation algorithm methods in order to determine the optimal coefficients for the system and control models.

In Chapter 7, we have studied the behavior of IMC tracking system taking into account Doppler effects. The initial simulation results have shown that the transient performance for the standard IMC structure is disappointing, and the tracking extremely suffers from Doppler effects. Hence, in this chapter we have developed the IMC-DLL structure to be able to keep the satisfactory synchronization even in the presence of Doppler effects. This chapter has consisted of three main points: First, we have proposed the multi-model approach as a potential solution for Doppler problem; where the tracking region is divided into several sub-regions, the DLL system within each sub-region is represented by an appropriate local-model. Switching between these local-models occurs at each time when the operating point exceeds certain threshold. Second, we noticed that Doppler shift produces also the residual steady-state error at the system output, and this undesired static error may impact on the system quality. To eliminate the residual steady-state error, the moving average filter has been built within the IMMC system. Third, the simulation has shown that the transient performance of the modified IMC system is improved as the threshold value decreases. So, instead of switching between local-models with dramatic jumps, the model switches have been smoothly achieved by using adaptive control technique. To sum up, the IMC structure in this chapter has been developed by adding three essential concepts: multiple models, moving average filter and adaptive filter. Finally, we have compared the tracking performance between the conventional DLL, which has been proposed in Chapter 5, and the DLL combined with IMC technique. The results have shown that as the relative velocity between transmitter and receiver is constant the tracking performance of the conventional DLL system is better. However, when the relative motion accelerates or decelerates the IMC DLL system has so fast transient response and is capable to

effectively follow the input variation, while the conventional DLL system has high opportunity to loss the tracking.

This work has led to three published journal papers, four published international conference papers, and one chapter in book. Furthermore, two additional journal papers are under process.

Perspectives

There are several critical issues that should be studied in the future work on the synchronization of UWB communications and its related topics. We present here some of these research directions that need to be further investigated:

1. In chapter 2, we have seen that timing synchronization with dirty template technique can be applied to UWB receiver even in the presence of time-hopping code or inter-frame interference, provided that inter-symbol interference (ISI) is absent. As mentioned in chapter 2, one solution to avoid ISI is to respect the condition (2.15). But in the case of ISI, the behavior of timing with dirty template system needs to be studied in detail in the future. Besides, time-hopping code in this thesis is used to improve the synchronization performance. It can also enable different users to transmit their digital information in the same propagation channel. The TDT method has been studied for single user only. Further future research could analysis the timing method in UWB communication systems for multiple users, taking into account inter-user interference.
2. This thesis has not included any experimental results, demonstrating practically the efficiency of the proposed timing algorithms. Therefore, in the future work, we plan to implement the dirty template UWB system (including signal detection, timing acquisition and tracking stages) on DSP or FPGA circuits. Initially, the platform will consist of two nodes (transmitter and receiver), and the tests will be achieved for an indoor wireless environment and since the influence of mobility between the transmitter and receiver is approximately neglected. This platform allows us to evaluate the UWB circuit performance and to validate our synchronization process concepts.
3. As mentioned before in Chapter 7, the tracking performance of the conventional DLL system is better than IMC DLL. However, when the relative motion accelerates or accelerates the IMC DLL system has so fast transient response and is able to effectively follow the input variation, while the conventional DLL system has high opportunity to lose the tracking. So, for the future work we can improve the timing performance of UWB systems by combining conventional and IMC methods together to get a hybrid DLL system which has the potential advantages of the two approaches.

4. In this PhD research work, we have studied the performance of synchronization system in the case of linear form of Doppler shift. It seems interesting for the future work to analyze the performance of DT-UWB system for other forms of Doppler shift.

5. Finally, we can employ IMC technique for controlling the DLL tracking structure for other narrowband communication systems.

Appendix 2.A

Band Pass Noise Model

Let the receiver frontend be modeled as an ideal bandpass filter with double sided bandwidth $B(\gg 1/T_s)$ and center frequencyf_0. The noise $w(t)$ across symbols T_s is bandpass filtered white Gaussian noise with double-sided power spectral density $N_0/2$ (see Figure 2.A.1). The noise spectral density function is given by:

$$S_w(f) = \begin{cases} N_0/2, & f \in (f_0 \mp B/2) \; and \; f \in (-f_0 \mp B/2) \\ 0, & otherwise \end{cases} \tag{2.A.1}$$

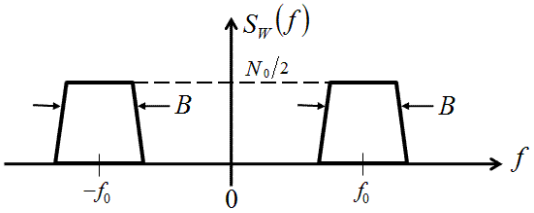


Figure 2.A.1 Noise power spectral density

The autocorrelation function is the inverse Fourier transform of the spectral density function:

$$R_w(\tau) = \int_{-\infty}^{\infty} e^{j2\pi f\tau} S_w(f) df$$

$$R_w(\tau) = 2\int_{0}^{\infty} cos(2\pi f\tau) S_w(f) df$$

$$R_w(\tau) = 2\int_{f_0-B/2}^{f_0+B/2} \frac{N_0}{2} cos(2\pi f\tau)\, df$$

$$R_w(\tau) = \frac{N_0}{2\pi\tau}[sin(2\pi f\tau)]_{f=f_0-B/2}^{f=f_0+B/2}$$

$$R_w(\tau) = \frac{N_0}{2\pi\tau}[sin(2\pi(f_0 + B/2)\tau) - sin(2\pi(f_0 - B/2)\tau)]$$

$$R_w(\tau) = \frac{N_0}{\pi\tau} sin(\pi B\tau) cos(2\pi f_0\tau)$$

$$R_w(\tau) = N_0 B sinc(B\tau) cos(2\pi f_0 \tau) \quad (2.A.2)$$

Thus, the noise N_0 represents a zero-mean Gaussian noise with power $R_w(0) = N_0 B$.

Appendix 2.B

Dirty Template Noise

In this section, we derive the Dirty Template noise $\breve{\omega}[n]$ model and calculate its statistic properties. Looking at the equation (2.21), it could be represented as following:

$$x(t + nT_s) = w(t + nT_s) + f(t + nT_s, \tau) \tag{2.B.1}$$

where:

$$f(t + nT_s, \tau) = \sqrt{\mathcal{E}_s} s[n - n_s - 1] p_R(t + T_s - \tau)\Big|_{t=0}^{t=\tau} + \sqrt{\mathcal{E}_s} s[n - n_s] p_R(t - \tau)\Big|_{t=\tau}^{t=T_s} \tag{2.B.2}$$

and $w(t)$ represents the bandpass filtered zero-mean additive white Gaussian noise (AGWN) with power spectral density (PSD) $N_0/2$ and with double sided bandwidth B ($\gg 1/T_s$) (see Appendix 2.A). The autocorrelation function of $w(t)$ had been derived in (2.A.2) as following:

$$R_w(t) = N_0 B sinc(B\tau) cos(2\pi f_0 \tau) = TF^{-1}\left\{\frac{N_0}{2}\right\}_{\pm f_0 - B/2}^{\pm f_0 + B/2} \tag{2.B.3}$$

f_0 represents the center frequency of receiver bandwidth filter. To simplify $f(t + nT_s, \tau)$, we could rewrite (2. B. 2) as:

$$f(t + nT_s, \tau) = \sqrt{\mathcal{E}_s} \sum_{m=0}^{1} s[n - n_s - m] p_R(t + mT_s - \tau) \tag{2.B.4}$$

Figure 2.B.1 represents the noise-free parts of the received signal. To simplify, $p_R(t)$ is plotted as a triangle.

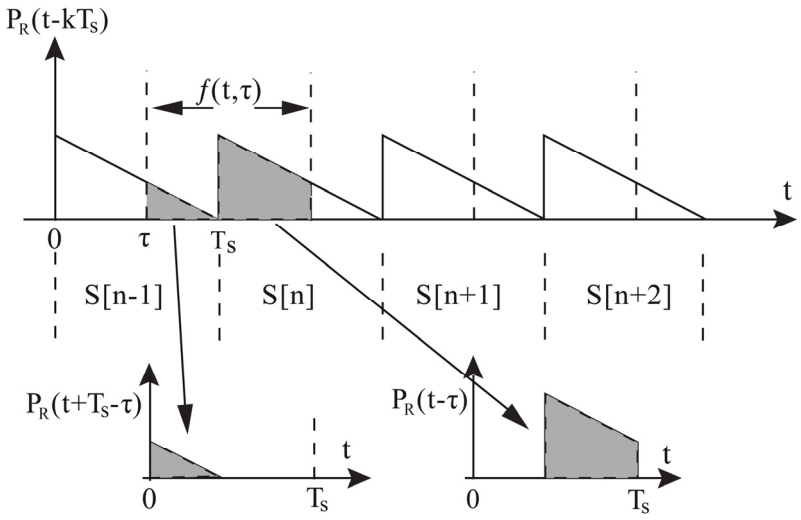


Figure 2.B.1 The recieved waveform

Note: The function $p_R(t-\tau)$ is given here by (Figure 2.B.1):

$$p_R(t-\tau) = \begin{cases} 0 & : t \in \{0, \tau) \\ p_R(t-\tau) & : t \in \{\tau, T_s) \end{cases}$$

The function $p_R(t+T_s-\tau)$ is given by (Figure 2.B.1):

$$p_R(t+T_s-\tau) = \begin{cases} p_R(t+T_s-\tau) & : t \in \{0, \tau) \\ 0 & : t \in \{\tau, T_s) \end{cases}$$

---

1. Dirty Template Noise Model $\breve{\omega}[n]$

From (2.26), the sampled noise $\breve{\omega}[n]$ is composed of three terms, as following:

$$\breve{\omega}[n] = \xi_1[n] + \xi_2[n] + \xi_3[n] \tag{2.B.5}$$

1.1. $\xi_1[n]$:

$$\xi_1[n] = \sqrt{\mathcal{E}_s} \sum_{m=0}^{1} s[n - n_s - m] \int_0^{T_s} p_R(t + mT_s - \tau) w(t + (n+1)T_s)\, dt \tag{2.B.6}$$

Substituting (2.B.4) in (2.B.6), the term $\xi_1[n]$ could be simplified by:

$$\xi_1[n] = \int_0^{T_s} f(t + nT_s, \tau) w(t + (n+1)T_s)\, dt \tag{2.B.7}$$

1.2. $\boldsymbol{\xi_2[n]}$:

$$\xi_2[n] = \sqrt{\mathcal{E}_s}\sum_{m=0}^{1} s[n-n_s-m+1]\int_0^{T_s} p_R(t+mT_s-\tau)w(t+nT_s)\,dt \qquad (2.B.8)$$

Substituting (2.B.4) in (2.B.8), the term $\xi_2[n]$ could be simplified by:

$$\xi_2[n] = \int_0^{T_s} f(t+(n+1)T_s,\tau)w(t+nT_s)\,dt \qquad (2.B.9)$$

1.3. $\boldsymbol{\xi_3[n]}$:

$$\xi_3[n] = \int_0^{T_s} w(t+nT_s)\,w(t+(n+1)T_s)dt \qquad (2.B.10)$$

Clearly, the statistic properties of the DT noise $\breve{\omega}[n]$ depends on information-bearing symbols $\{s[n]\}$. Therefore, to simplify calculus of the mean and the autocorrelation of $\breve{\omega}[n]$, we suppose that the symbols $\{s[n]\}$ are independent identically distributed (i.i.d.) PAM (binary pulse amplitude modulation) symbols with zero-mean and unit-energy $E\{s[k]s[l]\} = \delta_{k,l}$ (see Table 2.B.1).

Table 2.B.1 The autocorrelation function of $s[k]$

$\boldsymbol{s[k]}$	$+1$	$+1$	-1	-1
$\boldsymbol{s[l]}$	$+1$	-1	$+1$	-1
$\boldsymbol{s[k]s[l]}$	$+1$	-1	-1	$+1$
$\boldsymbol{s[k]s[k]}$	$+1$	$+1$	$+1$	$+1$
$\boldsymbol{E\{s[k]s[l]\}}$		0		$: k \neq l$
$\boldsymbol{E\{s[k]s[k]\}}$		$+1$		

2. The mean of Dirty Template Noise

$$E\{\breve{\omega}[n]\} = E\{\xi_1[n]\} + E\{\xi_2[n]\} + E\{\xi_3[n]\}$$

2.1. $\boldsymbol{E\{\xi_1[n]\}}$ & $\boldsymbol{E\{\xi_2[n]\}}$

$$E\{\xi_1[n]\} = \sqrt{\mathcal{E}_s}\sum_{m=0}^{1} s[n-n_s-m]\int_0^{T_s} p_R(t+\ mT_s-\tau)\underbrace{E\{w(t+(n+1)T_s)\}}_{=0}dt = 0$$

The mean of the first and the second terms ($E\{\xi_2[n]\} = E\{\xi_1[n]\} = 0$) is zero, because the signal and the noise are uncorrelated.

2.2. $E\{\xi_3[n]\}$

$$E\{\xi_3[n]\} = \int_0^{T_s} \underbrace{E\{w(t+nT_s)w(t+(n+1)T_s)\}}_{R_w(T_s)}\, dt$$

The pulse repetition time in UWB systems typically ranges from hundred to thousand times the pulse (monocycle) width ($T_s \gg T_p \approx 1/B$). In this situation, the noise auto-correlation proposed in (2.B.3), is approximately zero for time late T_s second ($R_w(T_s) \approx 0$). This leads to $E\{\xi_3[n]\} = 0$.

So the mean of DT noise $E\{\breve{\omega}[n]\}$ is approximately zero.

3. The autocorrelation of Dirty Template Noise

$$E\{\breve{\omega}[k]\breve{\omega}[l]\} = E\{\xi_1[k]\xi_1[l]\} + E\{\xi_2[k]\xi_2[l]\} + E\{\xi_3[k]\xi_3[l]\} + 2E\{\xi_1[k]\xi_2[l]\} + 2E\{\xi_1[k]\xi_3[l]\} + 2E\{\xi_2[k]\xi_3[l]\}$$

3.1. $E\{\xi_1[k]\xi_1[l]\}$ *&* $E\{\xi_2[k]\xi_2[l]\}$

$$= E\left\{\left(\int_0^{T_s} f(t+kT_s,\tau)w(t+(k+1)T_s)\,dt\right)\cdot\left(\int_0^{T_s} f(t+lT_s,\tau)\times w(t+(l+1)T_s)\,dt\right)\right\}$$

$$= \int_0^{T_s}\int_0^{T_s} E\{f(t_1+kT_s,\tau)f(t_2+lT_s,\tau)\}\cdot E\{w(t_1+(k+1)T_s)\times w(t_2+(l+1)T_s)\}\,dt_2dt_1$$

Let's find the value of $E\{f(t_1+kT_s,\tau)f(t_2+lT_s,\tau)\}$ and substituting it in the last equation:

$$E\{f(t_1+kT_s,\tau)f(t_2+lT_s,\tau)\} = \mathcal{E}_s\sum_{m_1=0}^{1}\sum_{m_2=0}^{1}\underbrace{E\{s[k-n_s-m_1]s[l-n_s-m_2]\}}_{=0 \quad \forall\, k-m_1\neq l-m_2}\times \underbrace{p_R(t_1+m_1T_s-\tau)p_R(t_2+m_2T_s-\tau)}_{=0 \quad :\, m_1\neq m_2}$$

$$= \mathcal{E}_s\delta_{k,l}\sum_{m=0}^{1} p_R(t_1+mT_s-\tau)p_R(t_2+mT_s-\tau)$$

$$\Longrightarrow E\{f(t_1+kT_s,\tau)f(t_2+lT_s,\tau)\} = \mathcal{E}_s\delta_{k,l}p_R(t_1)p_R(t_2)$$

So $E\{\xi_1[k]\xi_1[l]\}$ becomes:

$$E\{\xi_1[k]\xi_1[l]\} = \mathcal{E}_s\delta_{k,l}\int_0^{T_s}\int_0^{T_s} p_R(t_1)p_R(t_2)R_w(t_1-t_2)\,dt_2\,dt_1$$

$$= \mathcal{E}_s\delta_{k,l}\int_0^{T_s}\int_0^{T_s} p_R(t_1)p_R(t_2)\,N_0Bsinc\big(B(t_1-t_2)\big)cos\big(2\pi f_0(t_1-t_2)\big)dt_2dt_1$$

$$= \mathcal{E}_s \delta_{k,l} \int_0^{T_s} \int_0^{T_s} p_R(t_1) p_R(t_2)\, N_0 \left(\int_{-B/2}^{B/2} e^{j2\pi f(t_1-t_2)}\, df \right) \left(\frac{e^{-j2\pi f_0(t_1-t_2)} + e^{j2\pi f_0(t_1-t_2)}}{2} \right) dt_2 dt_1$$

$$= \mathcal{E}_s \delta_{k,l} \int_0^{T_s} \int_0^{T_s} p_R(t_1) p_R(t_2) \frac{N_0}{2} \left(\int_{-B/2}^{B/2} e^{j2\pi (f-f_0)(t_1-t_2)}\, df + \int_{B/2}^{B/2} e^{j2\pi (f+f_0)(t_1-t_2)}\, df \right) dt_2 dt_1$$

$$= \mathcal{E}_s \delta_{k,l} \int_0^{T_s} \int_0^{T_s} p_R(t_1) p_R(t_2) \frac{N_0}{2} \left(\int_{-f_0-B/2}^{-f_0+B/2} e^{j2\pi f(t_1-t_2)}\, df + \int_{f_0-B/2}^{f_0+B/2} e^{j2\pi f(t_1-t_2)}\, df \right) dt_2 dt_1$$

The signal $p_R(t)$ has the duration T_s, so $\int_0^{T_s} p_R(t) e^{-j2\pi ft} dt = \int_{-\infty}^{\infty} p_R(t) e^{-j2\pi ft} dt = TF\{p_R(t)\}$, and the bandwidth of $p_R(t)$ is $1/T_p$. Recalling that the bandwidth of the receiver bandpass frontend $B \approx 1/T_p$. Hence, $E\{\xi_1[k]\xi_1[l]\}$ is developed as following:

$$E\{\xi_1[k]\xi_1[l]\} = \frac{N_0}{2} \mathcal{E}_s \delta_{k,l} \int_0^{T_s} p_R(t_1) \times \left(\int_{-f_0-B/2}^{-f_0+B/2} \underbrace{\int_0^{T_s} p_R(t_2) e^{j2\pi f(t_1-t_2)} dt_2}_{TF\{p_R(t)\} e^{j2\pi f t_1}}\, df + \int_{f_0-B/2}^{f_0+B/2} \underbrace{\int_0^{T_s} p_R(t_2) e^{j2\pi f(t_1-t_2)} dt_2}_{TF\{p_R(t)\} e^{j2\pi f t_1}}\, df \right) dt_1$$

$$= \frac{N_0}{2} \mathcal{E}_s \delta_{k,l} \int_0^{T_s} p_R(t) \left(\underbrace{\int_{-f_0-B/2}^{-f_0+B/2} TF[p_R(t)]\, e^{j2\pi ft} df + \int_{f_0-B/2}^{f_0+B/2} TF[p_R(t)]\, e^{j2\pi ft} df}_{TF^{-1}\{TF\{p_R(t)\}\} = p_R(t)} \right) dt$$

$$= \frac{N_0}{2} \mathcal{E}_s \delta_{k,l} \underbrace{\int_0^{T_s} p_R^2(t)\, dt}_{\mathcal{E}_{max}}$$

Hence:

$$E\{\xi_1[k]\xi_1[l]\} = E\{\xi_2[k]\xi_2[l]\} = \frac{N_0}{2} \underbrace{\mathcal{E}_s \mathcal{E}_{max}}_{\mathcal{E}_r} \delta_{k,l} \tag{2.B.11}$$

Where $\mathcal{E}_r$ is an unknown constant received symbol energy, depends on the transmitted channel.

3.2. $E\{\boldsymbol{\xi_3[k]\xi_3[l]}\}$

$$= E\left\{ \left(\int_0^{T_s} w(t_1 + kT_s)\, w(t_1 + (k+1)T_s) dt_1 \right) \times \left(\int_0^{T_s} w(t_2 + lT_s)\, w(t_2 + (l+1)T_s) dt_2 \right) \right\}$$

$$= \int_0^{T_s} \int_0^{T_s} E\{w(t_1 + kT_s) \cdot w(t_1 + (k+1)T_s) \cdot w(t_2 + lT_s) \times w(t_2 + (l+1)T_s)\}\, dt_2 dt_1$$

For finding $E\{\xi_3[k]\xi_3[l]\}$, we have employed the fact that the expectation of the product of zero mean, jointly Gaussian random variables $\{n(t_1), n(t_2), n(t_3), n(t_4)\}$ is given as follows:

$$E\{n(t_1)n(t_2), n(t_3)\, n(t_4)\} = R_w(t_1 - t_2)R_w(t_3 - t_4) + R_w(t_1 - t_3)R_w(t_2 - t_4) + R_w(t_1 - t_4)R_w(t_2 - t_3)$$

where $n(t_1) = w(t_1 + kT_s)$, $n(t_2) = w(t_1 + (k+1)T_s)$, $n(t_3) = w(t_2 + lT_s)$ and $n(t_4) = w(t_2 + (l+1)T_s)$. Thus, we can represent $E\{\xi_3[k]\xi_3[l]\}$ as following:

$$E\{\xi_3[k]\xi_3[l]\} = \int_0^{T_s}\int_0^{T_s}\Bigg\{\underbrace{R_w^2(T_s)}_{\approx 0} + \underbrace{R_w^2(t_1 - t_2 + (k-l)T_s)}_{R_w^2(t_1-t_2)\delta_{k,l}} + \underbrace{R_w(t_1 - t_2 + (k-l-1)T_s)R_w(t_1 - t_2 + (k-l+1)T_s)}_{\approx 0: one\ of\ these\ two\ correlations\ is\ \text{approximately zero}}\Bigg\} dt_2 dt_1$$

$$= \delta_{k,l}\int_0^{T_s}\int_0^{T_s} R_w^2(t_1 - t_2)dt_2 dt_1$$

$$= \delta_{k,l}\int_0^{T_s}\int_0^{T_s}\left[N_0 B sinc\big(B(t_1 - t_2)\big)cos\big(2\pi f_0(t_1 - t_2)\big)\right]^2 dt_2 dt_1$$

Since the symbol time $T_s \gg 1/B$ is large enough so that most of the energy of the pulse concentrates within T_s, we can express $E\{\xi_3[k]\xi_3[l]\}$ in frequency domain by applying Parseval's theorem $\int_{-\infty}^{\infty}|x(t)|^2 dt = \int_{-\infty}^{\infty}|X(f)|^2 df$ and using (2.B.3), yields:

$$E\{\xi_3[k]\xi_3[l]\} = \delta_{k,l}\int_0^{T_s}\int_{-B/2}^{B/2} 2(N_0/2)^2 df\, dt \quad = \delta_{k,l}\frac{{N_0}^2}{2}BT_s$$

Thus:

$$E\{\xi_3[k]\xi_3[l]\} = \frac{{N_0}^2}{2}BT_s\delta_{k,l} \tag{2.B.12}$$

3.3. $E\{\xi_1[k]\xi_2[l]\}$

$$= E\left\{\left(\int_0^{T_s} f(t + kT_s, \tau)w(t + (k+1)T_s)\, dt\right)\cdot\left(\int_0^{T_s} f(t + (l+1)T_s, \tau)\times w(t + lT_s)dt\right)\right\}$$

$$= \int_0^{T_s}\int_0^{T_s} E\{f(t_1 + kT_s, \tau)f(t_2 + (l+1)T_s, \tau)\}\underbrace{E\{w(t_1 + (k+1)T_s)w(t_2 + lT_s)\}}_{R_w(t_1-t_2+(k-l+1)T_s)} dt_2\, dt_1$$

The first product is given as:

$$E\{f(t_1 + kT_s, \tau)f(t_2 + (l+1)T_s, \tau)\} = \mathcal{E}_s\sum_{m_1=0}^{1}\sum_{m_2=0}^{1}\underbrace{E\{s[k - n_s - m_1]s[l + 1 - n_s - m_2]\}}_{=0 \quad \forall\, k-m_1\neq l+1-m_2}\times$$

$$\underbrace{E\{p_R(t_1+m_1T_s-\tau)p_R(t_2+m_2T_s-\tau)\}}_{=0 \quad : m_1\neq m_2}$$

$$= \mathcal{E}_s\delta_{k,l+1}\sum_{m=0}^{1} p_R(t_1+mT_s-\tau)p_R(t_2+mT_s-\tau)$$

$$= \mathcal{E}_s\delta_{k,l+1}p_R(t_1)p_R(t_2)$$

So $E\{\xi_1[k]\xi_2[l]\}$ is :

$$E\{\xi_1[k]\xi_2[l]\} = \mathcal{E}_s\delta_{k,l+1}\int_0^{T_s}\int_0^{T_s} p_R(t_1)p_R(t_2)\underbrace{R_w(t_1-t_2+2T_s)}_{\approx 0\,:\,t_1\,\&\,t_2\in[0,T_s)}dt_2\,dt_1 = 0$$

3.4. $E\{\xi_1[k]\xi_3[l]\}$ & $E\{\xi_2[k]\xi_3[l]\}$

$$= E\left\{\left(\int_0^{T_s} f(t_1+kT_s,\tau)w(t_1+(k+1)T_s)\,dt_1\right)\cdot\left(\int_0^{T_s} w(t_2+lT_s)\,w(t_2+(l+1)T_s)dt_2\right)\right\}$$

$$= \int_0^{T_s}\int_0^{T_s} E\{f(t_1+kT_s,\tau)\cdot w(t_1+(k+1)T_s)\cdot w(t_2+lT_s)\cdot w(t_2+(l+1)T_s)\}\;dt_2dt_1$$

$$E\{\xi_1[k]\xi_3[l]\} =$$
$$\int_0^{T_s}\int_0^{T_s}\underbrace{E\{f(t_1+kT_s,\tau)w(t_1+(k+1)T_s)\}}_{=0}\underbrace{E\{w(t_2+lT_s)w(t_2+(l+1)T_s)\}}_{R_w(T_s)\approx 0}dt_2dt_1 +$$
$$\int_0^{T_s}\int_0^{T_s}\underbrace{E\{f(t_1+kT_s,\tau)w(t_2+lT_s)\}}_{=0}\underbrace{E\{w(t_1+(k+1)T_s)w(t_2+(l+1)T_s)\}}_{R_w(t_1-t_2)\delta_{k,l}}dt_2dt_1 +$$
$$\int_0^{T_s}\int_0^{T_s}\underbrace{E\{f(t_1+kT_s,\tau)w(t_2+(l+1)T_s)\}}_{=0}\underbrace{E\{w(t_1+(k+1)T_s)w(t_2+lT_s)\}}_{R_w(t_1-t_2)\delta_{k+1,l}}dt_2dt_1 = 0$$

In the UWB system, using the dirty template approach, the sampled noise $w(t+kT_s)$ is only weakly correlated with its adjacent $w(t+(k+1)T_s)$ since $(B \gg 1/T_s)$. We have also found that the three noise terms $\{\xi_1[n],\xi_2[n],\xi_3[n]\}$ could be treated approximately as mutually uncorrelated zero mean noises. Furthermore, $\xi_3[n]$ contains the product of two uncorrelated Gaussian noise components, the Central Limit Theorem (CLT) asserts that it will also be approximately Gaussian as a consequence of the integration. In other word, $\xi_3[n]$ is also approximately white Gaussian noise $\left(\xi_3[n]\sim\mathcal{N}(0,\frac{{N_0}^2}{2}BT_s)\right)$.

Finally, $\breve{\omega}[n]$ is zero-mean Gaussian ($\breve{\omega}[n]\sim\mathcal{N}(0,\frac{{N_0}^2}{2}BT_s + N_0\mathcal{E}_r)$) with correlation function $\left(\frac{{N_0}^2}{2}BT_s + N_0\mathcal{E}_r\right)\delta_{k,l}$.

Appendix 3.A

Matlab Program for Monte Carlo Computer Simulation

```
% This program is a Monte Carlo computer simulation
% This program was used to generate figure 3.5
%
% Set up the value of iteration number N, SNR
% and data aided symbol number M1
%
N=1000000;
SNR_dB=-5;
M1=8;
%
% Generate the UWB Gaussian Pulse 3rd Order
% fs is the sampling frequency
% Tp=1ns UWB pulse duration
% Nf number of UWB pulses (frames) in the waveform
% Tf=100ns the frame duration
% the normalized energy of the transmitted
%
fs=10e9;
Tp=1e-9;
Nf=24;
Tf=100;
Es=1;
t=-0.5*Tp:1/fs:0.5*Tp;
Y=(1-(4*pi.*(t.^2))/(Tp /2.5)^2) .* exp(-2*pi.*(t.^2)/(Tp /2.5)^2);
Y=sqrt(Es)*Y/sqrt(Nf*sum(Y.*Y));
%
% Compute UWB pulse variance var{Y(t)},equivalent noise bandwidth B,
% waveform duration Ts, and the capture energy Er
%
Var_Y=var(Y);
B=1/Tp;
Ts=Nf*Tf*Tp;
Er=1;
%
% Compute noise variance N0/2
```

```
%
SNR= 10^(SNR_dB/10);
Var_Noise=Var_Y/SNR;
N0=2*Var_Noise;
%
% Matched Filter Noise
%
var0= ((N0^2)/2)*B*Ts;
var1= ((N0^2)/2)*B*Ts + N0*Er;
%
% Initialize the test statistic array with N dimension, then
% compute Rxx and T under each hypothesis {H0,H1}
%
T0=zeros(N,1);
T1=zeros(N,1);
for i=1:N
    Rxx0=   sqrt(var0)*randn(M1,1);
    Rxx1=Er+sqrt(var1)*randn(M1,1);
    T0(i)=sum( Rxx0.^2+(2*var0/N0)*Rxx0 );
    T1(i)=sum( Rxx1.^2+(2*var0/N0)*Rxx1 );
end
%
% Set number of values of the threshold Xi, and set up Xi array
%
Num_Xi=1000;
Xi_min=min(min(T0),min(T1));
Xi_max=max(max(T0),max(T1));
delta=(Xi_max-Xi_min)/Num_Xi;
Xi=[Xi_min:delta:Xi_max]';
%
% Find Detection and False alarm probabilities
%
Pr_FA=zeros(length(Xi),1);
Pr_D =zeros(length(Xi),1);
for i=1:length(Xi)
    Pr_FA(i)=length( find(T0>Xi(i)) )/N;
    Pr_D(i) =length( find(T1>Xi(i)) )/N;
end
%
% Plot the results
%
figure

subplot(2,1,1);
plot(Xi,Pr_FA,'-')
xlabel('Xi')
ylabel('Pr(T0>Xi)')
grid

subplot(2,1,2);
plot(Xi,Pr_D,'--')
xlabel('Xi')
ylabel('Pr(T1>Xi)')
grid
```

Appendix 4.A

First Form of Deriving ML Estimator for DA Mode

Considering the training sequence samples mentioned in (4.20). We assert clearly that ($A_n = -B_n$) and the equation (4.29) becomes:

$$\sum_{n=0}^{M_2-1}\{2A_nR_{x,x}[n,\tau_0] - 4A_n^2\gamma(\tau_0) + 2A_n^2\mathcal{E}_{max}\} = 0 \quad (4.A.1)$$

From (4.3) we have $A_n^2 = \mathcal{E}_s^2$, substituting it in (4.A.1):

$$4\mathcal{E}_s^2M_2\gamma(\tau_0) = 2\mathcal{E}_s^2M_2\mathcal{E}_{max} + 2\sum_{n=0}^{M_2-1}A_nR_{x,x}[n,\tau_0]$$

Hence:

$$\gamma(\tau_0) = \frac{\mathcal{E}_{max}}{2} + \frac{1}{2\mathcal{E}_s^2M_2}\sum_{n=0}^{M_2-1}A_nR_{x,x}[n,\tau_0] \quad (4.A.2)$$

Next, we wish to determine the average term $\frac{1}{M_2}\sum_{n=0}^{M_2-1}A_nR_{x,x}[n,\tau_0]$ and to substitute it into (4.A.2). From (4.24) we have:

$$R_{x,x}[n,\tau_0] = \breve{\omega}[n] + 2A_n\gamma(\tau_0) - A_n\mathcal{E}_{max} \quad (4.A.3)$$

Hence:

$$\frac{1}{M_2}\sum_{n=0}^{M_2-1}A_nR_{x,x}[n,\tau_0] = \frac{1}{M_2}\sum_{n=0}^{M_2-1}A_n\big(\breve{\omega}[n] + A_n(2\gamma(\tau_0) - \mathcal{E}_{max})\big)$$

$$= \underbrace{\frac{1}{M_2}\sum_{n=0}^{M_2-1}A_n\breve{\omega}[n]}_{\approx 0} + \frac{1}{M_2}\sum_{n=0}^{M_2-1}A_n^2(2\gamma(\tau_0) - \mathcal{E}_{max})$$

Thus:

$$\frac{1}{M_2}\sum_{n=0}^{M_2-1} A_n R_{x,x}[n,\tau_0] = \mathcal{E}_s^2(2\gamma(\tau_0) - \mathcal{E}_{max}) \tag{4.A.4}$$

On the other hand, in DA mode, by considering $A_{n+1} = B_n = -A_n$ (see Table 2.2), so $R_{x,x}[n,\tau_0]$ and $R_{x,x}[n+1,\tau_0]$ could be represented as:

$$R_{x,x}[n,\tau_0] = \breve{\omega}[n] + A_n(2\gamma(\tau_0) - \mathcal{E}_{max}) \tag{4.A.5}$$

$$R_{x,x}[n+1,\tau_0] = \breve{\omega}[n+1] - A_n(2\gamma(\tau_0) - \mathcal{E}_{max}) \tag{4.A.6}$$

Hence:

$$\frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}[n,\tau_0]\cdot R_{x,x}[n+1,\tau_0] = \underbrace{\frac{1}{M_2}\sum_{n=0}^{M_2-1}\breve{\omega}[n]\,\breve{\omega}[n+1] + \frac{1}{M_2}\sum_{n=0}^{M_2-1}(\breve{\omega}[n+1]-\breve{\omega}[n])\,A_n(2\gamma(\tau_0)-\mathcal{E}_{max})}_{\approx 0} - \frac{1}{M_2}\sum_{n=0}^{M_2-1} A_n^2(2\gamma(\tau_0)-\mathcal{E}_{max})^2$$

$$\Longrightarrow \frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}[n,\tau_0]\cdot R_{x,x}[n+1,\tau_0] = -\mathcal{E}_s^2(2\gamma(\tau_0)-\mathcal{E}_{max})^2 \tag{4.A.7}$$

Comparing between (4.A.4) and (4.A.7) yields:

$$\frac{1}{M_2}\sum_{n=0}^{M_2-1} A_n R_{x,x}[n,\tau_0] = \frac{-1}{M_2(2\gamma(\tau_0)-\mathcal{E}_{max})}\sum_{n=0}^{M_2-1} R_{x,x}[n,\tau_0]\cdot R_{x,x}[n+1,\tau_0] \tag{4.A.8}$$

Substituting (4.A.8) in (4.A.2), we get:

$$\gamma(\tau_0) = \frac{\mathcal{E}_{max}}{2} - \frac{1}{2\mathcal{E}_s^2 M_2(2\gamma(\tau_0)-\mathcal{E}_{max})}\sum_{n=0}^{M_2-1} R_{x,x}[n,\tau_0]\cdot R_{x,x}[n+1,\tau_0]$$

Then:

$$\left(\gamma(\tau_0) - \frac{\mathcal{E}_{max}}{2}\right)^2 = -\frac{1}{4\mathcal{E}_s^2 M_2}\sum_{n=0}^{M_2-1} R_{x,x}[n,\tau_0]\cdot R_{x,x}[n+1,\tau_0]$$

Hence $\gamma(\tau_0)$ becomes:

$$\gamma(\tau_0) = \frac{\mathcal{E}_{max}}{2} \pm \frac{1}{2\mathcal{E}_s}\sqrt{-\frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}[n,\tau_0]\cdot R_{x,x}[n+1,\tau_0]} \tag{4.A.9}$$

Let's study the estimator for the case of the root in (4.A.9) multiplied by minus (for achieving the synchronization, we have two choices. The first is to search the beginning of the symbol at

$\tau_0 = 0$ & $\gamma(\tau_0) = 0$, so we take the root in (4.A.9) multiplied by minus sign. The second choice is to search the ending of the symbol at $\tau_0 = T_s$ & $\gamma(\tau_0) = \mathcal{E}_{max}$, by taking the root in (4.A.9) multiplied by positive sign). From (4.24) we have $\gamma(\tau_0) = \int_{T_s-\tau_0}^{T_s} p_R^2(t)\,dt$, and by using Trapezoidal interpolating rule, the integral $\int_{T_s-\tau_0}^{T_s} p_R^2(t)\,dt$ could be approximately represented as $\frac{\tau_0}{2}\left(p_R^2(T_s) + p_R^2(T_s-\tau_0)\right)$.

$$\gamma(\tau_0) = \int_{T_s-\tau_0}^{T_s} p_R^2(t)\,dt = \frac{\tau_0}{2}\left(p_R^2(T_s) + p_R^2(T_s-\tau_0)\right); \quad \gamma(\tau_0) \in [0, \mathcal{E}_{max}) \tag{4.A.10}$$

Substituting (4.A.10) in (4.A.9), $\hat{\tau}_0$ is given by:

$$\hat{\tau}_0(\text{DA}) = \frac{\mathcal{E}_{max}}{\left(p_R^2(T_s)+p_R^2(T_s-\tau_0)\right)}\left(1 - \frac{1}{\mathcal{E}_s\mathcal{E}_{max}}\sqrt{-\frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}[n,\tau_0]\cdot R_{x,x}[n+1,\tau_0]}\right) \tag{4.A.11}$$

Note: From (4.41), we assert that when timing synchronization is perfectly achieved ($\tau_0 = 0$) the interfering fraction $\gamma(\tau_0)$ will vanish. So we can estimate $\hat{\tau}_0$ by searching the minimum of $\gamma(\tau_0)$ without the need to apply Trapezoidal interpolating rule.

In contrast, without mistiming $\tau_0 = 0$, it leads to cancel this interference and make $\gamma(\tau_0)$ equal to zero.

Appendix 4.B
Second Form of Deriving ML Estimator for DA Mode

Appendix 4.A has derived the maximum likelihood estimator (MLE) for DA mode by substituting the term $\sum_{n=0}^{M_2-1} A_n R_{x,x}[n,\tau_0]$ in (4.A.2) for $\sum_{n=0}^{M_2-1} R_{x,x}[n,\tau_0]\cdot R_{x,x}[n+1,\tau_0]$. Here, we will try to replace $\sum_{n=0}^{M_2-1} A_n R_{x,x}[n,\tau_0]$ by $\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0]$. For DA mode the correlation output samples proposed in (4.24) becomes:

$$R_{x,x}[n,\tau_0] = \breve{\omega}[n] + 2A_n\gamma(\tau_0) - A_n\mathcal{E}_{max} \tag{4.B.1}$$

Thus:

$$\frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0] = \underbrace{\frac{1}{M_2}\sum_{n=0}^{M_2-1}\breve{\omega}^2[n]}_{\sigma_w^2} + \underbrace{\frac{1}{M_2}\sum_{n=0}^{M_2-1} 2\breve{\omega}[n]A_n(2\gamma(\tau_0)-\mathcal{E}_{max})}_{\approx 0} + \frac{1}{M_2}\sum_{n=0}^{M_2-1} A_n^2(2\gamma(\tau_0)-\mathcal{E}_{max})^2$$

$$\Longrightarrow \frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0] = \sigma_w^2 + \mathcal{E}_s^2(2\gamma(\tau_0)-\mathcal{E}_{max})^2 \tag{4.B.2}$$

Comparing between (4.A.4) and (4.B.2) yields:

$$\frac{1}{M_2}\sum_{n=0}^{M_2-1} A_n R_{x,x}[n,\tau_0] = \frac{1}{(2\gamma(\tau_0)-\mathcal{E}_{max})}\left(\frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0] - \sigma_w^2\right) \tag{4.B.3}$$

Substituting (4.B.3) in (4.A.2), we get:

$$\gamma(\tau_0) = \frac{\mathcal{E}_{max}}{2} - \frac{1}{2\mathcal{E}_s^2(2\gamma(\tau_0)-\mathcal{E}_{max})}\left(\frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0] - \sigma_w^2\right)$$

Then:

$$\left(\gamma(\tau_0) - \frac{\mathcal{E}_{max}}{2}\right)^2 = \frac{1}{4\mathcal{E}_s^2}\left(\frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0] - \sigma_w^2\right)$$

Hence $\gamma(\tau_0)$ becomes:

$$\gamma(\tau_0) = \frac{\mathcal{E}_{max}}{2} \pm \frac{1}{2\mathcal{E}_s}\sqrt{\frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0] - \sigma_w^2} \tag{4.B.4}$$

Let's study the estimator for the case of the root in (4.B.4) multiplied by minus. Based on the Trapezoidal interpolating theorem mentioned in (4.A.10), the timing offset estimator is given as follows:

$$\hat{\tau}_0(\text{DA}) = \frac{\mathcal{E}_{max}}{\left(p_R^2(T_s)+p_R^2(T_s-\tau_0)\right)}\left(1 - \frac{1}{\mathcal{E}_s\mathcal{E}_{max}}\sqrt{\frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0] - \sigma_w^2}\right) \tag{4.B.5}$$

But the variables $\mathcal{E}_{max}$ and $p_R^2(T_s - \tau_0)$ are unknown, intuitively the $\hat{\tau}_0$ estimation occurs at the minimum point of the following expression for different values of $\tau = m\Delta_t \in [0, T_s)$:

$$\hat{\tau}_0(\text{DA}) = \underset{m}{arg\,min}\ \frac{\mathcal{E}_{max}}{\left(p_R^2(T_s)+p_R^2(T_s-\tau_0)\right)}\left(1 - \frac{1}{\mathcal{E}_s\mathcal{E}_{max}}\sqrt{\frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0 - m\Delta_t] - \sigma_w^2}\right)$$

Considering the variables σ_w^2, $p_R^2(T_s - \tau_0)$, ε_s and ε_{max} independent of the change of τ, the estimator can be developed as follows:

$$\hat{\tau}_0(\text{DA}) = \underset{m}{arg\,max}\ \frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0 - m\Delta_t] \tag{4.B.6}$$

Appendix 4.C

Deriving ML Estimator for NDA Mode

From (4.29) we have:

$$\frac{1}{M_2}\sum_{n=0}^{M_2-1}\left\{(A_n - B_n)R_{x,x}[n,\tau_0] - (A_n - B_n)^2\gamma(\tau_0) - (A_n - B_n)B_n\mathcal{E}_{max}\right\} = 0$$

$$\frac{1}{M_2}\sum_{n=0}^{M_2-1}(A_n - B_n)R_{x,x}[n,\tau_0] = \underbrace{\frac{1}{M_2}\sum_{n=0}^{M_2-1}(A_n - B_n)^2\gamma(\tau_0)}_{\approx 2\mathcal{E}_s^2\gamma(\tau_0)} + \underbrace{\frac{1}{M_2}\sum_{n=0}^{M_2-1}(A_n - B_n)B_n\mathcal{E}_{max}}_{\approx -\mathcal{E}_s^2\mathcal{E}_{max}}$$

$$\frac{1}{M_2}\sum_{n=0}^{M_2-1}(A_n - B_n)R_{x,x}[n,\tau_0] = \mathcal{E}_s^2(2\gamma(\tau_0) - \mathcal{E}_{max}) \tag{4.C.1}$$

Hence:

$$\gamma(\tau_0) = \frac{\mathcal{E}_{max}}{2} + \frac{1}{2\mathcal{E}_s^2 M_2}\sum_{n=0}^{M_2-1}(A_n - B_n)R_{x,x}[n,\tau_0] \tag{4.C.2}$$

Next, we wish to determine the average term $\frac{1}{M_2}\sum_{n=0}^{M_2-1}(A_n - B_n)R_{x,x}[n,\tau_0]$ and to substitute it into (4.C.2). From (4.24) we have:

$$R_{x,x}[n,\tau_0] = \breve{\omega}[n] + (A_n - B_n)\gamma(\tau_0) + B_n\mathcal{E}_{max} \tag{4.C.3}$$

Thus:

$$\frac{1}{M_2}\sum_{n=0}^{M_2-1}R_{x,x}^2[n,\tau_0] = \underbrace{\frac{1}{M_2}\sum_{n=0}^{M_2-1}\breve{\omega}^2[n]}_{\sigma_w^2} + \underbrace{\frac{1}{M_2}\sum_{n=0}^{M_2-1}2\breve{\omega}[n]\left((A_n - B_n)\gamma(\tau_0) + B_n\mathcal{E}_{max}\right)}_{\approx 0} + \frac{1}{M_2}\sum_{n=0}^{M_2-1}\{(A_n - B_n)^2\gamma^2(\tau_0) + B_n^2\mathcal{E}_{max}^2 + 2(A_n - B_n)B_n\gamma(\tau_0)\mathcal{E}_{max}\}$$

$$\frac{1}{M_2}\sum_{n=0}^{M_2-1}R_{x,x}^2[n,\tau_0] = \sigma_w^2 + 2\mathcal{E}_s^2\gamma^2(\tau_0) + \mathcal{E}_s^2\mathcal{E}_{max}^2 - 2\mathcal{E}_s^2\gamma(\tau_0)\mathcal{E}_{max}$$

$$\frac{1}{M_2}\sum_{n=0}^{M_2-1}R_{x,x}^2[n,\tau_0] = \sigma_w^2 + \frac{\mathcal{E}_s^2}{2}(4\gamma^2(\tau_0) - 4\gamma(\tau_0)\mathcal{E}_{max} + \mathcal{E}_{max}^2) + \frac{\mathcal{E}_s^2}{2}\mathcal{E}_{max}^2$$

$$\Rightarrow \frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0] = \sigma_w^2 + \frac{1}{2}\mathcal{E}_s^2\mathcal{E}_{max}^2 + \frac{\mathcal{E}_s^2}{2}(2\gamma(\tau_0) - \mathcal{E}_{max})^2 \quad (4.C.4)$$

Comparing between (4.C.4) and (4.C.1) yields:

$$\frac{1}{M_2}\sum_{n=0}^{M_2-1}(A_n - B_n)R_{x,x}[n,\tau_0] = \frac{2}{(2\gamma(\tau_0)-\mathcal{E}_{max})}\left(\frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0] - \sigma_w^2 - \frac{1}{2}\mathcal{E}_s^2\mathcal{E}_{max}^2\right)$$

Substituting the last statement in (4.C.2), we get:

$$\gamma(\tau_0) = \frac{\mathcal{E}_{max}}{2} + \frac{1}{\mathcal{E}_s^2(2\gamma(\tau_0)-\mathcal{E}_{max})}\left(\frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0] - \sigma_w^2 - \frac{1}{2}\mathcal{E}_s^2\mathcal{E}_{max}^2\right)$$

Then:

$$\left(\gamma(\tau_0) - \frac{\mathcal{E}_{max}}{2}\right)^2 = \frac{1}{2\mathcal{E}_s^2}\left(\frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0] - \sigma_w^2 - \frac{1}{2}\mathcal{E}_s^2\mathcal{E}_{max}^2\right)$$

Hence $\gamma(\tau_0)$ becomes:

$$\gamma(\tau_0) = \frac{\mathcal{E}_{max}}{2} \pm \frac{1}{2\mathcal{E}_s}\sqrt{\frac{2}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0] - 2\sigma_w^2 - \mathcal{E}_s^2\mathcal{E}_{max}^2} \quad (4.C.5)$$

Let's study the estimator for the case of the root in (4.C.5) multiplied by minus. Based on the Trapezoidal interpolating theorem mentioned in (4.A.10), the timing offset estimator is given as follows:

$$\hat{\tau}_0 = \frac{\mathcal{E}_{max}}{\left(p_R^2(T_s)+p_R^2(T_s-\tau_0)\right)}\left(1 - \frac{1}{\mathcal{E}_s\mathcal{E}_{max}}\sqrt{\frac{2}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0] - 2\sigma_w^2 - \mathcal{E}_s^2\mathcal{E}_{max}^2}\right) \quad (4.C.6)$$

But the variables $\mathcal{E}_{max}$ and $p_R^2(T_s - \tau_0)$ are unknown, intuitively the $\hat{\tau}_0$ estimation occurs at the minimum point of the following expression for different values of $\tau = m\Delta_t \in [0, T_s)$:

$$\hat{\tau}_0(\text{NDA}) = \underset{m}{arg\,min}\ \frac{\mathcal{E}_{max}}{\left(p_R^2(T_s)+p_R^2(T_s-\tau_0)\right)} \times \left(1 - \frac{1}{\mathcal{E}_s\mathcal{E}_{max}}\sqrt{\frac{2}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0 - m\Delta_t] - 2\sigma_w^2 - \mathcal{E}_s^2\mathcal{E}_{max}^2}\right)$$

Considering the variables σ_w^2, $p_R^2(T_s - \tau_0)$, ε_s and ε_{max} independent of the change of τ, the estimator can be developed as follows:

$$\hat{\tau}_0(NDA) = \underset{m}{arg\,max}\ \frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0 - m\Delta_t] \quad (4.C.7)$$

Appendix 4.D

Deriving Timing Offset Estimator via Mean-Square Sense

In this section, we will seek to estimate the timing offset τ_0 using the mean-square sense (mss) concept. From (4.24) we have:

$$R_{x,x}[n,\tau_0] = \breve{\omega}[n] + (A_n - B_n)\gamma(\tau_0) + B_n\mathcal{E}_{max} \tag{4.D.1}$$

where $\mathcal{E}_{max} = \int_0^{T_s} p_R^2(t)\,dt$ indicates the maximum useful energy of the received waveform $r(t)$ which is captured at the correlator output. $\gamma(\tau_0) = \int_{T_s-\tau_0}^{T_s} p_R^2(t)\,dt$ represents the fraction of total energy $\mathcal{E}_{max}$ captured by the interfering symbol s[n − 1], for more details go back to (4.3) and Figure 4.1. Hence, in the case of mistiming ($\tau_0 \neq 0 \ \leq T_s$), the interfering effect of the symbol s[n − 1] presents in $R_{x,x}[n,\tau_0]$. In contrast, without mistiming $\tau_0 = 0$, it leads to cancel this interference and make $\gamma(\tau_0)$ equal to zero. Now, we wish to demonstrate how to estimate the timing offset based on the mean-square sense for DA and NDA modes.

1. DA timing offset estimator via mss:

Let's start with DA mode where $A_n = -B_n$. $R_{x,x}[n,\tau_0]$ is given as:

$$R_{x,x}[n,\tau_0] = \breve{\omega}[n] + B_n\mathcal{E}_{max} - 2B_n\gamma(\tau_0) \tag{4.D.2}$$

Thus:

$$\frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0] = \underbrace{\frac{1}{M_2}\sum_{n=0}^{M_2-1} \breve{\omega}^2[n]}_{\sigma_w^2} + \underbrace{\frac{2}{M_2}\sum_{n=0}^{M_2-1} \breve{\omega}[n]\left(B_n\mathcal{E}_{max} - 2B_n\gamma(\tau_0)\right)}_{0} + \underbrace{\frac{1}{M_2}\sum_{n=0}^{M_2-1} B_n^2\mathcal{E}_{max}^2}_{\mathcal{E}_s^2\mathcal{E}_{max}^2} + \underbrace{\frac{4}{M_2}\sum_{n=0}^{M_2-1} B_n^2\gamma^2(\tau_0)}_{4\mathcal{E}_s^2\gamma^2(\tau_0)} - \underbrace{\frac{4}{M_2}\sum_{n=0}^{M_2-1} B_n^2\mathcal{E}_{max}\gamma(\tau_0)}_{4\mathcal{E}_s^2\mathcal{E}_{max}\gamma(\tau_0)}$$

Assuming $M_2 \to +\infty$, then $\frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0]$ becomes:

$$\frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0] = \sigma_w^2 + 4\mathcal{E}_s^2\left(\frac{\mathcal{E}_{max}^2}{4} - \mathcal{E}_{max}\gamma(\tau_0) + \gamma^2(\tau_0)\right)$$

$$\frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0] = \sigma_w^2 + 4\mathcal{E}_s^2\left(\frac{\mathcal{E}_{max}}{2} - \gamma(\tau_0)\right)^2$$

Then:

$$\gamma(\tau_0) = \frac{\mathcal{E}_{max}}{2} \pm \frac{1}{2\mathcal{E}_s}\sqrt{\frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0] - \sigma_w^2} \tag{4.D.3}$$

Comparing between (4.D.3) and (4.B.4), we notice that we have the same results either by using ML approach or mss approach. The timing offset estimator thus is deduced as follows:

$$\hat{\tau}_0(\text{DA}) = \underset{m}{arg\,max}\ \frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0 - m\Delta_t] \tag{4.D.4}$$

2. NDA timing offset estimator via mss:

$$\frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0] = \underbrace{\frac{1}{M_2}\sum_{n=0}^{M_2-1} \breve{\omega}^2[n]}_{\sigma_w^2} + \underbrace{\frac{2}{M_2}\sum_{n=0}^{M_2-1} \breve{\omega}[n]\left((A_n - B_n)\gamma(\tau_0) + B_n\mathcal{E}_{max}\right)}_{0} + \underbrace{\frac{1}{M_2}\sum_{n=0}^{M_2-1}(A_n - B_n)^2\gamma^2(\tau_0)}_{2\mathcal{E}_s^2\gamma^2(\tau_0)} + \underbrace{\frac{1}{M_2}\sum_{n=0}^{M_2-1} B_n^2\mathcal{E}_{max}^2}_{\mathcal{E}_s^2\mathcal{E}_{max}^2} + \underbrace{\frac{2}{M_2}\sum_{n=0}^{M_2-1}(A_n - B_n)B_n\mathcal{E}_{max}\gamma(\tau_0)}_{-2\mathcal{E}_s^2\mathcal{E}_{max}\gamma(\tau_0)}$$

Assuming $M_2 \to +\infty$, then $\frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0]$ becomes:

$$\frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0] = \sigma_w^2 + 2\mathcal{E}_s^2\left(\gamma^2(\tau_0) - \mathcal{E}_{max}\gamma(\tau_0) + \frac{\mathcal{E}_{max}^2}{2}\right)$$

Then:

$$\gamma^2(\tau_0) - \mathcal{E}_{max}\gamma(\tau_0) + \frac{\mathcal{E}_{max}^2}{2} + \frac{\sigma_w^2}{2\mathcal{E}_s^2} - \frac{1}{2\mathcal{E}_s^2 M_2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0] = 0$$

Thus: $\Delta = \mathcal{E}_{max}^2 - 4\mathcal{E}_{max}^2\left(\frac{1}{2} + \frac{\sigma_w^2}{2\mathcal{E}_s^2\mathcal{E}_{max}^2} - \frac{1}{2M_2\mathcal{E}_s^2\mathcal{E}_{max}^2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0]\right)$

$$\gamma(\tau_0) = \frac{\mathcal{E}_{max} \pm \sqrt{\Delta}}{2} = \frac{\mathcal{E}_{max}}{2} \pm \frac{\mathcal{E}_{max}}{2}\sqrt{1 - 2\left(1 + \frac{\sigma_w^2}{\mathcal{E}_s^2\mathcal{E}_{max}^2} - \frac{1}{M_2\mathcal{E}_s^2\mathcal{E}_{max}^2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0]\right)}$$

$$\Rightarrow \gamma(\tau_0) = \frac{\mathcal{E}_{max}}{2} \pm \frac{1}{2\mathcal{E}_s}\sqrt{\frac{2}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0] - 2\sigma_w^2 - \mathcal{E}_s^2\mathcal{E}_{max}^2} \quad (4.D.5)$$

Comparing between (4.D.5) and (4.C.5), we notice that we have the same results either by using ML approach or mss approach. The timing offset estimator thus is deduced as follows:

$$\hat{\tau}_0(\text{DA}) = \underset{m}{arg\,max}\ \frac{1}{M_2}\sum_{n=0}^{M_2-1} R_{x,x}^2[n,\tau_0 - m\Delta_t] \quad (4.D.6)$$

Appendix 5.A

Noise at the Output of ELG Correlators

In this section, we derive the Dirty Template noise $n_{e,l}[\epsilon, n]$ model and calculate its statistical properties. From (5.7), the sampled noise $n_{e,l}[\epsilon, n]$ is composed of three terms, as following:

$$n_{e,l}[\epsilon, n] = \xi_1 + \xi_2 + \xi_3$$

where: $$n_{e,l}[\epsilon, n]: \begin{cases} \xi_1 = \int_0^{T_s} \sqrt{\mathcal{E}_s} p_R(t - \epsilon \pm \Delta) w_n(t)\, dt \\ \xi_2 = \int_0^{T_s} \sqrt{\mathcal{E}_s} s[n] p_R(t) \overline{w}(t)\, dt \\ \xi_3 = \int_0^{T_s} w_n(t) \overline{w}(t)\, dt \end{cases}$$

Clearly, the statistical properties of the DT noise $n_{e,l}[\epsilon, n]$ depends on information-bearing symbols $\{s[n]\}$. Therefore, to simplify the calculation of the mean and the autocorrelation of $n_{e,l}[\epsilon, n]$, we assume that the symbols $\{s[n]\}$ are independent identically distributed (i.i.d.) PAM (binary pulse amplitude modulation) symbols with zero-mean and unit-energy $E\{s[k]s[l]\} = \delta_{k,l}$ (see Table 2.2).

1. The mean of Dirty Template Noise

$$E\{n_{e,l}[\epsilon, n]\} = E\{\xi_1\} + E\{\xi_2\} + E\{\xi_3\}$$

1.1. $E\{\xi_1\}$

$$E\{\xi_1\} = \int_0^{T_s} \sqrt{\mathcal{E}_s} p_R(t - \epsilon \pm \Delta) \underbrace{E\{w_n(t)\}}_{=0} dt = 0$$

1.2. $E\{\xi_2\}$

$$E\{\xi_2\} = \int_0^{T_s} \sqrt{\mathcal{E}_s}\, \underbrace{E\{s[n]\}}_{=0}\, p_R(t)\, \underbrace{E\{\overline{w}(t)\}}_{=0}\, dt = 0$$

The mean of the first and the second terms ($E\{\xi_2\} = E\{\xi_1\} = 0$) is zero, because the signal and the noise are uncorrelated.

1.3. $E\{\xi_3\}$

$$E\{\xi_3\} = \int_0^{T_s} E\{w_n(t)\overline{w}(t)\}\, dt$$

$$E\{\xi_3\} = \int_0^{T_s} E\left\{w_n(t)\, \frac{1}{M}\sum_{k=1}^{M} w_k(t)\right\} dt$$

In data-aided mode, M training sequence, which is used to extract the reference signal, is sent prior to the data to be conveyed. So the order of data segment is always bigger than M ($n > M$). Hence:

$$E\{\xi_3\} = \int_0^{T_s} \frac{1}{M}\sum_{k=1}^{M} \underbrace{E\{w_n(t)w_k(t)\}}_{R_w(mT_s):\, m=1,2,\ldots}\, dt.$$

Since the time difference between the noise components is mT_s (equal or bigger than T_s), and the symbol time T_s in UWB systems typically ranges from hundred to thousand times the pulse (monocycle) width ($T_s \gg T_p \approx 1/B$), it follows that the noise autocorrelation, mentioned in (equation 2.A.2 in Appendix 2.A), is approximately zero ($R_w(mT_s) \approx 0$). This leads to $E\{w_n(t)\overline{w}(t)\} = 0$ & $E\{\xi_3\} = 0$.

Hence, the mean of DT noise $E\{n_{e,l}[\epsilon, n]\}$ is approximately zero.

2. The autocorrelation of Dirty Template Noise

$$E\{n_{e,l}[\epsilon,k]n_{e,l}[\epsilon,l]\} = E\{\xi_1[k]\xi_1[l]\} + E\{\xi_2[k]\xi_2[l]\} + E\{\xi_3[k]\xi_3[l]\} + 2E\{\xi_1[k]\xi_2[l]\} + 2E\{\xi_1[k]\xi_3[l]\} + 2E\{\xi_2[k]\xi_3[l]\} \quad (5.A.1)$$

2.1. $E\{\xi_1[k]\xi_1[l]\}$

$$= E\left\{\left(\int_0^{T_s} \sqrt{\mathcal{E}_s} p_R(t_1 - \epsilon \pm \Delta) w_k(t_1)\, dt_1\right) \cdot \left(\int_0^{T_s} \sqrt{\mathcal{E}_s} p_R(t_2 - \epsilon \pm \Delta) w_l(t_2)\, dt_2\right)\right\}$$

$$= \int_0^{T_s} \int_0^{T_s} \mathcal{E}_s p_R(t_1 - \epsilon \pm \Delta) p_R(t_2 - \epsilon \pm \Delta) \cdot \underbrace{E\{w_k(t_1) w_l(t_2)\}}_{R_w(t_1 - t_2)\delta_{k,l}} \, dt_2 dt_1$$

To simplify, $p_R(t - \epsilon \pm \Delta)$ is replaced by $\acute{p}_R(t)$. So $E\{\xi_1[k]\xi_1[l]\}$ becomes:

$$E\{\xi_1[k]\xi_1[l]\} = \delta_{k,l}\mathcal{E}_s \int_0^{T_s} \int_0^{T_s} \acute{p}_R(t_1)\acute{p}_R(t_2) R_w(t_1 - t_2)\, dt_2\, dt_1$$

$$= \delta_{k,l} \int_0^{T_s} \int_0^{T_s} \acute{p}_R(t_1)\acute{p}_R(t_2)\, N_0 B sinc\big(B(t_1 - t_2)\big) cos\big(2\pi f_0 (t_1 - t_2)\big) dt_2 dt_1$$

$$= \delta_{k,l}\mathcal{E}_s \int_0^{T_s} \int_0^{T_s} \acute{p}_R(t_1)\acute{p}_R(t_2)\, N_0 \left(\int_{-B/2}^{B/2} e^{j2\pi f(t_1 - t_2)}\, df \right) \cdot \left(\frac{e^{-j2\pi f_0 (t_1 - t_2)} + e^{j2\pi f_0 (t_1 - t_2)}}{2} \right) dt_2 dt_1$$

$$= \delta_{k,l}\mathcal{E}_s \int_0^{T_s} \int_0^{T_s} \acute{p}_R(t_1)\acute{p}_R(t_2) \frac{N_0}{2} \left(\int_{-B/2}^{B/2} e^{j2\pi (f - f_0)(t_1 - t_2)}\, df + \int_{B/2}^{B/2} e^{j2\pi (f + f_0)(t_1 - t_2)}\, df \right) dt_2 dt_1$$

$$= \delta_{k,l}\mathcal{E}_s \int_0^{T_s} \int_0^{T_s} \acute{p}_R(t_1)\acute{p}_R(t_2) \frac{N_0}{2} \left(\int_{-f_0 - B/2}^{-f_0 + B/2} e^{j2\pi f(t_1 - t_2)}\, df + \int_{f_0 - B/2}^{f_0 + B/2} e^{j2\pi f(t_1 - t_2)}\, df \right) dt_2 dt_1$$

The signal $\acute{p}_R(t)$ has the duration T_s, so $\int_0^{T_s} \acute{p}_R(t) e^{-j2\pi f t} dt = \int_{-\infty}^{\infty} \acute{p}_R(t) e^{-j2\pi f t} dt = TF\{\acute{p}_R(t)\}$, and the bandwidth of $\acute{p}_R(t)$ is $1/T_p$. Recalling that the bandwidth of the bandpass frontend $B \approx 1/T_p$. Hence, $E\{\xi_1[k]\xi_1[l]\}$ is developed as follows:

$$= \frac{N_0}{2}\mathcal{E}_s \delta_{k,l} \int_0^{T_s} \acute{p}_R(t_1) \left(\int_{-f_0 - B/2}^{-f_0 + B/2} \underbrace{\int_0^{T_s} \acute{p}_R(t_2) e^{j2\pi f(t_1 - t_2)} dt_2}_{TF\{\acute{p}_R(t)\} e^{j2\pi f t_1}} df + \right.$$

$$\left. \int_{f_0 - B/2}^{f_0 + B/2} \underbrace{\int_0^{T_s} \acute{p}_R(t_2) e^{j2\pi f(t_1 - t_2)} dt_2}_{TF\{\acute{p}_R(t)\} e^{j2\pi f t_1}} df \right) dt_1$$

$$= \frac{N_0}{2}\mathcal{E}_s \delta_{k,l} \int_0^{T_s} \acute{p}_R(t) \left(\underbrace{\int_{-f_0 - B/2}^{-f_0 + B/2} TF[\acute{p}_R(t)]\, e^{j2\pi f t} df + \int_{f_0 - B/2}^{f_0 + B/2} TF[\acute{p}_R(t)]\, e^{j2\pi f t} df}_{TF^{-1}\{TF\{\acute{p}_R(t)\}\} = \acute{p}_R(t)} \right) dt$$

$$= \frac{N_0}{2}\mathcal{E}_s \delta_{k,l} \underbrace{\int_0^{T_s} \acute{p}_R^2(t)\, dt}_{\mathcal{E}_{max}}$$

Hence:

$$E\{\xi_1[k]\xi_1[l]\} = \frac{N_0}{2}\mathcal{E}_s \mathcal{E}_{max} \delta_{k,l} \tag{5.A.2}$$

Where $\mathcal{E}_{max}$ is an unknown constant received symbol energy, depends on the transmitted channel.

2.2. $E\{\xi_2[k]\xi_2[l]\}$

$$= E\left\{\left(\int_0^{T_s}\sqrt{\mathcal{E}_s}s[k]p_R(t_1)\overline{w}(t_1)\,dt_1\right)\cdot\left(\int_0^{T_s}\sqrt{\mathcal{E}_s}s[l]p_R(t_2)\overline{w}(t_2)\,dt_2\right)\right\}$$

$$= \mathcal{E}_s\int_0^{T_s}\int_0^{T_s}\underbrace{E\{s[k]s[l]\}}_{\delta_{k,l}}p_R(t_1)p_R(t_2)\cdot\underbrace{E\{\overline{w}(t_1)\overline{w}(t_2)\}}_{R_{\overline{w}}(t_1-t_2)}\,dt_2dt_1$$

So $E\{\xi_2[k]\xi_2[l]\}$ becomes:

$$E\{\xi_2[k]\xi_2[l]\} = \mathcal{E}_s\delta_{k,l}\int_0^{T_s}\int_0^{T_s}p_R(t_1)p_R(t_2)R_{\overline{w}}(t_1-t_2)\,dt_2\,dt_1$$

Likewise, $E\{\xi_2[k]\xi_2[l]\}$ is given as follows:

$$E\{\xi_2[k]\xi_2[l]\} = \frac{N_0}{2M}\mathcal{E}_s\mathcal{E}_{max}\delta_{k,l} \tag{5.A.3}$$

2.3. $E\{\xi_3[k]\xi_3[l]\}$

$$E\{\xi_3[k]\xi_3[l]\} = E\left\{\left(\int_0^{T_s}w_k(t_1)\overline{w}(t_1)\,dt_1\right)\cdot\left(\int_0^{T_s}w_l(t_2)\overline{w}(t_2)\,dt_2\right)\right\}$$

$$= \int_0^{T_s}\int_0^{T_s}E\{w_k(t_1)\cdot\overline{w}(t_1)\cdot w_l(t_2)\cdot\overline{w}(t_2)\}\,dt_2dt_1$$

For finding $E\{\xi_3[k]\xi_3[l]\}$, we have employed the fact that the expectation of the product of zero mean, jointly Gaussian random variables $\{n(t_1), n(t_2), n(t_3), n(t_4)\}$ is given as the following:

$$E\{n(t_1)n(t_2), n(t_3)\,n(t_4)\} =$$
$$R_w(t_1-t_2)R_w(t_3-t_4) + R_w(t_1-t_3)R_w(t_2-t_4) + R_w(t_1-t_4)R_w(t_2-t_3)$$

where $n(t_1) = w_k(t_1)$, $n(t_2) = \overline{w}(t_1)$, $n(t_3) = w_l(t_2)$ and $n(t_4) = \overline{w}(t_2)$. In addition, we have found earlier that $E\{w_n(t)\overline{w}(t)\} = 0$. Thus, $E\{\xi_3[k]\xi_3[l]\}$ becomes as follows:

$$E\{\xi_3[k]\xi_3[l]\} = \int_0^{T_s}\int_0^{T_s}R_w(t_1-t_2+(k-l)T_s)R_{\overline{w}}(t_1-t_2)dt_2dt_1$$

$$= \delta_{k,l}\int_0^{T_s}\int_0^{T_s}R_w(t_1-t_2)R_{\overline{w}}(t_1-t_2)dt_2dt_1$$

$$= \delta_{k,l}\int_0^{T_s}\int_0^{T_s}\left[N_0Bsinc\big(B(t_1-t_2)\big)cos\big(2\pi f_0(t_1-t_2)\big)\right]\times$$

$$\left[\frac{N_0}{M}Bsinc\big(B(t_1-t_2)\big)cos\big(2\pi f_0(t_1-t_2)\big)\right]dt_2dt_1$$

$$=\frac{\delta_{k,l}}{M}\int_0^{T_s}\int_0^{T_s}\left[N_0Bsinc\big(B(t_1-t_2)\big)cos\big(2\pi f_0(t_1-t_2)\big)\right]^2 dt_2dt_1$$

From Appendix 2.A, we found clearly that:

$$R_w(\tau)=N_0Bsinc(B\tau)cos(2\pi f_0\tau)=TF^{-1}\left\{\frac{N_0}{2}\right\}_{\pm f_0-B/2}^{\pm f_0+B/2}$$

Since the symbol time $T_s \gg 1/B$ is large enough so that most of the energy of the pulse concentrates within T_s, we can express $E\{\xi_3[k]\xi_3[l]\}$ in frequency domain by applying Parseval's theorem $\int_{-\infty}^{\infty}|x(t)|^2dt=\int_{-\infty}^{\infty}|X(f)|^2df$ and using the previous equation, yields:

$$E\{\xi_3[k]\xi_3[l]\}=\delta_{k,l}\int_0^{T_s}\int_{-B/2}^{B/2}\frac{2(N_0/2)^2}{M}df\,dt=\delta_{k,l}\frac{N_0^2}{2M}BT_s$$

Thus:

$$E\{\xi_3[k]\xi_3[l]\}=\frac{N_0^2}{2M}BT_s\delta_{k,l} \tag{5.A.4}$$

2.4. $E\{\xi_1[k]\xi_2[l]\}$

$$=E\left\{\left(\int_0^{T_s}\sqrt{\mathcal{E}_s}p_R(t_1-\epsilon\pm\Delta)w_k(t_1)\,dt_1\right)\cdot\left(\int_0^{T_s}\sqrt{\mathcal{E}_s}s[l]p_R(t_2)\bar{w}(t_2)\,dt_2\right)\right\}$$

$$=\int_0^{T_s}\int_0^{T_s}\mathcal{E}_sE\{s[l]\}\,p_R(t_1-\epsilon\pm\Delta)p_R(t_2)\underbrace{E\{w_k(t_1)\bar{w}(t_2)\}}_{\approx 0}dt_2\,dt_1$$

Thus:

$$E\{\xi_1[k]\xi_2[l]\}=0 \tag{5.A.5}$$

2.5. $E\{\xi_1[k]\xi_3[l]\}$

$$=E\left\{\left(\int_0^{T_s}\sqrt{\mathcal{E}_s}p_R(t_1-\epsilon\pm\Delta)w_k(t_1)\,dt_1\right)\cdot\left(\int_0^{T_s}w_l(t_2)\bar{w}(t_2)\,dt_2\right)\right\}$$

$$=\int_0^{T_s}\int_0^{T_s}\sqrt{\mathcal{E}_s}E\{p_R(t_1-\epsilon\pm\Delta)\cdot w_k(t_1)\cdot w_l(t_2)\cdot\bar{w}(t_2)\}\,dt_2dt_1$$

$$=\int_0^{T_s}\int_0^{T_s}\sqrt{\mathcal{E}_s}\underbrace{E\{p_R(t_1-\epsilon\pm\Delta)w_k(t_1)\}}_{=0}\underbrace{E\{w_l(t_2)\bar{w}(t_2)\}}_{\approx 0}dt_2dt_1+$$

$$\int_0^{T_s}\int_0^{T_s}\sqrt{\mathcal{E}_s}\underbrace{E\{p_R(t_1-\epsilon\pm\Delta)w_l(t_2)\}}_{=0}\underbrace{E\{w_k(t_1)\overline{w}(t_2)\}}_{\approx 0}dt_2dt_1 +$$
$$\int_0^{T_s}\int_0^{T_s}\sqrt{\mathcal{E}_s}\underbrace{E\{p_R(t_1-\epsilon\pm\Delta)\overline{w}(t_2)\}}_{=0}\underbrace{E\{w_k(t_1)w_l(t_2)\}}_{R_w(t_1-t_2)\delta_{k,l}}dt_2dt_1 = 0$$

Thus:

$$E\{\xi_1[k]\xi_3[l]\} = 0 \tag{5.A.6}$$

2.6. $E\{\boldsymbol{\xi_2[k]\xi_3[l]}\}$

$$= E\left\{\left(\int_0^{T_s}\sqrt{\mathcal{E}_s}s[k]p_R(t_1)\overline{w}(t_1)\,dt_1\right)\cdot\left(\int_0^{T_s}w_l(t_2)\overline{w}(t_2)\,dt_2\right)\right\}$$

$$= \int_0^{T_s}\int_0^{T_s}\sqrt{\mathcal{E}_s}E\{s[k]p_R(t_1)\cdot\overline{w}(t_1)\cdot w_l(t_2)\cdot\overline{w}(t_2)\}\;dt_2dt_1 = 0$$

Thus:

$$E\{\xi_2[k]\xi_3[l]\} = 0 \tag{5.A.7}$$

Substituting (5.A.2 until 5.A.7) in (5.A.1), yields:

$$E\{n_{e,l}[\epsilon,k]n_{e,l}[\epsilon,l]\} = \frac{N_0}{2}\mathcal{E}_s\mathcal{E}_{max}\delta_{k,l} + \frac{N_0}{2M}\mathcal{E}_s\mathcal{E}_{max}\delta_{k,l} + \frac{N_0^2}{2M}BT_s\delta_{k,l} + 0 + 0 + 0$$

$$E\{n_{e,l}[\epsilon,k]n_{e,l}[\epsilon,l]\} = \left(\frac{N_0(M+1)}{2M}\mathcal{E}_s\mathcal{E}_{max} + \frac{N_0^2}{2M}BT_s\right)\delta_{k,l} \tag{5.A.8}$$

We have found that the three noise terms $\{\xi_1,\xi_2,\xi_3\}$ could be treated approximately as mutually uncorrelated zero mean noises. Furthermore, ξ_3 contains the product of two uncorrelated Gaussian noise components, the Central Limit Theorem (CLT) asserts that it will also be approximately Gaussian as a consequence of the integration. In other words, ξ_3 is also approximately white Gaussian noise $\left(\xi_3\sim\mathcal{N}(0,\frac{N_0^2}{2M}BT_s)\right)$.

Finally, $n_{e,l}[\epsilon,n]$ is also zero-mean Gaussian ($n_{e,l}[\epsilon,n]\sim\mathcal{N}(0,\sigma_{e,l}^2 = \frac{N_0(M+1)}{2M}\mathcal{E}_s\mathcal{E}_{max} + \frac{N_0^2}{2M}BT_s)$) with correlation function $\left(\frac{N_0(M+1)}{2M}\mathcal{E}_s\mathcal{E}_{max} + \frac{N_0^2}{2M}BT_s\right)\delta_{k,l}$.

Appendix 5.B

DLL Noise

In this section, we study the statistical properties of DLL noise $\eta[n] = n_e[\epsilon, n] - n_l[\epsilon, n]$. From (5.10), the noise η is got by subtracting two zero-mean Gaussian noises. So η is also considered as a zero-main Gaussian variable, and its variance is extracted as the following:

$$var\{\eta[n]\} = var\{n_e[\epsilon, n]\} + var\{n_l[\epsilon, n]\} - 2cov\{n_e[\epsilon, n], n_l[\epsilon, n]\}$$

Based on Appendix 5.A, we deduce:

$$var\{\eta[n]\} = \frac{N_0(M+1)}{M}\mathcal{E}_s\mathcal{E}_{max} + \frac{N_0^2}{M}BT_s - 2cov\{n_e[\epsilon, n], n_l[\epsilon, n]\} \quad (5.B.1)$$

In line with the work in Appendix 5.A, the covariance between n_e & n_l is given as follows:

$$cov\{n_e[\epsilon, n], n_l[\epsilon, n]\} = \Lambda_1 + \Lambda_2 + \Lambda_3$$

where:
$$\begin{cases} \Lambda_1 = E\left\{\left(\int_0^{T_s}\sqrt{\mathcal{E}_s}p_R(t-\epsilon-\Delta)w_n(t)\,dt\right)\cdot\left(\int_0^{T_s}\sqrt{\mathcal{E}_s}p_R(t-\epsilon+\Delta)w_n(t)\,dt\right)\right\} \\ \Lambda_2 = E\left\{\left(\int_0^{T_s}\sqrt{\mathcal{E}_s}s[n]p_R(t)\bar{w}(t-\Delta)\,dt\right)\cdot\left(\int_0^{T_s}\sqrt{\mathcal{E}_s}s[n]p_R(t)\bar{w}(t+\Delta)\,dt\right)\right\} \\ \Lambda_3 = E\left\{\left(\int_0^{T_s}w_n(t)\bar{w}(t-\Delta)\,dt\right)\cdot\left(\int_0^{T_s}w_n(t)\bar{w}(t+\Delta)\,dt\right)\right\} \end{cases}$$

1. Λ_1

$$= E\left\{\left(\int_0^{T_s}\sqrt{\mathcal{E}_s}p_R(t-\epsilon-\Delta)w_n(t)\,dt\right)\cdot\left(\int_0^{T_s}\sqrt{\mathcal{E}_s}p_R(t-\epsilon+\Delta)w_n(t)\,dt\right)\right\}$$

$$= \mathcal{E}_s\int_0^{T_s}\int_0^{T_s}p_R(t_1-\epsilon-\Delta)p_R(t_2-\epsilon+\Delta)\cdot\underbrace{E\{w_n(t_1)w_n(t_2)\}}_{R_w(t_1-t_2)}\,dt_2dt_1$$

$$= \frac{N_0}{2} \underbrace{\mathcal{E}_s \int_0^{T_s} p_R(t-\epsilon-\Delta) p_R(t-\epsilon+\Delta)\, dt}_{\Gamma_{rr}(2\Delta)}$$

Hence:

$$\Lambda_1 = \frac{N_0}{2} \Gamma_{rr}(2\Delta) \tag{5.B.2}$$

2. $\boldsymbol{\Lambda_2}$

$$= E\left\{\left(\int_0^{T_s} \sqrt{\mathcal{E}_s} s[n] p_R(t) \bar{w}(t-\Delta)\, dt\right) \cdot \left(\int_0^{T_s} \sqrt{\mathcal{E}_s} s[n] p_R(t) \bar{w}(t+\Delta)\, dt\right)\right\}$$

$$= \int_0^{T_s} \int_0^{T_s} \mathcal{E}_s p_R(t_1) p_R(t_2) R_{\bar{w}}(t_1 - t_2 - 2\Delta)\, dt_2\, dt_1$$

$$= \frac{N_0}{M} \mathcal{E}_s \int_0^{T_s} \int_0^{T_s} p_R(t_1) p_R(t_2)\, B sinc\big(B(t_1 - t_2 - 2\Delta)\big) cos\big(2\pi f_0 (t_1 - t_2 - 2\Delta)\big) dt_2 dt_1$$

$$= \frac{N_0}{M} \mathcal{E}_s \int_0^{T_s} \int_0^{T_s} p_R(t_1) p_R(t_2) \left(\int_{-B/2}^{B/2} e^{j2\pi f(t_1 - t_2 - 2\Delta)}\, df\right) \left(\frac{e^{-j2\pi f_0 (t_1 - t_2 - 2\Delta)} + e^{j2\pi f_0 (t_1 - t_2 - 2\Delta)}}{2}\right) dt_2 dt_1$$

$$= \frac{N_0}{2M} \mathcal{E}_s \int_0^{T_s} \int_0^{T_s} p_R(t_1) p_R(t_2) \times \left(\int_{-B/2}^{B/2} e^{j2\pi (f - f_0)(t_1 - t_2 - 2\Delta)}\, df + \int_{B/2}^{B/2} e^{j2\pi (f + f_0)(t_1 - t_2 - 2\Delta)}\, df\right) dt_2 dt_1$$

$$= \frac{N_0}{2M} \mathcal{E}_s \int_0^{T_s} \int_0^{T_s} p_R(t_1) p_R(t_2) \left(\int_{-f_0 - B/2}^{-f_0 + B/2} e^{j2\pi f(t_1 - t_2 - 2\Delta)}\, df + \int_{f_0 - B/2}^{f_0 + B/2} e^{j2\pi f(t_1 - t_2 - 2\Delta)}\, df\right) dt_2 dt_1$$

$$= \frac{N_0}{2M} \mathcal{E}_s \int_0^{T_s} p_R(t_1) \left(\int_{-f_0 - B/2}^{-f_0 + B/2} \underbrace{\int_0^{T_s} p_R(t_2) e^{j2\pi f(t_1 - t_2 - 2\Delta)} dt_2}_{TF\{p_R(t_2)\} e^{j2\pi f(t_1 - 2\Delta)}}\, df + \int_{f_0 - B/2}^{f_0 + B/2} \underbrace{\int_0^{T_s} p_R(t_2) e^{j2\pi f(t_1 - t_2 - 2\Delta)} dt_2}_{TF\{p_R(t_2)\} e^{j2\pi f(t_1 - 2\Delta)}}\, df \right) dt_1$$

$$= \frac{N_0}{2M} \mathcal{E}_s \int_0^{T_s} p_R(t) \left(\underbrace{\int_{-f_0 - B/2}^{-f_0 + B/2} TF[p_R(t)]\, e^{j2\pi f(t - 2\Delta)} df + \int_{f_0 - B/2}^{f_0 + B/2} TF[p_R(t)]\, e^{j2\pi f(t - 2\Delta)} df}_{TF^{-1}\{TF\{p_R(t)\} e^{j2\pi f(-2\Delta)}\} = p_R(t - 2\Delta)} \right) dt$$

$$= \frac{N_0}{2M} \underbrace{\mathcal{E}_s \int_0^{T_s} p_R(t)\, p_R(t - 2\Delta) dt}_{\Gamma_{rr}(2\Delta)}$$

Hence:

$$\Lambda_2 = \frac{N_0}{2M} \Gamma_{rr}(2\Delta) \tag{5.B.3}$$

3. $\boldsymbol{\Lambda_3}$

$$= E\left\{\left(\int_0^{T_s} w_n(t)\overline{w}(t-\Delta)\,dt\right)\cdot\left(\int_0^{T_s} w_n(t)\overline{w}(t+\Delta)\,dt\right)\right\}$$

$$= \int_0^{T_s}\int_0^{T_s} E\{w_n(t_1)\cdot\overline{w}(t_1-\Delta)\cdot w_n(t_2)\cdot\overline{w}(t_2+\Delta)\}\,dt_2 dt_1$$

As in Appendix 5.A, $E\{w_n(t)\overline{w}(t)\} = 0$. Thus:

$$\Lambda_3 = \int_0^{T_s}\int_0^{T_s} R_w(t_1-t_2)R_{\overline{w}}(t_1-t_2-2\Delta)dt_2 dt_1$$

Depending on the demonstration in Appendix 5.A, we can deduce that:

$$\Lambda_3 = \int_0^{T_s}\int_{-B/2}^{B/2} \frac{2(N_0/2)^2}{M} df\,dt \quad = \frac{N_0^2}{2M} BT_s \tag{5.B.4}$$

Substituting (5.B.2 until 5.B.4) in (5.B.1) yields:

$$var\{\eta[n]\} = \sigma_\eta^2 = \frac{N_0(M+1)}{M}(\mathcal{E}_s\mathcal{E}_{max} - \Gamma_{rr}(2\Delta)) \tag{5.B.5}$$

Therefore, $\eta[n]$ is zero-mean Gaussian noise ($\eta[n]\sim\mathcal{N}(0,\sigma_\eta^2)$).

Bibliography

[1] Federal Communications Commission (FCC), Revision of part 15 of the commission's rules regarding ultra-wideband transmission systems, First report and order, ET Docket 98–153, FCC 02–48, February 14, 2002

[2] M. G. Di Benedetto, T. Kaiser, A. F. Molisch, I. Oppermann, C. Politano and D. Porcino, "UWB communications systems: a comprehensive overview," Darmstadt, Germany: EURASIP, 2005

[3] P. Martigne, "UWB for low data rate applications: technology overview and regulatory aspects," in Proc. IEEE International Symposium on Circuits and Systems (ISCAS), pp. 2425-2428, May 21-24, 2006

[4] K. D. Colling and P. Ciorciari, "Ultra wideband communications for sensor networks," in Proc. IEEE Military Communications Conference (MILCOM), pp. 1-7, 2005

[5] S. Gezici, Z. Tian, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. Poor and Z. Sahinoglu, "Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks," IEEE Signal Processing Magazine, vol. 22, pp. 70-84, July 2005

[6] T. Norimatsu, R. Fujiwara, M. Kokubo, M. Miyazaki, A. Maeki, Y. Ogata, S. Kobayashi, N. Koshizuka and K. Sakamura, "A UWB-IR transmitter with digitally controlled pulse generator," IEEE Journal of Solid-State Circuits, vol. 42, no. 6, pp. 1300-1309, June 2007

[7] J. Zhang, P.V. Orlik, Z. Sahinoglu, A. F. Molisch and P. Kinney, "UWB Systems for wireless sensor networks," Proceedings of the IEEE, vol. 97, no. 2, pp. 313 – 331, February 2009

[8] A. Anwarul, M. A. Matin and Asaduzzaman and A. Nowshad, "Novel applications of the UWB technologies: Chapter 8", edited by B. Lembrikov, Published by InTech, ISBN: 9789533073248, pp. 159-174, August 01, 2011

[9] P. Pagani, F. Talom, P. Pajusco and B. Uguen, "Communications ultra large bande: Le canal de propagation radioélectrique," Hermès, Lavoisier, ISBN: 2746216361, 2007

[10] I. Oppermann, M. Hämäläinen and J. Iinatti, "UWB theory and applications," Wiley, ISBN: 0470869178, October 2004

[11] R.S. Thoma, O. Hirsch, J. Sachs and R. Zetik, "UWB sensor networks for position location and imaging of objects and environments," in Proc. 2nd European Conference Antennas and Propagation (EuCAP), pp. 1-9, 2007

[12] L. Yuheng, L. Chao, Y. He, J. Wu and Z. Xiong, "A perimeter intrusion detection system using dual-mode wireless sensor networks," in Proc. 2nd International Conference on Communications and Networking, China, pp. 861–865, August 22-24, 2007

[13] X. Huang, E. Dutkiewicz, R. Gandia and D. Lowe, "Ultra-wideband technology for video surveillance sensor networks," in Proc. IEEE International Conference on Industrial Informatics, pp. 1012-1017, August 16-18, 2006

[14] J. Li and T. Talty, “Channel characterization for ultra-wideband intra-vehicle sensor networks,” in Proc. Military Communications Conference (MILCOM), pp. 1-5, 2006

[15] M. Ghavami, L.B. Michael and R. Kohno, “Ultra wideband signals and systems in communication engineering,” 2nd edition, John Wiley & Sons, Ltd, ISBN: 0470027630, March 6, 2007

[16] H. Arslan, Z.N. Chen and M.G. Di Benedetto, “Ultra wideband wireless communication,” Wiley-Interscience, 1st edition, ISBN: 0471715212, October 13, 2006

[17] Z. Tian and G.B. Giannakis, “Timing synchronization for UWB Impulse Radios,” Chapter4, In: UWB Wireless Communication and Networks, edited by X. Shen, M. Guizani, R. Caiming, Qiu and T. L. Ngoc, John Wiley & Sons, Ltd, ISBN: 0470011440, pp. 53-81, 2006

[18] F. Nekoogar, “Ultra-wideband communications: fundamentals and applications,” published by Prentice Hall, ISBN: 0131463268, August 31, 2005

[19] J. H. Reed, “An introduction to ultra wideband communication systems,” Prentice-Hall, ISBN: 0131481037, April 05, 2005

[20] M.Z. Win and R.A. Scholtz, “On the energy capture of ultra wide bandwidth signals in dense multipath environments,” IEEE Communications Letters, vol. 2, no. 9, pp. 245–247, September 1998

[21] E.A. Homier and R.A. Scholtz, “Rapid acquisition of ultra-wideband signals in the dense multipath channel,” Proc. IEEE Conference on Ultra Wideband Systems and Technologies, pp. 105-110, 2002

[22] R. Fleming, C. Kushner, G. Roberts and U. Nandiwada, “Rapid acquisition for ultra-wideband localizers,” Proc. IEEE Conference on Ultra Wideband Systems and Technologies, pp. 245-250, 2002

[23] L. Yang, Z. Tian and G.B. Giannakis, “Non-data aided timing acquisition of ultra-wideband transmissions using cyclo-stationarity,” Proc. IEEE Conference ASSP, pp. 121-124, 2003

[24] T. Strohmer, M. Emami, J. Hansen, G. Papanicolaou and A.J. Paulraj, “Application of time-reversal with MMSE equalizer to UWB communications,” IEEE Global Communications Conference, pp. 3123-3127, 2004

[25] H. Zhang and D.L. Goeckel, “Generalized transmitted-reference UWB Systems,” Proc. IEEE Conference on Ultra Wideband Systems and Technologies, pp. 147-151, 2003

[26] J.A. Pardinas-Mir, M. Muller, R. Lamberti and C. Gimenes, “TR-UWB detection and synchronization - using the time delayed sampling and correlation detection method,” Proc. 8th EuRAD Conf. pp.202-205, 2011

[27] R. Alhakim, K. Raoof and E. Simeu, “A novel fine synchronization method for dirty template UWB timing acquisition,” Proc. 6th IEEE Conference on Wireless Communications, Networking and Mobile Computing, Chengdu, Chine, pp. 1-4, September 23-25, 2010

[28] R. Alhakim, K. Raoof, E. Simeu and Y. Serrestou, “Cramer-Rao lower bounds and maximum likelihood timing synchronization for dirty template UWB communications,” Signal Image and Video Processing Journal, pp. 1-17, 2011

[29] R. Alhakim, K. Raoof, E. Simeu and Y. Serrestou, “Data-aided timing estimation in UWB communication systems using dirty templates,” IEEE International Conference on Ultra Wideband (ICUWB), pp. 435-439, September 14-16, 2011

[30] R. Alhakim, E. Simeu and K. Raoof, “Internal model control for a self-tuning delay-locked loop in UWB communication systems,” 17th IEEE International On-Line Testing Symposium (IOLTS), Athens, Greece, pp. 121-126, July 13-15, 2011

[31] R. Alhakim, E. Simeu and K. Raoof, "Design of tracking loop with dirty templates for UWB communication systems," Signal Image and Video Processing Journal, pp.1-17, June 27, 2012

[32] R. Alhakim, K. Raoof and E. Simeu, "Design of tracking loop for UWB systems," International Conference on Information Processing and Wireless Systems (IP-WIS), Sousse, Tunisia, March 16-18, 2012

[33] R. Alhakim, E. Simeu and K. Raoof, "Timing synchronisation for IR-UWB communication systems," Chapter 2, In: Ultra wideband - current status and future trends, edited by M. A. Matin, Published by InTech, ISBN: 9789535107811, pp. 15-40, October 03, 2012

[34] R. Alhakim, K. Raoof and E. Simeu, "Detection of UWB signal using dirty template approach," Signal Image and Video Processing Journal, Ed. Springer, 2012 (accepted)

[35] N. Gross, 21 ideas for the 21st century, Business Week, pp. 78–167, August 30, 1999

[36] R. Shorey, A. Anand, M.C. Chan and W.T. Ooi, "Mobile, wireless and sensor networks: technology, applications and future directions," Wiley-IEEE Press, 1st edition, ISBN: 0471718165, March 23, 2006

[37] F. Martincic and L. Schwiebert, "Introduction to wireless sensor networking," Chapter1, In: Handbook of sensor networks: algorithms and architectures, edited by I. Stojmenovic, Wiley-Interscience, 1st edition, pp. 1-40, ISBN: 0471684724, October 5, 2005

[38] Q. Wang and I. Balasingham, "Wireless sensor networks - an introduction," Chapter 1, In: Wireless sensor networks: application - centric design, edited by G.V. Merrett and Y.K. Tan, Published by InTech, ISBN: 978-953-307-321-7, pp. 1-16, December 14, 2010

[39] O. Gatin and B. Lhenoret, "WSN and energy harvesting opportunities for railway applications at SNCF," IDTechEx: Energy Harvesting & Storage Europe in Munich, Germany, May 26, 2010

[40] J. Zhang, P. Orlik, Z. Sahinoglu, A. Molisch and P. Kinney, "UWB systems for wireless sensor networks," Mitsubishi Electric Research Laboratories, http://www.merl.com/, March 2009

[41] J. Zhang, P.V. Orlik, Z. Sahinoglu, A.F. Molisch and P. Kinney, "UWB systems for wireless sensor networks," Proceedings of the IEEE, vol. 97, issue 2, pp. 313-331, February 2009

[42] C. Shannon, "A mathematical theory of communication," The Bell System Technical Journal 27, pp. 379–423 and 623–656, July/October 1948

[43] J.D. Parsons, "The mobile radio propagation channel," 2nd edition, John Wiley & Sons, Ltd, ISBN: 9780471988571, 2000

[44] J.F. M. Gerrits and J.R. Farserotu, "Wavelet generation circuit for UWB impulse radio applications," Electronics Letters, vol. 38, no. 25, pp. 1737-1738, December 2002

[45] F. Nekoogar, "Ultra-wideband communications: fundamentals and applications," Prentice Hall, ISBN: 0131463268, August 31, 2005

[46] A. Reisenzahn, T. Buchegger, D. Scherrer, S. Matzinge, S. Hantscher and C. G. Diskus, "A ground penetrating UWB radar system," The Third International Conference of UWB and Ultra-short Impulse Signals, pp. 116-118, September 2006

[47] H.T. Li, H.Y. Chan, C. W. Chao and W.P. Lin, "Performance study of wireless universal serial bus transmission system," 11th International Conference on Advanced Communication Technology (ICACT), vol. 1, pp. 193-195, February 15-18, 2009

[48] R. Aiello and A. Batra, "Ultra wideband systems: technologies and applications," Communications Engineering Series, Newnes, 2006

[49] IEEE 802.15 WPAN Low Rate Alternative PHY Task Group 4a (TG4a). Available: http://www.ieee802.org/15/pub/TG4a.html

[50] R. A. Scholtz, "Multiple access with time-hopping impulse modulation," In IEEE MILCOM93, vol. 2, October 1993

[51] J. T. Conroy, J. L. LoCicero and D. R. Ucci, "Communication techniques using mono-pulse waveforms," In IEEE MILCOM99, vol. 2, November 1999

[52] M. Ghavami, L.B. Michael, S. Haruyama and R. Kohno, "A novel UWB pulse shape modulation system," Kluwer Wireless Personal Communications Journal, pp. 105-120, 2002

[53] R.M. Buehrer, W.A. Davis, A. Safaai-Jazi and D. Sweeney, "Ultra-wideband propagation measurements and modelling," DARPA NETEX Program Final Report, January 2004

[54] A. Stephan, "Resource allocation strategies and linear pre-coded OFDM optimization for ultra-wideband communications," Thesis, INSA & IETR, Rennes, December 15, 2008

[55] D. Cassioli, M.Z. Win and A. F. Molisch, "The ultra-wide bandwidth indoor channel: from statistical model to simulations," IEEE Journal on Selected Areas in Communications, vol. 20, pp. 1247–1257, August 2002

[56] M.Z. Win and R.A. Scholtz, "Impulse radio: how it works," IEEE Communications Letters, vol. 2, no. 2, pp. 36-38, February 1998

[57] S.S. Ghassemzadeh et al., "A statistical path loss model for in-home UWB channels," Proc. IEEE Conference on Ultra Wideband Systems and Technologies, pp. 59-64, May 2002

[58] R.J. Cramer, R. A. Scholtz, and M.Z. Win, "Evaluation of an ultra-wide-band propagation channel," IEEE Transactions on Antennas and Propagations, vol. 50, pp. 561–570, May 2002

[59] J. Kunisch and J. Pamp, "Measurement results and modelling aspects for the UWB radio channels," Proc. IEEE Conference on Ultra Wideband Systems and Technologies, pp. 19-23, May 2002

[60] A. Street, L. Lukama and D. Edwards, "Use of VNAs for wideband propagation measurements," IEEE Proc. Communications, vol. 148, no. 6, pp. 411-415, December 2001

[61] A. A. Saleh and R. A. Valenzuela, "A statistical model for indoor multipath propagation," IEEE Journal on Selected Areas in Communications, vol. SAC-5, no. 2, pp. 128-137, February 1987

[62] W.P. Siriwongpairat and K.J.R. Liu, "Ultra-wideband communications systems: multiband OFDM approach," Published by Wiley-IEEE Press, 1st edition, ISBN: 0470074698, November 27, 2007

[63] F.M. Andreas and R.F. Jeffrey, "Channel models for ultra-wideband personal area networks," IEEE wireless communications, vol. 10, no. 6, pp. 14-21, December 2003

[64] L. Yang and G. B. Giannakis, "Low-complexity training for rapid timing acquisition in ultra-wideband communications", Proc. IEEE Global Telecommunications Conference, San Francisco, CA, pp. 769-773, December 1-5, 2003

[65] M.Z. Win and R.A. Scholtz, "Ultra-wide bandwidth time-hopping spread-spectrum impulse radio for wireless multiple-access communications," IEEE Transactions on Communications, vol. 48, no. 4, pp. 679-691, April 2000

[66] J.G. Proakis and M. Salehi, "Communication systems engineering," 2nd edition, Section 7.8: Symbol Synchronization, Prentice Hall, Upper Saddle River, New Jersey, 2002

[67] V. Lottici, A. D'Andrea and U. Mengali, "Channel estimation for ultra-wideband communications," IEEE Journal on Select Areas in Communications, vol. 20, no. 12, pp. 1638–1645, December 2002

[68] L. Yang and G.B. Giannakis, "Timing ultra-wideband signals with dirty templates," Proc. IEEE Transactions On Communications, Vol. 53, No. 11, pp. 1952-1963, November 2005

[69] Kay S.M, "Fundamentals of statistical signal processing: detection theory." Prentice Hall, 1998

[70] A. Schranzhofer, "Acquisition for a transmitted reference UWB receiver," Thesis, Delft University of Technology, Netherlands, May 2007

[71] B. A. Bernd, "Markov chain Monte Carlo simulations and their statistical analysis: with web-based Fortran code," Hackensack, NJ: World Scientific, ISBN: 981-238-935-0, October 1, 2004

[72] W. T. Eadie, D. Drijard, F.E. James, M. Roos and B. Sadoulet, "Statistical methods in experimental physics," North-Holland Publishing Co. Amsterdam, pp. 269–334, ISBN 0-444-10117-9, 1971

[73] G. Marsaglia, W. Tsang and J. Wang, "Evaluating Kolmogorov's distribution," Journal of Statistical Software, Vol. 8, Issue 18, 2003

[74] Matlab help: Kolmogorov-Smirnov test, http://www.mathworks.fr/fr/help/stats/kstest.html

[75] L. Cairone, "Synchronization in UWB systems," Thesis, University of Pisa, Italy, 2006-2007

[76] F. Salem, R. Pyndiah and A. Bouallegue, "Synchronization using an adaptive early-late algorithm for IR-TH-UWB transmission in multipath scenarios," Proc. International Symposium on Wireless Communication Systems, pp. 268-271, September 7, 2005

[77] S.M. Kay, "Fundamentals of statistical signal processing: estimation theory," Prentice Hall, Chapter 3, pp. 27-67, 1993

[78] H.L.V. Trees, "Detection estimation and modulation theory: part I," John Wiley and Sons, Chapter 2, pp. 65-73, 2001

[79] H. Meyr, M. Moeneclaey and S.A. Fechtel, "Digital communication receivers: synchronization, channel estimation and signal processing," John Wiley & Sons Inc, Chapter 2, pp. 89-99, 1997

[80] A.N. D'Andrea, U. Mengali and R. Reggiannini, "The modified Cramer-Rao bound and its application to synchronization problems," IEEE Trans. Communications, vol. 42, no. 234, pp. 1391-1399, 1994

[81] E.G. Ström, S. Parkvall, S.L. Miller and B.E. Ottersten, "Propagation delay estimation in asynchronous direct sequence code-division multiple access systems," IEEE Trans. Communications, vol. 44, no. 1, pp. 84-93, 1996

[82] E.G. Ström and F. Malmsten, "A maximum likelihood approach for estimating DS-CDMA multipath fading channels," IEEE Journal on Selected Areas in Communications, vol. 18, no. 1, pp. 132-140, 2000

[83] H. Steendam and M. Moeneclaey, "Low-SNR limit of the Cramer-Rao bound for estimating the time delay of a PSK, QAM, or PAM waveform," IEEE Communications Letters, vol. 5, no. 1, pp. 31-33, 2001

[84] F. Rice, B. Cowley, B. Moran and M. Rice, "Cramer-Rao lower bounds for QAM phase and frequency estimation," IEEE Trans. Communications, vol. 49, no. 9, pp. 1582-1591, 2001

[85]M. Moeneclaey, “A simple lower bound on the linearized performance of practical symbol synchronizers,” IEEE Trans. Communications, vol. 31, no. 9, pp. 1029-1032, 1983

[86] M. Moeneclaey, “A fundamental lower bound on the performance of practical joint carrier and bit synchronizers,” IEEE Trans. Communications, vol. 32, no. 9, pp. 1007-1012, 1984

[87] M. Moeneclaey, “On the true and the modified Cramer-Rao bounds for the estimation of a scalar parameter in the presence of nuisance parameters,” IEEE Trans. Communications, vol. 46, no. 11, pp. 1536-1544, 1998

[88] H. Meyr and G. Ascheid, “Synchronization in digital communications, volume I: phase-, frequency-locked loops and amplitude control,” John Wiley & Sons Inc., Chapter 2-3, pp. 19-161, 1990

[89] J. S. Lee and L. E. Miller, “CDMA systems engineering handbook,” Artech House Publishers, Chapter 7, pp. 790-802, 1998

[90] W. C. Lindsey and M. K. Simon, “Telecommunication systems engineering,” Dover Publications Inc. New York, Chapter 4, pp. 159-165, 1992

[91] J. G. Proakis, “Digital communications,” 4th ed. McGraw Hill Higher Education, Chapter 6, pp. 358-365, 2001

[92] S. Sumi and S. Tachikawa, “On a delay locked loop for UWB-IR system,” Natl. Conv. Rec. IEICE Japan, A-5-26, pp. 165, March 2003

[93] W. Zhang, H. Shen and K. S. Kwak, “Improved delay-locked loop in a UWB impulse radio time-hopping spread-spectrum system,” Proc. ETRI journal, vol. 29, no. 6, pp. 716-724, December 2007

[94] U. Mengali and A.N.D. Andrea, “Synchronization techniques for digital receivers,” Springer, Chapter 3, pp. 120-128, 1997

[95] S. Farahmand, L. Xiliang and G.B. Giannakis, “Demodulation and tracking with dirty templates for UWB impulse radio: algorithms and performance”, Proc. IEEE Trans. On Vehicular Technology, vol. 54, no. 5, pp. 1595-1608, september 2005

[96] R. Jaffe and E. Rechtin, “Design and performance of phase-lock circuits capable of near-optimum performance over a wide range of input signal and noise levels”, Proc. IRE Trans. Inf. Theory, vol. 1, pp. 66–76, March 1955

[97] T.S. Rappaport, “Wireless communications: principles and practice,” 2nd *Edition*, Prentice Hall, 2001, Chapter 4, pp. 139-167

[98] C.S. Ling, M.D. Brown, P.F. Weston and C. Roberts, “Gain tuned internal model control for handling saturation in actuators,” American Control Conference, Boston, Massachusetts, vol.5, pp. 4692 - 4697 vol. 5, June 30 - July 2, 2004

[99] B. Hamed and W. Issa, “A new approach for robust internal model control of nonlinear systems: control systems application,” LAP LAMBERT Academic Publishing, ISBN: 3846556904, November 2011

[100] F. Mahbub, “System identification of internal model control,” LAP LAMBERT Academic Publishing, ISBN: 3846523526, October 2011

[101] G. FENG and R. LOZANO, “Adaptive control systems,” Newnes, ISBN: 0750639962, Jun. 1999

[102] C. Brosilow and B. Joseph, “Techniques of model-based control,” Prentice Hall PTR, ISBN: 013028078, April 2002

[103] L. Ljung, System Identification: Theory for the User, 2nd Ed, Prentice Hall, ISBN: 0136566952, January 1999

[104] J. Fox, "Applied regression analysis and generalized linear models," SAGE Publications Inc. 2nd edition, ISBN: 0761930426, juin 2008

[105] H. Sahai and M. I. Ageel, "The analysis of variance: fixed, random and mixed models," Birkhauser Boston Inc, ISBN: 0817640126, February 2000

[106] A.C. Cameron and F.A.G. Windmeijer, "An R-squared measure of goodness of fit for some common nonlinear regression models," Journal of Econometrics, Elsevier, vol. 77, Issue 2, pp. 329-342, April 1997

[107] T. Ohta, T. Ogita, S. Rump and S. Oishi, "Numerical verification method for arbitrarily ill-conditioned linear systems," Transactions on the Japan Society for Industrial and Applied Mathematics (JSIAM), vol. 15, no. 3, pp. 269-287, 2005

[108] C.R. Cutler and F.H. Yocum, "Experience with the DMC inverse for identification," Proc. of the 4th International Conference on Chemical Process Control, Padre Island, Texas, AIChE, NY, pp. 297–317, July 1991

[109] E. Simon, L. Ros and K. Raoof, "Synchronization over rapidly time-varying multipath channel for CDMA downlink RAKE receivers in time-division mode," Vehicular Technology, IEEE Transactions on, vol. 56, no. 4, pp. 2216-2225, July 2007

[110] L. Jun, W. Wan-li and D. Xiu-hua, "Multiple model internal model control based on fuzzy membership function", Proceedings of the IEEE International Conference on Automation and Logistics Qingdao, 2008

[111] K. Lahmari, "Contribution à la commande multi-modle des processus complexes," Thesis, UST de Lille, 1999

[112] S. Talmoudi, K. Abderrahim, R. Ben Abdennour and M. Ksouri, "Multi-model approach using neural networks for complex systems modeling and identification," The Internationa Journal of Nonlinear Dynamics and Systems Theory, pp. 299-316, 2008

[113] P.J. Gawthrop, "Continuous-time local state local model networks," Proceeding of IEEE Conference on Systems, Man and Cybernetics, pp. 852-857, 1995

[114] K.S. Narendra and Balakrishnan, "Adaptative control using multiple models," IEEE Transactions on Automatic Control, pp. 171-187, 1997

[115] S. Mezghani, A. Elkamel and P. Borne, "Multi-model control of discrete systems with uncertainties," The Electronic International Journal of Advance Modilling and Optimization, pp. 7-17, 2001

[116] A. Zribi, M. Chtourou and M. Djemel, "Internal multiple models control based on robust clustering algorithm," Nonlinear Dynamics and Systems Theory, pp. 99-112, 2011

[117] K. Gasso, "Identification des systmes dynamiques non linaires: approche multimodle," Thesis, Institut National Polytechnique de Lorraine, France, 2000

[118] T.A. Johansen and B.A. Foss, "Constructing NARMAX models using ARMAX models," International Journal of Control, vol. 58, pp. 1125-1153, 1993

[119] R. Murray-Smith, T.A. Johansen and T. Johanson, "Multiple model approaches to modelling and control," Taylor & Francis Ltd, 19 March 1997

[120] K. Atkinson, "An introduction to numerical analysis," New York: John Wiley & Sons, 2nd edition, ISBN: 978-0-47150-023-0, January 17, 1989

[121] Y.L. Chou, “Statistical analysis,” Holt,Rinehart & Winston of Canada Ltd; 2nd edition, ISBN: 978-0-03089-422-0, July 1975

[122] Http://en.wikipedia.org/wiki/Moving_average#cite_note-1

[123] S.T. Karris, “Introduction to simulink with engineering applications,” Orchard Publications, ISBN: 0974423971, section 5.1.14, 30 Mai 2006

[124] T. Hagglund, “PID controllers: theory, design, and tuning,” Published by ISA: The Instrumentation, Systems, and Automation Society, 2nd edition, ISBN: 1556175167, 1 January 1995

[125] K.J. Astrom and T. Hagglund, “Advanced PID control,” Published by ISA: The Instrumentation, Systems, and Automation Society, ISBN: 1556179421, 15 August 2005

[126] S. Haykin, “Adaptive filter theory,” Prentice Hall, 4th edition, ISBN: 0-130-90126-1, 24 Septembre 2001

[127] P.S.R. Diniz, “Adaptive filtering: algorithms and practical implementation,” Springer, 4th edition, ISBN: 1-461-44105-6, 13 August 2012

[128] S. Haykin and B. Widrow,” Least-mean-square adaptive filters,” Wiley-Interscience, 1st edition, ISBN 0-471-21570-8, 8 September 2003

[129] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, “Chapter 9: Root finding and nonlinear sets of equations,” in Book: Numerical Recipes: The Art of Scientific Computing, New York, Cambridge University Press, 3rd ed., ISBN: 978-0-521-88068-8, 10 September 2007

[130] C.T. Kelley, “Solving nonlinear equations with newton's method,” Society for Industrial and Applied Mathematics, ISBN: 0-89871-546-6,1 January 1987

[131] M. Greenberg, “Advanced engineering mathematics,” Prentice Hall, 2nd edition, ISBN: 0-13-321431-1, 18 January 1998